Google Code

Picasa Web Albums Data API

Picasa Web Albums Data API Reference Guide

This document provides detailed reference documentation for the Picasa Web Albums data API.

Contents

Audience

Picasa Web Albums feed types
- Visibility values
- Projection values
- Path values
- Kind values

Picasa Web Albums query parameters reference

Picasa Web Albums elements reference
- Media RSS namespace
- GeoRSS and GML (georss and gml namespace) element reference
- Exif namespace
- Picasa Web Albums extension namespace

Audience

This document is intended for programmers who want to write client applications that can interact with Picasa Web Albums.

It's a reference document; it assumes that you understand the concepts presented in the developer's guide, and the general ideas behind the Google data APIs protocol.

Picasa Web Albums feed types

Picasa Web Albums provides a variety of representations of photo- and album-related data. There are three independent axes for specifying what you want when you request data: visibility, projection, and path/kind.

Visibility values let you request data at various levels of sharing. For example, a visibility value of `public` requests publicly visible data. For a list of values, see Visibility values, below. The default visibility depends on whether the request is authenticated or not.

Projection values let you indicate what elements and extensions should appear in the feed you're requesting. For example, a projection value of `base` indicates that the representation is a basic Atom feed without any extension elements, suitable for display in an Atom reader. For a list of values, see Projection values, below. You must specify a projection value.

Path and kind values let you indicate what type of items you want information about. For example, a path of `user/liz` and a kind value of `tag` requests a feed of tags associated with the user whose username is `liz`. Path and kind values are separate parts of the URI, but they're used together to indicate the item type(s) desired. For a list of path values, see Path values, below; for a list of kind values, see Kind values, below. You must specify a path, but kind is optional; the default kind depends on the path.

To request a particular representation, you specify a visibility value, a projection value, and a path and kind in the URI that you send to Picasa Web Albums.

The URI of a representation of a Picasa Web Albums feed takes the following form:

```
http://picasaweb.google.com/data/feed/projection/path?kind=kind&access=visibili
```

For example, the following URI is for a feed listing a user's private albums. The projection is `api`, which requests a full feed with all relevant extension elements, and the visibility is `private`, which requests only albums that aren't publicly visible:

```
http://picasaweb.google.com/data/feed/api/user/liz?kind=album&access=private
```

If you don't specify a visibility value, then the visibility depends on your authentication. For authenticated requests, the default is `all`. For unauthenticated requests, the default is `public`.

The projection value is required.

As a developer, you'll usually want to use the `api` projection. You'll normally use only two of the visibility levels: `public` or `all`.

For an authenticated feed, instead of specifying a user ID, you can specify the special value `default`, which tells Picasa Web Albums to use the user ID and password that you provided during authentication.

Some feed types are read/write (that is, you can post new data to them, and edit or delete existing data), while others are read-only. Read-only feeds don't include a `<link rel="http://schemas.google.com/g/2005#post">` element (to tell you where to post new events), and entries in read-only feeds don't include `<link rel="edit">` or `<link rel="edit-media">` elements.

Visibility values

The following table describes the supported visibility values:

Visibility	Description	Security Notes
`all`	Shows both public and private data.	Requires authentication. Default for authenticated users.
`private`	Shows only private data.	Requires authentication.
`public`	Shows only public data.	Does not require authentication. Default for unauthenticated users.

Projection values

Although the examples in this documentation use only the `api` projection, Picasa Web Albums also supports a `base` projection for use by feed readers.

The following table describes the supported projection values:

Projection Name	Description	Updatability

`api`	Feed including all relevant gphoto extension elements. All properties contain plain text, not HTML.	Read/write if authenticated user is owner of the content.
`base`	Basic Atom feed without any extension elements. Its `<atom:summary>` and `<atom:content>` properties contain entity-encoded HTML.	Always read-only.

Path values

There are three basic types of feeds: user-based, album-based, and photo-based. You indicate which type you want by specifying a path. For each of those types, you can additionally specify which kind or kinds of associated data you want.

Note: Each path includes a user ID, which is an ID associated with a user's Google account. The user ID can be either a username or an email address, so it may or may not include an @ sign and a domain name.

User-based feed

The user-based feed represents data associated with a particular user. A user-based feed can contain either `album`, or `tag` or `photo` kinds, which you can request using the `kind` parameter.

