AWS Import/Export

Developer Guide

API Version 2010-06-03

AWS Import/Export: Developer Guide

Table of Contents

Welcome

Topics

This guide contains information for you to create import and export jobs quickly, and includes guidelines for shipping your storage devices to AWS. This guide also contains conceptual and reference information about manifest options, which allow you to configure your import and export jobs to meet your specific needs.

What Is AWS Import/Export?

Topics

Overview of AWS Import/Export

AWS Import/Export accelerates transferring large amounts of data between the AWS cloud and portable storage devices that you mail to us. AWS transfers data directly onto and off of your storage devices using Amazon's high-speed internal network. Your data load typically begins the next business day after your storage device arrives at AWS. After the data export or import completes, we return your storage device. For large data sets, AWS Import/Export can be significantly faster than Internet transfer and more cost effective than upgrading your connectivity.

AWS Import/Export supports:

- Import to Amazon S3
- Export from Amazon S3
- Import to Amazon EBS

- Import to Amazon Glacier

AWS Import/Export does not currently support export from Amazon EBS or Amazon Glacier.

Important
This guide assumes that you are an Amazon S3 user, an Amazon Glacier user, or an Amazon Elastic Compute Cloud (EC2) user for Amazon EBS data. If that is not the case, go to the Amazon S3 Getting Started Guide, Amazon Glacier Developer Guide, or Amazon EC2 Getting Started Guide for Amazon EBS data.

Advantages to AWS Import/Export

Using AWS Import/Export provides the following major advantages:

- **Migrate Large Amounts of Data into the AWS Cloud—**Mail us one or more storage devices with any amount of data. We load your data to the AWS cloud and return your storage device.
- **Content Distribution—**We can export data from the AWS cloud onto one or more of your storage devices for delivery to your customers.
- **Direct Data Interchange—**If you regularly receive content on portable storage devices from your business associates, you can have them send it directly to AWS for import into Amazon EBS, Amazon S3, or Amazon Glacier.
- **Off-Site Backup—**Send full or incremental backups to Amazon S3 for reliable and redundant off-site storage.
- **Disaster Recovery—**In the event you need to quickly retrieve a large backup stored in Amazon S3, use AWS Import/Export to transfer the data to a portable storage device and deliver it to your site.

Available Tools, Libraries, and Interfaces

AWS Import/Export provides a command line tool for creating import and export jobs, without writing any code. For applications requiring programming interface, AWS SDK provides Java, .NET and PHP libraries to create and manage import and export jobs. You can also optionally use the REST interface when working with AWS Import/Export.

There are also third party tools that support AWS Import/Export. For more information, go to http://aws.amazon.com/importexport/tools.

AWS Import/Export Jobs

Topics

This section describes AWS Import/Export jobs and job types.

Job

You create an AWS Import/Export job in two steps. First, you submit a job request to AWS. Each AWS Import/Export job corresponds to exactly one storage device. You submit your job request by using the AWS Import/Export command line tool, which requires no programming. If your application requires a programmatic interface, you can use the AWS SDK or the REST API. Second, you send your storage

device to AWS. We use our high bandwidth network to import your data into AWS or export data from AWS to your storage device, then return your device.

You must submit a separate job request for each device.

> **Note**
> You can send multiple devices in the same shipment. If you do, however, there are specific guidelines and limitations that govern what devices you can ship and how your devices must be packaged. If your shipment is not prepared and packed correctly, AWS Import/Export cannot process your jobs. Regardless of how many devices you ship at one time, you must submit a separate job request for each device. For complete details about packaging requirements when shipping multiple devices, see Shipping Multiple Devices (p. 68).

Job Type

For each job type, the following table summarizes the source of the data, either your device or an Amazon S3 bucket, where the data will be moved to, and the result you can expect when the job is complete.

Import to Amazon S3		
Source	Target	Result
• Files on a device file system. • One device per import job. • We recommend encrypting your data using TrueCrypt open-source encryption software. See Using TrueCrypt Encryption (p. 66).	• Objects in an existing Amazon S3 bucket. • One bucket per import job. • If you encrypted your data using TrueCrypt, AWS decrypts the data before performing the import.	• One object for each file. • Files and directories can be excluded or renamed using manifest file options. • AWS erases your device after every import job prior to shipping.
Export from Amazon S3		
Source	Target	Result
• Objects in one or more Amazon S3 buckets. • You can export from multiple buckets for each export job. • You must provide a password that AWS will use to encrypt your data.	• Files on your storage device. • One device per export job. • AWS formats your device. • AWS copies your data to an encrypted file container on your device. See Using TrueCrypt Encryption (p. 66).	• One file for each object. • You can export from more than one bucket and specify which files to export using manifest file options. • You will need use TrueCrypt to access the encrypted files.
Import to Amazon Glacier		
Source	Target	Result
• Entire device. • One device per import job. • We recommend encrypting your device.	• One archive in an existing Amazon Glacier vault. • One archive per import job. • AWS does not decrypt your device.	• A device image stored as a single archive. • The device's file system is never mounted. • AWS erases your device after every import job prior to shipping.

Import to Amazon EBS (Device capacity less than or equal to 1 TB)		
Important An import to Amazon EBS will have different results depending on whether the capacity of your storage device is less than or equal to 1 TB or greater than 1 TB. The maximum size of an Amazon EBS snapshot is 1 TB, so if the device image is larger than 1 TB, the image is chunked and stored on Amazon S3. The target location is determined based on the total capacity of the device, not the amount of data on the device.		
Source	Target	Result
• Entire device (device capacity less than or equal to 1 TB). • One device per import job. • We recommend encrypting your device.	• One Amazon EBS snapshot. • One snapshot per import job. • AWS does not decrypt your device.	• A device image stored as a single snapshot. • The device's file system is never mounted. • If the device was encrypted, the device image is encrypted. • AWS erases your device after every import job prior to shipping.
Import to Amazon EBS (Device capacity greater than 1 TB)		
Source	Target	Result
• Entire device (device capacity greater than 1 TB). • One device per import job. • We recommend encrypting your data.	• Multiple objects in an existing Amazon S3 bucket. • One device image, multiple Amazon S3 objects per import job. • AWS does not decrypt your data.	• A device image chunked as a series of 1 TB snapshots that are stored as objects in the Amazon S3 bucket specified in the `logBucket` manifest file option. • The device's file system is never mounted. • If the device was encrypted, the device image is encrypted. • AWS erases your device after every import job prior to shipping.

Related Concepts

- To learn more about Amazon S3, see Introduction to Amazon S3.
- To learn more about Amazon EBS, go to Amazon Elastic Block Store.
- To learn more about Amazon Glacier, go to Amazon Glacier.
- If you need assistance with your AWS Import/Export job, contact AWS Support Center.

Pricing

You are not charged for the time required to erase your device following an import, and you are not charged for the time required to decrypt or encrypt your device. You can approximate your costs by using the AWS Import/Export Calculator.

Return Shipping

Return shipping charges are dependent on the location of your AWS storage and your return shipping destination. Any applicable return shipping expenses will be charged once your package is ready for shipment to your return shipping address. You can estimate your return shipping costs by using the AWS Import/Export Calculator.

Next Step

Your next step is to get set up and to create your first job. For more information, see Getting Started (p. 6).

Getting Started

The Getting Started section provides step-by-step instructions to set up and create an AWS Import/Export job.

Before You Use AWS Import/Export

AWS Import/Export supports data upload and download from Amazon S3 buckets, and data upload to Amazon Elastic Block Store (Amazon EBS) snapshots and Amazon Glacier vaults. The AWS Import/Export Getting Started steps assume you already use Amazon S3, Amazon EBS, or Amazon Glacier.

To upload or download data from Amazon S3, you need to have an existing Amazon S3 account and be familiar with Amazon S3. For more information, see the Amazon S3 Getting Started Guide.

To upload data to an Amazon EBS snapshot for Amazon EC2, you need to have an existing Amazon EC2 instance to associate with the data, and an Amazon S3 bucket to store AWS Import/Export log files. For more information about Amazon EC2 and Amazon EBS, see Amazon Elastic Block Store.

To upload data to an Amazon Glacier vault, you need to have an Amazon Glacier vault to associate with the data, and an Amazon S3 bucket to store AWS Import/Export log files. For more information about Amazon Glacier, see Amazon Glacier.

Next Sign Up for AWS (p. 7).

Sign Up for AWS

To use AWS Import/Export, you need an AWS account. If you already have an AWS account, you are automatically signed up for the AWS Import/Export service. If you do not have an AWS account, go to the Amazon Web Services home page to sign up.

Next: Download the AWS Import/Export Web Service Tool (p. 7).

Download the AWS Import/Export Web Service Tool

You can create an AWS Import/Export job using a command line tool, which requires no programming. If your application requires programmatic access, you can use the AWS SDK for Java or the REST API to send requests to AWS. This getting started uses the AWS Import/Export Web Service Tool, an open-source command line tool for creating and managing your import and export jobs.

To download the AWS Import/Export Web Service Tool

1. Click http://awsimportexport.s3.amazonaws.com/importexport-webservice-tool.zip.
2. Extract the zipped content to a directory on your computer.

Your next task is to add your AWS credentials to the *`AWSCredentials.properties`* file.

Next: Save Your Credentials to a File (p. 8).

Save Your Credentials to a File

AWS Import/Export uses AWS Identity and Access Management (IAM) to control which users in your AWS account can create, modify, view, or cancel AWS Import/Export jobs. We strongly recommend that you create IAM users and use the credentials for those IAM users to create and manage AWS Import/Export jobs. For more information, see Controlling Access to AWS Import/Export Jobs (p. 26).

> **Important**
> The IAM user that creates an AWS Import/Export job must also have read and write access to the Amazon S3 bucket associated with the job and, for import to Amazon Glacier, the Amazon Glacier vault. That user must have access to the necessary resources until the AWS Import/Export job is completed. For listings of the specific permissions required, see Granting Access to Related AWS Resources (p. 28).

In this step, you will add your AWS IAM user credentials to the *`AWSCredentials.properties`* file that the AWS Import/Export Web Service Tool provides for you. The following procedure takes you through the necessary steps.

To save your access credentials to a file

1. Locate your IAM access credentials. If you do not have IAM access credentials, you will need to create them or have someone who has permission to administer IAM create them for you. For more information, go to Administering Access Keys for IAM Users.

   > **Note**
   > When you create an access key, it's the only opportunity you have to view and save the secret access key for that user. Make sure you download the access key and save it in a secure location. You will not be able to view the secret access key using the AWS Management Console again later; if it becomes unavailable, you will have to create a new access key.

2. Go to the directory where you installed the AWS Import/Export Web Service tool.
3. Open the *`AWSCredentials.properties`* file from this directory using any text editor.
4. Update the file by providing your *`accessKeyId`* and *`secretKey`*. The following is an example of a credentials file.

```
# Fill in your AWS Access Key ID and Secret Access Key
# http://aws.amazon.com/security-credentials
accessKeyId:<Your Access Key ID>
secretKey:<Your Secret Key>
```

5. Save the file.

   > **Important**
   > To protect your credentials, be sure to save the file to a secure location. This Getting Started assumes that you saved your credentials in the same directory as the Import/Export Web

Service Tool, so that directory should be in a secure location. If your credentials file is in a different location, you can modify your command line to reference that location.

Now, you are ready to create a sample import or export job.

Next:

Create an Import Job (p. 9).

Create Your First Amazon S3 Export Job (p. 21).

Create an Import Job

You can create an import job to move data from your hardware storage device to an Amazon S3 bucket, an Amazon EBS snapshot, or an Amazon Glacier vault.

The following sections present the steps to accomplish the tasks for each type of import.

- To upload data to an Amazon S3 bucket: Create Your First Amazon S3 Import Job (p. 9).
- To upload data to an Amazon EBS snapshot: Create Your First Amazon EBS Import Job (p. 13).
- To upload data to an Amazon Glacier vault: Create Your First Amazon Glacier Import Job (p. 17).

Create Your First Amazon S3 Import Job

With an Amazon S3 import, AWS uploads individual files from your device to objects in an Amazon S3 bucket. If you encrypt your data using TrueCrypt, the data is decrypted using the password you provide. One device is loaded to one bucket. Your device's file system is mounted on an AWS Import/Export data loading station and each file is uploaded as an object in your Amazon S3 bucket. You can optionally manage directories and filenames as part of the load process.

After the import operation completes, AWS Import/Export erases your device before shipping it.

You must submit a separate job request for each device.

> **Note**
> You can send multiple devices in the same shipment. If you do, however, there are specific guidelines and limitations that govern what devices you can ship and how your devices must be packaged. If your shipment is not prepared and packed correctly, AWS Import/Export cannot process your jobs. Regardless of how many devices you ship at one time, you must submit a separate job request for each device. For complete details about packaging requirements when shipping multiple devices, see Shipping Multiple Devices (p. 68).

For more information, see Guidelines and Limitations (p. 130)

You must submit a separate job request for each device.

To create an Amazon S3 import job

1. Create a manifest file.
2. Prepare your device for shipping.
3. Send a `CreateJob` request.

   AWS Import/Export sends a response with a `SIGNATURE` file and job information.
4. Copy the `SIGNATURE` file to your device.
5. Ship your device.

Tip
You can create an Amazon S3 bucket using AWS Management Console. For more information, go to http://docs.aws.amazon.com/AmazonS3/latest/UG/CreatingaBucket.html.

These steps assume that you have signed up for an AWS account and created your `AWScredentials.properties` file as described in the earlier tasks.

Create a Manifest File

The manifest file is a YAML-formatted text file that tells AWS how to handle your job. It consists of a set of name-value pairs that supply required information, such as your device ID, TrueCrypt password, and return address. For more information about YAML, go to http://yaml.org.

The AWS Import/Export Web Service Tool provides three examples of manifest files that you can modify for your import and export jobs.

1. Go to the `Examples` folder in the folder where you installed the AWS Import/Export Web Service Tool.
2. Open the `S3-import-manifest.txt` file in a text editor.

```
manifestVersion:2.0
generator:Text editor
bucket:[Enter import bucket]
deviceId:[Enter device serial number]
eraseDevice:yes
notificationEmail: [Email addresses, semicolon separated]
trueCryptPassword: [Optional – password]
acl:private
serviceLevel:standard
returnAddress:
    name:[Your name]
    company:[Optional - your company]
    street1:[Your street]
    street2:[Optional - additional street info]
    street3:[Optional - additional street info]
    city:[Your city]
    stateOrProvince:[Required for USA and Canada. Your state or province.]
    postalCode:[Your postal code]
    country:[Your country]
    phoneNumber: [Contact number]
```

Important
When shipping devices internationally except within the European Union you must include the customs option in the manifest. For more information about the customs manifest option, see Customs Manifest File Option (p. 79).

3. In the file, replace all text in brackets with the appropriate values. Delete any unused optional lines.
4. Provide the name of the existing Amazon S3 bucket where you want to upload your data. AWS Import/Export loads the data to a single bucket. For the *bucket* option provide only the name of your bucket, for example *s3examplebucket*.
5. For the `eraseDevice` field, specify `yes` to acknowledge that AWS will erase your device following the import operation before shipping it. If the `eraseDevice` field is missing or if the value is not `yes`, the job request will fail.
6. For the `notificationEmail` field, enter one or more email addresses, separated by semicolons, so we can contact you if there are any problems. If the `notificationEmail` field is missing or empty, the job request will fail.
7. Save the file as `MyS3ImportManifest.txt` in the same folder as your `AWSCredentials.properties` file.

For more information about manifest file options, see Creating Import Manifests (p. 37).

Prepare Your Device for Import

Next, you prepare your device for import.

1. Optionally encrypt your data using TrueCrypt.

   For added security, we strongly recommend that you encrypt your data. You can encrypt the drive or create an encrypted file container on your device. For more information, see Using TrueCrypt Encryption (p. 66).
2. Verify that all file names are valid. File names must use standard ASCII or UTF-8 character encoding. Any file with a name that is not a valid ASCII or UTF-8 string is not imported.
3. Copy your data to the device. Do not ship the only copy of your data. AWS will erase your device, even if we cannot perform an import.

If your device is not properly prepared for import, AWS might erase the device and return it without processing the request. If we are unable to erase the data on the device, we will schedule it for destruction and our support team will contact you using the email address specified in the manifest file. For more information, see *Shipping Your Storage Device* (p. 63).

Send a CreateJob Request

Now that you have your credentials file and manifest file in place, you send a `CreateJob` request to AWS Import/Export. You submit a separate `CreateJob` request for each device.

1. Open a command prompt (or, on a Mac, use the Terminal application), and change to the directory where you unzipped the AWS Import/Export tool.
2. Enter the following `CreateJob` request on the command line.

```
CmdPrompt>java -jar lib/AWSImportExportWebServiceTool-1.0.jar CreateJob Import
 MyS3ImportManifest.txt .
```

   The `CreateJob` request includes a directory name for where you want the `SIGNATURE` file saved. In this example, the closing period (.) at the end of the command refers to the current directory. Either include the period, as shown above, or replace it with a directory name.

If there is an error, AWS Import/Export returns an error message.

If there are no errors in the request, AWS Import/Export returns a *JOB CREATED* response. The following is a sample response.

```
JOB CREATED
 JobId: ABCDE
 JobType: Import
 ****************************************
 * AwsShippingAddress *
 ****************************************
 AWS Import/Export C/O AnyCompany
 JOBID ABCDE
 123 Any Street
 Anytown, VA 22153
 ****************************************
 * SignatureFileContents *
 ****************************************
 version:2.0
 signingMethod:HmacSHA1
 jobId:ABCDE-VALIDATE-ONLY
 signature:cbfdUuhhmauYS+ABCl5R9heDK/V=
 Writing SignatureFileContents to cmdPrompt\.\SIGNATURE
```

Your job expires after 30 days. If you do not send a device, there is no charge.

Copy the Signature File

In response to your `CreateJob` request, AWS writes a file named `SIGNATURE` to your computer. The `SIGNATURE` file contains a signature, which is an encrypted string that allows us to authenticate your device and match it to your job request.

Locate the `SIGNATURE` file in the directory you specified in your job request and copy it to the root directory of your storage device. The file name must be named `SIGNATURE` and it must be in the device's root directory.

Each device you send must include the unique `SIGNATURE` file for that device and that JOBID. AWS Import/Export validates the `SIGNATURE` file on your storage device before starting the data load. If the `SIGNATURE` file is missing or invalid (for example, if it is associated with a different job request), AWS Import/Export will not perform the data load and we will return your storage device.

Ship Your Device

It is essential that you prepare your device properly before shipping.

> **Note**
> You can send multiple devices in the same shipment. If you do, however, there are specific guidelines and limitations that govern what devices you can ship and how your devices must be packaged. If your shipment is not prepared and packed correctly, AWS Import/Export cannot process your jobs. Regardless of how many devices you ship at one time, you must submit a separate job request for each device. For complete details about packaging requirements when shipping multiple devices, see Shipping Multiple Devices (p. 68).

1. Pack the *power supply*, *power cable*, and *interface cables* with your storage device. Without these, we can't transfer your data and will return your device.
2. Fill out the packing slip and include the packing slip with your device and cables.

   Download the packing slip from http://s3.amazonaws.com/awsimportexport/AWS_Import_Export_Packing_Slip.pdf.

3. Send your package to the address provided in the `CreateJob` response.

   For detailed instructions for preparing and shipping your device, see Shipping Your Storage Device (p. 63).

   Important
   Your job expires after 30 days. If you do not send a device, there is no charge. You are billed only after AWS Import/Export receives your device. If we receive your package after your job expires, we will return your device. You will only be charged for the shipping fees, if any.

Now, you have explored the getting started steps of AWS Import/Export. To learn more about the product, see Where Do I Go from Here? (p. 25).

Create Your First Amazon EBS Import Job

With an Amazon EBS import job, AWS creates a virtual copy of your device as a single image. The device's file system is never mounted. If your device is encrypted, then the device image is also encrypted. AWS does not decrypt your device before importing the device image.

The capacity of your device determines how the device image is stored. The maximum capacity of an Amazon EBS volume is 1 TB. If the capacity of the device is greater than 1 TB, we cannot import the disk directly into an EBS snapshot.

Note
The maximum capacity of a device is independent of the amount of data stored on the device.

If your device capacity is 1 TB or less, we store the device image directly to an EBS snapshot.

If your device capacity is greater than 1 TB, we store the device image as a series of objects in an Amazon S3 bucket. You can then create a RAID of Amazon EBS volumes using software such as Logical Volume Manager, and copy the image from Amazon S3 to the new volume.

After the import operation completes, AWS Import/Export erases your device before shipping it.

You must submit a separate job request for each device.

Note
You can send multiple devices in the same shipment. If you do, however, there are specific guidelines and limitations that govern what devices you can ship and how your devices must be packaged. If your shipment is not prepared and packed correctly, AWS Import/Export cannot process your jobs. Regardless of how many devices you ship at one time, you must submit a separate job request for each device. For complete details about packaging requirements when shipping multiple devices, see Shipping Multiple Devices (p. 68).

For more information, see Guidelines and Limitations (p. 130)

To create an Amazon EBS import job

1. Create a manifest file.
2. Prepare your device for shipping.
3. Send a `CreateJob` request.

   AWS Import/Export sends a response and a PDF file with a barcode and job information.
4. Attach the barcode to your device.
5. Ship your device.

In your request, you must identify an existing Amazon S3 bucket for logging. You must have write permissions on the bucket.

Tip
You can create a bucket using AWS S3 Console. For more information, go to http://docs.aws.amazon.com/AmazonS3/latest/UG/CreatingaBucket.html.

These steps assume that you have signed up for an AWS account and created your `AWSCredentials.properties` file as described in the earlier tasks.

Create a Manifest File

The manifest file is a YAML-formatted text file that tells AWS how to handle your job. It consists of a set of name-value pairs that supply required information, such as your device ID, log bucket, and return address. For more information about YAML, go to http://yaml.org.

The AWS Import/Export Web Service Tool provides three examples of manifest files that you can modify for your import and export jobs.

1. Go to the `Examples` folder in the folder where you installed the AWS Import/Export Web Service Tool.
2. Open the `ebs-import-manifest.txt` file in a text editor.

```
manifestVersion: 3.0
deviceId:[Enter device serial number]
logPrefix: [Optional - a prefix for your log bucket]
logBucket:[Your log bucket]
generator: AWS Import/Export Command Line Tool
eraseDevice:yes
notificationEmail: [Email addresses, semicolon separated]
returnAddress:
    name:[Your name]
    company:[Optional - your company]
    street1:[Your street]
    street2:[Optional - additional street info]
    street3:[Optional - additional street info]
    city:[Your city]
    stateOrProvince:[Required for USA and Canada. Your state or province.]
    postalCode:[Your postal code]
    country:[Your country]
    phoneNumber:[Contact number]
operations:
    - destination: ebs-snapshot
      source: device
      region: us-east-1 | us-west-1 | us-west-2 | eu-west-1 | ap-southeast-1

      deviceCapacityGreaterThan1TB: [yes|no]
```

Important
When shipping devices internationally except within the European Union you must include the customs option in the manifest. For more information about the customs-related manifest options, see Customs Manifest File Option (p. 79).

3. In the file, replace all text in brackets with the appropriate values. Delete any unused optional lines.
4. Provide the name of the existing Amazon S3 bucket for logging. For the *`logBucket`* option, provide only the name of your bucket, for example `s3examplebucket`.

If your device capacity is greater than 1 TB, your files will be stored in the Amazon S3 bucket specified in the *logBucket* option.

5. For the `eraseDevice` field, specify `yes` to acknowledge that AWS will erase your device following the import operation before shipping it. If the `eraseDevice` field is missing or if the value is not `yes`, the job request will fail.
6. For the `notificationEmail` field, enter one or more email addresses, separated by semicolons, so we can contact you if there are any problems. If the `notificationEmail` field is missing or empty, the job request will fail.
7. The *destination* subfield in the *operations* option specifies that the data is imported into an Amazon EBS snapshot. Change the *region* parameter to specify the correct region. For more information about regions go to the AWS Import/Export Calculator. For more information about other *operations* options, see Creating Import Manifests (p. 37).
8. If your device capacity is 1 TB or less, specify `no` for the *deviceCapacityGreaterThan1TB* option. If your device capacity is greater than 1 TB, specify `yes` for the *deviceCapacityGreaterThan1TB* option.

   If the *deviceCapacityGreaterThan1TB* option is not set correctly for your device, AWS Import/Export will not perform the import and we will return your storage device.
9. Save the file as `MyEBSImportManifest.txt` in the same folder as your `AWSCredentials.properties` file.

