

C# S3 EXAMPLES

CREATING A CONNECTION

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

```
using System;
using Amazon;
using Amazon.S3;
using Amazon.S3.Model;

string accessKey = "put your access key here!";
string secretKey = "put your secret key here!";

AmazonS3Config config = new AmazonS3Config();
config.ServiceURL = "objects.dreamhost.com";

AmazonS3 client = Amazon.AWSClientFactory.CreateAmazonS3Client(
    accessKey,
    secretKey,
    config
);
```

LISTING OWNED BUCKETS

This gets a list of Buckets that you own. This also prints out the bucket name and creation date of each bucket.

```
ListBucketResponse response = client.ListBuckets();
foreach (S3Bucket b in response.Buckets)
{
    Console.WriteLine("{0}\t{1}", b.BucketName, b.CreationDate);
}
```

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

```
PutBucketRequest request = new PutBucketRequest();
request.BucketName = "my-new-bucket";
client.PutBucket(request);
```

LISTING A BUCKET'S CONTENT

This gets a list of objects in the bucket. This also prints out each object's name, the file size, and last modified date.

```
ListObjectsRequest request = new ListObjectsRequest();
request.BucketName = "my-new-bucket";
ListObjectsResponse response = client.ListObjects(request);
foreach (S3Object o in response.S3Objects)
{
    Console.WriteLine("{0}\t{1}\t{2}", o.Key, o.Size, o.LastModified);
}
```

```
}
```

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!

```
DeleteBucketRequest request = new DeleteBucketRequest();
request.BucketName = "my-new-bucket";
client.DeleteBucket(request);
```

FORCED DELETE FOR NON-EMPTY BUCKETS

Attention: not available

CREATING AN OBJECT

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

```
PutObjectRequest request = new PutObjectRequest();
request.Bucket      = "my-new-bucket";
request.Key         = "hello.txt";
request.ContentType = "text/plain";
request.ContentBody = "Hello World!";
client.PutObject(request);
```

CHANGE AN OBJECT'S ACL

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

```
SetACLRequest request = new SetACLRequest();
request.BucketName = "my-new-bucket";
request.Key        = "hello.txt";
request.CannedACL  = S3CannedACL.PublicRead;
client.SetACL(request);

SetACLRequest request2 = new SetACLRequest();
request2.BucketName = "my-new-bucket";
request2.Key        = "secret_plans.txt";
request2.CannedACL  = S3CannedACL.Private;
client.SetACL(request2);
```

DOWNLOAD AN OBJECT (TO A FILE)

This downloads the object `perl_poetry.pdf` and saves it in `C:\Users\larry\Documents`

```
GetObjectRequest request = new GetObjectRequest();
request.BucketName = "my-new-bucket";
request.Key        = "perl_poetry.pdf";
GetObjectResponse response = client.GetObject(request);
response.WriteResponseStreamToFile("C:\\Users\\larry\\Documents\\perl_poetry.pdf");
```

DELETE AN OBJECT

This deletes the object `goodbye.txt`

```
DeleteObjectRequest request = new DeleteObjectRequest();
request.BucketName = "my-new-bucket";
request.Key        = "goodbye.txt";
client.DeleteObject(request);
```

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. This then 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 C# S3 Library does not have a method for generating unsigned URLs, so the following example only shows generating signed URLs.

```
GetPreSignedUrlRequest request = new GetPreSignedUrlRequest();
request.BucketName = "my-bucket-name";
request.Key        = "secret_plans.txt";
request.Expires    = DateTime.Now.AddHours(1);
request.Protocol   = Protocol.HTTP;
string url = client.GetPreSignedURL(request);
Console.WriteLine(url);
```

The output of this will look something like:

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