To retrieve a user-based feed, you can make a request to the appropriate feed URL:

```
http://picasaweb.google.com/data/feed/projection/user/userID/?kind=kinds
```

Album-based feed

The album-based feed represents an album and any kinds associated with the album. An album-based feed can contain either `photo` or `tag` kinds, which you can request using the `kind` parameter.

To retrieve an album-based feed, you can make a request to the appropriate feed URL. To retrieve an album by ID:

```
http://picasaweb.google.com/data/feed/projection/user/userID/albumid/albumID?ki
```

To retrieve an album by name:

```
http://picasaweb.google.com/data/feed/projection/user/userID/album/albumName?ki
```

Photo-based feed

The photo-based feed provides a list of comments or tags associated with the specified photo.

To retrieve a photo-based feed, send a request to the appropriate feed URL, specifying the photo's album either by ID or by name:

```
http://picasaweb.google.com/data/feed/projection/user/userID/albumid/albumID/ph
```

Or:

```
http://picasaweb.google.com/data/feed/projection/user/userID/album/albumName/ph
```

Community search feed

The community search feed allows for searching all public, searchable photos. The feed is similar to that of an album-based feed.

To retrieve a community search feed, send a request to the appropriate feed URL, specifying the q query parameter and value:

```
http://picasaweb.google.com/data/feed/projection/all?kind=photo&q=searchTerm
```

Kind values

The `kind` parameter lets you request information about a particular kind of item. The parameter value should be a comma-separated list of requested kinds.

If you omit the `kind` parameter, Picasa Web Albums chooses a default kind depending on the level of feed you're requesting. For a user-based feed, the default kind is `album`; for an album-based feed, the default kind is `photo`; for a photo-based feed, the default kind is `comment`; for a community search feed, the default kind is `photo`.

Note: You can't edit comments or tags. Sending a PUT to the edit URI of a comment or tag entry has no effect.

The following table describes the supported kind values:

Kind	Description	Supported Feeds
`album`	Feed includes some or all of the albums the specified user has in their gallery. Which albums are returned depends on the visibility value specified.	User-based feed
`photo`	Feed includes the photos in an album (album-based), recent photos uploaded by a user (user-based) or photos uploaded by all users (community search).	Album-based, user-based and community search feeds
`comment`	Feed includes the comments that have been made on a photo.	Photo-based and user-based feeds
`tag`	Includes all tags associated with the specified user, album, or photo. For user-based and album-based feeds, the tags include a weight value indicating how often they occurred.	User-based, album-based, and photo-based feeds

Picasa Web Albums query parameters reference

Picasa Web Albums supports the standard GData `start-index` and `max-results` parameters. The user-based and community search feeds support the q full-text search query parameter. Picasa Web Albums does not currently support the other standard parameters.

The Picasa Web Albums data API supports the following parameters:

Parameter	Meaning	Notes
`kind`	Picasa Web Albums-specific query parameter for kind queries.	See the Kind values section for valid values.
`access`	Visibility parameter	See the Visibility values section for valid values.
`thumbsize`	Thumbnail size parameter	Valid with `album` or `photo` kinds; specifies what image size to use for thumbnails. Multiple values may be specified using a comma-delimited list. If multiple values are specified, multiple `media:thumbnail` elements will be returned in the feed. Refer to the list of valid values below.
`imgmax`	Image size parameter	Valid with `album` or `photo` kinds; specifies what image size to use for the `media:content`. Only a single value may be specified. Refer to the list of valid values below.
`tag`	Tag filter parameter	Filters photos in the user-based and album-based feeds by the specified tag.
`q`	Full-text query string	Searches the title, caption and tags for the specified string value. See the Google data query parameter reference. This parameter can only be used with the user-based and community search feeds.
`max-results`	Maximum number of results to be retrieved	See the Google data query parameter reference.
`start-index`	1-based index of the first result to be retrieved	Used to page through the result set. See the Google data query parameter reference.

The following values are valid for the `thumbsize` and `imgmax` query parameters:

Size	Cropped	Embeddable	Size	Cropped	Embeddable
`32`	Yes	Yes	`576`	No	Yes
`48`	Yes	Yes	`640`	No	Yes
`64`	Yes	Yes	`720`	No	Yes
`72`	No	Yes	`800`	No	Yes
`144`	No	Yes	`912`	No	No
`160`	Yes	Yes	`1024`	No	No
`200`	No	Yes	`1152`	No	No
`288`	No	Yes	`1280`	No	No
`320`	No	Yes	`1440`	No	No
`400`	No	Yes	`1600`	No	No
`512`	No	Yes	`d`	No	No

The *embeddable*
values above indicate whether the image can be embedded in a webpage. If not, the resulting image may only be referenced directly.