For more information about manifest file options, see Manifest File Options Reference (p. 76).

Prepare Your Device for Import

Next, you need to prepare your device for import.

1. Optionally encrypt your data. For added security, we strongly recommend that you encrypt your data. AWS does not decrypt your data before importing to Amazon EBS or Amazon Glacier. The entire device image is imported.
2. Copy your data to the device. Do not ship the only copy of your data. AWS will erase your device, even if we cannot perform an import.

Send a CreateJob Request

Now that you have your credentials file and manifest file in place, you send a `CreateJob` request to AWS Import/Export.

You send a separate CreateJob request for each device.

1. Open a command prompt (or, on a Mac, use the Terminal application), and change to the directory where you unzipped the AWS Import/Export tool.
2. Enter the following `CreateJob` request on the command line.

```
CmdPrompt>java -jar lib/AWSImportExportWebServiceTool-1.0.jar CreateJob Import
 MyEBSImportManifest.txt .
```

   The `CreateJob` request includes a directory name for where you want the shipping instructions PDF file saved. In this example, the closing period (.) at the end of the command refers to the current directory. Either include the period, as shown above, or replace it with a directory name.

If there is an error, AWS Import/Export returns an error message.

If there are no errors in the request, AWS Import/Export returns a *JOB CREATED* response.The following is a sample response.

```
JOB CREATED

JobId: ABCDE
JobType: Import

Shipping instructions saved to C:\DirectoryName\.\shipping-instructions-ABCDE.pdf

SHIPPING INSTRUCTIONS:

1. Open the Shipping Instructions file with a PDF reader.
2. Print out the Shipping Instructions which includes the AWS Shipping Address

and Signature barcode.
3. Follow the directions in the PDF file, which includes cutting out and securely

 attaching the Signature barcode to your device.

*IMPORTANT* - Failure to attach the Signature barcode to your device will prevent

AWS from loading your data.  This barcode can only be used with 1 device.
```

Your job expires after 30 days. If you do not send a device, there is no charge.

Attach the Barcode

After the `CreateJob` request completes successfully, AWS Import/Export places the shipping instructions and a barcode as a PDF file in the directory you specified in the `CreateJob` request. AWS Import/Export also places a copy of the PDF file in the Amazon S3 bucket that you specified with the logBucket manifest option. The name of the PDF file is `shipping-instructions-`*`YOURJOBID`*`.pdf` where *`YOURJOBID`* is the name of the job for your request.

The shipping instructions contain information about how and where to ship your device and a barcode that uniquely associates your device with your job request.

Print the PDF and attach the barcode securely to your device.

You must submit a separate job request for each device, and each job request generates a unique barcode. AWS Import/Export validates the barcode on your storage device before starting the data load. If the barcode is invalid (for example, it doesn't match your device), it's missing, or it's illegible, AWS Import/Export will not perform the data load and we will return your storage device.

> **Important**
> Attach the barcode securely to your device, with tape on all four sides. Do not shrink the barcode or obscure it in any way. If the barcode is separated from your device, we cannot validate it and we will return your device without performing the data load.

Ship Your Device

It is essential that you prepare your device properly before shipping.

> **Note**
> If an error occurs during the `CreateJob` process, an error message will indicate the cause. You can then either get the shipping instructions from the Amazon S3 bucket, or you can retry the download using the `GetShipInfo` operation. For more information, see Getting Shipping Information (p. 73).

1. Attach the barcode securely to your device.
2. Pack the *power supply*, *power cable*, and *interface cables* with your storage device. Without these, we can't transfer your data and will return your device.
3. Send your package to the address provided in the PDF.

   For detailed instructions for preparing and shipping your device, see Shipping Your Storage Device (p. 63).

   Important
   Your job expires after 30 days. If you do not send a device, there is no charge. You are billed only after AWS Import/Export receives your device. If we receive your package after your job expires, we will return your device. You will only be charged for the shipping fees, if any.

If your device is not properly prepared for shipping and import, AWS might need to erase the device and return it without processing the request. If we are unable to erase the data on the device, we will schedule it for destruction and our support team will contact you using the email address specified in the manifest file. For more information, see *Shipping Your Storage Device* (p. 63).

Now, you have explored the getting started steps of AWS Import/Export. To learn more about the product, see Where Do I Go from Here? (p. 25).

Create Your First Amazon Glacier Import Job

With an Amazon Glacier import job, AWS creates a virtual copy of your device as a single image. The device's file system is never mounted. The entire image for one device is uploaded to one Amazon Glacier archive in an existing Glacier vault. If your device is encrypted, then the device image is also encrypted. AWS does not decrypt your device before importing the device image.

After the import operation completes, AWS Import/Export erases the device before shipping it.

You must submit a separate job request for each device.

Note
You can send multiple devices in the same shipment. If you do, however, there are specific guidelines and limitations that govern what devices you can ship and how your devices must be packaged. If your shipment is not prepared and packed correctly, AWS Import/Export cannot process your jobs. Regardless of how many devices you ship at one time, you must submit a separate job request for each device. For complete details about packaging requirements when shipping multiple devices, see Shipping Multiple Devices (p. 68).

For more information, see Guidelines and Limitations (p. 130)

To create an Amazon Glacier import job

1. Create a manifest file.
2. Prepare your device for shipping.
3. Send a `CreateJob` request.

   AWS Import/Export sends a response and a PDF file with a barcode and job information.
4. Attach the barcode to your device.
5. Ship your device.

In your request, you must identify an existing Amazon S3 bucket for logging. You must have write permissions on the bucket.

Tip
You can create a bucket using AWS S3 Console. For more information, go to http://docs.aws.amazon.com/AmazonS3/latest/UG/CreatingaBucket.html.

These steps assume that you have signed up for an AWS account and created your `AWScredentials.properties` file as described in the earlier tasks.

Create a Manifest File

The manifest file is a YAML-formatted text file that instructs AWS about how to handle your job. It consists of a set of name-value pairs that supply required information such as your device ID, log bucket, return address, and so on. For more information about YAML, go to http://yaml.org.

The AWS Import/Export Web Service Tool provides examples of manifest files that you can modify for your import and export jobs.

1. Go to the `Examples` folder in the folder where you installed the AWS Import/Export Web Service Tool.
2. Open the `glacier-import-manifest.txt` file in a text editor.

```
manifestVersion: 3.0
deviceId:[Enter device serial number]
logPrefix: [Optional - a prefix for your log bucket]
logBucket:[Your log bucket]
generator: AWS Import/Export Command Line Tool
eraseDevice: yes
notificationEmail: [Email addresses, semicolon separated]
returnAddress:
    name:[Your name]
    company:[Optional - your company]
    street1:[Your street]
    street2:[Optional - additional street info]
    street3:[Optional - additional street info]
    city:[Your city]
    stateOrProvince:[Required for USA and Canada. Your state or province.]
    postalCode:[Your postal code]
    country:[Your country]
    phoneNumber:[Contact number]
operations:
    - destination: glacier
      source: device
      region: [us-east-1|us-west-1|us-west-2|eu-west-1]
      vaultName: [Your Glacier vault]
      archiveComment: [Your description for the new Glacier archive]
```

Important
When shipping devices internationally except within the European Union you must include the customs option in the manifest. For more information about the customs-related manifest options, see Customs Manifest File Option (p. 79).

3. In the file, replace all text in brackets with the appropriate values. Delete any unused optional lines.
4. Provide the name of the existing Amazon S3 bucket for logging. For the *`logBucket`* option, provide only the name of your bucket, for example `s3examplebucket`.
5. For the `eraseDevice` field, specify `yes` to acknowledge that AWS will erase your device following the import operation before shipping it. If the `eraseDevice` field is missing or if the value is not `yes`, the job request will fail.

6. For the `notificationEmail` field, enter one or more email addresses, separated by semicolons, so we can contact you if there are any problems. If the `notificationEmail` field is missing or empty, the job request will fail.
7. The *`destination`* subfield in the *`operations`* option specifies that the data is imported into an Amazon Glacier vault. Change the *`region`* parameter to specify the correct region. For more information about regions go to the AWS Import/Export Calculator. For more information about how to specify the *`operations`* option, see Creating Import Manifests (p. 37).
8. Save the file as `MyGlacierImportManifest.txt` in the same folder as your `AWSCredentials.properties` file.

For more information about manifest file options, see Manifest File Options Reference (p. 76).

Prepare Your Device for Import

Next, you need to prepare your device for import.

1. Optionally encrypt your data. For added security, we strongly recommend that you encrypt your data. AWS does not decrypt your data before importing to Amazon EBS or Amazon Glacier. The entire device image is imported.
2. Copy your data to the device. Do not ship the only copy of your data. AWS will erase your device, even if we cannot perform an import.

If your device is not properly prepared for shipping and import, AWS might erase the device and return it without processing the request. If we are unable to erase the data on the device, we will schedule it for destruction and our support team will contact you using the email address specified in the manifest file. For more information, see *Shipping Your Storage Device* (p. 63).

Send a CreateJob Request

Now that you have your credentials file and manifest file in place, you send a `CreateJob` request to AWS Import/Export.

1. Open a command prompt (or, on a Mac, use the Terminal application), and change to the directory where you unzipped the AWS Import/Export tool.
2. Enter the following `CreateJob` request on the command line.

```
CmdPrompt>java -jar lib/AWSImportExportWebServiceTool-1.0.jar CreateJob Import
 MyGlacierImportManifest.txt .
```

   The `CreateJob` request includes a directory name for where you want the shipping instructions PDF file saved. In this example, the closing period (.) at the end of the command refers to the current directory. Either include the period, as shown above, or replace it with a directory name.

If there is an error, AWS Import/Export returns an error message.

If there are no errors in the request, AWS Import/Export returns a *JOB CREATED* response.The following is a sample response.

```
JOB CREATED

JobId: ABCDE
JobType: Import
```

```
Shipping instructions saved to C:\DirectoryName\.\shipping-instructions-ABCDE.pdf

SHIPPING INSTRUCTIONS:

1. Open the Shipping Instructions file with a PDF reader.
2. Print out the Shipping Instructions which includes the AWS Shipping Address

and Signature barcode.
3. Follow the directions in the PDF file, which includes cutting out and securely

 attaching the Signature barcode to your device.

*IMPORTANT* - Failure to attach the Signature barcode to your device will prevent

AWS from loading your data.  This barcode can only be used with 1 device.
```

Attach the Barcode

After the `CreateJob` request completes successfully, AWS Import/Export places the shipping instructions and a barcode as a PDF file in the directory you specified in the `CreateJob` request. AWS Import/Export also places a copy of the PDF file in the Amazon S3 bucket that you specified with the logBucket manifest option. The name of the PDF file is `shipping-instructions-`*`YOURJOBID`*`.pdf` where *`YOURJOBID`* is the name of the job for your request.

The shipping instructions contain information about how and where to ship your device and a barcode that uniquely associates your device with your job request.

Print the PDF and attach the barcode securely to your device.

You must submit a separate job request for each device, and each job request generates a unique barcode. AWS Import/Export validates the barcode on your storage device before starting the data load. If the barcode is invalid (for example, it doesn't match your device), it's missing, or it's illegible, AWS Import/Export will not perform the data load and we will return your storage device.

> **Important**
> Attach the barcode securely to your device, with tape on all four sides. Do not shrink the barcode or obscure it in any way. If the barcode is separated from your device, we cannot validate it and we will return your device without performing the data load.

Ship Your Device

It is essential that you prepare your device properly before shipping.

> **Note**
> If an error occurs during the `CreateJob` process, an error message will indicate the cause. You can then either get the shipping instructions from the Amazon S3 bucket, or you can retry the download using the `GetShipInfo` operation. For more information, see Getting Shipping Information (p. 73).

1. Attach the barcode securely to your device.
2. Pack the *power supply*, *power cable*, and *interface cables* with your storage device. Without these, we can't transfer your data and will return your device.
3. Send your package to the address provided in the PDF.

For detailed instructions for preparing and shipping your device, see Shipping Your Storage Device (p. 63).

Important
Your job expires after 30 days. If you do not send a device, there is no charge. You are billed only after AWS Import/Export receives your device. If we receive your package after your job expires, we will return your device. You will only be charged for the shipping fees, if any.

Now, you have explored the getting started steps of AWS Import/Export. To learn more about the product, see Where Do I Go from Here? (p. 25).

Create an Export Job

Now that you're signed up and have downloaded the AWS Import/Export Web Service Tool, you can create an export job to download your data from Amazon S3 buckets to your storage device.

Important
You must create the job and get a response from AWS before you can send your storage device to us for download.

Now, you are ready to submit your create export job request.

Create Your First Amazon S3 Export Job (p. 21).

Create Your First Amazon S3 Export Job

An Amazon S3 export transfers individual objects from Amazon S3 buckets to your device, creating one file for each object. You can export from more than one bucket and you can specify which files to export using manifest file options.

For added security for your data while the device is in transit, AWS Import/Export formats your device before beginning the data load. Any data that is stored on the device when you sent the device to AWS will be lost. AWS copies your data to an encrypted file container on your device. You will need to decrypt the data using TrueCrypt encryption software. For more information, see Using TrueCrypt Encryption (p. 66).

Important
You must provide an encryption password with your job request. If you lose your password, you will not be able to decrypt your device. AWS will not provide your password if you lose it.

You must submit a separate job request for each device.

Note
You can send multiple devices in the same shipment. If you do, however, there are specific guidelines and limitations that govern what devices you can ship and how your devices must be packaged. If your shipment is not prepared and packed correctly, AWS Import/Export cannot process your jobs. Regardless of how many devices you ship at one time, you must submit a separate job request for each device. For complete details about packaging requirements when shipping multiple devices, see Shipping Multiple Devices (p. 68).

For more information, see Guidelines and Limitations (p. 130)

Note
Export from Amazon EBS or Amazon Glacier to a storage device is not currently supported.

Tip
You can create a bucket using the AWS Management Console. For more information, go to Creating a Bucket.

To create an Amazon S3 export job

1. Create a manifest file.
2. Prepare your device for shipping.
3. Send a `CreateJob` request.

   AWS Import/Export sends a response with a `SIGNATURE` file and job information.
4. Copy the `SIGNATURE` file to your device.
5. Ship your device.

These steps assume that you have signed up for an AWS account and created your `AWScredentials.properties` file as described in the earlier tasks.

Create a Manifest File

The manifest file is a YAML-formatted text file that instructs AWS about how to handle your job. It consists of a set of name-value pairs that supply required information such as your device ID, log bucket, return address, and so on. For more information about YAML, go to http://yaml.org.

The AWS Import/Export Web Service Tool provides three examples of manifest files that you can modify for your import and export jobs.

1. Go to the `Examples` folder in the folder where you installed the AWS Import/Export Web Service Tool.
2. Open the `S3-export-manifest.txt` file in a text editor.

```
manifestVersion:2.0
generator:text editor
deviceId:[Enter device serial number]
fileSystem:[NTFS| EXT4]
trueCryptPassword: [password]
serviceLevel:expeditedShipping
targetDirectory:/
recoveryDirectory:EXPORT-RECOVERY
logBucket:[Your log bucket]
notificationEmail: [Email addresses, semicolon separated]
operations:
  - exportBucket:[Your bucket]
returnAddress:
    name:[Your name]
    company:[Optional - your company]
    street1:[Your street]
    street2:[Optional - additional street info]
    street3:[Optional - additional street info]
    city:[Your city]
    stateOrProvince:[Required for USA and Canada. Your state or province.]
    postalCode:[Your postal code]
```

```
country:[Your country]
phoneNumber:[Contact number]
```

Important
When shipping devices internationally except within the European Union you must include the customs option in the manifest. For more information about the customs-related manifest options, see Customs Manifest File Option (p. 79).

3. In the file, replace all text in brackets with the appropriate values. Delete any unused optional lines.
4. Provide an encryption password. AWS will encrypt your data with TrueCrypt using the password you provide. You will need the same password to decrypt your data after you receive it back from AWS. If you do not provide a password, the `CreateJob` request will fail.
5. Provide the name of the existing Amazon S3 bucket containing the data to export, a bucket name to save your log data. For the *`logBucket`* and *`exportBucket`* options provide only the name of the bucket, for example *`s3examplebucket`*.
6. Save the file as `MyS3ExportManifest.txt` in the same folder as your `AWSCredentials.properties` file.

For more information about manifest file options, see Creating Export Manifests (p. 57).

Send a CreateJob Request

Now that you have your credentials file and manifest file in place, you send a `CreateJob` request to AWS Import/Export. You send a separate `CreateJob` request for each device.

1. Open a command prompt (or, on a Mac, use the Terminal application), and change to the directory where you unzipped the AWS Import/Export tool.
2. Enter the following `CreateJob` request on the command line.

```
cmdPrompt>java -jar lib/AWSImportExportWebServiceTool-1.0.jar CreateJob Export
 MyS3ExportManifest.txt .
```

The `CreateJob` request includes a directory name for where you want the `SIGNATURE` file saved. In this example, the closing period (.) at the end of the command refers to the current directory. Either include the period, as shown above, or replace it with a directory name.

If there is an error, AWS Import/Export returns an error message.

If there are no errors in the request, AWS Import/Export returns a *JOB CREATED* response.The following is a sample response.

```
JOB CREATED
JobId:   TC44P
JobType: Export
**************************************
* AwsShippingAddress                 *
**************************************
AWS Import/Export C/O AnyCompany
JOBID TC44P
123 Any Street
Anytown, VA 22153
**************************************
* SignatureFileContents              *
```

```
**************************************
version:2.0
signingMethod:HmacSHA1
jobId:TC44P
signature:BBAqQ/E+R7bdXRVR1s9vTi5t7Aps=

Writing SignatureFileContents to C:\ImportExportWebServiceTool\.\SIGNATURE
```

Your job expires after 30 days. If you do not send a device, there is no charge.

Copy the Signature File

In response to your `CreateJob` request, AWS writes a file named `SIGNATURE` to your computer. The `SIGNATURE` file contains a signature, which is an encrypted string that allows us to authenticate your device and match it to your job request.

Locate the `SIGNATURE` file in the directory that you specified in your job request and copy it to the root directory of your storage device. The file name must be named `SIGNATURE` and it must be in the device's root directory.

Each device you send must include the unique `SIGNATURE` file for that device and that JOBID. AWS Import/Export validates the `SIGNATURE` file on your storage device before starting the data load. If the `SIGNATURE` file is missing or invalid (for example, if it is associated with a different job request), AWS Import/Export will not perform the data load and we will return your storage device.

Ship Your Device

It is essential that you prepare your device properly before shipping. If your device is not properly prepared for shipping and export, AWS might need to erase the device and return it without processing the request. If we are unable to erase the data on the device, we will schedule it for destruction and our support team will contact you using the email address specified in the manifest file. For more information, see *Shipping Your Storage Device* (p. 63).

1. Pack the *power supply*, *power cable*, and *interface cables* with your storage device. Without these, we can't transfer your data and will return your device.
2. Fill out the packing slip and include the packing slip with your device and cables.

   Download the packing slip from http://s3.amazonaws.com/awsimportexport/AWS_Import_Export_Packing_Slip.pdf.
3. Send your package to the address provided in the `CreateJob` response.

   For detailed instructions for preparing and shipping your device, see Shipping Your Storage Device (p. 63).

   Important
   Your job expires after 30 days. If you do not send a device, there is no charge. You are billed only after AWS Import/Export receives your device. If we receive your package after your job expires, we will return your device. You will only be charged for the shipping fees, if any.

Now, you have explored the getting started steps of AWS Import/Export. To learn more about the product, see Where Do I Go from Here? (p. 25).

Where Do I Go from Here?

Topics

The Getting Started section provided a step-by-step experience of creating a job. Now you can learn more about the service.

Working with Import Jobs

This section describes the process of importing data into AWS. You can import data to Amazon S3 buckets, Amazon Elastic Block Store (Amazon EBS) snapshots, and Amazon Glacier vaults. Amazon EBS snapshots can be converted into volumes for use with Amazon Elastic Compute Cloud (EC2). For more information, see Working with Import Jobs (p. 30).

Working with Export Jobs

This section describes the process of exporting data from AWS. AWS Import/Export doesn't support export jobs from Amazon EBS or Amazon Glacier. For more information, see Working with Export Jobs (p. 55).

Managing Your Jobs

AWS Import/Export tools and API provide you ways to manage your job easily. For example, after creating a job, you can update or cancel it. You can also get list of all your jobs or get status of a specific job. For more information, see Managing Your Jobs (p. 71).

Pricing Information

As with all AWS services, you pay only for the resources that you use. For more information on AWS Import/Export pricing model, go to AWS Import/Export detail page.

Using IAM with AWS Import/Export

Topics

AWS Identity and Access Management is a web service that enables Amazon Web Services (AWS) customers to manage users and user permissions in AWS. AWS Import/Export users must have certain permissions to access AWS Import/Export actions, such as CreateJob and GetStatus. An IAM user that creates an import or export must also have access to the AWS resources, such as the Amazon S3 buckets and Amazon Glacier vault, that will used for the import or export operations.

Controlling Access to AWS Import/Export Jobs

AWS Import/Export integrates with AWS Identity and Access Management (IAM), which allows you to control which actions a user can perform.

By default, IAM users have no access to AWS Import/Export actions. If you want IAM users to be able to work with AWS Import/Export, you must grant them permissions. You do this by creating an IAM policy that defines which Import/Export actions the IAM user is allowed to perform. You then attach the policy to the IAM user or to an IAM group that the user is in.

You can give IAM users of your AWS account access to all AWS Import/Export actions or to a subset of them. The following is the full list of AWS Import/Export actions:

```
importexport:CancelJob
importexport:CreateJob
importexport:GetStatus
importexport:ListJobs
importexport:UpdateJob
```

For information about how to create a policy and attach it to a group, user, or role, go to Managing IAM Policies in the AWS Identity and Access Management documentation.

For general information about IAM, go to:

- AWS Identity and Access Management (IAM) detail page

- What Is IAM? in the AWS Identity and Access Management documentation

Example IAM User Policies for AWS Import/Export

This section shows three simple policies for controlling access to AWS Import/Export. AWS Import/Export service does not support resource-level permissions, so in policies for Import/Export, the `"Resource"` element is always `"*"`, which means all resources.

Allow read-only access to the jobs created under the AWS account

The following policy only allows access to the `ListJobs` and `GetStatus` actions, which are read-only actions.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "importexport:ListJobs",
        "importexport:GetStatus"
      ],
      "Resource": "*"
     }
   ]
}
```

Allow full access to all AWS Import/Export jobs created under the AWS account

The following policy allows access to all AWS Import/Export actions.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "importexport:*",
      "Resource": "*"
    }
  ]
}
```

Deny a set of actions from an IAM user

By default, all permissions are denied; if you do not explicitly grant access to Import/Export actions, users are not allowed to perform those actions. It's also possible to explicitly deny access to specific actions. This might be useful if one policy (or statement in a policy) grants access to a set of actions, but you want to exclude one or more individual actions.

