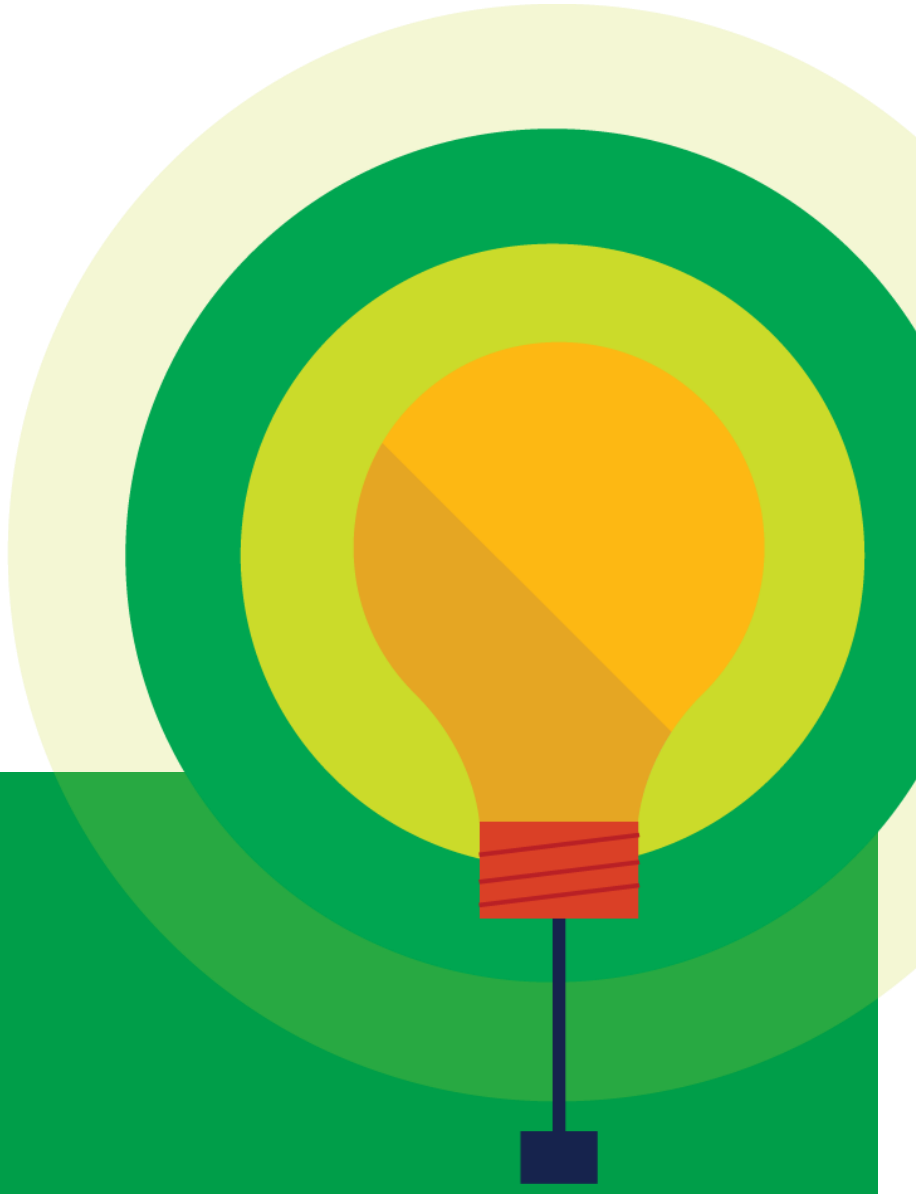
Amazon S3 to Azure Storage

Command line cross cloud copy tool



Microsoft

Abstract

The following is a 'how to' code sample and library that can be used for recursive copy storage structure from Amazon S3 to Azure storage. The library also can be used and integrated in any custom solution.

CONTENTS

Prerequisites

- Azure storage account
- Amazon S3 account
- Visual Studio 2015

Introduction

The S3 to Azure sample illustrates basic principles of copying from Amazon S3 bucket to Azure Storage blob container. The sample tool enables multithreaded asynchronous recursive copying of bucket folder structure and files matching the provided path prefix. A local connection is used to data transfer. The server side copying available on Azure is out of score for this sample. Also this sample is an illustration and is not intended to be a production ready application, but it can be used as is and extended for specific user scenarios.

Source Code

The sample uses Amazon AWS SDK for .NET to work with Amazon S3 service. Microsoft *WindowsAzure.Storage* client library is used to work with Microsoft Azure Blob Storage. A *CommandLineParser* library is a part of the project and is used for command line parsing.

The sample is developed using C# as a Visual Studio 2015 solution, which includes two projects:

- *BasicCloudCopy* is a library providing sample code of the copying engine. This library can be utilized in variety of custom applications.

- *S3ToAzure* is a command line tool built on the top of *BasicCloudCopy* library. It provides command line parameters parsing and console output.

BasicCloudCopy library

The sample library BasicCloudCopy provides the following concepts:

- *Copy job* is an asynchronous single file copying operation.
- *Copy job* factory creates copy jobs.
- *Copy job runner* uses factory to construct subsequent jobs, executes jobs in parallel according to provided limit and handles job competitions.