Note: The d size value results in the `<media:content>` elements referencing the original uploaded photo, including all original Exif data. It is valid only for use with the `imgmax` query parameter.

Picasa Web Albums elements reference

In addition to the standard GData elements, the Picasa Web Albums data API uses the `media` and `gphoto` namespaces.

Media RSS (media namespace) element reference

GeoRSS and GML (georss and gml namespace) element reference

Exif (exif namespace) element reference

Picasa Web Albums extension (gphoto namespace) element reference

For information about the standard GData elements, see the Atom specification and the Kinds document.

Media RSS (media namespace) element reference

Picasa Web Albums uses the `media`
namespace for elements defined in the Media RSS specification. For information about the `media` namespace, see the Media RSS specification.

Note: Despite the name "Media RSS," the Picasa Web Albums data API uses elements from the media namespace in both Atom and RSS feeds.

Picasa Web Albums uses the following `media` elements:

media:content

media:credit

media:description

media:group

media:keywords

media:thumbnail

media:title

As usual in the GData documentation, a question mark after an attribute name indicates that the attribute is optional. Almost all of the attributes we list for `media` elements are required.

media:content

Contains the URL and other information about the full version of the entry's media content.

The `<media:content>` element appears as a child of a `<media:group>` element.

There can be multiple `<media:content>` elements for a given `<media:group>`. For example, a video may have a `<media:content medium="image">` element that specifies a JPEG representation of the video, and a `<media:content medium="video">` element that specifies the URL of the video itself.

Attributes

Attribute	Description
`url`	The URL of the full version of the media for the current entry.
`type`	The MIME type of the media.
`medium`	Either `image` or `video`. Somewhat redundant, as `type` contains a more specific MIME type, but `medium` may be simpler for the client to interpret.
`height`	The height of the image or video represented by this `<media:content>` element.
`width`	The width of the image or video represented by this `<media:content>` element.
`fileSize?`	The file size in bytes of the image or video represented by this `<media:content>` element.

Example

```
<media:content url="http://lh3.google.com/image/.../MyImage.jpg"
  fileSize="12216320" type="image/jpeg" medium="image"
  height="1600" width="1200"><media:content>
```

media:credit

Contains the nickname of the user who created the content. This is a user-specified value that should be used when referring to the user by name.

The `<media:credit>` element appears as a child of a `<media:group>` element.

```
<media:credit>Liz Bennet</media:credit>
```

media:description

Contains a description of the entry's media content. For `api` projections, the description is in plain text; for `base` projections, the description is in HTML.

The `<media:description>` element appears as a child of a `<media:group>` element.

Attributes

Attribute	Description
`type`	Either `plain` or `html`. Is set to `plain` for `api` projections, `html` for `base` projections.

Example

```
<media:description type="plain">A set of photographs I took while
  vacationing in Italy.  Florence is beautiful!</media:description>
```

media:group

Container element for all `media` elements.

The `<media:group>` element can appear as a child of an `album` or `photo` entry.

media:keywords

Lists the tags associated with the entry. Contains a comma-separated list of tags that have been added to the photo, or all tags that have been added to photos in the album.

The `<media:keywords>` element appears as a child of a `<media:group>` element.

```
<media:keywords>italy, vacation, sunset</media:keywords>
```

media:thumbnail

Contains the URL of a thumbnail of a photo or album cover. If the `thumbsize` parameter is set, this element points to thumbnails of the requested sizes; otherwise the thumbnails are the default thumbnail size.

The `<media:thumbnail>` element appears as a child of a `<media:group>` element. There can be multiple `<media:content>` elements for a given `<media:group>`; for example, a given item may have multiple thumbnails at different sizes. Photos generally have two thumbnails at different sizes; albums generally have one cropped thumbnail.

Attributes

Attribute	Description
`url`	The URL of the thumbnail image.
`height`	The height of the thumbnail image.
`width`	The width of the thumbnail image.

Example

```
<media:thumbnail
  url="http://picasaweb.google.com/image/liz/.../Wonderful.jpg?imgmax=160&crop=
  height='160' width='160'></media:thumbnail>
```

media:title

Contains the title of the entry's media content, in plain text.

The `<media:title>` element appears as a child of a `<media:group>` element.

Attributes

Attribute	Description
`type`	Always set to `plain`.