The following policy contains two statements. The first statement allows access to all the AWS Import/Export actions. The second statement explicitly denies access to `UpdateJob`. If new actions are added to AWS

Import/Export, this policy automatically grants permission for those new actions because of the first statement. However, the user will always be denied access to the `UpdateJob` action, even if that action is explicitly allowed in another policy.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect":"Allow",
      "Action":"importexport:*"
    },
    {
      "Effect":"Deny",
      "Action":"importexport:UpdateJob",
      "Resource": "*"
    }
  ]
}
```

Granting Access to Related AWS Resources

The IAM user that creates an AWS Import/Export job must have permissions to access the Amazon S3 buckets and, for import to Amazon Glacier, the Amazon Glacier vault, that will be used for the import or export operations. If the user does not have the necessary permissions, the CreateJob request will fail.

For all import and export jobs, the IAM user must have the following access permissions on the Amazon S3 log bucket:

```
s3:GetBucketLocation
s3:PutObject
s3:AbortMultipartUpload
s3:ListMultipartUploadParts
s3:ListBucketMultipartUploads
```

For import to Amazon S3, the IAM user must have the following access permissions on the Amazon S3 import bucket:

```
s3:GetBucketLocation
s3:PutObject
s3:AbortMultipartUpload
s3:ListMultipartUploadParts
s3:ListBucketMultipartUploads
```

For export from Amazon S3, the IAM user must have the following access permissions on the Amazon S3 export buckets:

```
s3:GetBucketLocation
s3:GetObject
s3:ListBucket
```

For import to Amazon Glacier, the IAM user must have the following access permissions on the Amazon Glacier vault:

```
glacier:AbortMultipartUpload
glacier:CompleteMultipartUpload
glacier:DescribeVault
glacier:InitiateMultipartUpload
glacier:ListMultipartUploads
glacier:ListParts,
glacier:UploadArchive
glacier:UploadMultipartPart
```

The only permissions required for import to Amazon EBS are the previously listed permissions for the Amazon S3 log bucket.

For more information, go to:

- Amazon S3: Using IAM Policies
- Amazon Glacier: Access Control Using AWS Identity and Access Management (IAM)

Working with Import Jobs

Topics

This section summarizes the process you use to import data from your storage device to an Amazon S3 bucket, an Amazon Elastic Block Store (Amazon EBS) snapshot, or an Amazon Glacier vault.

The AWS Import/Export Web Service Tool discussed in Getting Started (p. 6) is the easiest way to create import jobs. It is a command line tool and requires no programming. However, if your application needs a programming interface, you can use the AWS SDK for Java or the REST API to create jobs.

Creating Amazon S3 Import Jobs

With an Amazon S3 import, AWS uploads individual files from your device to objects in an Amazon S3 bucket. One device is loaded to one bucket. Your device's file system is mounted on an AWS Import/Export data loading station and each file is uploaded as an object in your Amazon S3 bucket. You can optionally manage directories and filenames as part of the load process.

You must submit a separate job request for each device.

For more information, see Guidelines and Limitations (p. 130)

Import to Amazon S3 Process

Import Job Process

1	You create an `AWSCredentials.properties` file that contains your *`accessKeyId`* and *`secretKey`* associated with your AWS account. For more information, see Save Your Credentials to a File (p. 8).

2	You create an import manifest file that specifies how to transfer data from your storage device to a single Amazon S3 bucket. For more information, see Creating an Import Manifest File (p. 37).
3	You initiate an import job by sending a `CreateJob` request that includes the manifest file. You send a separate CreateJob request for each device. Your job expires after 30 days. If you do not send a device, there is no charge. You can send a CreateJob request using the AWS Import/Export Tool, the AWS Command Line Interface (CLI), the AWS SDK for Java, or the AWS REST API. The easiest method is the AWS Import/Export Tool. For details, see Sending a CreateJob Request Using the AWS Import/Export Web Service Tool (p. 46) Sending a CreateJob Request Using the AWS SDK for Java (p. 48) Sending a CreateJob Request Using the REST API (p. 51)
4	AWS Import/Export sends a response that includes a job ID, a signature value, and an AWS shipping address. The response also saves a `SIGNATURE` file to your computer. You will need this information in subsequent steps.
5	Optionally, you encrypt your device or create an encrypted file container using TrueCrypt. If you use an encrypted container, name the container `<JOBID>.tc`, using the Job ID from the CreateJob request. For example, `KSVDJ.tc`. For more information, see Using TrueCrypt Encryption (p. 66).
6	You copy the `SIGNATURE` file to the root directory of your storage device. You can use the file AWS sent or copy the signature value from the response into a new text file named `SIGNATURE`. The file name must be `SIGNATURE` and it must be in the device's root directory. Each device you send must include the unique `SIGNATURE` file for that device and that JOBID. AWS Import/Export validates the `SIGNATURE` file on your storage device before starting the data load. If the `SIGNATURE` file is missing invalid (for example, if it is associated with a different job request), AWS Import/Export will not perform the data load and we will return your storage device.
7	You copy your data to your storage device. AWS Import/Export loads the data from your device to a single Amazon S3 bucket. Note that when uploading data to Amazon S3, the file path on your storage device determines the corresponding object key. For example, if your storage device has an `images` top level folder with a file `sample.jpg`, the uploaded file/object key will be images/sample.jpg. File names must use UTF-8 character encoding. Any file with a name that is not a valid UTF-8 string is not imported. For information on storage device requirements, see Storage Device Requirements (p. 63).
8	You fill out the AWS Import/Export packing slip to send your device. You must send the device to the address returned in the `CreateJob` response. You must also add the Job ID to the packing slip. For more information see Filling Out the Packing Slip (p. 65) and Storage Device Requirements (p. 63). You ship the device, the cables and packing slip to AWS. Make sure to include your job ID on the shipping label and on the device you are shipping. Otherwise, your download might be delayed. **Note** You can send multiple devices in the same shipment. If you do, however, there are specific guidelines and limitations that govern what devices you can ship and how your devices must be packaged. If your shipment is not prepared and packed correctly, AWS Import/Export cannot process your jobs. Regardless of how many devices you ship at one time, you must submit a separate job request for each device. For complete details about packaging requirements when shipping multiple devices, see Shipping Multiple Devices (p. 68). Your job expires after 30 days. If we receive your package after your job expires, we will return your device. You will only be charged for the shipping fees, if any.

9	AWS Import/Export validates the `SIGNATURE` file on the root drive of your storage device. If the `SIGNATURE` file doesn't match the `CreateJob` response, AWS can't load your data and we will return your device. Once your storage device arrives at AWS, your data load typically begins by the end of the next business day. The time line for loading your data depends on a number of factors, including decryption, the availability of an import station, the amount of data to load, and the data transfer rate of your device. We import all of the data into a single Amazon S3 bucket.
10	If you encrypted your data, AWS decrypts the data using the password you provided in the manifest file. If we cannot decrypt the data, we erase the device and ship it to the return address you provided.
11	AWS imports all of the data into a single Amazon S3 bucket.
12	Following the import operation, AWS will erase the device, even if the import was not successful. In cases where we are unable to erase the data on the device, we will schedule it for destruction. In this case, our support teams will contact you over the email address specified in the Manifest file.
13	AWS repacks your storage device and ship it to the return address you provided in your manifest file. We do not ship to post office boxes.

Creating Amazon EBS Import Jobs

AWS Import/Export uploads your device to Amazon Elastic Block Store (Amazon EBS) snapshots. Amazon EBS snapshots can be converted into volumes for use with Amazon Elastic Compute Cloud (Amazon EC2).

If you are using AWS Import/Export to upload data to Amazon EBS, we assume you are an Amazon EC2 user and are familiar with the Amazon EBS concepts volumes and snapshots. To learn more about Amazon EBS, go to Amazon Elastic Block Store.

An Amazon EBS import creates a virtual copy of your device as a single image. The device's file system is never mounted.

The capacity of your device determines how the device image is stored. The maximum capacity of an Amazon EBS volume is 1 TB. If the capacity of the device is greater than 1 TB, we cannot import the disk directly into an EBS snapshot.

Note
The maximum capacity of a device is independent of the amount of data stored on the device.

If your device capacity is 1 TB or less, we store the device image directly to an EBS snapshot.

If your device capacity is greater than 1 TB, we store the device image as a series of objects in an Amazon S3 bucket. You can then create a RAID of Amazon EBS volumes using software such as Logical Volume Manager, and copy the image from Amazon S3 to the new volume.

You must submit a separate job request for each device.

For more information, see Guidelines and Limitations (p. 130)

Import to Amazon EBS Process

Import to Amazon EBS Process

1	You copy your data to your storage device. Do not send the only copy of your data. AWS will erase your device. For added security, we strongly recommend using encryption. AWS does not decrypt your data.
2	You create an `AWSCredentials.properties` that contains the *`accessKeyId`* and *`secretKey`* associated with your AWS account. For more information, see Save Your Credentials to a File (p. 8).
3	You create an import manifest file that specifies the operation, Amazon EBS, the region, and return shipping address. For more information, see Import to Amazon EBS Manifest File Options (p. 91). **Important** If your device capacity is 1 TB or less, specify `no` for the `deviceCapacityGreaterThan1TB` option. If your device capacity is greater than 1 TB, specify `yes` for the `deviceCapacityGreaterThan1TB` option. If the `deviceCapacityGreaterThan1TB` option is not set correctly for your device, AWS Import/Export will not perform the import and we will return your storage device. The maximum capacity of a device is independent of the amount of data stored on the device.
4	You initiate an import job by sending a `CreateJob` request that includes the manifest file. You must submit a separate job request for each device. Your job expires after 30 days. If you do not send a device, there is no charge. You can send a `CreateJob` request using the AWS Import/Export Tool, the AWS Command Line Interface (CLI), the AWS SDK for Java, or the AWS REST API. The easiest method is the AWS Import/Export Tool. For details, see Sending a CreateJob Request Using the AWS Import/Export Web Service Tool (p. 46) Sending a CreateJob Request Using the AWS SDK for Java (p. 48) Sending a CreateJob Request Using the REST API (p. 51)
5	AWS Import/Export sends a response that includes a job ID, shipping instructions, and a PDF file. The PDF file contains job information, shipping instructions, and a barcode. You must submit a separate job request for each device, and each job request generates a unique barcode. AWS Import/Export validates the barcode on your storage device before starting the data load. If the barcode is invalid (for example, it doesn't match your device), it's missing, or it's illegible, AWS Import/Export will not perform the data load and we will return your storage device.
6	You print the PDF. For an example of the PDF, see The Import PDF File (p. 34). You cut out the barcode portion of the printed PDF, and tape it to your device. You must send the device to the address indicated in the shipping instructions. **Important** Attach the barcode securely to your device, with tape on all four sides. Do not shrink the barcode or obscure it in any way. If the barcode is separated from your device, we cannot validate it and we will return your device without performing the data load.

7	You ship the device, the cables, and the printed bar code to AWS. Make sure to include your job ID on the shipping label. Otherwise, your device might be returned unprocessed. **Note** You can send multiple devices in the same shipment. If you do, however, there are specific guidelines and limitations that govern what devices you can ship and how your devices must be packaged. If your shipment is not prepared and packed correctly, AWS Import/Export cannot process your jobs. Regardless of how many devices you ship at one time, you must submit a separate job request for each device. For complete details about packaging requirements when shipping multiple devices, see Shipping Multiple Devices (p. 68). **Important** Your job expires after 30 days. If you do not send a device, there is no charge. You are billed only after AWS Import/Export receives your device. If we receive your package after your job expires, we will return your device. You will only be charged for the shipping fees, if any.
8	AWS Import/Export uses the bar code to identify and authenticate your data load. Once your storage device arrives at AWS, your data load typically begins by the end of the next business day. The time line for loading your data depends on a number of factors, including the availability of an import station, the amount of data to load, and the data transfer rate of your device. AWS displays your Amazon EBS snapshot in the list of EBS Snapshots for your account within the AWS Management Console for Amazon EC2. For instructions to find your snapshots in the Amazon EC2 Console, see Your Amazon EBS Snapshot in the AWS Management Console (p. 35).
9	Following the import operation, AWS will erase the device, even if the import was not successful. In cases where we are unable to erase the data on the device, we will schedule it for destruction. In this case, our support teams will contact you over the email address specified in the Manifest file.
10	We repack your storage device and ship it to the return address you provided in your manifest file. We do not ship to post office boxes.

The Import PDF File

The AWS response to the `CreateJob` request includes your job ID, AWS shipping address, and saves a PDF file to your computer. The PDF file contains shipping instructions with a unique bar code to identify and authenticate your device. Following is an example PDF file.

AWS Import/Export Shipping Instructions

Create Date: 7/5/11
JOBID: XXXX
Device ID: XXXX
Amazon EC2 Region: us-east-1

To prepare and send your job:

1. Cut out the barcode below and securely attach it to your storage device (external hard drive, internal SATA drive, or USB flash drive) with clear tape.
2. Pack your storage device. For external hard drives please make sure to enclose your USB or eSATA connector and powersupply.
3. Ship your storage device to:

AWS Import/Export
JOBID XXXX
2646 Rainier Ave South Suite 1060
Seattle, WA 98144
United States
206-266-6868
awsimportexport@amazon.com

IMPORTANT
We cannot process your job without the bar code below attached to your device.
---------- CUT HERE ----------

XXXX

1.0:XXXX:axu2jXnimvw3rrszlHG88Y0OQ=

As noted in Import to Amazon EBS Process (p. 33), you must print the PDF, tape the bar code to your device, and follow the shipping instructions on the page.

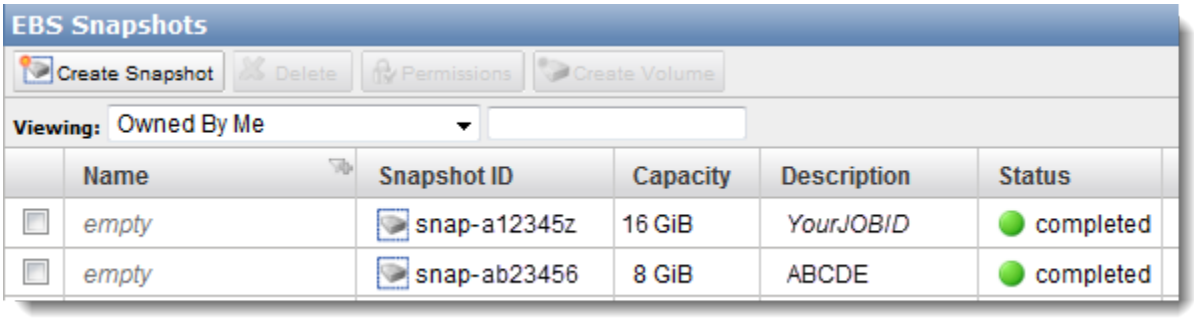
Your Amazon EBS Snapshot in the AWS Management Console

You can monitor the status of your Amazon EBS import job in the AWS Management Console. Once your data begins importing to an Amazon EBS snapshot, you can find the Snapshot ID and its status in the AWS Management Console for Amazon EC2.

To find your Amazon EBS Snapshot

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. Select **Snapshots** in the left side Navigation pane.

   The **EBS Snapshots** pane appears.



The Description field includes the job ID provided by AWS Import/Export as a response to your `CreateJob` request.

Creating Amazon Glacier Import Jobs

An Amazon Glacier import job creates a virtual copy of your device as a single image. The entire image for one device is uploaded to one Glacier archive in an existing Glacier vault. One archive holds the contents of one device. The device's file system is never mounted. You send a separate `CreateJob` request for each device.

If you want the lower cost storage provided by Amazon Glacier, but you want your individual files to be stored as separate objects, first import your files to Amazon S3, then use Amazon S3 Object Lifecycle Management to transition your files to the Amazon Glacier Storage Class. When you transition objects to the Glacier storage class, Amazon S3 internally uses Amazon Glacier for durable storage at lower cost. Amazon S3 objects that are associated with the Glacier object class are visible and available only in Amazon S3. For more information, see Object Lifecycle Management.

If you are using AWS Import/Export to upload data to Amazon Glacier, we assume you are an Amazon Glacier user and are familiar with the Amazon Glacier concepts *vaults* and *archives*. To learn more about Amazon Glacier, go to Amazon Glacier.

For more information, see Guidelines and Limitations (p. 130)

Import to Amazon Glacier Process

Import Job Process

1	You copy your data to your storage device. Do not send the only copy of your data. AWS will erase your device. For added security, we strongly recommend using encryption. AWS does not decrypt your data.
2	You create an `AWSCredentials.properties` file that contains the *`accessKeyId`* and *`secretKey`* associated with your AWS account. For more information, see Save Your Credentials to a File (p. 8).
3	You create an import manifest file that specifies the Amazon Glacier region and return shipping address. For more information, see Import to Amazon Glacier Manifest File Options (p. 94).

4	You initiate an import job by sending a `CreateJob` request that includes the manifest file. You must submit a separate job request for each device. Your job expires after 30 days. If you do not send a device, there is no charge. You can send a CreateJob request using the AWS Import/Export Tool, the AWS Command Line Interface (CLI), the AWS SDK for Java, or the AWS REST API. The easiest method is the AWS Import/Export Web Services Tool. For details, see Sending a CreateJob Request Using the AWS Import/Export Web Service Tool (p. 46) Sending a CreateJob Request Using the AWS SDK for Java (p. 48) Sending a CreateJob Request Using the REST API (p. 51)
5	AWS Import/Export sends a response that includes a job ID, shipping instructions, and a PDF file. The PDF file contains job information, shipping instructions, and a barcode. You must submit a separate job request for each device, and each job request generates a unique barcode. AWS Import/Export validates the barcode on your storage device before starting the data load. If the barcode is invalid (for example, it doesn't match your device), it's missing, or it's illegible, AWS Import/Export will not perform the data load and we will return your storage device.
6	You print the PDF. For an example of the PDF, see The Import PDF File (p. 34). You cut out the barcode portion of the printed PDF, and tape it to your device. You must send the device to the address indicated in the shipping instructions. **Important** Attach the barcode securely to your device, with tape on all four sides. Do not shrink the barcode or obscure it in any way. If the barcode is separated from your device, we cannot validate it and we will return your device without performing the data load.
7	AWS Import/Export uses the bar code to identify and authenticate your data load. Once your storage device arrives at AWS, your data load typically begins by the end of the next business day. The time line for loading your data depends on a number of factors, including the availability of an import station, the amount of data to load, and the data transfer rate of your device.
8	Following the import operation, AWS will erase the device, even if the import was not successful. In cases where we are unable to erase the data on the device, we will schedule it for destruction. In this case, our support teams will contact you over the email address specified in the Manifest file.
9	We repack your storage device and ship it to the return address you provided in your manifest file. We do not ship to post office boxes.

Creating Import Manifests

Topics

You provide your job information and specify how to transfer your data between your storage device and AWS by creating a manifest file. The manifest file is a YAML-formatted text file that instructs AWS about how to handle your job. It consists of a set of name-value pairs that supply required information such as your device ID, log bucket, return address, and so on. For more information about YAML, go to http://yaml.org.

Import Manifest Examples

Each type of import job has a set of required options.

This sample Amazon S3 import manifest file includes only the required options.

```
manifestVersion: 2.0;
returnAddress:
   name: Jane Roe
   company: Example Corp.
   street1: 123 Any Street
   city: Anytown
   stateOrProvince: WA
   postalCode: 91011-1111
   phoneNumber: 206-555-1111
   country: USA
deviceId: 49382
eraseDevice: yes
notificationEmail: john.doe@example.com;jane.roe@example.com
bucket: myBucket
```

This sample Amazon EBS import manifest file includes only the required options.

```
manifestVersion: 3.0
deviceId: 49382
logBucket: myBucket
eraseDevice: yes
notificationEmail: john.doe@example.com;jane.roe@example.com
generator: AWS Import/Export Command Line Tool
returnAddress:
   name: Jane Roe
   company: Example Corp.
   street1: 123 Any Street
   city: Anytown
   stateOrProvince: WA
   postalCode: 91011-1111
   phoneNumber: 206-555-1111
   country: USA
operations:
    - destination: ebs-snapshot
      source: device
      region: us-east-1
      deviceCapacityGreaterThan1TB: yes
```

This sample Amazon Glacier import manifest file includes only the required options.

```
manifestVersion: 3.0
deviceId: 49382
logBucket: myBucket
```

```
generator: AWS Import/Export Command Line Tool
notificationEmail: john.doe@example.com;jane.roe@example.com
returnAddress:
   name: Jane Roe
   company: Example Corp.
   street1: 123 Any Street
   city: Anytown
   stateOrProvince: WA
   postalCode: 91011-1111
   phoneNumber: 206-555-1111
   country: USA
operations:
    - destination: glacier
      source: device
      region: us-east-1
      vaultName: MyGlacierVault
      archiveComment: July 24, 2012 daily archive
```

For a complete list of manifest options, see Manifest File Options Reference (p. 76). For examples of import manifest options, see Configuring Common Import Manifest Options (p. 39).

Configuring Common Import Manifest Options

These options are common to Amazon S3, Amazon EBS, and Amazon Glacier import jobs.

Topics

For a complete list of manifest options, see Manifest File Options Reference (p. 76). For examples of import manifest options, see Configuring Common Import Manifest Options (p. 39).

Shipping Devices Internationally

When shipping devices internationally except within the European Union you must include the customs option in the manifest. For more information about the customs related manifest options, see Customs Manifest File Option (p. 79).

Expediting the Return of Your Storage Device

When returning your device to a U.S. address from a U.S. region bucket, you can use the *`serviceLevel`* manifest option to upgrade your shipping. The default *`serviceLevel`* option is standard shipping, which includes free ground domestic shipping. When you specify *`expeditedShipping`*, your device is delivered using two-day shipping for an additional fee. For information about return shipping fees and services, go to the AWS Import/Export Calculator.

Specifying Device Erase After Import

After every import operation, even if the import was not successful, AWS Import/Export will erase the contents of your storage device to safeguard the data during return shipment. AWS overwrites all writable blocks on your device with zeros. You will need to repartition and format your device after we return it to you. You must include `eraseDevice: Yes` in your import manifest to acknowledge that your device will

be erased. If we are unable to erase the data on the device, we will schedule it for destruction and our support team will contact you using the email address specified in the manifest file `notificationEmail` field.

```
eraseDevice: yes
notificationEmail: john.doe@example.com
```

Preventing Log File Name Collisions

The AWS Import/Export process generates a log file. The log file name always ends with the phrase `import-log-` followed by your *`JobId`*. There is a remote chance that you already have an object with this name. To avoid a key collision, you can add an optional prefix to the log file by adding the *`logPrefix`* option in the manifest. AWS Import/Export takes the string value specified for this option and inserts it between the bucket name and log report name. The following manifest option sets the prefix for the log key.

```
logPrefix: logs/
```

For example, if your job ID is 53TX4, the log file is saved to `http://s3.amazonaws.com/mybucket/logs/import-log-53TX4`.

> **Note**
> We do not include a forward slash (/) automatically. If you don't include the slash at the end of the value for *`logPrefix`*, the value is concatenated to the log file name. For example, if your *`logPrefix`* is `logs` the log file key your key would become `logsimport-log-jobId` instead of `logs/import-log-JobId`.

`logPrefix` + `import-log-JOBID` cannot be longer than 1024 bytes. If it is, AWS Import/Export returns an InvalidManifestField error from the `CreateJob` action.

Working with Amazon S3 Import Manifests

These options are used with Amazon S3 import jobs.

Topics

Handling Gzip Files

When importing Gzip compressed files you can set the *`setContentEncodingForGzFiles`* option to `yes`. This adds the Content-Encoding header, in addition to the Content-Type header, when the Gzip compressed files are uploaded. The Content-Encoding header helps most browsers render these files correctly.

```
setContentEncodingForGzFiles: yes
```

Additionally, the extensions, `.gz` and `.gzip` are ignored when setting the Content-Type header if the file has been compressed with Gzip.

For example, if *setContentEncodingForGzFiles* is set to "yes", the compressed file compressed using Gzip `text1.html.gz,` is uploaded with the following HTTP headers:

- Content-Encoding: gzip
- Content-Type: text/html

The gzip compressed file, `text2.html,` is uploaded with the following HTTP headers:

- Content-Encoding: gzip
- Content-Type: text/html

The non-compressed file, `text3.html`, is uploaded with the following HTTP headers:

- Content-Type: text/html

Note
When *setContentEncodingForGzFiles* is set to `yes`, only files that are gzip compressed will contain a Content-Encoding header. We look at the first few bytes of all imported files to see if they are compressed using Gzip. If so, they get the Content-Encoding header regardless of the file extension.

The gzip compressed file, `text.gzip`, is uploaded with the following HTTP headers using the *defaultContentType* specified in the manifest file:

- Content-Encoding: gzip
- Content-Type: binary/octet-stream

Setting ACL on Imported Amazon S3 Objects

When importing data to Amazon S3, the permissions on the imported objects is set as private. You can specify the `acl` manifest option to specify the access control list (ACL) on the imported objects. The following manifest option sets the ACL value on the uploaded objects to public-read.

```
acl: public-read
```

Excluding Files and Directories

Topics

You can instruct AWS Import/Export not to import some of the directories and files on your storage device. This is a convenient feature that allows you to keep a directory structure intact, but avoid uploading unwanted files and directories. Use the *ignore* option to specify directories, files, or file types on your storage device to ignore. Use standard Java regular expressions to specify naming patterns. For information about Java regular expressions, go to http://download.oracle.com/javase/tutorial/essential/regex/. The following examples show Java regular expressions that are commonly used in a manifest.

Excluding Files

The following example uses the ignore option with two Java regular expressions to exclude files with suffix ending with a tilde and `.swp`.

```
ignore:
  -  .*~$
  -  .*\.swp$
```

The following `ignore` option excludes all the files on the storage device with the `.psd` extension.

