PERL S3 EXAMPLES

CREATING A CONNECTION

This creates a connection so that you can interact with the server.

```
use Amazon::S3;
my $access_key = 'put your access key here!';
my $secret_key = 'put your secret key here!';

my $conn = Amazon::S3->new({
        aws_access_key_id     => $access_key,
        aws_secret_access_key => $secret_key,
        host                  => 'objects.dreamhost.com',
        secure                => 1,
        retry                 => 1,
});
```

LISTING OWNED BUCKETS

This gets a list of Amazon::S3::Bucket objects that you own. We'll also print out the bucket name and creation date of each bucket.

```
my @buckets = @{$conn->buckets->{buckets} || []};
foreach my $bucket (@buckets) {
        print $bucket->bucket . "\t" . $bucket->creation_date . "\n";
}
```

The output will look something like this:

```
mahbuckat1   2011-04-21T18:05:39.000Z
mahbuckat2   2011-04-21T18:05:48.000Z
mahbuckat3   2011-04-21T18:07:18.000Z
```

CREATING A BUCKET

This creates a new bucket called `my-new-bucket`

```
my $bucket = $conn->add_bucket({ bucket => 'my-new-bucket' });
```

LISTING A BUCKET'S CONTENT

This gets a list of hashes with info about each object in the bucket. We'll also print out each object's name, the file size, and last modified date.

```
my @keys = @{$bucket->list_all->{keys} || []};
foreach my $key (@keys) {
        print "$key->{key}\t$key->{size}\t$key->{last_modified}\n";
}
```

The output will look something like this:

```
myphoto1.jpg 251262  2011-08-08T21:35:48.000Z
myphoto2.jpg 262518  2011-08-08T21:38:01.000Z
```

DELETING A BUCKET

Note: The Bucket must be empty! Otherwise it won't work!

```
$conn->delete_bucket($bucket);
```

FORCED DELETE FOR NON-EMPTY BUCKETS

Attention: not available in the Amazon::S3 perl module

CREATING AN OBJECT

This creates a file `hello.txt` with the string `"Hello World!"`

```
$bucket->add_key(
        'hello.txt', 'Hello World!',
        { content_type => 'text/plain' },
);
```

CHANGE AN OBJECT'S ACL

This makes the object `hello.txt` to be publicly readable and `secret_plans.txt` to be private.

```
$bucket->set_acl({
        key       => 'hello.txt',
        acl_short => 'public-read',
});
$bucket->set_acl({
        key       => 'secret_plans.txt',
        acl_short => 'private',
});
```

DOWNLOAD AN OBJECT (TO A FILE)

This downloads the object `perl_poetry.pdf` and saves it in `/home/larry/documents/`

```
$bucket->get_key_filename('perl_poetry.pdf', undef,
        '/home/larry/documents/perl_poetry.pdf');
```

DELETE AN OBJECT

This deletes the object `goodbye.txt`

```
$bucket->delete_key('goodbye.txt');
```

GENERATE OBJECT DOWNLOAD URLS (SIGNED AND UNSIGNED)

This generates an unsigned download URL for `hello.txt`. This works because we made `hello.txt` public by setting the ACL above. Then this generates a signed download URL for `secret_plans.txt` that will work for 1 hour. Signed download URLs will work for the time period even if the object is private (when the time period is up, the URL will stop working).

Note: The Amazon::S3 module does not have a way to generate download URLs, so we are going to be using another module instead. Unfortunately, most modules for generating these URLs assume that you are using Amazon, so we have had

to go with using a more obscure module, Muck::FS::S3. This should be the same as Amazon's sample S3 perl module, but this sample module is not in CPAN. So, you can either use CPAN to install Muck::FS::S3, or install Amazon's sample S3 module manually. If you go the manual route, you can remove `Muck::FS::` from the example below.

```
use Muck::FS::S3::QueryStringAuthGenerator;
my $generator = Muck::FS::S3::QueryStringAuthGenerator->new(
        $access_key,
        $secret_key,
        0, # 0 means use 'http'. set this to 1 for 'https'
        'objects.dreamhost.com',
);

my $hello_url = $generator->make_bare_url($bucket->bucket, 'hello.txt');
print $hello_url . "\n";

$generator->expires_in(3600); # 1 hour = 3600 seconds
my $plans_url = $generator->get($bucket->bucket, 'secret_plans.txt');
print $plans_url . "\n";
```

The output will look something like this:

```
http://objects.dreamhost.com:80/my-bucket-name/hello.txt
http://objects.dreamhost.com:80/my-bucket-name/secret_plans.txt?Signature=XXXXXXXXXXXXXXXXXXX
```