Example

```
<media:title type="plain">My Album</media:title>
```

GeoRSS and GML (georss and gml namespace) element reference

Picasa Web Albums uses the `georss` and `gml` namespaces for elements defined in the GeoRSS and Geography Markup Language specifications.

Note: Despite the name "GeoRSS," the Picasa Web Albums data API uses elements from these namespaces in both Atom and RSS feeds.

Specifically, Picasa Web Albums uses the following elements:

- **georss:where**
- **gml:Point**
- **gml:pos**

Picasa Web Albums also accepts geographic-location data in two other formats: W3C format and plain-GeoRSS (without GML) format. But those formats are beyond the scope of this document. All geo data that appears in feeds generated by Picasa Web Albums uses the GeoRSS-plus-GML format documented here.

As usual in the GData documentation, a question mark after an attribute name indicates that the attribute is optional. Almost all of the attributes we list for these elements are required.

georss:where

Specifies a geographical location or region. (Not to be confused with `<gd:where>`.) A container element, containing a single `<gml:Point>` element.

Appears as a child of a photo or album entry.

Attributes

None.

Example

```
<georss:where>
  <gml:Point>
    <gml:pos>35.669998 139.770004</gml:pos>
  </gml:Point>
</georss:where>
```

gml:Point

Specifies a particular geographical point, by means of a `<gml:pos>` element.

Appears as a child of a `<georss:where>` element.

Attributes

None.

Example

```
<gml:Point>
  <gml:pos>35.669998 139.770004</gml:pos>
</gml:Point>
```

gml:pos

Specifies a latitude and longitude, separated by a space.

Appears as a child of a `<gml:Point>` element.

Attributes

None.

Example

```
<gml:pos>35.669998 139.770004</gml:pos>
```

Exif (exif namespace) element reference

Picasa Web Albums uses the `exif` namespace to represent Exif data encoded in a photo. The schema URL for the `exif` namespace is `http://schemas.google.com/photos/exif/2007`.

The `exif` elements all appear inside the container element called `<exif:tags>`.

Picasa Web Albums uses the following `exif` elements:

- **exif:distance**
- **exif:exposure**
- **exif:flash**
- **exif:focallength**
- **exif:fstop**
- **exif:imageUniqueID**
- **exif:iso**
- **exif:make**
- **exif:model**
- **exif:tags**
- **exif:time**

exif:distance

The distance to the subject.

The `<exif:distance>` element can appear as a child of `<exif:tags>`.

Example

```
<exif:distance>0.0</exif:distance>
```

Schema

```
namespace exif = "http://schemas.google.com/photos/exif/2007"
start = distance

distance =
  element exif:distance {
    xsd:float
  }
```

exif:exposure

The exposure time used.

The <exif:exposure> element can appear as a child of <exif:tags>.

Examples

```
<exif:exposure>8.0E4</exif:exposure>
```

```
<exif:exposure>0.025</exif:exposure>
```

Schema

```
namespace exif = "http://schemas.google.com/photos/exif/2007"
start = exposure

exposure =
  element exif:exposure {
    xsd:float
  }
```

exif:flash

Boolean value indicating whether the flash was used.

The <exif:flash> element can appear as a child of <exif:tags>.

Examples

```
<exif:flash>true</exif:flash>
```

Schema

```
namespace exif = "http://schemas.google.com/photos/exif/2007"
start = flash

flash =
  element exif:flash {
    xsd:boolean
  }
```

exif:focallength

The focal length used.

The <exif:focallength> element can appear as a child of <exif:tags>.

Examples

```
<exif:focallength>23.7</exif:focallength>
```

Schema

```
namespace exif = "http://schemas.google.com/photos/exif/2007"
start = focallength
```

```
focallength =
  element exif:focallength {
    xsd:float
  }
```

exif:fstop

The fstop value used.

The <exif:fstop> element can appear as a child of <exif:tags>.

Examples

```
<exif:fstop>5.0</exif:fstop>
```

Schema

```
namespace exif = "http://schemas.google.com/photos/exif/2007"
start = fstop

fstop =
  element exif:fstop {
    xsd:float
  }
```

exif:imageUniqueID

The unique image ID for the photo.

The <exif:imageUniqueID> element can appear as a child of <exif:tags>.

Examples

```
<exif:imageUniqueID>5.0</exif:imageUniqueID>
```

Schema

```
namespace exif = "http://schemas.google.com/photos/exif/2007"
start = imageUniqueID

imageUniqueID =
  element exif:imageUniqueID{
    xsd:string
  }
```

exif:iso

The iso equivalent value used.