```
ignore:
  - \.psd$
  - \.PSD$
```

The log report includes all ignored files, including the `SIGNATURE` file you added at the root of your storage device.

Excluding Directories

The following `ignore` option specifies that the `backup` directory at the root of your storage device will be excluded from the import.

```
ignore:
  - ^backup/
```

Important
When specifying a path that includes a file separator, for example, `images/myImages/sampleImage.jpg`, make sure to use a forward slash, "/", and not a back slash.

The following ignore option ignores all the content in the images/myImages directory.

```
ignore:
  - ^images/myImages/
```

Excluding Recycle Bin

Many storage devices include recycle bins. You may not want to upload the recycle bin in the import process. To skip the recycle bin on Windows computers you specify the following ignore option. The regular expression in the first line applies to NTFS file systems formatted for Windows Vista and Windows 7. The regular expression in the second line applies to NTFS file systems on Windows 2000, Windows XP and Windows NT. And the regular expression in the third line applies to the FAT file system.

```
ignore:
  - ^\$Recycle\.bin/
  - ^RECYCLER/
  - ^RECYCLED/
```

Excluding Lost+Found

The Java regular expression in the following `ignore` statement prevents the `lost+found` directory from being uploaded.

```
ignore:
  - ^lost\+found/
```

Manipulating Amazon S3 Object Key Names

Topics

The file path on your storage device determines the corresponding object key when AWS loads your data to an Amazon S3 bucket. For example, if your storage device has an `images` top-level folder with a file `sample.jpg`, the uploaded file/object key will be `images/sample.jpg`.

You can use the substitutions option to change object key names when importing to Amazon S3. For example, you can add prefixes or suffixes to file names. For more information, see Working with Amazon S3 Substitutions.

Specifying Key Prefix

The AWS Import/Export prefix mechanism allows you to create a logical grouping of the objects in a bucket. The *`prefix`* value is similar to a directory name that enables you to store similar data under the same directory in a bucket. For example, if your Amazon S3 bucket name is `my-bucket`, and you set *`prefix`* to `my-prefix/`, and the file on your storage device is `/jpgs/sample.jpg`, then `sample.jpg` would be loaded to `http://s3.amazonaws.com/my-bucket/my-prefix/jpgs/sample.jpg`. If the prefix is not specified, `sample.jpg` would be loaded to `http://s3.amazonaws.com/my-bucket/jpgs/sample.jpg`. You can specify a prefix by adding the `prefix` option in the manifest.

> **Important**
> We do not include a forward slash (/) automatically. If you don't include the slash at the end of the value for *`prefix`*, the value is concatenated to the file name. For example, if your prefix is `images` and you import the file `sample.jpg`, your key would become `imagessample.jpg` instead of `images/sample.jpg`.

```
prefix: my-prefix/
```

Working with Amazon S3 Substitutions

The *`substitutions`* option is helpful for changing file names, appending prefixes or suffixes to file names, or other name changes during an Amazon S3 import or export job. For import jobs, use the *`substitutions`* option to change object key names. For export jobs, use the *`substitutions`* option to change names to write to your file system. For example, use the following entry to replace all the uppercase "A", "B" and "C" letters in your file names and directories with lowercase letters on the object key names before the data is uploaded to your Amazon S3 bucket.

The *`substitutions`* option can only be used when *`manifestVersion`* is set to 2.0, and is not available for Amazon EBS or Amazon Glacier import jobs.

```
substitutions:
    "A" : "a"
    "B" : "b"
    "C" : "c"
```

Important
Avoid attempting to replace an entire value with an empty string. For import jobs, an empty string causes the import to fail and report a "400 InvalidSubstitution" error in the log. Also, avoid substitutions that result in multiple files mapping to the same file name. For import jobs, when multiple files map to the same file name, the behavior is undefined. For export jobs, when multiple files map to the same file name, the object falls into the recovery process. For more information about the recovery process, see Collecting Files That Have Naming Errors (p. 61).

The *substitutions* option is applied after any *prefix* options are applied. First, the *prefix* option determines what Amazon S3 data to copy, then, the *substitutions* option determines the value to write. For export jobs, if a *targetDirectory* option is specified, the substitutions option is applied to the *targetDirectory* value while it is writing to your device.

Example: Using Both Prefix and Substitutions Options

```
prefix: myprefix$COLON$
substitutions:
    "$COLON$" : ":"
    "$QUOTE$" : "\""
```

When you use the above options, you can enter a fully qualified file name on your device, such as:

```
/backup/$COLON$/$QUOTE$computer1$QUOTE$/c$COLON$/image.dat
```

AWS Import/Export performs the substitution and the file name becomes the following object name on Amazon S3.

```
myprefix:backup/:/"computer1"/c:/image.dat
```

Example: Using Substitutions, Prefix, and targetDirectory Options

```
substitutions:
    ":" : "$COLON$"
    "," : "$COMMA$"
operations:
  - exportBucket: mybucket
    prefix: myprefix,
    targetDirectory: data,
```

When you use the above options, you can enter an Amazon S3 object name, such as:

```
mybucket/myprefix,/backup/:/computer1/c:/image.dat
```

AWS Import/Export performs the substitution, and the object name becomes the following fully qualified file name on your device.

```
/data$COMMA$/myprefix$COMMA$/backup/$COLON$/computer1/c$COLON$/image.dat
```

Example: Changing Name Space Encoding

In most cases the file system name space requirements are more restrictive than the Amazon S3 name space requirements, and you might need to map between valid file system names and the Amazon S3 key name space. For an import job, you can take advantage of the more expressive name space support

in Amazon S3 by encoding the file name with information that is decoded before upload. In the following example, character encoding for a colon is converted to a colon in the Amazon S3 name space:

```
substitutions:
    $COLON$ : ":"
```

For an export job, you can handle characters that don't map to the file system.

```
substitutions:
    ":" : "$COLON$"
```

Mapping Uppercase Characters to Lowercase Characters

You can define a rule to substitute all uppercase characters for file names with the equivalent lowercase characters for object names in your import job. For example, use the following entry to replace all the uppercase characters in your file names with lowercase letters for an entire alphabet. List all the letters in the alphabet (you need to specify each one) with the uppercase letters on the left side of the option parameter and lowercase letters on the right side of the option parameter (where "..." represents all the characters between C and Y):

```
substitutions:
    "A" : "a"
    "B" : "b"
    "C" : "c"
       ...
    "Y" : "y"
    "Z" : "z"
```

For more information, see the *substitutions* option in Common Manifest File Options (p. 76).

Mapping File Directories to the Amazon S3 Root

Amazon S3 performs well even when there are millions of files in the same bucket. To import your data efficiently into Amazon S3 using AWS Import/Export, you might decide to eliminate your subdirectories. If you name your directories carefully, such that none of the names of the directories are substrings of your file names, you can use the *substitutions* manifest option to remove the directory from the key name. The following example assumes you have a directory structure that divides your data across the three subdirectories, *ZZ1*, *ZZ2*, *ZZ3* in your file system.

```
ZZ1/
ZZ2/
ZZ3/
```

To remove the directory name from the Amazon S3 key names, define the following *substitutions* option in your manifest file:

```
substitutions:
    "ZZ1/" : ""
    "ZZ2/" : ""
    "ZZ3/" : ""
```

All of the files will be stored in the Amazon S3 bucket root.

Important
None of the files within the subdirectories should contain the *substitutions* strings in their file names (such as *"ZZ1/"*, *"ZZ2/"*, or *"ZZ3/"*).
If two files have the same name, both files are uploaded to Amazon S3, but you will only retain the bytes of the last file transferred.
Use the forward slash (/) as the file separator character. Don't use the back-slash (\) or double back-slash (\\).

Sending CreateJob Requests

Topics

This section summarizes the tools for sending a `CreateJob` request to AWS Import/Export.

After you have created an `AWSCredentials.properties` file and a manifest file, you initiate an import job or export job by sending a `CreateJob` request that specifies the job type, Import or Export, and specifies the manifest file.

You can send a `CreateJob` request using the AWS Import/Export Web Service Tool, the AWS SDK for Java, or the AWS REST API. The easiest method is by using the AWS Import/Export Web Service Tool.

Sending a CreateJob Request Using the AWS Import/Export Web Service Tool

The easiest way to send a `CreateJob` request is by using the AWS Import/Export Tool.

To send a CreateJob request using the AWS Import/Export Web Service Tool

1. Click the following link to download the AWS Import/Export Web Service Tool:

   Click http://awsimportexport.s3.amazonaws.com/importexport-webservice-tool.zip.

   Extract the zipped content to a directory on your computer.
2. Open a command prompt (or a terminal application), and change to the directory where you unzipped the AWS Import/Export Web Service Tool.
3. Enter the `CreateJob` request on the command line, specifying either Import or Export and the manifest file name.

   For an Import job, enter the following `CreateJob` request on the command line.

```
CmdPrompt>java -jar lib/AWSImportExportWebServiceTool-1.0.jar CreateJob Import
 MyImportManifest.txt .
```

   For an Export job, enter the following `CreateJob` request on the command line.

```
CmdPrompt>java -jar lib/AWSImportExportWebServiceTool-1.0.jar CreateJob Export
 MyExportManifest.txt .
```

The `CreateJob` request includes a directory name for where you want the `SIGNATURE` file saved. In this example, the closing period (.) at the end of the command refers to the current directory. Either include the period, as shown above, or replace it with a directory name.

If there is an error, AWS Import/Export returns an error message.

If there are no errors in the request, AWS Import/Export returns a *JOB CREATED* response.

For an Amazon S3 import or export job, the response includes a job ID, a signature value, and an AWS shipping address. The response also saves a SIGNATURE file to your computer.

For an Amazon EBS or Amazon Glacier import job, the response includes a job ID and shipping instructions. It also places a PDF file on your computer and in the Amazon S3 log bucket you identified in your manifest. The PDF file contains job information, shipping instructions, and a barcode.

The following examples show an import job response and an export job response.

Example import job response

```
JOB CREATED
 JobId: ABCDE
 JobType: Import
 ***************************************
 * AwsShippingAddress *
 ***************************************
 AWS Import/Export C/O AnyCompany
 JOBID ABCDE
 123 Any Street
 Anytown, VA 22153
 ***************************************
 * SignatureFileContents *
 ***************************************
 version:2.0
 signingMethod:HmacSHA1
 jobId:ABCDE-VALIDATE-ONLY
 signature:cbfdUuhhmauYS+ABCl5R9heDK/V=
 Writing SignatureFileContents to cmdPrompt\.\SIGNATURE
```

Example export job response

```
JOB CREATED

******************************
* RECEIVED SUCCESSFUL RESPONSE *
******************************
jobId: ETU5V
signature: f7+7Dabe30tqvZkFFjlIOK5KUHg=
jobType: Export

*****************************************************
* signatureFileContents - write this to a file called *
* SIGNATURE in the root of your disk                  *
*****************************************************
version:2.0
signingMethod:HmacSHA1
jobId:ETU5V
signature:f7+7Dabe30tqvZkFFjlIOK5KUHg=


*****************************************
* SEND YOUR STORAGE DEVICE TO THIS ADDRESS *
*****************************************
AWS Import/Export
JOBID ETU5V
2646 Rainier Ave South Suite 1060
Seattle, WA 98144
United States
206-266-6868
```

The Getting Started (p. 6) section provides step-by-step examples of creating import and export jobs using the AWS Import/Export Web Services Tool.

Sending a CreateJob Request Using the AWS SDK for Java

The AWS SDK for Java exposes an API to create an import job or an export job. You can then use the information in the response to send your storage device to AWS.

Create Job Process

1	Create an instance of the `AmazonImportExportClient` class by providing your AWS credentials.
2	Create an instance of the `CreateJobRequest` class by providing job information, such as the job type and the manifest.
3	Send the create job request by calling the `AmazonImportExportClient.createRequest` method.

The following Java code sample demonstrates creating an import job using the preceding steps.

```
AmazonImportExportClient client =
   new AmazonImportExportClient(
   new PropertiesCredentials(
       ImportCreateJobSample.class.getResourceAsStream("AWSCredentials.proper
ties")));
String manifest = readManifestFile(manifestFilePath);
CreateJobRequest createRequest = new CreateJobRequest().withManifest(mani
fest).withJobType("Import");
CreateJobResult createResult = client.createJob(createRequest);
```

The following Java code sample demonstrates creating an export job using the preceding steps.

```
AmazonImportExportClient client =
   new AmazonImportExportClient(
   new PropertiesCredentials(
       ImportCreateJobSample.class.getResourceAsStream("AWSCredentials.proper
ties")));
String manifest = readManifestFile(manifestFilePath);
CreateJobRequest createRequest = new CreateJobRequest().withManifest(mani
fest).withJobType("Export");
CreateJobResult createResult = client.createJob(createRequest);
```

To send a CreateJob request and print the response using Java

1. Create a properties file, `AWSCredentials.properties`, and provide the Access Key ID and Secret Key value for your IAM user. For more information, see Using IAM with AWS Import/Export (p. 26).

   Following is an example credentials file.

   ```
   accessKey:<Your Access Key ID>
   secretKey:<Your Secret Key>
   ```

   Note
   You cannot use the same `AWSCredentials.properties` file that you use with the AWS Import/Export Web Service tool. The fields are not the same.

   The `AWSCredentials.properties` file must be saved in the same directory as the produced .class file. Thus, if your class is in a package, or if your IDE puts your .class files in a different location, you will have to determine the appropriate directory to save your properties file. Also, the path to your manifest file must be explicit, including the full path to the file (such as *`C:\\Users\\UserName\\MyManifest.txt`*).
2. Create a manifest file describing your job.

   For an example manifest, see Create Your First Amazon S3 Import Job (p. 9) or Create Your First Amazon S3 Import Job (p. 21).
3. Copy the following Java class to a file in your project.

   ```
   import java.io.BufferedReader;
   import java.io.FileReader;
   import java.io.IOException;
   import com.amazonaws.auth.PropertiesCredentials;
   import com.amazonaws.services.importexport.AmazonImportExportClient;
   import com.amazonaws.services.importexport.model.CreateJobRequest;
   import com.amazonaws.services.importexport.model.CreateJobResult;
   ```

```
public class ImportExportCreateJobSample {
 private static String manifestFilePath = "[Provide-Explicit-Manifest-File-
Path]";

    public static void main(String args[]) throws IOException {
          AmazonImportExportClient client =
            new AmazonImportExportClient(
                new PropertiesCredentials(
                    ImportExportCreateJobSample.class.getResourceAs
Stream("AWSCredentials.properties")));

          String manifest = readManifestFile(manifestFilePath);

          CreateJobRequest createRequest = new CreateJobRequest().withMani
fest(manifest).withJobType("[Import|Export]");

          // Call service.
          CreateJobResult createResult = client.createJob(createRequest);

          // Process result.
          System.out.println();
          System.out.println();
          System.out.println("*********************************");
          System.out.println("* RECEIVED SUCCESSFUL RESPONSE *");
          System.out.println("*********************************");
          System.out.println("jobId: " + createResult.getJobId());
          System.out.println("signature: " + createResult.getSignature());
          System.out.println("jobType: " + createResult.getJobType());
          System.out.println();
          Sys
tem.out.println("*****************************************************");

          System.out.println("* signatureFileContents - write this to a file
 called *");
          System.out.println("* SIGNATURE in the root of your disk
          *");
          Sys
tem.out.println("*****************************************************");

          System.out.println(createResult.getSignatureFileContents());
          System.out.println();
         System.out.println("*****************************************");

          System.out.println("* SEND YOUR STORAGE DEVICE TO THIS ADDRESS *");

         System.out.println("*****************************************");

          System.out.println(createResult.getAwsShippingAddress());
    }

   public static String readManifestFile(String filename) throws IOException
 {
          StringBuilder manifest = new StringBuilder();

         BufferedReader input = new BufferedReader(new FileReader(filename));

          try {
```

```
            String line = null;
            while ((line = input.readLine()) != null) {
                manifest.append(line);
                manifest.append(System.getProperty("line.separator"));
            }
        } finally {
            input.close();
        }
        return manifest.toString();
    }
}
```

4. Update the code by providing your manifest file path and specifying Import or Export for the job type .
5. Run the code.

   For an Amazon S3 import or export job, AWS Import/Export sends a response that includes a job ID, a signature value, and an AWS shipping address. The response also saves a SIGNATURE file to your computer.

   For an AWS EBS or Amazon Glacier import job, AWS Import/Export sends a response that includes a job ID and shipping instructions. It also places a PDF file in the Amazon S3 log bucket you identified in your manifest. The PDF file contains job information, shipping instructions, and a barcode.

Sending a CreateJob Request Using the REST API

If your application requires programmatic access to AWS Import/Export and the AWS SDK does not meet your requirements you can program directly using the REST API.

To create a job using the REST API

1. Create a manifest file describing your job.

   For an example manifest, see Create Your First Amazon S3 Import Job (p. 9) or Create Your First Amazon S3 Import Job (p. 21).
2. URL encode the manifest file.
3. Sign the request.

   For more information, go to Making Query Requests (p. 103).
4. Send a `CreateJob` request, specifying `JobType=Import` or `JobType=Export`.

   Following is a sample CreateJob request for an import job. The request includes the URL encoded manifest file in the body of the request as the *`Manifest`* option value. The request also shows the other required string parameters.

```
POST / HTTP/1.1
content-type:application/x-www-form-urlencoded;charset=utf-8
host: https://importexport.amazonaws.com
content-length:579

Action=CreateJob&Manifest=manifestVersion%3A%202.0%0Abucket%3A%20myBucket%0
eraseDevice%3A%20yes%0AreturnAddress%3A%0A%20%20%20%20name%3A%20Amazon.com%20AT
TN%20Joe%20Random%20%0A%20%20%20%20
street1%3A%201200%2012th%20Ave%20S.%0A%20%20%20%20
```

```
city%3A%20Seattle%0A%20%20%20%20stateOr
Province%3A%20WA%0A%20%20%20%20postalCode%3A%2098114%0A%20%20%20%20phoneNum
ber%3A%20206-266-0000%0A%20%20%20%20country%3A%20USAJobType=ImportSignature
Version=2&SignatureMethod=HmacSHA256&Version=2010-06-01&Signature=%2FVfkltRBOo
SUi1sWxRzN8rw%3D
```

For an Amazon S3 import or export job, AWS Import/Export sends a response that includes a job ID, a signature value, and an AWS shipping address.

For an AWS EBS or Amazon Glacier import job, AWS Import/Export sends a response that includes a job ID and shipping instructions. It also places a PDF file in the Amazon S3 log bucket you identified in your manifest. The PDF file contains job information, shipping instructions, and a barcode.

For more information about creating a job using the REST API, go to CreateJob (p. 109). You can validate the manifest and parameter values without creating a job by setting the ValidateOnly parameter to *`true`*.

Viewing Import Logs

After successfully transferring data between AWS and your portable storage device, AWS Import/Export generates a log and saves it to Amazon S3 as an object.

Topics

Log Files

The log file is a UTF-8 encoded CSV file that contains, among other things, information about each file loaded to or from your storage device.

With Amazon S3 import jobs, AWS Import/Export saves the log to the same Amazon S3 bucket as your data.

With Amazon EBS and Amazon Glacier jobs, AWS Import/Export saves the log to the bucket you specified in the *`logBucket`* option in your manifest file.

For an import job, the log name ends with the phrase `import-log-` followed by your *`JOBID`*. For an export job, the log name ends with the phrase `export-log-` followed by your *`JOBID`*. For example, if the import *`JOBID`* is *53TX4*, the log name ends in `import-log-53TX4`. By default, if you do not set *`logPrefix`* in the manifest file, a job loaded to *`mybucket`* with the *`JOBID`* of *53TX4* loads the logs to `http://mybucket.s3.amazonaws.com/import-log-53TX4`. If you set *`logPrefix`* to `logs/`, the log location is `http://s3.amazonaws.com/mybucket/logs/import-log-53TX4`.

For an export job, the log name ends with the phrase `export-log-` followed by your *`JOBID`*. For example, if the *`JOBID`* is *53TX4*, the log file name would end in `http://s3.amazonaws.com/mybucket/logs/export-log-53TX4`.

> **Note**
> If you have a log object with the same key name as an existing Amazon S3 object, the new log overwrites the existing object. You can use the *`logPrefix`* option to prevent object collisions.

Viewing the log files

You can download your log using the AWS Management Console. For more information, go to http://aws.amazon.com/console/.

Amazon S3 and Amazon EBS Log File Fields

The following table describes the fields in an Amazon S3 or Amazon EBS log file.

Field	Description
DateTime	The date and time when we processed a file, for example, `Wed Nov 12 11:07:34 PST 2008`.
File	The name of the file in the root directory of your storage device, for example, `/images/image.jpg`.
Status	Specifies either an AWS Import/Export status or one of the standard Amazon S3 REST web server HTTP status codes, for example, `200`. For more information, see Error and Status Codes (p. 125) .
Code	Specifies either an AWS Import/Export code, such as `Ignored` when the *`Status`* is 103, or the standard Amazon S3 REST error codes, for example, `OK`.
Key	Specifies where AWS Import/Export loads the file in Amazon S3. The key includes the bucket name and any prefix settings set in the manifest file, for example, `/mybucket/myprefix/images/image.jpg`.
MD5	The checksum of the object stored in Amazon S3 identified by `Key`, for example, `d2a22fcab097sample32c544`.
Bytes	The number of bytes stored in Amazon S3 for the object identified by `Key`, for example, `57344`.
Content-Type	The value of the `Content-Type` header stored in Amazon S3, for example, `image/jpeg`.
ETag	The ETag value of the object in Amazon S3.

Sample Amazon S3 Log File

The following shows a sample Amazon S3 import log file.

DateTime	File	Status	Code	Key	MD5	Bytes	Content-Type
Tue 03 Feb 200921:57:43 GMT	SIGNATURE	103	Ignore			90	application/ octet-stream
Tue 03 Feb 200921:57:43 GMT	Notice.txt	200	OK	MyBucket/ images/ Notice.txt	f60fe317bc4 97b1204b327 490959eb64	13	text/plain
Tue 03 Feb 200921:57:43 GMT	README. txt	200	OK	MyBucket/ images/ README.txt	d5a46fa2259 6d2464effba bb19000e	13	text/plain

DateTime	File	Status	Code	Key	MD5	Bytes	Content-Type
Tue 03 Feb 200921:57:43 GMT	images/ wookie1.jpg	200	OK	MyBucket/ images/images-wookie1.jpg	ccac0d0316 03ff1c299770 3a64845	497	image/jpeg
Tue 03 Feb 200921:57:43 GMT	images/ chewie.GIF	200	OK	MyBucket/ images/images-chewie.GIF	5fc22dc5 94e0c5929c0 b9585ba9e70	878	image/gif
Tue 03 Feb 200921:57:43 GMT	images/ resources/ chewie.psd	200	OK	MyBucket/ images/images-resources-chewie.psd	130b64d1 71e9bbc2be11 7f189796c	524	application/ octet-stream
Tue 03 Feb 200921:57:43 GMT	images/ chewie.TIF	200	OK	MyBucket/ images/images-chewie.TIF	823d17b 05321b96f91 74e3dcace7	458	image/tiff

Amazon Glacier Log File

The Amazon Glacier log file is uploaded to the logBucket specified in the manifest once the import completes.

An Amazon Glacier log file is a two-line text file in the following format:

```
DATE,VAULT,ARCHIVE ID,SIZE,TREE HASH,ARCHIVE COMMENT
<Date>,<Vault Name>,/<AWS Account ID>/vaults/<Vault Name>/archives/<Archive ID>,<Archive Comment>
```

Status Codes

For information about Amazon log file status codes, see Error and Status Codes (p. 125).

Working with Export Jobs

Topics

This section summarizes the process you use to export data from an Amazon S3 bucket to your storage device.

> **Note**
> AWS Import/Export doesn't support export jobs from Amazon Elastic Block Store or Amazon Glacier.

Creating Amazon S3 Export Jobs

An Amazon S3 export transfers individual objects from Amazon S3 buckets to your device, creating one file for each object. You can export from more than one bucket and you can specify which files to export using manifest file options.

> **Note**
> You cannot export Amazon S3 objects that have been transitioned to Amazon Glacier Storage Class using Amazon S3 Object Lifecycle Management. Before sending an Amazon S3 export job, check your lifecycle configuration rules to ensure that your objects will be available in Amazon S3 standard class until the export is complete. For more information, see Object Lifecycle Management in the *Amazon S3 Developer Guide*.

You must submit a separate job request for each device.

For more information, see Guidelines and Limitations (p. 130)

Amazon S3 Export Process

Export Job Process

1	You create an export manifest file that specifies how to load data onto your device, including an encryption password and details such as the name of the bucket that contains the data to export. For more information, see The Export Manifest File (p. 57). If you are going to mail us multiple storage devices, you must create a manifest file for each storage device.
2	You initiate an export job by sending a `CreateJob` request that includes the manifest file. You must submit a separate job request for each device. Your job expires after 30 days. If you do not send a device, there is no charge. You can send a CreateJob request using the AWS Import/Export Tool, the AWS Command Line Interface (CLI), the AWS SDK for Java, or the AWS REST API. The easiest method is the AWS Import/Export Tool. For details, see Sending a CreateJob Request Using the AWS Import/Export Web Service Tool (p. 46) Sending a CreateJob Request Using the AWS SDK for Java (p. 48) Sending a CreateJob Request Using the REST API (p. 51)
3	AWS Import/Export sends a response that includes a job ID, a signature value, and an AWS shipping address. The response also saves a `SIGNATURE` file to your computer. You will need this information in subsequent steps.
4	You copy the `SIGNATURE` file to the root directory of your storage device. You can use the file AWS sent or copy the signature value from the response into a new text file named `SIGNATURE`. The file name must be `SIGNATURE` and it must be in the device's root directory. Each device you send must include the unique `SIGNATURE` file for that device and that JOBID. AWS Import/Export validates the `SIGNATURE` file on your storage device before starting the data load. If the `SIGNATURE` file is missing invalid (if, for instance, it is associated with a different job request), AWS Import/Export will not perform the data load and we will return your storage device.
5	You fill out the AWS Import/Export packing slip to send your device. You must send the device to the address returned in the `CreateJob` response. You must also add the job ID to the packing slip. For more information see Filling Out the Packing Slip (p. 65) and Storage Device Requirements (p. 63).
6	You ship the device, the cables and packing slip to AWS. Make sure to include your job ID on the shipping label and on the device you are shipping. Otherwise, your job might be delayed. Your job expires after 30 days. If we receive your package after your job expires, we will return your device. You will only be charged for the shipping fees, if any. You must submit a separate job request for each device. **Note** You can send multiple devices in the same shipment. If you do, however, there are specific guidelines and limitations that govern what devices you can ship and how your devices must be packaged. If your shipment is not prepared and packed correctly, AWS Import/Export cannot process your jobs. Regardless of how many devices you ship at one time, you must submit a separate job request for each device. For complete details about packaging requirements when shipping multiple devices, see Shipping Multiple Devices (p. 68).

7	AWS Import/Export validates the signature on the root drive of your storage device. If the signature doesn't match the signature from the `CreateJob` response, AWS Import/Export can't load your data. Once your storage device arrives at AWS, your data transfer typically begins by the end of the next business day. The time line for exporting your data depends on a number of factors, including the availability of an export station, the amount of data to export, and the data transfer rate of your device.
8	AWS reformats your device and encrypts your data using the password you provided in your manifest.
9	We repack your storage device and ship it to the return shipping address listed in your manifest file. We do not ship to post office boxes.
10	You use TrueCrypt to decrypt your device with the password you provided in your manifest. For more information, see Using TrueCrypt Encryption (p. 66)

Creating Export Manifests

You provide your job information and specify how to transfer your data between your storage device and AWS by creating a manifest file. The manifest file is a YAML-formatted text file. It consists of a set of name-value pairs that supply required information such as your device ID, log bucket, return address, and so on. For more information about YAML, go to http://yaml.org.

Import and export manifest files use a common set of options as well as options that are unique to each type.

A set of customs manifest file options is required for any storage device shipped internationally.

For a complete list of manifest options, go to Manifest File Options Reference (p. 76).

Export Manifest Example

An S3 export job has a set of required options.

This sample Amazon S3 export manifest file includes only the required options.

```
manifestVersion:2.0
generator:text editor
deviceId:ABCDE
trueCryptPassword: [password]
fileSystem:NTFS
serviceLevel:expeditedShipping
targetDirectory:/
recoveryDirectory:EXPORT-RECOVERY
logBucket:ImportExportBucket
notificationEmail: jane.roe@example.net
operations:
  - exportBucket: ImportExportBucket
returnAddress:
   name: Jane Roe
   company: Example Corp.
   street1: 123 Any Street
   city: Anytown
```

```
stateOrProvince: WA
postalCode: 91011-1111
phoneNumber: 206-555-1111
country: USA
```

For a complete list of manifest options, see Manifest File Options Reference (p. 76). For examples of commonly used manifest options, see Working with Amazon S3 Export Manifests (p. 58).

Working with Amazon S3 Export Manifests

Topics

This section gives examples of using export manifest options.

Shipping Internationally

When shipping devices internationally except within the European Union you must include the customs option in the manifest. For more information about the customs related manifest options, see Customs Manifest File Option (p. 79).

Expediting the Return of Your Storage Device

When returning your device to a U.S. address from a U.S. region bucket, you can use the *`serviceLevel`* manifest option to upgrade your shipping. The default *`serviceLevel`* option is standard shipping, which includes free ground domestic shipping. When you specify *`expeditedShipping`*, your device is delivered using two-day shipping for an additional fee. For information about return shipping fees and services, go to the AWS Import/Export Calculator.

Preventing Log File Name Collisions

The *`logPrefix`* option defines a string AWS Import/Export inserts between the bucket name and log report name to prevent name collisions. The log file name always ends with the phrase `export-log-` followed by your *`JobId`*.

Example path and file name without *`logPrefix`*:

`http://mybucket.s3.amazonaws.com/export-log-53TX4`.

Example path and file name with *`logPrefix`* set to `logs/`:

`http://mybucket.s3.amazonaws.com/logs/export-log-53TX4`

> **Note**
> We do not include a slash (/) automatically. If you don't include the slash at the end of the value for *`logPrefix`*, the value is concatenated to the log file name. For example, if your *`logPrefix`* is `taxonomy` and your root directory contains the file `phylum.jpg`, your key would become `taxonomyphylum.jpg` instead of `taxonomy/phylum.jpg`.

`logPrefix` + `export-log-JOBID` cannot be longer than 1024 bytes. If it is, AWS Import/Export returns an *`InvalidManifestField`* error from the `CreateJob` action.

Managing File Exports

Topics

You may need to limit which files are exported from a bucket. For instance, you might not want all of the files in a bucket, or you may have too many files for a single device and need to export to multiple devices.

This section describes how to configure the *operations* manifest file option to manage those tasks.

If you need to export files from multiple buckets, you can specify multiple exportBucket options in the same manifest file.

Operations Export Manifest Option

The *operations* manifest option specifies a list of one or more export operations. AWS Import/Export exports keys in alphabetical order. That is, it carries out the operations in the order listed and then in the alphabetical order of the keys affected by each export operation. The files on the storage device remain in strict alphabetical order even if the data to export exceeds the capacity of the storage device; AWS Import/Export does not, for example, squeeze in an extra file or two that are out of order just because there is room on the storage device. This alphabetical order enables you to continue the export easily: create a new export job using the key of the last file exported as the *beginMarker* in your new export job.

The *operations* subfields *prefix*, *beginMarker*, and *endMarker* limit the data exported from a bucket.

prefix	Restricts the keys exported to those located in the path specified by *prefix*.
beginMarker	Limits the keys exported to those that occur alphabetically *after* this option value.
endMarker	A subfield of *operations* that limits the keys exported to those that occur alphabetically *before and including* the key that matches *endMarker*. Any keys that are prefixed by *endMarker* are not included.

AWS Import/Export uses the entire key when naming the objects exported to your storage device. For example, AWS Import/Export would map the key `starwars.s3.amazonaws.com/images/jedi.jpg` to the following file on your storage device `/starwars/images/jedi.jpg`. For Microsoft Windows, AWS Import/Export would map the same key to `\starwars\images\jedi.jpg`.

The following sections talk in depth about some of the export fields you use with *operations*.

Limiting the Exported Data

Use *beginMarker* and *endMarker* to specify the beginning and the ending of the keys to export.

The export job starts with the key after the *beginMarker*. In other words, the *beginMarker* is not included in the export job, although the *endMarker* is included. For example, the following options export all of the keys that occur alphabetically between `html/index.html` and `images/kate.jpg`.

The export will not include `html/index.html`, will include `images/kate.jpg`, and will not include `images/kate.png`.

```
operations:
  - exportBucket: mybucket
    beginMarker: html/index.html
    endMarker: images/kate.jpg
```

Note
It's possible to construct an export operation that includes no keys. For example, if you specify `image` for *`prefix`* and `foo` for *`endMarker`*, there would be no matching keys because `foo` comes alphabetically before `image`.

Exporting Data From Multiple Buckets

To export multiple subsets of data from one or more buckets, specify multiple *`exportBucket`* operations followed by appropriate export options. Each dash (-) after *`operations`* specifies a new set of data to export. Each entry requires an *`exportBucket`* manifest option. The following example shows four data sets to export from three different buckets: `mybucket`, `lost`, and `threescompany`.

All buckets must be in the same Amazon S3 region and you must have read permissions on each bucket and each object.

To export data from more than one bucket

Use *`exportBucket`* to specify each bucket you want to export data from.

In the following example, AWS Import/Export exports all keys under `/images` in `mybucket` to `starwars/image-backup` on the storage device.

```
operations:
  - exportBucket: mybucket
    prefix: images
    beginMarker: images/starwars/jedi.jpg
    endMarker: images/starwars/masterwindoo.jpg
  - exportBucket: lost
  - exportBucket: threescompany
    prefix: images
    endMarker: images/characters/mrfurley.jpg
```

The order in which AWS Import/Export exports keys is the order of the operations you specify in the *`operations`* manifest option, and then within each operation, by the alphabetical order of the keys to be exported. In the previous procedure, for example, AWS Import/Export exports all of the keys from `mybucket` in alphabetical order, and then from `lost` in alphabetical order.

Note
AWS Import/Export exports keys in parallel, but the finished order of files on the storage device is by operation and then alphabetical. This functionality becomes important when the storage device has insufficient capacity to hold all of the data that needs to be exported. This ordering enables you to easily continue the export with a new export job using the key of the last key exported as your *`beginMarker`* in your new export job.

Specifying the Target Directory

AWS Import/Export gives you the ability to specify the location for the exported files on your storage device using the export manifest options *`operations`* and *`targetDirectory`*.

To export data to a specific directory

Use the *`targetDirectory`* to specify the directory on the storage device you want to export files into.

In the following example, AWS Import/Export exports the contents of the bucket named `bucket-data-comes-from` to the directory, `myexport/stuff`, on the storage device.

The default *`targetDirectory`* is the bucket name.

```
operations:
  - exportBucket: bucket-data-comes-from
    targetDirectory: myexport/stuff
```

Collecting Files That Have Naming Errors

Amazon S3 object keys might contain characters that are incompatible with the file system you specified in your manifest. When AWS Import/Export exports files to your device, we attempt to remove illegal characters and map the Amazon S3 keys as closely as possible to a file name that is permitted by the file system. If a file name cannot be mapped, the file is copied to a directory named EXPORT-RECOVERY. You can optionally specify a different directory name by specifying the *`recoveryDirectory`* option in your export manifest.

Files are written to the recovery directory beginning in a subdirectory named 0000. The files are named sequentially from 0000 to 9999, then a new subdirectory named 0001 is created, and so on. This process continues until one of the following conditions occurs:

- All files are stored
- The file system is full
- 100 million files are written to the *`recoveryDirectory`* directory
- The *`recoveryDirectory`* directory exceeds the file system's maximum file size limit

For example, if your recovery directory is named EXPORT-RECOVERY, your hierarchy of files and directories will look similar to this:

```
EXPORT-RECOVERY/
  0000/
    0000
    0001
    …
    9999
  0001/
  …
  9999/
```

Viewing Import/Export Logs

After transferring data between AWS and your portable storage device, AWS Import/Export generates a log file. The log file is a CSV file that contains, among other things, information about each file loaded to or from your storage device. With export jobs, AWS Import/Export saves the log files to the bucket you specify with the *`logBucket`* manifest option.

The log file name of an export job always ends with the phrase `export-log-` followed by your *`JOBID`*. For example, if the *`JOBID`* is *53TX4*, the log file name would end in `export-log-53TX4`.

Note
If you have a log file name with the same name as an existing Amazon S3 object, the new log file overwrites the existing object. You can use the *logPrefix* field to prevent object collisions. For more information about the *logPrefix*, see Preventing Log File Name Collisions (p. 58).

Sample Log File

AWS Import/Export uses the same set of fields in an import or an export log. For more information about log files and descriptions of log file fields, see Amazon S3 and Amazon EBS Log File Fields (p. 53).

Shipping Your Storage Device

Topics

The following sections explain how to prepare and ship your storage device to AWS.

You can use any carrier of your choice to send packages to AWS, however we highly recommend that you use a shipper that provides a tracking number. For data loads in US Region buckets, AWS returns United States-bound packages via standard UPS ground and international-bound packages via DHL Express Worldwide. For data loads in the EU (Ireland) Region, AWS returns packages via An Post. For data loads in Asia Pacific (Singapore) buckets, AWS returns packages via DHL Express Worldwide.

Your device will be delivered to an AWS sorting facility and then forwarded by Amazon to the AWS data center. Your shipper will only have tracking information to the sorting facility. AWS updates the job status when the package is received by the AWS data center. When you send a `GetStatus` command, the status of your job will not show `At AWS` until the shipment has been received at the AWS data center.

Storage Device Requirements

To connect your storage device to one of our AWS Import/Export stations, your device must meet the following requirements:

- Compatible with Red Hat Linux and warranted by the manufacturer to support eSATA, USB 3.0, or USB 2.0 interface on Linux.
- Maximum device size is 14 inches/35 centimeters high by 19 inches/48 centimeters wide by 36 inches/91 centimeters deep (8Us in a standard 19 inch/48 centimeter rack).
- Maximum weight is 50 pounds/22.5 kilograms.
- Maximum device capacity is 16 TB

- Power requirements vary by region. For more information, go to the Selecting Your Storage Device section on the AWS Import/Export Product Details page.

Important
You must ship your storage device with its *power supply*, *power cable*, and *interface cables*. Without these we can't transfer your data and will return your device.

Preparing Your Storage Device for Amazon S3 Import and Export Jobs

The process for preparing your storage device is as follows. Individual steps are described in detail in the following topics.

1	Ensure that your storage device conforms to AWS requirements. For more information, see Storage Device Requirements (p. 63).
2	For import to Amazon S3, you can encrypt your data before you send it the device. For export to Amazon S3, you provide a password that we will use to encrypt your data. For more information, see Using TrueCrypt Encryption (p. 66).
3	Label both your device and package with the *`JobId`* value from the `CreateJob` response.
4	Copy the `SIGNATURE` value to a file named `SIGNATURE` in the root directory of your storage device.
5	Print out and fill in the packing slip with your JobId, unique device identifier, and contact information.
6	Pack your *power supply*, *power cable*, and *interface cables* along with your device. **Important** Secure your device within the shipping box to prevent shifting in transit. For example, wrapping your device in bubble wrap will help prevent shifting and give added protection from damage.

For a checklist that covers the steps for shipping a device for an Amazon S3 import or export job, go to the AWS Import/Export Pack and Ship Checklist on our web site.

Copying the Signature Value

For Amazon S3 import or export jobs, your storage device must have on its root directory a file named `SIGNATURE`. The `CreateJob` response includes the *`SignatureFileContents`* text and a file named `SIGNATURE` in its response. The file name must be `SIGNATURE`, all caps, and it must be in the device's root directory.

To copy the signature value to your storage device

1. Locate the `SIGNATURE` on your computer, or create a file named `SIGNATURE` and copy the *`SignatureFileContents`* text into it.

   You can also get the *`SignatureFileContents`* value by submitting a `GetStatus` request.

   The file name, `SIGNATURE`, must be all caps.

2. Copy the `SIGNATURE` file to your storage device in the root directory.

   Each device you send must include the unique `SIGNATURE` file for that device and that JOBID. AWS Import/Export validates the `SIGNATURE` file on your storage device before starting the data load. If the `SIGNATURE` file is missing or invalid (if, for instance, it is associated with a different job request), AWS Import/Export will not perform the data load and we will return your storage device.

Filling Out the Packing Slip

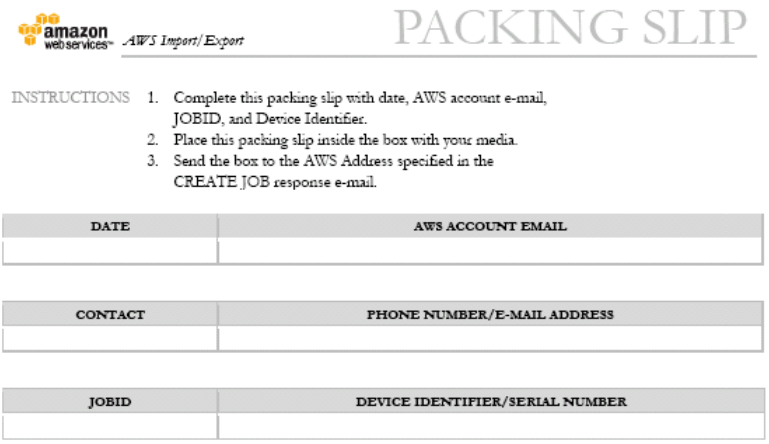
For Amazon S3 import or export jobs only, you must fill out the AWS Import/Export packing slip.

To fill out your packing slip

1. Go to http://s3.amazonaws.com/awsimportexport/AWS_Import_Export_Packing_Slip.pdf and print the packing slip.
2. Fill in the date, your AWS account email, a contact name, a contact phone number or email, and a storage device ID.

   Each package can contain only one storage device and each storage device can have only one device ID, such as a serial number.

The following is an example packing slip.



amazon web services — AWS Import/Export

PACKING SLIP

INSTRUCTIONS
1. Complete this packing slip with date, AWS account e-mail, JOBID, and Device Identifier.
2. Place this packing slip inside the box with your media.
3. Send the box to the AWS Address specified in the CREATE JOB response e-mail.

DATE	AWS ACCOUNT EMAIL

CONTACT	PHONE NUMBER/E-MAIL ADDRESS

JOBID	DEVICE IDENTIFIER/SERIAL NUMBER

Preparing Your Storage Device for Amazon EBS and Amazon Glacier Import Jobs

The process for preparing your storage device is as follows. Individual steps are described in detail in the following topics.

1	Ensure that your storage device conforms to AWS requirements. For more information, see Storage Device Requirements (p. 63).

2	Print the PDF file from the `CreateJob` response and tape the bar code portion to your device. Be especially careful to attach the right barcode to the right device. You submit a separate job request for each device, and each job request generates a unique barcode. **Important** Attach the barcode securely to your device, with tape on all four sides. Do not shrink the barcode or obscure it in any way. AWS Import/Export uses the signature barcode to validate your storage device before starting the data load. If the barcode is separated from your device, we cannot validate it and we will return your device without performing the data load.
3	From the PDF file from the `CreateJob` response, put the shipping information portion on the shipping box containing your device.
4	Pack your *power supply*, *power cable*, and *interface cables* along with your device. **Important** Secure your device within the shipping box to prevent shifting in transit. For example, wrapping your device in bubble wrap will help prevent shifting and give added protection from damage.

Using TrueCrypt Encryption

For added security, AWS Import/Export supports data encryption using TrueCrypt for import to Amazon S3 and export from Amazon S3. TrueCrypt is an open-source disk encryption application.

TrueCrypt is the only device encryption supported by AWS Import/Export. For information about how to download, install, and use TrueCrypt, go to www.truecrypt.org.

For import to Amazon S3, you can use TrueCrypt to encrypt your data before sending it to AWS Import/Export. You will need to include your TrueCrypt password in your import manifest.

For import to Amazon EBS or Amazon Glacier, you can use any encryption method you choose. AWS does not decrypt your data for import to Amazon EBS or Amazon Glacier. We strongly encourage you to encrypt your data.

For export from Amazon S3, AWS always encrypts your data using TrueCrypt with the TrueCrypt password in your export manifest.

The following sections detail the encryption process for import to Amazon S3 and export from Amazon S3.

Encryption for Import to Amazon S3

Follow the instructions in the TrueCrypt documentation to create a new TrueCrypt volume. AWS Import/Export supports only TrueCrypt volumes created as non-system partitions or encrypted file containers. Do not use the **Encrypt the system partition or the entire system drive** option.

To ensure that we can decrypt your device, choose the following options when creating a TrueCrypt volume:

- Select either the **Create an encrypted file container** option or the **Encrypt a non-system partition/drive** option.
- For **Volume Type**, select **Standard TrueCrypt volume**. Do not create a hidden volume.

- If you are creating an encrypted file container, for **Volume Location**, create a file in your device's home directory and name the file `<JOBID>.tc`, where <JOBID> is the job ID associated with the device. For example, `1B23C.tc`.
- If you are encrypting non-system partition, for **Volume Location**, select the partition to be encrypted. For **Volume Creation Mode**, if your volume already contains files, select **Encrypt partition in place**. TrueCrypt will not format the partition or erase any files. Encrypt partition in place takes longer than encrypting an empty volume. If the partition is empty, select **Create encrypted volume and format it**.
- For **Encryption Options**, use **AES** for the **Encryption Algorithm** and **RIPEMD-160** for the **Hash Algorithm**.
- Create a password for your TrueCrypt volume. You will include this password in the job manifest that you create with the AWS Import/Export tool. Without this password, we won't be able to access any data on the encrypted partition.

  Important
  Do not lose your password, or you will not be able to decrypt your device. AWS Import/Export will not send the password with the encrypted device.
- If you are prompted to select a volume format, select **NTFS**.

After you create your TrueCrypt volume, use TrueCrypt to mount the volume and then copy your files into the volume. Your device must contain only one partition or container and no other files. If you use a file container, the container must be named `<JOBID>.tc`, using the job ID associated with the device. Copy the SIGNATURE file to the root directory of the encrypted volume.

When you complete your manifest file in preparation for submitting a job request, include the password for the `trueCryptPassword` option. When AWS receives your device, we will attempt to mount the device. If the volume is encrypted, we decrypt it using the password you supplied. If the volume is not encrypted, we will look for a container file named `<JOBID>.tc`, using the job ID associated with the device. We will decrypt the container using the password you supplied.

Following a successful import, we will erase the device and ship it to the address provided on your manifest.

If any of the following conditions occur, AWS will erase your device without performing the import and ship it to the address provided on your manifest:

- You specified a TrueCrypt password in your manifest, but the partition is not encrypted or no encrypted container named `<JOBID>.tc` exists on the device.
- More than one container exists on the device.
- More than one partition exists on the device.
- AWS is not able to decrypt the partition or the container.

If we are unable to erase the data on the device, we will schedule it for destruction and our support team will contact you using the email address specified in the manifest file.

Encryption with Export from Amazon S3

When you create an Export to Amazon S3 job, you specify a TrueCrypt password in the export manifest. AWS creates a TrueCrypt encrypted volume in a file container on your device using the password you provided, and then copies the data to the container. The container is named `JOBID.tc`, using the job ID associated with your device. When you receive your device, you will use TrueCrypt with the password you provided with your export manifest to decrypt the container and mount the volume.

Important
Do not lose your password, or you will not be able to decrypt your device. AWS Import/Export will not send the password with the encrypted device.

For information about how to install and use TrueCrypt, go to www.truecrypt.org.

Packing Your Storage Device

This section describes how to pack your media.

To pack your storage device

1. Make sure that you package your equipment appropriately. Your shipper can provide packing guidelines.

   Important
   Although AWS has a number of internal controls and procedures to prevent loss, damage or disclosure of your data, AWS is not responsible for damages associated with loss or inadvertent disclosure of data; or the loss, damage, or destruction of the physical hardware. You should always retain a back-up copy of your data. For information about failed jobs, see Failed Jobs (p. 75).

2. Enclose your device and your device's *power supply*, *power cable*, and *interface cables*.
3. For an Amazon S3 import or export job, enclose your filled-out packing slip. For an Amazon EBS or Amazon Glacier import job, make sure the bar code from the PDF is taped to your device, and the shipping information is on the outside of the box containing your device.

AWS Shipping Addresses

You will ship your storage device to an address in the United States, Europe, or Asia. The *AwsShippingAddress* value in the `CreateJob` and `GetStatus` responses contain the correct address to ship your storage device to.

Your device will be delivered to an AWS sorting facility and then forwarded by Amazon to the AWS data center. Your shipper will only have tracking information to the sorting facility. AWS updates the job status when the package is received by the AWS data center. When you send a `GetStatus` command, the status of your job will not show `At AWS` until the shipment has been received at the AWS data center.

Important
To access Amazon S3 buckets, Amazon EBS snapshots, or Amazon Glacier vaults that are in the EU Region, the shipping device must originate from and be shipped back to a location in the European Union.

Important
If you ship your storage device to an address not specified in the `CreateJob` response, AWS Import/Export will return your storage device without performing the job and you will be charged for any applicable return shipping charges and device-handling fees. You should always verify the address before shipping.

Shipping Multiple Devices

As a general rule, you must ship each device in a separate package. AWS Import/Export will, however, accept multiple devices in a single shipment if you follow the guidelines and limitations that are outlined in this topic.

Note
These requirements are only for shipping multiple devices per package. If you are shipping each device separately, see Packing Your Storage Device (p. 68).

If you ship multiple devices in a single shipment, your shipment must meet all of these criteria:

- Create a separate Import/Export job request, with a unique job ID, for each device.
- You may ship up to eight devices per container.
- The devices must be packed in a hard case designed to safely ship the devices, with a separate slot or compartment for each device.
- Label each slot and each hard drive with the corresponding job ID.
- If the devices require separate power supplies, power cables, and interface cables, be sure all of the necessary components are packaged with each device. Label each component with the corresponding job ID.
- Be sure all the labels are attached securely.

The following types of devices have additional specific requirements.

- **Bare SATA drives** — Bare drives require special care. Ship bare SATA drives in specially designed hard cases with a separate slot or compartment for each device.
- **Single-spindle external hard disks** — Pack your devices securely with bubble wrap or similar material so that the devices cannot shift in the container. It is especially important to include all cables, cords, and power supplies.
- **RAID devices** — These devices are often not designed for shipping with drives already mounted. Consult the manufacturer's recommendations for preparing the device for shipping. Pack your devices securely with bubble wrap or similar material so that the devices cannot shift in the container. It is especially important to include all cables, cords, and power supplies.
- **Multi-RAID devices** — Multi-RAID devices present special packaging problems. We recommend that you take extra precautions to pack these devices securely. Shipping multiple devices per shipment may incur larger return shipping costs.

Note
Shipping multiple devices per shipment may incur larger return shipping costs.

Mailing Your Storage Device

This section describes how to fill out the shipping label and send your storage device in a package to AWS. There may be a charge for shipping your storage device back to you. For more information, see Return Shipping (p. 5).

To mail your storage device

1. Fill out the shipping label with the AWS shipping address specified in the `CreateJob` response along with the unique job ID. For example, if your job ID is HBNJB and you're shipping your storage device to Virginia, your shipping label would be:

```
AWS Import/Export C/O Dynamex
JOBID HBNJB
7664-A Fullerton Road
Springfield, VA 22153
United States
703-440-8069
```

Important
The shipping label must include a valid job ID. If that information is missing, AWS will refuse delivery of your storage device and your carrier will process it as an undelivered package. For more information about the job ID, see Job (p. 2).

2. Fill in your return shipping address.

   Include a contact phone number in case we have a problem processing your storage device.

   Note
   You can ship your storage device to AWS from one address and have the device returned to another address.

   Important
   If AWS Import/Export refuses the delivery of your package, the carrier uses the return shipping address to return your storage device. After AWS processes your job, however, we return your storage device to the return address you specified in your manifest file, not the return address listed on your shipping label.

3. When shipping storage devices internationally to and from the United States or Singapore, you must include the customs manifest options. For example, if you are shipping a storage device from a non-U.S. address or are requesting that AWS return your device to a non-U.S. address when doing a data load to a US Region bucket, you must include manifest options for United States Customs. This also applies for Singapore Customs when doing data load to a Singapore Region bucket. For more information, go to Customs Manifest File Option (p. 79).

   Important
   To access Amazon S3 buckets, Amazon EBS snapshots, or Amazon Glacier vaults that are in the EU Region, the shipping device must originate from and be shipped back to a location in the European Union.
   A customs manifest is not required for shipments within the EU.

4. Ship your package with the carrier of your choice.

Important
If you are shipping internationally to AWS, you must pay all duties and taxes on your shipment. AWS will not accept any packages that have unpaid duties and taxes. You are responsible for shipping charges to AWS (including any applicable duty and taxes). Packages with unpaid duty and/or taxes will be refused upon arrival. If your Amazon S3 bucket is located in the US, Northern California, or the Singapore Region, you are also responsible for any applicable duties and taxes on return shipments sent to addresses outside of the United States or Singapore. These duties and taxes will be charged directly by our return shipping carrier upon delivery of your package.

Managing Your Jobs

Topics

AWS Import/Export service provides ways for you to manage your import and export jobs. You can get status of a job or list the jobs you have created, and you can modify or delete a job. If your job fails, AWS will notify you.

Listing Your Jobs

You use `ListJobs` to list jobs. You can only list the jobs that you own. You might list jobs so that you can then get their status.

To list your jobs using the command line client

Use the `ListJobs` command.

```
> java -jar lib\AWSImportExportWebServiceTool-1.0.jar ListJobs
```

To list your jobs using REST

Submit a `ListJobs` request.

```
POST / HTTP/1.1
content-type:application/x-www-form-urlencoded;charset=utf-8
host: https://importexport.amazonaws.com
content-length:155
Action=ListJobs&SignatureVersion=2
```

```
&SignatureMethod=HmacSHA256&Version=2010-06-01&Signature=%2FVfkltRBOo
SUi1sWxRzN8rw%3D
```

You can limit the responses you get by using the *MaxJobs* parameter. The following example shows how to list up to 25 import jobs. These jobs are listed from newest to oldest.

```
POST / HTTP/1.1
content-type:application/x-www-form-urlencoded;charset=utf-8
host: https://importexport.amazonaws.com
content-length:163

Action=ListJobs&MaxJobs=25&SignatureVersion=2
&SignatureMethod=HmacSHA256&Version=2010-06-01&Signature=%2FVfkltRBOo
SUi1sWxRzN8rw%3D
```

Updating a Job

You use `UpdateJob` to update the specifics of how you want a job done. You specify a job using the job ID returned in the `CreateJob` response. You update a job request by replacing the original manifest with the manifest you include in this request. You can either modify the original manifest you submitted or start over and create a new manifest file. In either case, the original manifest is discarded.

You can only use `UpdateJob` after submitting a `CreateJob` request, but before the data transfer starts, and you can only use it on jobs you own.

To update a job using the command line client

Use the `UpdateJob` command with the new or updated manifest.

```
>  java -jar lib\AWSImportExportWebServiceTool-1.0.jar UpdateJob JOBID Export
C:\DIRECTORY\updatedmanifest.txt
```

To update a job using REST

1. Modify the original manifest you submitted or start over and create a new manifest file.
2. URL-encode the manifest file.
3. Send an `Update` request.

The following example submits the above manifest file to update the job identified by *JOBID*.

```
POST / HTTP/1.1
content-type:application/x-www-form-urlencoded;charset=utf-8
host: https://importexport.amazonaws.com
content-length:741

Action=UpdateJob&JobId=JOBID&Manifest=manifestVersion
%3A%201.0%0Abucket%3A%20XMSAGE
AreturnAddress%3A%0A%20%20%20%20name%3A%20Amazon.com%20ATTN%20
Joe%20Random%20%0A%20%20%20%20street1%3A%201200%111%20Nosuch%20Ave%20
S.%0A%20%20%20%20city%3A%20Seattle%0A%20%20%20%20
stateOrProvince%3A%20WA%0A%20%20%20%20postalCode%3A%2098114%0A%20%20%20%20
phoneNumber%3A%20206-266-0000%0A%20%20%20%20country%3A%20USA%0
```

```
AsetContentEncodingForGzFiles%3A%20yes%0Aexpires%3A%20tomorrow%0
AcontentLanguage%3A%20Spanish%0Aacl%3A%20public-read%0AcacheControl%3A%20
max-age%3D3600&JobType=Import
&SignatureVersion=2&SignatureMethod=HmacSHA256&Version=2010-06-01
&Signature=%2FVfkltRBOoSUi1sWxRzN8rw%3D
```

Amazon Import/Export returns a response to notify you whether we were able to update your job.

Getting Shipping Information

You use `GetShipInfo` to get the shipping information for a job. For manifestVersion 3.0 jobs, you get a PDF containing shipping instructions with a unique bar code to identify and authenticate your device.

The `GetShipInfo` action is not supported for REST requests, and can only be issues using the command line client. The command line client supports `GetShipInfo` for both *`manifestVersion`* 2.0 and *`manifestVersion`* 3.0 jobs.

To get shipping information using the command line client

Use the `GetShipInfo` command.

The following command retrieves the shipping information for the job *`JOBID`*.

```
>java -jar lib\AWSImportExportWebServiceTool-1.0.jar GetShipInfo JOBID
```

If your manifest file for the original order specified *`manifestVersion:2.0`* , the shipping information is returned in the current window. If your manifest file specified *`manifestVersion:3.0`* the current window displays the path to a PDF file containing your shipping information with some instructions on what to do with the PDF file.

Getting the Status of Your Jobs

You might like to know if the job is waiting to start, in progress, or completed. For some operations, such as `CancelJob` and `UpdateJob`, you might use `GetStatus` first because jobs cannot be canceled or updated if their status is `completed`.

When you ship your device, it will be delivered to a sorting facility, and then forwarded on to an AWS data center. When you send a GetStatus command, the status of your job will not show `At AWS` until the shipment has been received at the AWS data center.

> **Note**
> If you need assistance with your AWS Import/Export job, contact AWS Support Center.

`GetStatus` returns the status of a specified job. You specify a job using the job ID returned in the `CreateJob` response. Example job statuses are pending, completed, in progress, device error, and expired. For more information about statuses, go to GetStatus (p. 113). You can only get the status of jobs you own.

To get the status of a job using the command line tool

Use the `GetStatus` command.

The following command returns the status of the job identified by *`JOBID`*.

```
> java -jar lib\AWSImportExportWebServiceTool-1.0.jar GetStatus JOBID
```

To get the status of a job using REST

Submit a `GetStatus` request.

The following request returns the status of the job identified by *`JOBID`*.

```
POST / HTTP/1.1
content-type:application/x-www-form-urlencoded;charset=utf-8
host: https://importexport.amazonaws.com
content-length:189

Action=GetStatus&JobId=JOBID&JobType=Import
&SignatureVersion=2&SignatureMethod=HmacSHA256&Version=2010-06-01
&SignatureMethod=%2FVfkltRBOoSUi1sWxRzN8rw%3D
```

Amazon Import/Export returns the status of your job and some fields from the manifest. For more information about the API operation, go to GetStatus (p. 113).

Canceling a Job

You use `CancelJob` to cancel a job. You specify the job to cancel using the job ID returned in the `CreateJob` response. The operation fails if the job is already complete. We can, however, cancel a job in progress. You can only cancel jobs you own.

To cancel a job using the command line client

Use the `CancelJob` command.

The following command cancels the job *`JOBID`*.

```
> java -jar lib\AWSImportExportWebServiceTool-1.0.jar CancelJob JOBID
```

To cancel a job using REST

Submit a `CancelJob` request.

```
POST / HTTP/1.1
content-type:application/x-www-form-urlencoded;charset=utf-8
host: https://importexport.amazonaws.com
content-length:189

Action=CancelJob&JobId=JOBID&JobType=Import
&SignatureVersion=2&SignatureMethod=HmacSHA256&Version=2010-06-01
&SignatureMethod=%2FVfkltRBOoSUi1sWxRzN8rw%3D
```

Amazon Import/Export returns a response to notify you whether we were able to cancel your job. For more information about the API operation, go to CancelJob (p. 107).

Failed Jobs

Sometimes Import/Export jobs may fail due to various reasons like bad sectors, unsupported file systems, damaged enclosures, shipping damages, or old worn out hardware. AWS will notify you via the email address specified in the manifest file if a job fails.

It is AWS policy that all devices shipped from AWS facilities must be completely erased or only contain data encrypted by AWS. Failed Import/Export jobs are treated according to this policy.

If an import job fails, we will erase the contents of your device and return it to you. If device erasure cannot be performed, our support team will contact you with further information. Please note that some files may have already been imported into AWS before the job is cancelled.

If an export job fails, we will stop the export job and return your device to you. Please note that some files may have already been exported onto your device before this job was cancelled. These files are encrypted with the encryption password you provided. If we are unable to initiate the export job, our support team will contact you with further information.

Manifest File Options Reference

The following sections describe the options in the manifest file.

The common manifest file options apply to all manifest files, including Amazon S3 import and export jobs and Amazon EBS and Amazon Glacier import jobs.

Include the customs manifest file option if you are shipping your device internationally.

Topics

Common Manifest File Options

The manifest file includes information related to processing your data and returning your storage device. The following table explains options that are common to all manifest files.

Option	Description	Required
accessKeyId	The `accessKeyId` option is deprecated. It is retained only for backward compatibility. Do not use `accessKeyId` with new AWS Import/Export jobs.	No

Option	Description	Required
deviceId	Your storage device's serial number or other unique identifier. If your device does not have a unique number, please place a sticker with its own unique identifier onto the device and then enter that identifier as the *deviceId*. The following is an example device ID. `deviceId: MyCompany489732` Type: String Default: None	Yes
generator	A field describing the application that put together the manifest file and submitted the `CreateJob` or `UpdateJob` request. The following is an example of generator. `generator: AWS Import/Export Sample App 1.0` Type: String Default: None	No
logBucket	The Amazon S3 bucket where AWS Import/Export saves your log. You must create a bucket before we can save log data to it; we do not create the bucket for you. The IAM user that creates the job must have write permissions on this bucket. The following is an example of a *logBucket* value. **Note** For import to Amazon EBS, if your storage device's capacity exceeds 1TB, a device image is stored as two or more objects to your specified Amazon S3 log bucket. `logBucket: logging-bucket` Type: String Default: The import *bucket* for import to Amazon S3, otherwise None. Constraints: Must be an existing Amazon S3 bucket name.	Yes, except for import to Amazon S3.
logPrefix	A string to insert between the bucket name and log report name to prevent name collisions. For more information, go to Preventing Log File Name Collisions (p. 58). The following is an example of a *logPrefix*. `logPrefix: logs/` Type: String Default: None	No

Option	Description	Required
manifestVersion	The version of the manifest specification used to write the manifest file. The manifest version must be 2.0 for import and export to Amazon S3 and 3.0 for import to Amazon EBS and Amazon Glacier. Type: String Default: None Valid Values: 2.0 | 3.0	Yes
notificationEmail	A semicolon delimited list of email addresses to which you want us to send job completed notification emails. `notificationEmail: joe.doe@example.com` Type: String Default: None	Yes

returnAddress	The address we return your storage device to after completing the job.
	
	The returnAddress option includes subfields, as shown in the following example.
	
	`returnAddress:`
	`  name: Jane Roe`
	`  company: Example Corp.`
	`  street1: 123 Any Street`
	`  city: Anytown`
	`  stateOrProvince: WA`
	`  postalCode: 91011-1111`
	`  phoneNumber: 206-555-1111`
	`  country: USA`
	
	The `stateOrProvince` subfield is required for USA and Canada. The subfields *`street2`* and *`street3`* are optional.
	
	Type: String
	
	Default: None
	
	Constraints: Your return address must be a physical street address and not a post office box.
		Yes

Option	Description	Required
serviceLevel	The service level for the job. For more information about specifying your job's service levels, go to Expediting the Return of Your Storage Device (p. 39). `serviceLevel: expeditedShipping` Type: String Default: standard Valid Values: standard | expeditedShipping	No

Example Import to Amazon S3 Manifest Using Common Options

The following example is the manifest file content for an import job showing most of the common manifest file options, including expedited shipping. It also includes the *`bucket`* option, the *`eraseDevice`* option, and the `notificationEmail` option because they required for Amazon S3 manifest files.

```
manifestVersion: 2.0
bucket: data-load-bucket
deviceId: ABCDE
generator: AWS ImportExport Web Service Tool 1.0
logPrefix: logs
logBucket: iemanifest-log-bucket
prefix: imported/
eraseDevice: yes
notificationEmail: john.doe@example.com;jane.roe@example.com
returnAddress:
   name: Jane Roe
   company: Example Corp.
   street1: 123 Any Street
   city: Anytown
   stateOrProvince: WA
   postalCode: 91011-1111
   phoneNumber: 206-555-1111
   country: USA
serviceLevel: expeditedShipping
```

Customs Manifest File Option

When shipping a device internationally, you must include the `customs` manifest file option with its required subfields in your manifest file. AWS Import/Export uses these values to validate your inbound shipment and prepare your outbound customs paperwork. If you exclude these fields from your manifest and the country that we are shipping your storage device to is outside the country of the data-loading facility, your job request will fail.

> **Note**
> This requirement does not apply to shipments in the European Union. When shipping your device between the EU (Ireland) Region's data-loading facility and a European Union member nation, you do not need to include the `customs` options.

To access Amazon S3 buckets that are in the EU Region, the shipping device must originate from and be shipped back to a location in the European Union.

All of the options in the following table are subfields of the *customs* option, for example:

```
customs:
     dataDescription: This device contains medical test results.
     encryptedData: yes
     encryptionClassification: 5D992
     exportCertifierName: John Doe
     requiresExportLicense: yes
     deviceValue: 250.00
     deviceCountryOfOrigin: Ghana
     deviceType: externalStorageDevice
```

Option	Description	Required

customs	Specify this option to provide information for AWS Import/Export to complete the customs-related paperwork when returning your storage device internationally from the United States, or Singapore.
	Note
	The customs option is not required when shipping devices within the European Union.
	All other options described in this table are subfields of this option. The following shows the `customs` option with its required subfields.
	`customs:`
	` dataDescription: Medical test results.`
	` encryptedData: yes`
	` encryptionClassification: 5D992`
	` exportCertifierName: John Doe`
	` requiresExportLicense: yes`
	` deviceValue: 250.00`
	` deviceCountryOfOrigin: Ghana`
	` deviceType: externalStorageDevice`
	` typeOfExport: return`
	Type: String
	Default: None
		Yes

dataDescription	Provide a brief description of the data you are importing or exporting from AWS Import/Export. `dataDescription: This device contains medical test results.` Type: String Default: None	Yes

Option	Description	Required
deviceCountryOfOrigin	The country where the device was manufactured. This is usually indicated somewhere on the device with a "made in" or "assembled in" label. Optionally, you can use the ISO country codes. For more information, go to http://www.iso.org/iso/home/standards/country_codes/iso-3166-1_decoding_table.htm. `deviceCountryOfOrigin: Japan` Type: String Default: None	Yes
deviceHTS	You may override the default Harmonized Tariff Schedule (HTS) code associated with the specified device type, you provided by using the `deviceType` option. This value of this option must be a valid 10 digit HTS code. `deviceHTS: 8471.70.4000` Type: String Constraints: 10 digit HTS code.	Yes
deviceValue	The monetary value (in U.S. dollars) of the storage device. When shipping internationally, the device value must be less than $2,500.00. The following example specifies that the storage device value is $250.00. `storageDeviceValue: 250.00` Type: Float Default: None	Yes

Option	Description	Required
deviceType	Specifies the type of device that you are sending. Each device type defaults to an HTS code that's used in your commercial invoice. We use the following default mapping between the device type and HTS code: • `externalStorageDevice` : `8471.70.4065` • `usbFlashDrive` : `8471.70.9000` • `sataDrive` : `8471.70.2000` You can override these default mappings by specifying the `deviceHTS` option. `deviceType: usbFlashDrive` Type: Enum Default: None Valid Values: `externalStorageDevice` | `usbFlashDrive` | `sataDrive`	Yes
encryptedData	Specifies whether the data on your device is encrypted. If yes, then you must include the `encryptionClassification` option. The following example specifies that the data is encrypted. `encryptedData: yes` Type: String Default: None Valid Values: `yes` | `no`	Yes
encryptionClassification	If the device includes encrypted data, this field allows you to specify the software's classification. When shipping encrypted data to or from the United States the encryption software must be classified as 5D992 under the United States Export Administration Regulations. `encryptionClassification: 5D992` Type: String Default: None Condition: Required when `encryptedData` value is yes.	Yes

Option	Description	Required
exportCertifierName	Specifies the name the person or company responsible for shipping the storage device. The following example specifies the certifier's name. `exportCertifierName: John Doe` Type: String Default: None	Yes
requiresExportLicense	Specifies whether the import or export of the device or data require a license under the regulations of any applicable country, such as the United States Export Administration Regulations or the International Traffic in Arms Regulations. Note that only devices and data which do not require a license, are classified as EAR99 under United States Export Administration Regulations, and are not subject to the International Traffic in Arms Regulations are eligible for AWS Import/Export. `requiresExportLicense: no` Type: String Default: None Valid Values: `yes` | `no`	Yes
typeOfExport	Specifies the type of export. The value `return` indicates the shipment is being returned to the country from which it was shipped. The value `temporary` indicates the shipment will be returning from the destination country to the AWS data loading facilities country. The value `permanent` means that the shipment will remain in the country it is being sent to, and the shipment didn't originate from that country. `typeOfExport: permanent` Type: String Default: return Valid Values: `return` | `permanent` | `temporary`	Optional

Example Import to Amazon S3 Manifest with Customs Option

The following example is the manifest file content for an import job to Amazon S3, including Customs option.