The library provides basic interfaces:

- *ICopyJob* is a basic interface for single thread copy job. This interface allows to start a copy operation using *CopyAsync*, check the progress using *ProgressChanged* event and provide a way how to determine does a retry operation possible after last failure using *RequestRetry*.
- *ICopyJobFactory* is a basic interface for the copy job factory. The factory is intended to construct copy job objects using *CreateNextCopyJobAsync*. This factory method creates a subsequent copy job object or returns null if no more objects available.

The library provides implementations of these interfaces for the case of local Amazon S3 to Azure copying using local memory streams.

- *S3ToAzureCopyJob* is an implementation of *ICopyJob* and provides copying an object from Amazon S3 bucket to Azure Storage container blob using local memory stream.
- *S3ToAzureCopyJobFactory* is an implementation of *ICopyJobFactory*. The factory creates a copy jobs for recursive copying of objects from Amazon S3 bucket to Azure Storage container.
- *CopyJobRunner* executes copy jobs and tracks competition, starts subsequent copy jobs according to maximum thread amount limit.

Most of Amazon/Azure related processing is performed in the *S3ToAzureCopyJob.CopyAsync*, *S3ToAzureCopyJob.UpdateSourceListAsync* methods and *S3ToAzureCopyJobFactory* constructor.

S3ToAzure application

The command line tool project is very simple:

- Options is a class describing available options to be handled by the *CommandLineParser* library. This class implements several interfaces used by S3ToAzureCopyJobFactory and *CopyJobRunner* as option sources.
- Program is a main application class.

Command line arguments are parsed by *Parser.Default.ParseArguments*. If options are lexically correct and constraints are passed, then a *CopyAsyncAndReturnExitCode* wrapper is executed. It starts and wait for asynchronous method *CopyAsync* and handles exceptions. The *CopyAsync* constructs the *S3ToAzureCopyJobFactory* and *CopyJobRunner*, connects a progress handler with console output and starts the copy job runner.

The following error screen will be shown if the required command line parameters are not provided:

```
e:\temp\Release>S3ToAzure.exe
S3ToAzure 0.1.0.0
Copyright (c) Microsoft 2016

ERROR(S):
  Required option 'i, keyid' is missing.
  Required option 's, secret' is missing.
  Required option 'r, region' is missing.
  Required option 'b, bucket' is missing.
  Required option 'a, account' is missing.
  Required option 'k, key' is missing.
  Required option 'c, container' is missing.

  -i, --keyid        Required. Amazon access key id

  -s, --secret       Required. Amazon secret access key

  -r, --region       Required. Amazon region like us-west-1, eu-central-1 and
                     etc.

  -b, --bucket       Required. Source bucket name

  -p, --prefix       (Default: /) Source path prefix. Recursive for matched
                     folders

  -a, --account      Required. Azure Storage account name

  -k, --key          Required. Azure Storage account access Key

  -c, --container    Required. Target container name

  -f, --folder       (Default: /) Target folder

  -l, --level        (Default: 0) Level of public access that is allowed on the
                     target container. 0 - Off, 1 - Container, 2 - Blob. Used
                     in case the container does not exist

  -R, --retries      (Default: 1) Maximum retry count

  -T, --threads      (Default: 4) Copying threads amount

  --help             Display this help screen.

  --version          Display version information.
```

If parameters are correct the application writes a copying progress output:

```
e:\temp\Release>S3ToAzure.exe -i [redacted] -s [redacted] -r eu-central-1 -b s3-azur
e-bucket1 -p /S3ToAzureSample -a sampletarget -k [redacted]
[redacted] -c target -f /Samples
Samples\S3ToAzureSample/BasicCloudCopy/BasicCloudCopy.csproj : 100%
Samples\S3ToAzureSample/BasicCloudCopy/ForwardOnlyReadStream.cs : 100%
Samples\S3ToAzureSample/BasicCloudCopy/CopyJobRunner.cs : 100%
Samples\S3ToAzureSample/BasicCloudCopy/ICopyJob.cs : 100%
Samples\S3ToAzureSample/BasicCloudCopy/ICopyJobFactory.cs : 100%
Samples\S3ToAzureSample/BasicCloudCopy/Interfaces.cs : 100%
Samples\S3ToAzureSample/BasicCloudCopy/Properties/AssemblyInfo.cs : 100%
Samples\S3ToAzureSample/BasicCloudCopy/S3ToAzureCopyJob.cs : 100%
Samples\S3ToAzureSample/BasicCloudCopy/S3ToAzureCopyJobFactory.cs : 100%
Samples\S3ToAzureSample/BasicCloudCopy/app.config : 100%
Samples\S3ToAzureSample/BasicCloudCopy/packages.config : 100%
Samples\S3ToAzureSample/S3ToAzure/App.config : 100%
Samples\S3ToAzureSample/S3ToAzure/Properties/AssemblyInfo.cs : 100%
Samples\S3ToAzureSample/S3ToAzure/Program.cs : 100%
Samples\S3ToAzureSample/S3ToAzure/Options.cs : 100%
Samples\S3ToAzureSample/S3ToAzure/S3ToAzure.csproj : 100%
Samples\S3ToAzureSample/S3ToAzure/packages.config : 100%
Samples\S3ToAzureSample/S3ToAzureSample.sln : 100%

DONE
```