The <exif:iso> element can appear as a child of <exif:tags>.

Examples

```
<exif:iso>200</exif:iso>
```

Schema

```
namespace exif = "http://schemas.google.com/photos/exif/2007"
start = iso

iso =
  element exif:iso {
    xsd:unsignedInt
  }
```

exif:make

The make of the camera used.

The <exif:make> element can appear as a child of <exif:tags>.

Examples

```
<exif:make>Fictitious Camera Company</exif:make>
```

Schema

```
namespace exif = "http://schemas.google.com/photos/exif/2007"
start = make

make =
  element exif:make {
    xsd:string
  }
```

exif:model

The model of the camera used.

The <exif:model> element can appear as a child of <exif:tags>.

Examples

```
<exif:model>AMAZING-100D</exif:model>
```

Schema

```
namespace exif = "http://schemas.google.com/photos/exif/2007"
start = model

model =
  element exif:model {
    xsd:string
  }
```

exif:tags

The container for all exif elements.

The <exif:tags> element can appear as a child of a photo entry.

Examples

```
<exif:tags>
  <exif:fstop>4.0</exif:fstop>
  <exif:make>Fictitious Camera Company</exif:make>
  <exif:model>AMAZING-100D</exif:model>
  <exif:distance>0.0</exif:distance>
  <exif:exposure>0.016667</exif:exposure>
  <exif:flash>true</exif:flash>
  <exif:focallength>25.0</exif:focallength>
  <exif:iso>400</exif:iso>
  <exif:time>1180294337000</exif:time>
  <exif:imageUniqueID>246244a35234432f20efb832d8ea3c28</exif:imageUniqueID>
</exif:tags>
```

exif:time

The date/time the photo was taken, represented as the number of milliseconds since January 1st, 1970.

The `<exif:time>` element can appear as a child of `<exif:tags>`.

Examples

```
<exif:time>1180294337000</exif:time>
```

Schema

```
namespace exif = "http://schemas.google.com/photos/exif/2007"
start = time

time =
  element exif:time {
    xsd:unsignedInt
  }
```

Note: The value of this element should always be identical to the value of the `<gphoto:timestamp>`.

Picasa Web Albums extension (gphoto namespace) element reference

The Picasa Web Albums data API uses the `gphoto` namespace for extension elements. The schema URL for the `gphoto` namespace is `http://schemas.google.com/photos/2007`.

Most of the gphoto elements can appear only with a particular kind. A few elements can appear with multiple different kinds.

Elements appearing with multiple kinds

- gphoto:albumid
- gphoto:commentCount
- gphoto:commentingEnabled
- gphoto:id

Elements appearing only with the `user` kind

- gphoto:maxPhotosPerAlbum
- gphoto:nickname
- gphoto:quotacurrent
- gphoto:quotalimit
- gphoto:thumbnail

gphoto:user

Elements appearing only with the `album` kind

gphoto:access
gphoto:bytesUsed
gphoto:location
gphoto:name
gphoto:numphotos
gphoto:numphotosremaining

Elements appearing only with the `photo` kind

gphoto:checksum
gphoto:client
gphoto:height
gphoto:position
gphoto:rotation
gphoto:size
gphoto:timestamp
gphoto:version
gphoto:width

Elements appearing only with the `comment` kind

gphoto:photoid

Elements appearing only with the `tag` kind

gphoto:weight

Elements appearing with multiple kinds

The following elements appear in feeds of multiple kinds.

gphoto:albumid

The album's ID.

The <gphoto:albumid> element can appear as a child of <atom:entry> or <atom:feed>. The <gphoto:albumid> element is valid with the photo and comment kinds.

Example

```
<gphoto:albumid>5024425138</gphoto:albumid>
```

Schema

```
namespace gphoto = "http://schemas.google.com/photos/2007"
start = albumid

albumid =
  element gphoto:albumid {
    xsd:string
  }
```

gphoto:commentCount

The number of comments on an element.

The <gphoto:commentCount> element can appear as a child of <atom:entry> or <atom:feed>. The <gphoto:commentCount> element is valid with the album and photo kinds.

Example

```
<gphoto:commentCount>11</gphoto:commentCount>
```

Schema

```
namespace gphoto = "http://schemas.google.com/photos/2007"
start = commentCount

commentCount =
  element gphoto:commentCount {
    xsd:unsignedInt
  }
```

gphoto:commentingEnabled

Boolean value indicating whether commenting is enabled for the element.