```
manifestVersion: 2.0
deviceId: ABCDE
eraseDevice: yes
bucket: my-amz-bucket
acl: authenticated-read
cacheControl: max-age=3600
contentDisposition: attachment
contentLanguage: en
contentTypes:
 csv: application/vnd.ms-excel
customs:
     dataDescription: This device contains medical test results.
     encryptedData: yes
     encryptionClassification: 5D992
     exportCertifierName: John Doe
     requiresExportLicense: yes
     deviceValue: 250.00
     deviceCountryOfOrigin: Brazil
     deviceType: externalStorageDevice
diskTimestampMetadataKey: disk-timestamp
generator: AWS Import Export Docs
ignore:
  - \.psd$
  - \.PSD$
logPrefix: logs
logBucket: iemanifest-bucket
notificationEmail: john.doe@example.com;jane.doe@example.com
prefix: imported/
setContentEncodingForGzFiles: yes
staticMetadata:
    import-timestamp: Wed, 23 Feb 2011 01:55:59 GMT
    application: AppImportedData
returnAddress:
    name: John Doe
    company: Example Corp
    street1: UG 8 Golf Course Road
    city: Anycity
    stateOrProvince: Haryana
    postalCode: 122002
    phoneNumber: +91 431 555 1000
    country: India
serviceLevel: standard
```

Import to Amazon S3 Manifest File Options

The S3 import manifest file includes information related to importing your data into Amazon S3 and returning your storage device. This section explains options that are unique to Amazon S3 import manifest files. For information about options that are common to all manifest files, see Common Manifest File Options (p. 76).

Amazon S3 import and export use manifest file version 2.0. Set *`manifestVersion`* to *`2.0`* for all Amazon Amazon S3 import and export manifest files.

Option	Description	Required
acl	Specifies the access control list (ACL) to apply to each object loaded to Amazon S3, for example, `acl: public-read`. The default value gives full permissions only the person who uploaded the object to the bucket. For descriptions of the valid ACL values, go to Canned ACL. The following example sets the ACL of the uploaded objects. `acl: public-read` Type: String Default: private Valid Values: private | public-read | public-read-write | authenticated-read	No
bucket	Specifies the Amazon S3 bucket where AWS Import/Export loads your data. You must create a bucket before we can load data into it; we do not create the bucket for you. The account doing the load must have write permissions on this bucket. The following is an example of a bucket name. `bucket: data-load-bucket` Type: String Default: None Constraints: Must be an existing Amazon S3 bucket name	Yes
cacheControl	Specifies the value of the *`Cache-Control`* HTTP header field for all imported files. We do not validate this value. For more information on *`cacheControl`*, please go to http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html#sec14.9. `cacheControl: max-age=3600` Type: String Default: none	No

Option	Description	Required
contentDisposition	Specifies the Content-Disposition HTTP header to apply to all loaded files. For more details, go to RFC 2616 section 19. The following is an example *contentDisposition* value. `contentDisposition: attachment` **Note** This example causes browsers to prompt to save the Amazon S3 object rather than displaying the object directly in the browser. Type: String Default: None	No
contentLanguage	Specifies the value of the HTTP Content-Language header field for all imported files. We do not validate this value. For more information about the Content-Language header, go to http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html#sec14.12. `contentLanguage: es` Type: String Default: none	No

contentTypes	Contains a map of case-insensitive strings that are key/value pairs that set the *Content-Type* header. Each key/value pair contains a key that is the file extension (for example, `jpg`) and a corresponding value for the *Content-Type* (for example, `image/jpeg`). These values take precedence over the default file extension to Content-Type mapping. For more details on the default mappings, see Appendix C: File Extension to Mime Types (p. 143).
	
	The following is an example set of key/value pairs.
	
	`contentTypes:`
	` jpg: image/jpeg`
	` JPG: image/jpeg`
	` gif: image/gif`
	` GIF: image/gif`
	` mp3: audio/mpeg`
	` txt: text/plain`
	` zip: application/zip`
	
	Type: Map of string to string
	
	Default: None
		No

Option	Description	Required

diskTimestampMetadataKey	A user-defined key for adding a storage device's last modified time stamp in the metadata of the content imported to Amazon S3. The value of this key has the same RFC 1123 format as the Amazon S3 Date header: *`[ DAY "," ] DD-MON-YYY HH:MM:SS zone time`* (for example, *`Wed, 23 Feb 2011 01:55:59 GMT`*). The key name is prefixed with *`x-amz-meta-`*. For example, if you specify:
	`diskTimestampMetadataKey: disk-timestamp`
	the following is the result of a *`GET`* operation on the Amazon S3 content:
	`x-amz-meta-disk-timestamp: Wed, 23 Feb 2011 01:55:59 GMT`
	Type: String
	Default: None
	Constraints: The user-defined key must be a valid HTTP header key (ASCII-only, no colons, no spaces). The key name chosen here must not also be a key name used in the *`staticMetadata`* feature.
		No

eraseDevice	Acknowledges that AWS will erase the contents of your storage device after importing the data. All writable blocks on your storage device will be overwritten with zeros. You will need to repartition and format your device after the wipe. The following is an example entry. `eraseDevice: yes` Type: String Default: None Valid Values: yes | true	Yes

Option	Description	Required
expires	Specifies the value of the HTTP Expires header field for all imported files. Expire values should be dates using the RFC 1123 date format. For more information about the Expires header, go to http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html#sec14.21. `expires: Thu, 07 Jan 2010 16:00:00 GMT` Type: String (24 bytes) Default: None Constraints: Must follow RFC 1123 date time format, [DAY ","] DD-MON-YYY HH:MM:SS zone time, for example "Mon , 12 Apr 2010 23:20:00 UT". For more information, go to RFC 1123 and RFC 822.	No
ignore	Specifies directories, files, or file types that we should not load from your storage device. For more information about *`ignore`*, see Excluding Files and Directories (p. 41). Type: Array of strings Default: We load all files except the `SIGNATURE` file Constraints: The dash followed by a space, "- ", is YAML syntax and must precede each filter	No
prefix	Specifies a string to insert between the bucket name and full path of each file located on your storage device. For more information, see Specifying Key Prefix (p. 43). Type: String Default: None Constraints: A key is a sequence of unicode characters whose UTF-8 encoding is at most 1024 bytes long. If the *`logPrefix`* + `import-log-JOBID` is longer than 1024 bytes. AWS Import/Export returns an *`InvalidmanifestField`* error from the `CreateJob` response.	No
setContentEncodingForGzFiles	Specifies that files compressed with gzip should have their *`Content-Encoding`* header set to gzip and that the *`Content-Type`* header should ignore the file endings `.gz` and `.gzip` when determining file content types. For more information, see Handling Gzip Files (p. 40) . We do not validate this value. `setContentEncodingForGzFiles: Yes` Type: String Default: None	No

Option	Description	Required

staticMetadata	This option defines a user-defined key/value pairs to apply to every file imported into Amazon S3. The key/value pair is prefixed with *x-amz-meta-*. For example, if you specify:
	`staticMetadata:`
	`  import-timestamp: Wed, 23 Feb 2011 01:55:59 GMT`
	`  application: MyBackupApp`
	The following is the result of a *GET* operation on the Amazon S3 content:
	`x-amz-meta-import-timestamp: Wed, 23 Feb 2011 01:55:59 GMT`
	`x-amz-meta-application: MyBackupApp`
	Type: Map of string to string
	Default: None
	Constraints: The user-defined keys must be valid HTTP header keys (ASCII-only, no colons, no spaces); their values must be ASCII-only. The key name chosen here must not also be a key name used in the *diskTimestampMetadataKey* feature.
		No

storageClass	Specifies the Amazon S3 storage class for the import job. For more information about the Amazon S3 storage class, go to the Amazon S3 Developer Guide. `storageClass: REDUCED_REDUNDANCY` Type: Enum Default: `STANDARD` Valid Values: `STANDARD` | `REDUCED_REDUNDANCY`	No

Option	Description	Required

substitutions	This option allows you to provide strings to replace existing values (such as object keys or file names) between the Amazon S3 bucket and the designated device during imports or exports. Matches for your provided values are substituted for the existing values before the import or export operation. The values specified on the left side of the colon are replaced with the values on the right side. Substitutions are applied in length order; the longest matching option is applied first. For more information about working with substitutions, see Working with Amazon S3 Substitutions (p. 43).
	
	`substitutions:`
	`  "," : "$COMMA$"`
	
	Type: String
	
	Default: None
		No

trueCryptPassword	If the data was encrypted using TrueCrypt, the password that was provided when the device was encrypted. For more information about encrypting data using TrueCrypt, go to Using TrueCrypt Encryption (p. 66). The following is an example of *`trueCryptPassword`* `trueCryptPassword: SimpleButEasi1yCracke dEXAMPLE` Type: String Default: none	No

Example Import to Amazon S3 Manifest

The following example is the manifest file content for an import job to Amazon S3, including expedited shipping and the required `eraseDevice` and `notificationEmail` options.

```
manifestVersion: 2.0
deviceId: ABCDE
eraseDevice: yes
notificationEmail: john.doe@example.com;jane.roe@example.com
bucket: my-amz-bucket
acl: authenticated-read
cacheControl: max-age=3600
contentDisposition: attachment
contentLanguage: en
contentTypes:
 csv: application/vnd.ms-excel
diskTimestampMetadataKey: disk-timestamp
generator: AWS Import Export Docs
ignore:
  - \.psd$
  - \.PSD$
logPrefix: logs
logBucket: iemanifest-bucket
prefix: imported/
setContentEncodingForGzFiles: yes
staticMetadata:
    import-timestamp: Wed, 23 Feb 2011 01:55:59 GMT
    application: AppImportedData
trueCryptPassword: SimpleButEasi1yCrackedEXAMPLE
returnAddress:
   name: Jane Roe
   company: Example Corp.
   street1: 123 Any Street
   city: Anytown
   stateOrProvince: WA
   postalCode: 91011-1111
   phoneNumber: 206-555-1111
   country: USA

serviceLevel: expeditedShipping
```

Import to Amazon EBS Manifest File Options

The Amazon EBS import manifest file includes information related to importing your data into Amazon EBS and returning your storage device. This section explains options that are unique to Amazon EBS import manifest files. For information about options that are common to all manifest files, see Common Manifest File Options (p. 76).

> **Note**
> Amazon EBS and Amazon Glacier use only manifest file version 3.0. Set *`manifestVersion`* to 3.0 for all Amazon EBS and Amazon Glacier manifest files.

An Amazon EBS import creates a virtual copy of your device as a single image. The device's file system is never mounted.

The capacity of your device determines how the device image is stored. The maximum capacity of an Amazon EBS volume is 1 TB. If the capacity of the device is greater than 1 TB, we cannot import the disk directly into an EBS snapshot.

Note
The maximum capacity of a device is independent of the amount of data stored on the device.

If your device capacity is 1 TB or less, we store the device image directly to an EBS snapshot.

If your device capacity is greater than 1 TB, we store the device image as a series of objects in an Amazon S3 bucket. You can then create a RAID of Amazon EBS volumes using software such as Logical Volume Manager, and copy the image from Amazon S3 to the new volume.

Option	Description	Required
eraseDevice	Acknowledges that AWS will erase the contents of your storage device after importing the data. All writable blocks on your storage device will be overwritten with zeros. You will need to repartition and format your device after the wipe. Example `eraseDevice: yes` Type: String Default: None Valid Values: yes | true	Yes

operations	The *`operations`* option is a container for one or more subfields. You specify operations parameters by a dash followed by a set of subfields.
	
	The following example shows the *`operations`* option with the required Amazon EBS subfields.
	
	`operations:`
	`  - destination: ebs-snapshot`
	`  source: device`
	`  region: us-east-1`
	`  deviceCapacityGreaterThan1TB: no`
	
	See Operations Subfields (p. 92) for details about each subfield.
	
	Type: String
	
	Default: None
		Yes

Operations Subfields

The following table describes the subfields of the Operations parameter

Operations Subfields	Description	Required
destination	The destination of the data. For an Amazon EBS import operation the destination is *ebs-snapshot*. Type: String Default: None Valid Values: ebs-snapshot	Yes
deviceCapacityGreaterThan1TB	Specifies if the device capacity is larger than the maximum Amazon EBS volume size of 1TB. The capacity of your device determines how the device image is stored. The maximum capacity of an Amazon EBS volume is 1 TB. If the capacity of the device is greater than 1 TB, we cannot import the disk directly into an EBS snapshot. The maximum capacity of a device is independent of the amount of data stored on the device. If your device capacity is 1 TB or less, we store the device image directly to an EBS snapshot. If your device capacity is greater than 1 TB, we store the device image as a series of objects in an Amazon S3 bucket. You can then create a RAID of Amazon EBS volumes using software such as Logical Volume Manager, and copy the image from Amazon S3 to the new volume. If the *deviceCapacityGreaterThan1TB* option is not set correctly for your device, AWS Import/Export will not perform the import and we will return your storage device. Type: String Default: None Valid Values: yes | no	Yes
region	The region to target for the import operation. Type: String Default: None Valid Values: us-east-1 | us-west-1 | us-west-2 | eu-west-1 | ap-southeast-1 For more information about regions go to the AWS Import/Export Calculator.	Yes

Operations Subfields	Description	Required
source	The originator of the data. For an import job the origin is the device. Type: String Default: None Valid Values: device	Yes

Example Import to Amazon EBS Manifest

The following example is the manifest file content for an import job to Amazon EBS, including expedited shipping.

```
manifestVersion: 3.0
deviceId: ABCDE
eraseDevice: yes
generator: AWS Import Export Docs
logPrefix: logs
logBucket: iemanifest-bucket
notificationEmail: john.doe@example.com;jane.roe@example.com
operations:
  - destination: ebs-snapshot
    source: device
    region: us-east-1
    deviceCapacityGreaterThan1TB: no
returnAddress:
   name: Jane Roe
   company: Example Corp.
   street1: 123 Any Street
   city: Anytown
   stateOrProvince: WA
   postalCode: 91011-1111
   phoneNumber: 206-555-1111
   country: USA
serviceLevel: expeditedShipping
```

Import to Amazon Glacier Manifest File Options

The Amazon Glacier import manifest file includes information related to importing your data into Amazon Glacier and returning your storage device. This section explains options that are unique to Amazon Glacier import manifest files. AWS Import/Export does not support exporting from Amazon EBS or Amazon Glacier. For information about options that are common to import and export manifest files, see Common Manifest File Options (p. 76).

> **Note**
> Amazon EBS and Amazon Glacier only use manifest file version 3.0. Set *`manifestVersion`* to 3.0 for all Amazon EBS and Amazon Glacier manifest files.

Option	Description	Required
eraseDevice	Acknowledges that AWS will erase the contents of your storage device after importing the data. All writable blocks on your storage device will be overwritten with zeros. You will need to repartition and format your device after the wipe. The following is an example entry. `eraseDevice: yes` Type: String Default: None Valid Values: yes | true	Yes

operations	The *`operations`* option is a container for one or more subfields. You specify operations parameters by a dash followed by a set of subfields.
	
	The following example shows *`operations`* with the required Amazon Glacier subfields.
	
	`operations:`
	`  - destination: glacier`
	`  source: device`
	`  region: us-east-1`
	`  vaultName: 2012-07_archive`
	`  archiveComment: daily_2012-07-24`
	
	For more information about each subfield, see Operations Subfields (p. 95).
	
	Type: String
	
	Default: None
		Yes

Operations Subfields

The following table describes the subfields of the Operations parameter

Operations Subfields	Description	Required
archiveComment	A description for the archive. Amazon Glacier creates an archive on the specified vault using this text as the archive description. The following is an example entry. `archiveComment: daily archive 2012-07-2` Constraints: Maximum length is 997 characters. Type: String Default: None	Yes
destination	The destination of the data. For an Amazon Glacier import operation the destination is *`glacier`*. Type: String Default: None Valid Values: glacier	Yes
region	The region to target for the import operation. Type: String Default: None Valid Values: us-east-1|us-west-1|us-west-2|eu-west-1 For more information about regions go to the AWS Import/Export Calculator.	Yes
source	The originator of the data. For an import job, the origin is the device. Type: String Default: None Valid Values: device	Yes
vaultName	Specifies the Amazon Glacier vault where AWS Import/Export loads your data. You must create a vault before we can load data into it; we do not create the vault for you. The account doing the load must have write permissions on this vault. Type: String Default: None Constraints: Must be an existing Amazon Glacier vault name	Yes

Example Import to Amazon Glacier Manifest

The following example is the manifest file content for an import job to Amazon Glacier, including expedited shipping.

```
manifestVersion: 3.0

deviceId: 49382
eraseDevice: yes
logBucket: myBucket
logPrefix: logs
generator: AWS Import/Export Command Line Tool
notificationEmail: john.doe@example.com;jane.roe@example.com
returnAddress:
   name: Jane Roe
   company: Example Corp.
   street1: 123 Any Street
   city: Anytown
   stateOrProvince: WA
   postalCode: 91011-1111
   phoneNumber: 206-555-1111
   country: USA
operations:
    - destination: glacier
      source: device
      region: us-east-1
      vaultName: MyGlacierVault
      archiveComment: July 24, 2012 daily archive
serviceLevel: expeditedShipping
```

Export from Amazon S3 Manifest File Options

The export manifest file contains information related to exporting your data from Amazon S3 onto your storage device. This section describes options that are unique to export manifest files. For information about options that are common to import and export manifest files, see Common Manifest File Options (p. 76).

Amazon S3 import and export only use manifest file version 2.0. Set **manifestVersion** to *`2.0`* for all Amazon S3 import and export manifest files.

Option	Description	Required
fileSystem	Specifies the file system on your storage device. The following is an example *`fileSystem`* entry. `fileSystem: NTFS` Type: String Default: None Valid Values: NTFS | EXT4	Yes

Option	Description	Required

operations	A container that takes subfields that specify the operations to perform on each bucket for this job. The subfields are name-value pairs. For example, *exportBucket* specifies the bucket from which to export data. For more information, see Operations Export Manifest Option (p. 59). The following example shows *operations* with two subfields, *exportBucket* and *endMarker*.
	
	`operations:`
	`  - exportBucket: bucket-data-comes-from`
	`    targetDirectory: myexport/stuff`
	`    beginMarker: images/starwars`
	`    endMarker: images/xena`
	`    prefix: exported/`
	
	For information about each subfield, see Operations Subfields (p. 99).
	
	Type: String
	
	Default: None
		Yes

recoveryDirectory	This is the directory on your storage device where we save files whose filenames AWS Import/Export must change from their keys. This happens often because characters that are valid in keys are not always valid for filenames. The following example names the recovery directory `LostAndFound`. `recoveryDirectory: LostAndFound` Type: String Default: EXPORT-RECOVERY	No

substitutions	This option allows you to provide strings to replace existing values (such as object keys or file names) between the Amazon S3 bucket and the designated device during imports or exports. Matches for your provided values are substituted for the existing values before the import or export operation. The values specified on the left side of the colon are replaced with the values on the right side. Substitutions are applied in length order; the longest matching option is applied first. For more information about working with substitutions, see Working with Amazon S3 Substitutions (p. 43).
	
	`substitutions:`
	`    "," : "$COMMA$"`
	
	Type: String
	
	Default: None
		No

Option	Description	Required
trueCryptPassword	AWS Import/Export will use the password to encrypt your data using TrueCrypt during the export. **Important** Do not lose your password, or you will not be able to decrypt your data. AWS Import/Export will not send the password with the encrypted device. For more information about encrypting data by using TrueCrypt, go to Using TrueCrypt Encryption (p. 66). The following is an example of *`trueCryptPassword`* `trueCryptPassword: SimpleButEasi1yCrackedEXAMPLE` Type: String Default: none	Yes

Operations Subfields

The following table describes the subfields of the Operations parameter

Operations Subfield	Description	Required

beginMarker	Indicates where in the bucket to begin exporting data. The export job only includes keys that occur alphabetically after, but not including, *`beginMarker`*. For example, if `beginMarker` is `alpha` and `endMarker` is `omega`, then keys `alpha-1` and `omega` are exported, but `alpha` and `omega-1` are not exported. This option is convenient for limiting the number of files to export and for splitting large export jobs between multiple devices. The following is an example *`beginMarker`* value.
	
	`operations:`
	`  - exportBucket: bucket-data-comes-from`
	`    beginMarker: images/alpha`
	`    endMarker: images/omega`
	
	Type: String
	
	Default: none
		No

Operations Subfield	Description	Required

endMarker	Indicates where in the bucket to stop exporting data. The export job only includes keys that occur alphabetically before and including the key that matches *endMarker*. For example, if `beginMarker` is `alpha` and `endMarker` is `omega`, then keys `alpha-1` and `omega` are exported, but `alpha` and `omega-1` are not exported. This option is convenient for limiting the number of files to export and splitting large export jobs across multiple devices. The following is an example *endMarker* value.
	
	`operations:`
	`  - exportBucket: bucket-data-comes-from`
	`  beginMarker: images/alpha`
	`  endMarker: images/omega`
	
	Type: String
	
	Default: none
		No

exportBucket	Specifies the Amazon S3 bucket where AWS Import/Export retrieves your data. Each listing under *operations* must have an *exportBucket* option. The AWS IAM user that created the job must have read permissions on the bucket and objects. If no other options are specified with the listing, AWS Import/Export exports all files in the bucket to your storage device. The following is an example of an *exportBucket* name.
	
	`operations:`
	`  - exportBucket: bucket-data-comes-from`
	
	Type: String
	
	Default: None
	
	Constraints: Must be an existing Amazon S3 bucket name
		Yes

Operations Subfield	Description	Required

prefix	Specifies the path in the *exportBucket* where AWS Import/Export finds the data to export. All keys that contain the prefix are saved to your storage device. For more information, go to Managing File Exports (p. 59). Note that *prefix* has a different meaning when used in import manifest files.
	
	If the *logPrefix* + `import-log-JOBID` is longer than 1024 bytes, AWS Import/Export returns an InvalidManifestField error from the `CREATE JOB` command.
	
	The following example exports all files under the directory *images*.
	
	`operations:`
	`  - prefix: images`
	
	Type: String
	
	Default: None
	
	Constraints: A key is a sequence of unicode characters whose UTF-8 encoding is at most 1024 bytes long.
		No

targetDirectory	Indicates where to export the data to on your storage device. In the following example, AWS Import/Export exports the contents of the bucket, `bucket-data-comes-from`, to `myexport/stuff` on the storage device.
	
	`operations:`
	`  - exportBucket: bucket-data-comes-from`
	`    targetDirectory: myexport/stuff`
	
	Type: String
	
	Default: The specified *exportBucket*.
		No

Example Export from Amazon S3 Manifest

The following example is the manifest file content for an export job from Amazon S3, including expedited shipping.

```
manifestVersion: 2.0
deviceId: ABCDE
fileSystem: NTFS
generator: AWS Import Export Docs
logPrefix: logs
logBucket: iemanifest-bucket
notificationEmail: john.doe@example.com;jane.roe@example.com
trueCryptPassword: SimpleButEasi1yCrackedEXAMPLE
operations:
  - exportBucket: my-amz-bucket
    targetDirectory: myexport/stuff
    beginMarker: images/starwars
    endMarker: images/xena
    prefix: exported/
recoveryDirectory: LostAndFound
returnAddress:
   name: Jane Roe
   company: Example Corp.
   street1: 123 Any Street
   city: Anytown
   stateOrProvince: WA
   postalCode: 91011-1111
   phoneNumber: 206-555-1111
   country: USA
serviceLevel: expeditedShipping
```

API Reference

Topics

Making Query Requests

Topics

For AWS Import/Export, query requests are HTTPS requests that use the HTTP verb POST and a query parameter named `Action`.

Endpoints

To send requests to AWS Import/Export, use the service endpoint:

```
https://importexport.amazonaws.com
```

> **Note**
> AWS Import/Export does not support requests over HTTP; the service only supports HTTPS requests.

Structure of a POST Request

This guide presents the AWS Import/Export POST requests, which consist of:

- **Endpoint—**The AWS Import/Export entry point:

```
https://importexport.amazonaws.com
```

- **Action—**The action you want to perform. For example, getting the status of an existing job (GetStatus). For a complete list, see Actions (p. 106).
- **Parameters—**A set of parameters common to all AWS Import/Export actions, and any request parameters supported by the AWS Import/Export API. For details, see Common Query Parameters (p. 124).

Sample POST Request and Response

POST Request

The following is an example POST request to get the status of a job.

```
POST / HTTP/1.1
content-type:application/x-www-form-urlencoded;charset=utf-8
host: https://importexport.amazonaws.com
content-length:207

Action=GetStatus&SignatureMethod=HmacSHA256&JobId=JOBID&SignatureVersion=2&Ver
sion=2010-06-03&Signature=%2FVfkltRBOoSUi1sWxRzN8rw%3D&Timestamp=2011-06-
20T22%3A30%3A59.556Z
```

The body of the request is all on one line. However, line feeds have been added to make the examples easier to read.

```
POST / HTTP/1.1
content-type:application/x-www-form-urlencoded;charset=utf-8
host: https://importexport.amazonaws.com
content-length:207

Action=GetStatus
&JobId=JOBID

&SignatureMethod=HmacSHA256
&SignatureVersion=2
&Version=2010-06-03
&Signature=lBP67vCvGlDMBQ1dofZxg8E8SUEXAMPLE
&Timestamp=2011-06-20T22%3A30%3A59.556Z
```

The first line represents the type of http request.

Lines 2-4 contain the HTTP headers, including the endpoint of the request.

After the HTTP headers, the body of the request contains the list of parameters. Each parameter is separated by an ampersand (&). The Action parameter indicates the action to perform For a list of the actions, see Actions (p. 106).

Response

The following is the sample response:

```
<CreateJobResponse xmlns="http://importexport.amazonaws.com/">
  <CreateJobResult>
    <JobId>TEST1</JobId>
      <SignatureFileContents>version:2.0
signingMethod:HmacSHA1
jobId:WXSPP
signature:jLbLVp54vBa/vuh/XW1MOMdZUxw=
      </SignatureFileContents>
      <AwsShippingAddress>AWS Import/Export
JOBID TEST1
2646 Rainier Ave South Suite 1060
Seattle, WA 98144</AwsShippingAddress>
    <Signature>O3qG7xnI46ZEm9gi+eOy9dnSS9k=</Signature>
  </CreateJobResult>
  <ResponseMetadata>
    <RequestId>ace8cb5b-0ada-11df-a381-155bec6f7c93</RequestId>
  </ResponseMetadata>
  </CreateJobResponse>
```

Query Authentication

You can send requests over HTTPS. When you do, you must include a signature in every request. This section describes how to create the signature. The method described in the following procedure is known as *signature version 2*.

Sample GetStatus Request

The following request gets the current status of an import job (modified to make the *`Signature`* inside the request invalid).

```
POST / HTTP/1.1
content-type:application/x-www-form-urlencoded;charset=utf-8
host: https://importexport.amazonaws.com
content-length:639

Action=GetStatus&SignatureMethod=HmacSHA256&JobId=JOBID&SignatureVersion=2&Ver
sion=2010-06-01&Signature=%2FVfkltRBOoSUi1sWxRzN8rw%3D&Timestamp=2011-06-
20T22%3A30%3A59.556Z
```

The following is the string to sign.

The body of the request is all on one line. However, line feeds have been added to make the examples easier to read.

```
POST\n
importexport.amazonaws.com\n
/\n
&Action=GetStatus
&JobId=JOBID
&SignatureMethod=HmacSHA256
&SignatureVersion=2
Version=2010-06-01&
```

The following is the signed request.

