

# Google Data APIs for Cocoa Applications part 2; Picasa example

Greg Robbins  
David Oster  
*Google Mac Client Team*



## Picasa Slide Show



## In a nutshell:

```
@interface QuartzComposerPicasaImageController : NSWindowController {
    IBOutlet QCView *qcView_; // image inputs i1,i2,..i6
    NSMutableArray *thumbnailURLs_;
    NSMutableArray *thumbnailImages_;
    int imageIndex_;
    int frameIndex_;
}
- (void)awakeFromNib;
- (void)startFetchingURLs; // fetch an array of thumbnail URLs
- (void)finishFetchingURLsWithServiceTicket:(GDataServiceTicketBase *)ticket
  feedPhoto:(GDataFeedPhoto *)photoFeed;
- (void)onHeartbeat:(NSTimer*)theTimer;
- (void)startFetchingNextImage; // thumbnail URL to NSImage
- (void)finishFetchingImageWithFetcher:(GDataHTTPFetcher*)fetcher data:(NSData
*)data;
- (void)showNextImage; // store an image into the QCView
@end
```



## Begin Fetching those URLs

```
- (void)awakeFromNib {
    [self startFetchingURLs];
}

- (void)startFetchingURLs {
    GDataServiceGooglePicasaWeb *s = [[[GDataServiceGooglePicasaWeb
alloc] initWithServiceTicket:nil];
    NSURL *feedURL = [NSURL URLWithString:kGDataPicasaWebAllFeed];
    GDataQueryPicasaWeb *query = [GDataQueryPicasaWeb
picasaWebQueryWithFeedURL:feedURL];
    [query setKind:kGDataPicasaWebKindPhoto];
    [query setFullTextQueryString:@"red"];
    [s fetchPicasaWebQuery:(*)query delegate:self
  didFinishSelector:@selector(finishFetchingURLs:)];
}
```



## Got them, start our heartbeat

```
- (void)finishFetchingURLsWithServiceTicket:
  (GDataServiceTicketBase *)ticket
  feedPhoto:(GDataFeedPhoto *)photoFeed {

  NSArray *photos = [photoFeed entries];
  thumbnailURLs_ = [[photos arrayByLargestThumbnail]
retain];
  thumbnailImages_ = [[NSMutableArray alloc] init];

  [NSTimer scheduledTimerWithTimeInterval:1. target:self
selector:@selector(onHeartbeat:) userInfo:nil repeats:YES];
}
```



## The Heartbeat

```
- (void)onHeartbeat:(NSTimer*)theTimer {
  if ([thumbnailURLs_ count]) {
    [self startFetchingNextImage];
  }
  if ([thumbnailImages_ count]) {
    [self showNextImage];
  }
}
```



## Why Not: initWithContentsOfURL: ?

```
// pop URL stack and start fetching the image.
- (void)startFetchingNextImage {
    NSURL *url = [thumbnailURLs_ lastObject];
    UIImage *imag = [[[UIImage alloc]
initWithContentsOfURL:: url] autorelease];
    if (imag) {
        [thumbnailImages_ addObject:imag];
    }
}
```



## Why Not: initWithContentsOfURL: ?

```
// pop URL stack and start fetching the image.
- (void)startFetchingNextImage {
    NSURL *url = [thumbnailURLs_ lastObject];
    UIImage *imag = [[[UIImage alloc]
initWithContentsOfURL:: url] autorelease];
    if (imag) {
        [thumbnailImages_ addObject:imag];
    }
}
```

**because it makes the animation suck.**




## startFetchingNextImage

```
// pop URL stack and start fetching the image.
- (void)startFetchingNextImage {
    NSURLRequest *request = [NSURLRequest
requestWithURL:[thumbnailURLs_ lastObject]];

    [thumbnailURLs_ removeLastObject]; // pop the URL
stack

    GDataHTTPFetcher* myFetcher = [GDataHTTPFetcher
httpFetcherWithRequest:request];

    [myFetcher beginFetchWithDelegate:self
didFinishSelector:
@selector(finishFetchingImageWithFetcher:data:)];
```



## finishFetchingImage

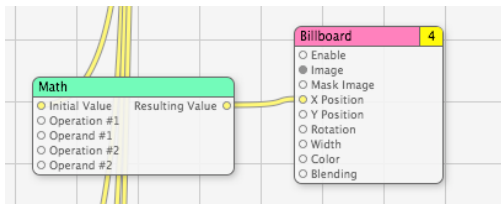
```
- (void)finishFetchingImageWithFetcher:
(GDataHTTPFetcher*)fetcher data:(NSData *)data {
    NSImage *imag = [[[NSImage alloc] initWithData: data]
autorelease];
    if (imag) {
        [thumbnailImages_ addObject:imag];
    }
}
```



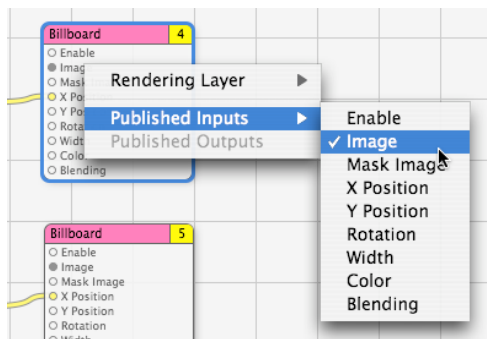


## Just 6 copies of:

add an offset to our oscillator, set as "x". "image" is an input to the whole composition.



## Publish an Input



## Where do I start?

### *Code and documentation*

<http://gdata-objectivec-client.googlecode.com/>

### *Discussion*

<http://groups.google.com/group/gdata-objectivec-client>



## End

GData Objective-C Client Library

for now, the source code from this presentation is at:

<http://turboZen.com/sourceCode/>