The <gphoto:commentingEnabled> element can appear as a child of <atom:entry> or <atom:feed>.

The <gphoto:commentingEnabled> element is valid with the album and photo kinds.

Example

```
<gphoto:commentingEnabled>true</gphoto:commentingEnabled>
```

Schema

```
namespace gphoto = "http://schemas.google.com/photos/2007"
start = commentingEnabled

commentingEnabled =
  element gphoto:commentingEnabled {
    xsd:boolean
  }
```

gphoto:id

The ID of the current element.

The <gphoto:id> element can appear as a child of <atom:entry> or <atom:feed>. The <gphoto:id> element is valid with the album, photo, and comment kinds.

Example

```
<gphoto:id>512131187</gphoto:id>
```

Schema

```
namespace gphoto = "http://schemas.google.com/photos/2007"
start = id

id =
  element gphoto:id {
```

```
    xsd:string
  }
```

Elements appearing with the user kind

The following elements appear only in feeds of the `user` kind.

gphoto:maxPhotosPerAlbum

The maximum number of photos allowed in an album.

The <gphoto:maxPhotosPerAlbum> element can appear as a child of <atom:entry> or <atom:feed>. The <gphoto:maxPhotosPerAlbum> element is valid with the `user` kind.

Note: This element appears only if the authenticated user is the owner of the feed.

Example

```
<gphoto:maxPhotosPerAlbum>1000</gphoto:maxPhotosPerAlbum>
```

Schema

```
namespace gphoto = "http://schemas.google.com/photos/2007"
start = maxPhotosPerAlbum

maxPhotosPerAlbum =
  element gphoto:maxPhotosPerAlbum {
    xsd:unsignedInt
  }
```

gphoto:nickname

The user's nickname. This is a user-specified value that should be used when referring to the user by name.

The <gphoto:nickname> element can appear as a child of <atom:entry> or <atom:feed>. The <gphoto:nickname> element is valid with the `user` kind.

Example

```
<gphoto:nickname>Liz Bennett</gphoto:nickname>
```

Schema

```
namespace gphoto = "http://schemas.google.com/photos/2007"
start = nickname

nickname =
  element gphoto:nickname {
    xsd:string
  }
```

gphoto:quotacurrent

The number of bytes of storage currently in use by the user.

The <gphoto:quotacurrent> element can appear as a child of <atom:entry> or <atom:feed>. The <gphoto:quotacurrent> element is valid with the user kind.

Note: This element appears only if the authenticated user is the owner of the feed.

Example

```
<gphoto:quotacurrent>312459331</gphoto:quotacurrent>
```

Schema

```
namespace gphoto = "http://schemas.google.com/photos/2007"
start = quotacurrent

quotacurrent =
  element gphoto:quotacurrent {
    xsd:unsignedLong
  }
```

gphoto:quotalimit

The total amount of space allotted to the user.

The <gphoto:quotalimit> element can appear as a child of <atom:entry> or <atom:feed>. The <gphoto:quotalimit> element is valid with the user kind.

Note: This element appears only if the authenticated user is the owner of the feed.

Example

```
<gphoto:quotalimit>1385222385</gphoto:quotalimit>
```

Schema

```
namespace gphoto = "http://schemas.google.com/photos/2007"
start = quotalimit

quotalimit =
  element gphoto:quotalimit {
    xsd:unsignedLong
  }
```

gphoto:thumbnail

The URL of a thumbnail-sized portrait of the user. Not to be confused with the <media:thumbnail> element.

The <gphoto:thumbnail> element can appear as a child of <atom:entry> or <atom:feed>. The <gphoto:thumbnail> element is valid with the user kind.

Example

```
<gphoto:thumbnail>http://picasaweb.google.com/image/.../Hello.jpg</gphoto:thumb
```

Schema

```
namespace gphoto = "http://schemas.google.com/photos/2007"
start = thumbnail

thumbnail =
  element gphoto:thumbnail {
    xsd:string
  }
```

gphoto:user

The user's username. This is the name that is used in all feed URLs.

The <gphoto:user> element can appear as a child of <atom:entry> or <atom:feed>. The <gphoto:user> element is valid with the user kind.

Example

```
<gphoto:user>liz</gphoto:user>
```

Schema

```
namespace gphoto = "http://schemas.google.com/photos/2007"
start = user

user =
  element gphoto:user {
    xsd:string
  }
```

Elements appearing with the album kind

The following elements appear only in feeds of the album kind.

gphoto:access

The album's access level. In this document, access level is also referred to as "visibility." Valid values are public or private.

The <gphoto:access> element can appear as a child of <atom:entry> or <atom:feed>. The <gphoto:access> element is valid with the album kind.

Example

```
<gphoto:access>public</gphoto:access>
```

Schema

```
namespace gphoto = "http://schemas.google.com/photos/2007"
start = access

access =
  element gphoto:access {
    xsd:string
  }
```

gphoto:bytesUsed

The number of bytes of storage that this album uses.

The <gphoto:bytesUsed> element can appear as a child of <atom:entry> or <atom:feed>. The <gphoto:bytesUsed> element is valid with the album kind.

Note: This element appears only if the authenticated user is the owner of the feed.

Example

```
<gphoto:bytesUsed>11876307</gphoto:bytesUsed>
```

Schema

```
namespace gphoto = "http://schemas.google.com/photos/2007"
start = bytesUsed

bytesUsed =
  element gphoto:bytesUsed {
    xsd:unsignedInt
  }
```

gphoto:location

The user-specified location associated with the album.

The <gphoto:location> element can appear as a child of <atom:entry> or <atom:feed>. The <gphoto:location> element is valid with the album kind.

Example

```
<gphoto:location>Tokyo, Japan</gphoto:location>
```

Schema

```
namespace gphoto = "http://schemas.google.com/photos/2007"
start = location

location =
  element gphoto:location {
    xsd:string
  }
```

gphoto:name

The name of the album, which is the URL-usable name derived from the title. This is the name that should be used in all URLs involving the album.

The <gphoto:name> element can appear as a child of <atom:entry> or <atom:feed>. The <gphoto:name> element is valid with the album kind.

Example

```
<gphoto:name>mytrip</gphoto:name>
```

Schema

```
namespace gphoto = "http://schemas.google.com/photos/2007"
start = name

name =
  element gphoto:name {
    xsd:string
  }
```

gphoto:numphotos

The number of photos in the album.

The <gphoto:numphotos> element can appear as a child of <atom:entry> or <atom:feed>. The <gphoto:numphotos> element is valid with the album kind.

Example

```
<gphoto:numphotos>237</gphoto:numphotos>
```

Schema

```
namespace gphoto = "http://schemas.google.com/photos/2007"
start = numphotos

numphotos =
  element gphoto:numphotos {
    xsd:unsignedInt
  }
```

gphoto:numphotosremaining

The number of remaining photo uploads allowed in this album. This is equivalent to the user's maximum number of photos per album (<gphoto:maxPhotosPerAlbum>) minus the number of photos currently in the album (<gphoto:numphotos>).

The <gphoto:numphotosremaining> element can appear as a child of <atom:entry> or <atom:feed>. The <gphoto:numphotosremaining> element is valid with the album kind.

Note: This element appears only if the authenticated user is the owner of the feed.

Example

```
<gphoto:numphotosremaining>763</gphoto:numphotosremaining>
```

Schema

```
namespace gphoto = "http://schemas.google.com/photos/2007"
start = numphotosremaining

numphotosremaining =
  element gphoto:numphotosremaining {
    xsd:unsignedInt
  }
```

Elements appearing with the photo kind

The following elements appear only in feeds of the `photo` kind.

gphoto:checksum

The checksum on the photo. This optional field can be used by uploaders to associate a checksum with a photo to ease duplicate detection.

The `<gphoto:checksum>` element can appear as a child of `<atom:entry>` or `<atom:feed>`. The `<gphoto:checksum>` element is valid with the `photo` kind.

Example

```
<gphoto:checksum>23512309abbs298</gphoto:checksum>
```

Schema

```
namespace gphoto = "http://schemas.google.com/photos/2007"
start = checksum

checksum =
  element gphoto:checksum {
    xsd:string
  }
```

gphoto:client

The client application that created the photo. (Optional element.)

The `<gphoto:client>` element can appear as a child of `<atom:entry>` or `<atom:feed>`. The `<gphoto:client>` element is valid with the `photo` kind.

Example

```
<gphoto:client>Picasa2.6</gphoto:client>
```

Schema

```
namespace gphoto = "http://schemas.google.com/photos/2007"
start = client

client =
  element gphoto:client {
    xsd:string
  }
```

gphoto:height

The height of the photo in pixels.

The `<gphoto:height>` element can appear as a child of `<atom:entry>` or `<atom:feed>`. The `<gphoto:height>` element is valid with the `photo` kind.

Example

```
<gphoto:height>1200</gphoto:height>
```

Schema

```
namespace gphoto = "http://schemas.google.com/photos/2007"
start = height

height =
  element gphoto:height {
    xsd:unsignedLong
  }
```

gphoto:position

The ordinal position of the photo within the parent album.

The <gphoto:position> element can appear as a child of <atom:entry> or <atom:feed>. The <gphoto:position> element is valid with the photo kind.

Example

```
<gphoto:position>10</gphoto:position>
```

Schema

```
namespace gphoto = "http://schemas.google.com/photos/2007"
start = position

position =
  element gphoto:position {
    xsd:float
  }
```

gphoto:rotation

The rotation of the photo in degrees, used to change the rotation of the photo. Will only be shown if the rotation has not already been applied to the requested images.

The <gphoto:rotation> element can appear as a child of <atom:entry> or <atom:feed>. The <gphoto:rotation> element is valid with the photo kind.

Example

```
<gphoto:rotation>90</gphoto:rotation>
```

Schema

```
namespace gphoto = "http://schemas.google.com/photos/2007"
start = rotation

rotation =
  element gphoto:rotation {
    xsd:unsignedInt
  }
```

gphoto:size

The size of the photo in bytes.

The <gphoto:size> element can appear as a child of <atom:entry> or <atom:feed>. The <gphoto:size> element is valid with the photo kind.

Example

```
<gphoto:size>149351</gphoto:size>
```

Schema

```
namespace gphoto = "http://schemas.google.com/photos/2007"
start = size

size =
  element gphoto:size {
    xsd:unsignedLong
  }
```

gphoto:timestamp

The photo's timestamp, represented as the number of milliseconds since January 1st, 1970. Contains the date of the photo either set externally or retrieved from the Exif data.

The <gphoto:timestamp> element can appear as a child of <atom:entry> or <atom:feed>. The <gphoto:timestamp> element is valid with the photo kind.

Note: The value of this element should always be identical to the value of the <exif:time> value if this Exif data existed in the uploaded photo.

Example

```
<gphoto:timestamp>1168640584000</gphoto:timestamp>
```

Schema

```
namespace gphoto = "http://schemas.google.com/photos/2007"
start = timestamp

timestamp =
  element gphoto:timestamp {
    xsd:long
  }
```

gphoto:version

The version number of the photo. Version numbers are based on modification time, so they don't increment linearly. Note that if you try to update a photo using a version number other than the latest one, you may receive an error; for more information, see Optimistic concurrency (versioning) in the GData reference document.

The <gphoto:version> element can appear as a child of <atom:entry> or <atom:feed>. The <gphoto:version> element is valid with the photo kind.

Example

```
<gphoto:version>22838</gphoto:version>
```

Schema

```
namespace gphoto = "http://schemas.google.com/photos/2007"
start = version

version =
  element gphoto:version {
    xsd:string
  }
```

gphoto:width

The width of the photo in pixels.

The <gphoto:width> element can appear as a child of <atom:entry> or <atom:feed>. The <gphoto:width> element is valid with the photo kind.

Example

```
<gphoto:width>1600</gphoto:width>
```

Schema

```
namespace gphoto = "http://schemas.google.com/photos/2007"
start = width

width =
  element gphoto:width {
    xsd:unsignedLong
  }
```

Elements appearing with the comment kind

The following element appears only in feeds of the comment kind.

gphoto:photoid

The ID of the photo associated with the current comment.

The <gphoto:photoid> element can appear as a child of <atom:entry> or <atom:feed>. The <gphoto:photoid> element is valid with the comment kind.

Example

```
<gphoto:photoid>301521187</gphoto:photoid>
```

Schema

```
namespace gphoto = "http://schemas.google.com/photos/2007"
start = photoid
```

```
photoid =
  element gphoto:photoid {
    xsd:string
  }
```

Elements appearing with the tag kind

The following element appears only in feeds of the `tag` kind.

gphoto:weight

The weight of the tag. The weight is the number of times the tag appears in photos under the current element. The default weight is 1.

The `<gphoto:weight>` element can appear as a child of `<atom:entry>` or `<atom:feed>`. The `<gphoto:weight>` element is valid with the `tag` kind.

Example

```
<gphoto:weight>3</gphoto:weight>
```

Schema

```
namespace gphoto = "http://schemas.google.com/photos/2007"
start = weight

weight =
  element gphoto:weight {
    xsd:unsignedInt
  }
```