```
POST\n
importexport.amazonaws.com\n
/\n
&Action=GetStatus
&JobId=JOBID
&SignatureMethod=HmacSHA256
&SignatureVersion=2
&Version=2010-06-01
&Signature=%2FVfkltRBOoSUi1sWxRzN8rw%3D
&Timestamp=2011-06-20T22%3A30%3A59.556Z
```

Actions

Topics

This section describes the web service actions for AWS Import/Export using the Query API request parameters.

> **Note**
> AWS Import/Export does not support SOAP.

CancelJob

Description

This action cancels a specified job. Only the job owner can cancel it. The action fails if the job has already started or is complete.

Request Parameters

For information about the common parameters that all actions use, see Common Query Parameters (p. 124)

Name	Description	Required
Action	Specifies the action to invoke. Type: String Default: None Valid Values: `CancelJob`	Yes
JobId	Identifier returned by `CreateJob` that specifies the job you want to cancel. Type: String Default: None	Yes

Response Elements

Name	Description
CancelJobResponse	Container for the response. Type: None Ancestor: None Children: CancelJobResult (p. 107), ResponseMetadata (p. 108)
CancelJobResult	Container for the result. Type: None Ancestor: CancelJobResponse (p. 107) Children: Success (p. 108)
RequestId	Identifier that uniquely identifies the request. Type: String Ancestor: CancelJobResponse (p. 107).ResponseMetadata (p. 108) Children: None

Name	Description
ResponseMetadata	Container for the response metadata. Type: String Ancestor: CancelJobResponse Children: RequestId (p. 107)
Success	Specifies whether (`true`) or not (`false`) AWS Import/Export canceled the job. Type: Boolean Valid Values: `true` | `false` Ancestor: CancelJobResponse (p. 107).CancelJobResult (p. 107) Children: None

Errors

For a list of common errors, see Error and Status Codes (p. 125).

Examples

Sample Request

```
POST / HTTP/1.1
content-type:application/x-www-form-urlencoded;charset=utf-8
host: https://importexport.amazonaws.com
content-length:188

Action=CancelJob&SignatureMethod=HmacSHA256&JobId=JOBID&
&SignatureVersion=2&
Signature=%2FVfkltRBOoSUi1sWxRzN8rw%3D&Timestamp=2011-06-20T23%3A53%3A37.232Z
```

Sample Response

```
<CancelJobResponse xmlns="http://importexport.amazonaws.com/">
  <CancelJobResult>
    <Success>true</Success>
  </CancelJobResult>
  <ResponseMetadata>
    <RequestId>a4f704fc-0ad8-11df-b085-4384d3e37ae5</RequestId>
  </ResponseMetadata>
</CancelJobResponse>
```

Related actions

The following actions are related to `CancelJob`.

- CreateJob (p. 109)

CreateJob

Description

This action initiates the process of scheduling an upload or download of your data. You include in the request a manifest that describes the data transfer specifics. The response to the request includes a job ID, which you can use in other actions, a signature that you use to identify your storage device, and the address you should ship your storage device to.

Request Parameters

For information about the common parameters that all actions use, see Common Query Parameters (p. 124)

Name	Description	Required
Action	Specifies the action to invoke. Type: String Default: None Valid Values: CreateJob	Yes
JobType	Specifies whether the job to initiate is an import or export job. Type: Enum Default: None Valid Values: Import | Export	Yes
Manifest	The UTF-8 encoded text of a manifest file. For information about manifest parameters, see Manifest File Options Reference (p. 76). Type: String Default: None	Yes
ValidateOnly	Validate the manifest and parameter values in the request, but do not actually create a job. Type: Boolean Default: `false` Valid Values: `true` | `false`	No

Response Elements

Name	Description
AwsShippingAddress	Address you ship your storage device to. Type: String Ancestor: CreateJobResponse (p. 110).CreateJobResult (p. 110)

Name	Description
CreateJobResponse	Container for the response. Type: None Ancestor: None Children: CreateJobResult (p. 110), ResponseMetadata (p. 110)
CreateJobResult	Container. Type: None Ancestor: CreateJobResponse Children: JobId (p. 110), AwsShippingAddress (p. 109), Signature (p. 110)
JobId	The ID created by AWS that uniquely identifies your AWS Import/Export job. Type: String Ancestor: CreateJobResponse (p. 110).CreateJobResult (p. 110)
JobType	Specifies whether the job is an import or export job. Type: String Ancestor: GetStatusResponse.GetStatusResult Children: None
RequestId	ID that uniquely identifies the request. Type: String Ancestor: CreateJobResponse (p. 110).ResponseMetadata (p. 110)
ResponseMetadata	Container. Type: None Ancestor: CreateJobResponse Children: RequestId (p. 107)
Signature	An encrypted code used to authenticate the request and response, for example, DV+TpDfx1/TdSE9ktyK9k/bDTVI=. Only use this value if you want to create the signature file yourself. Type: String Ancestor: CreateJobResponse (p. 110).CreateJobResult (p. 110)

Name	Description

SignatureFileContents	The actual text of the SIGNATURE file to be written to disk, for example:
	`version:2.0`
	`signingMethod:HmacSHA1`
	`jobId:<the customer's job id>`
	`signature:DV+TpDfx1/TdSE9ktyK9k/bDTVI=`
	Type: String
	Ancestor: CreateJobResponse (p. 110).CreateJobResult (p. 110)

WarningMessage	An optional message notifying you of non-fatal issues with the job, such as use of an incompatible Amazon S3 bucket name. Type: String Ancestor: CreateJobResponse (p. 110).CreateJobResult (p. 110)

Errors

For a list of common errors, see Error and Status Codes (p. 125).

Examples

Sample Request

```
POST / HTTP/1.1
content-type:application/x-www-form-urlencoded;charset=utf-8
host: https://importexport.amazonaws.com
content-length:639

Manifest=manifestVersion%3A%202.0%0Abucket%3A%20myawsbucket
%0AreturnAddress%3A%0A%20%20%20%20name%3A%20
Amazon.com%20ATTN%20Joe%20Ran
dom%20%0A%20%20%20%20street1%3A%201200%20AAAA%20Ave%20
S.%0A%20%20%20%20city%3A%20Seattle%0A%20%20%20%20stateOr
Province%3A%20WA%0A%20%20%20%20
postalCode%3A%2098114%0A%20%20%20%20phoneNumber%3A%20206-266-0000%0A%20%20%20%20
country%3A%20USA&JobType=Import&Action=CreateJob&SignatureMethod=HmacSHA256
&SignatureVersion=2&Version=2010-06-01
&Signature=%2F5UvjcJOE1PqUa%2BcmnzadQYs2frTAt8LSo3M5o%2BEH0%3D&Timestamp=
2011-06-20T18%3A49%3A14.981Z
```

Sample Response

```
<CreateJobResponse xmlns="http://importexport.amazonaws.com/">
  <CreateJobResult>
    <JobId>TEST1</JobId>
    <AwsShippingAddress>AWS Import/Export
JOBID TEST1
123 Any Street
Seattle, WA 98144</AwsShippingAddress>
```

```
    <Signature>O3qG7xnI46ZEm9gi+eOy9dnSS9k=</Signature>
  </CreateJobResult>
  <ResponseMetadata>
    <RequestId>ace8cb5b-0ada-11df-a381-155bec6f7c93</RequestId>
  </ResponseMetadata>
</CreateJobResponse>
```

Related actions

The following actions are related to `CreateJob`.

- CancelJob (p. 107)

GetStatus

Description

This action returns information about a job, including where the job is in the processing pipeline, the status of the results, and the signature value associated with the job. You can only return information about jobs you own.

Request Parameters

For information about the common parameters that all actions use, see Common Query Parameters (p. 124)

Name	Description	Required
Action	Specifies the action to invoke. Type: String Default: None Valid Values: GetStatus	Yes
JobId	Specifies the job to return information about. Type: String Default: Constraints: Must be a valid job ID for a job you own.	Yes

Response Elements

Name	Description
AwsShippingAddress	The physical address where you ship your storage device. Type: String Ancestor: GetStatusResponse (p. 114).GetStatusResult (p. 114)
Carrier	Name of the shipping company. This value is included when the *LocationCode* is `Returned`. Type: String Ancestor: GetStatusResponse (p. 114).GetStatusResult (p. 114) Children: None
CreationDate	Timestamp of the CreateJob request in ISO8601 date format. For example 2010-03-28T20:27:35Z. For more information, go to ISO 8601. Type: String Ancestor: GetStatusResponse (p. 114).GetStatusResult (p. 114) Children: None

Name	Description
GetStatusResponse	Container. Type: String Ancestor: None Children: GetStatusResult (p. 114)
GetStatusResult	Container. Type: String Ancestor: GetStatusResponse (p. 114) Children: Carrier (p. 113) | CreationDate (p. 113) | ErrorCount (p. 114) | JobId (p. 110) | JobType (p. 114) | LocationCode (p. 114) | LocationMessage (p. 115) | LogBucket (p. 115) | LogKey (p. 115) | ProgressCode (p. 115) | ProgressMessage (p. 115) | Signature (p. 110) | TrackingNumber (p. 116)
ErrorCount	Number of errors. We return this value when the *ProgressCode* is `Success` or `SuccessWithErrors`. Type: Integer Ancestor: GetStatusResponse (p. 114).GetStatusResult (p. 114) Children: None
JobId	Unique identifier created by AWS Import/Export that specifies a job you want the status of. Type: String Ancestor: GetStatusResponse (p. 114).GetStatusResult (p. 114) Children: None
JobType	Specifies whether the job is an import or export job. Type: String Ancestor: GetStatusResponse (p. 114).GetStatusResult (p. 114) Children: None
LocationCode	A token representing the location of the storage device, such as `AtAWS`. For more information, see Device Location Status Codes (p. 127). Type: Enum Valid Values: `AtAWS` | `NotReceived` | `Returned` Ancestor: GetStatusResponse (p. 114).GetStatusResult (p. 114) Children: None

Name	Description
LocationMessage	A human readable message describing the physical location of the storage device. Type: String Ancestor: GetStatusResponse (p. 114).GetStatusResult (p. 114) Children: None
LogBucket	Amazon S3 bucket for user logs. Type: String (63 bytes) Ancestor: GetStatusResponse (p. 114).GetStatusResult (p. 114) Children: None
LogKey	The key where the user logs were stored. Type: String (1024 bytes) Ancestor: GetStatusResponse (p. 114).GetStatusResult (p. 114) Children: None
ProgressCode	A token representing the state of the job, such as `Started`. For more information, see Job Progress Status Codes (p. 127). Type: Enum Ancestor: GetStatusResponse (p. 114).GetStatusResult (p. 114) Children: None
ProgressMessage	A more human readable form of the job status. Type: String Ancestor: GetStatusResponse (p. 114).GetStatusResult (p. 114) Children: None
RequestId	A string that uniquely identifies the request. Type: String Ancestor: GetStatusResponse (p. 114) Children: None
ResponseMetadata	Container. Type: String Ancestor: GetStatusResponse (p. 114) Children: RequestId (p. 107)

Name	Description
Signature	An encrypted code used to authenticate the request and response, for example, DV+TpDfx1/TdSE9ktyK9k/bDTVI=. Only use this value if you want to create the signature yourself. Type: String Ancestor: GetStatusResponse (p. 114).GetStatusResult (p. 114) Children: None

SignatureFileContents	The actual text of the `SIGNATURE` file to be written to disk, for example:
	
	`version:2.0`
	`signingMethod:HmacSHA1`
	`jobId:<the customer's job ID>`
	`signature:DV+TpDfx1/TdSE9ktyK9k/bDTVI=`
	
	Type: String
	
	Ancestor: GetStatusResponse (p. 114).GetStatusResult (p. 114)

TrackingNumber	The shipping tracking number assigned by AWS Import/Export to the storage device when it's returned to you. We return this value when the *`LocationCode`* is `Returned`. Type: String (32 bytes) Ancestor: GetStatusResponse (p. 114).GetStatusResult (p. 114) Children: None
WarningMessage	An optional message notifying you of non-fatal issues with the job, such as use of an incompatible Amazon S3 bucket name. Type: String Ancestor: GetStatusResponse (p. 114).GetStatusResult (p. 114)

Errors

For a list of common errors, see Error and Status Codes (p. 125).

Examples

Sample Request

```
POST / HTTP/1.1
content-type:application/x-www-form-urlencoded;charset=utf-8
host: https://importexport.amazonaws.com
content-length:207

Action=GetStatus&SignatureMethod=HmacSHA256&JobId=JOBID&SignatureVersion=2&Ver
sion=2010-06-01&Signature=
%2FVfkltRBOoSUi1sWxRzN8rw%3D&Timestamp=2011-06-20T22%3A30%3A59.556Z
```

Sample Response

```
<GetStatusResponse xmlns="http://importexport.amazonaws.com/">
  <GetStatusResult>
    <JobType>Import</JobType>
    <CurrentManifest>manifestVersion: 2.0
deviceId: TestDevice1
bucket: myawsbucket
returnAddress:
   name: Jane Roe
   company: Example Corp.
   street1: 123 Any Street
   city: Anytown
   stateOrProvince: WA
   postalCode: 91011-1111
   phoneNumber: 206-555-1111
   country: USA
eraseDevice: yes<
/CurrentManifest>
    <JobId>A36BC</JobId>
    <LocationMessage>AWS has not received your device.</LocationDescription>
    <ProgressCode>Pending</ProgressCode>
    <SignatureFileContents>version:2.0
signingMethod:HmacSHA1
jobId:A36BC
signature:O8WnNGMfs6cwbpQHvJ0qXLrUdqI=
</SignatureFileContents>
    <ErrorCount>0</ErrorCount>
    <ProgressMessage>The specified job has not started.</ProgressDescription>
    <LocationCode>NotReceived</LocationCode>
    <CreationDate>2010-04-17T00:09:02Z</CreationDate>
    <AwsShippingAddress>AWS Import/Export
JOBID A36BC
1111 Nosuch Ave South
Seattle, WA 92222</AwsShippingAddress>
    <Signature>O8WnNGMfs6cwbpQHvJ0qXLrUdqI=</Signature>
  </GetStatusResult>
  <ResponseMetadata>
    <RequestId>a5f302fa-4bd8-11df-b17e-6b34e254a573</RequestId>
  </ResponseMetadata>
</GetStatusResponse>
```

Related actions

The following actions are related to `GetStatus`.

- `ListJobs`

ListJobs

Description

This action returns the jobs associated with the requester. AWS Import/Export lists the jobs in reverse chronological order based on the date of creation. For example, if job Test1 was created on 30 December 2009 and Test2 was created 05 February 2010, the `ListJobs` action would return Test2 followed by Test1.

Request Parameters

For information about the common parameters that all actions use, see Common Query Parameters (p. 124).

Name	Description	Required
Action	Specifies the action to invoke. Type: String Default: None Valid Value: ListJobs	Yes
Marker	Specifies the *JOBID* to start after when listing the jobs created with your account. See *MaxJobs*. Type: String Default: None Constraints: Must be a valid job ID	No
MaxJobs	Sets the maximum number of jobs returned in the response. If there are additional jobs that were not returned because *MaxJobs* was exceeded, the response contains <IsTruncated>true</IsTruncated>. To return the additional jobs, see *Marker*. Type: String Default: None Constraints: Must be between 0 and 1000	No

Response Elements

Name	Description
CreationDate	Timestamp of the `CreateJob` request in ISO8601 date format. For example, 2010-03-28T20:27:35Z. For more information, go to ISO 8601. Type: String Ancestor: ListJobsResponse.ListJobsResult.Jobs.member

Name	Description
IsCanceled	Indicates whether the job was canceled. Type: String Valid Values: `true` | `false` Ancestor: ListJobsResponse.ListJobsResult.Jobs.member
IsTruncated	Indicates whether the list of jobs was truncated. If `true`, call `ListJobs` again using the last *`JobId`* element as the marker. Type: String Valid Values: `true` | `false` Ancestor: ListJobsResponse.ListJobsResult.Jobs.member
JobId	ID generated by AWS that uniquely identifies the AWS Import/Export job. Type: String Ancestor: ListJobsResponse.ListJobsResult.Jobs.member
Jobs	Container for all jobs. Type: None Ancestor: ListJobsResponse.ListJobsResult Children: member, IsTruncated
JobType	Specifies whether the job is an import or export job. Type: String Ancestor: ListJobsResponse.ListJobsResult.Jobs.member
ListJobsResponse	Container for the response. Type: None Ancestor: None Children: ListJobsResult, Response Metadata
ListJobsResult	Container for jobs. Type: None Ancestor: ListJobsResponse Children: Jobs
member	Container for job information. Type: None Ancestor: ListJobsResponse.ListJobsResult.Jobs Children: creationDate, JobId, IsCanceled

Name	Description
RequestId	ID generated by AWS that uniquely identifies the `ListJobs` request. Type: String Ancestor: ListJobsResponse.ListJobsResult.ResponseMetadata
ResponseMetadata	Container for *`RequestId`*. Type: None Ancestor: ListJobsResponse.ListJobsResult Children: RequestId

Errors

For a list of common errors, see Error and Status Codes (p. 125).

Examples

Sample Request

```
POST / HTTP/1.1
content-type:application/x-www-form-urlencoded;charset=utf-8
host: https://importexport.amazonaws.com
content-length:206

Action=ListJobs&MaxJobs=25&SignatureMethod=HmacSHA256
&SignatureVersion=2&Version=2010-06-01&Signature=%2FVfkltRBOoSUi1sWxRzN8rw%3D
&Timestamp=2011-06-18T02%3A29%3A39.202Z
```

Sample Response

```
<ListJobsResponse xmlns="http://importexport.amazonaws.com/">
  <ListJobsResult>
    <Jobs>
      <member>
        <JobType>Import</JobType>
        <CreationDate>2010-04-21T22:21:51Z</CreationDate>
        <IsCanceled>false</IsCanceled>
        <JobId>ADP7B</JobId>
      </member>
      <member>
        <JobType>Import</JobType>
        <CreationDate>2010-04-21T22:19:05Z</CreationDate>
        <IsCanceled>false</IsCanceled>
        <JobId>AVEYF</JobId>
      </member>
    </Jobs>
    <IsTruncated>true</IsTruncated>
  </ListJobsResult>
  <ResponseMetadata>
    <RequestId>cc6ea8c0-4da9-11df-81c3-e94d3ca214a8</RequestId>
```

```
  </ResponseMetadata>
</ListJobsResponse>
```

Related actions

The following actions are related to `ListJobs`.

- GetStatus (p. 113)

UpdateJob

Description

You use this action to change the parameters specified in the original manifest file by supplying a new manifest file. The manifest file attached to this request replaces the original manifest file. You can only use the action after a `CreateJob` request and before the data transfer starts, and you can only use it on jobs you own.

Request Parameters

For information about the common parameters that all actions use, see Common Query Parameters (p. 124)

Name	Description	Required
Action	Specifies the action to invoke. Type: String Default: None Valid Values: UpdateJob	Yes
JobId	Identifier that specifies the job you want to update. Type: String Default: None Constraints: Must be a valid job ID for a job you created.	Yes
JobType	Specifies import or export job. Type: String Default: None Valid Values: Import | Export	Yes
Manifest	The UTF-8 encoded text of the manifest file. For more information, see Manifest File Parameters (p. 76). Type: String Default: None	Yes

Response Elements

Name	Description
RequestId	ID that uniquely identifies the request. Type: String Ancestor: UpdateJobResponse (p. 123).ResponseMetadata (p. 123) Children: None

Name	Description
ResponseMetadata	Container. Type: None Ancestor: UpdateJobResponse (p. 123) Children: RequestId (p. 107)
Success	Specifies whether (`true`) or not (`false`) AWS Import/Export updated your job. Type: Boolean Valid values: `true` | `false` Ancestor: UpdateJobResponse (p. 123).UpdateJobResult (p. 123)
UpdateJobResponse	Container for the response. Type: None Ancestor: None Children: UpdateJobResult (p. 123), ResponseMetadata (p. 123)
UpdateJobResult	Container. Type: None Ancestor: UpdateJobResponse (p. 123) Children: Success (p. 108)
WarningMessage	An optional message notifying you of non-fatal issues with the job, such as use of an incompatible Amazon S3 bucket name. Type: String Ancestor: UpdateJobResponse (p. 123)

Errors

For a list of common errors, see Error and Status Codes (p. 125).

Examples

Sample Request

```
POST / HTTP/1.1
content-type:application/x-www-form-urlencoded;charset=utf-8
host: https://importexport.amazonaws.com
content-length:625

JobType=Import&Manifest=manifestVersion%3A%201.0%0Abucket%
3A%20XYZ%0AreturnAddress%3A%0A%20%20%20%20name
%3A%20Amazon.com%20ATTN%20Joe%20Ran
```

```
dom%20%0A%20%20%20%20street1%3A%201200%11111%20Nosuch%20Ave
%20S.%0A%20%20%20%20city%3A%20Seattle%0A%20%20%20%20stateOr
Province%3A%20WA%0A%20%20%20%20
postalCode%3A%2098114%0A%20%20%20%20phoneNumber%3A%20206-266-
0000%0A%20%20%20%20country%3A%20USA
%0AsetContentEncodingForGzFiles%3A%20yes%0Aexpires%3A%20tomorrow%0AcontentLan
guage%3A%20Spanish
%0Aacl%3A%20public-read%0AcacheControl%3A%20max-age%3D3600&Action=UpdateJob&Sig
natureMethod=HmacSHA256
&JobId=JOBID&SignatureVersion=2
&Version=2010-06-01Signature=%2FVfkltRBOoSUi1sWxRzN8rw%3D&Timestamp=2011-06-
21T00%3A28%3A12.023Z
```

Sample Response

```
<UpdateJobResponse xmlns="http://importexport.amazonaws.com/">
  <UpdateJobResult>
    <Success>true</Success>
  </UpdateJobResult>
  <ResponseMetadata>
    <RequestId>12144cd2-0adb-11df-a381-155bec6f7c93</RequestId>
  </ResponseMetadata>
</UpdateJobResponse>
```

Related actions

The following actions are related to `UpdateJob`.

- CreateJob (p. 109)

Common Query Parameters

This section lists the requests parameters that all actions use. Any action-specific parameters are listed in the topic for the action.

Name	Description	Required
Action	The action to perform. For a complete list, see Actions (p. 106). Type: String Default: None Valid Values: CancelJob | CreateJob | GetStatus | ListJobs | UpdateJob	Yes
AWSAccessKeyId	The `accessKeyId` option is deprecated. It is retained only for backward compatibility. Do not use `accessKeyId` with new AWS Import/Export jobs.	No

Name	Description	Required
Signature	The digital signature you created for the request. For more information see Query Authentication (p. 105). Default: None Type: String	Yes
SignatureMethod	The hash algorithm you used to create the request signature. Default: None Type: String Valid Values: `HmacSHA256` | *`HmacSHA1`*	Yes
SignatureVersion	The signature version you use to sign the request. Set this value to *2* or *3*. Default: None Types: String Valid Value: *2*|*3*	Yes
Timestamp	The date and time the request was signed, in the format YYYY-MM-DDThh:mm:ss.SSSZ, as specified in the ISO 8601 Standard. Default: None Type: String	Yes
Version	The API version to use, in the format YYYY-MM-DD. Default:None Type: String	Yes

Error and Status Codes

Topics

This section describes errors, job and device status codes, as well as status codes you might find in AWS Import/Export log files.

Error Response

The following is the format of an error response when using the Web Service operations.

```
<ErrorResponse xmlns="http://importexport.amazonaws.com/">
  <Error>
    <Type>Sender</Type>
    <Code>ExceptionCode</Code>
    <Message>ExceptionMessage</Message>
  </Error>
  <RequestId>a1f5a4c6-0adb-11df-a381-155bec6f7c93</RequestId>
</ErrorResponse>
```

Error Codes

There are a number of errors that can occur when using the web service operations. The following table describes those error codes.

Code	Description	HTTP Code
BucketPermissionException	The account specified does not have the appropriate bucket permissions.	400
CanceledJobIdException	The specified job ID has been canceled and is no longer valid.	400
ExpiredJobIdException	The specified job ID has expired.	400
InvalidAddressException	The address specified in the manifest is invalid.	400
InvalidClientTokenIdException	The AWS access key does not exist.	403
InvalidCustomsException	One or more customs parameters was invalid. Please correct and resubmit.	400
InvalidFileSystemException	File system specified in export manifest is invalid.	400
InvalidJobIdException	The *JOBID* was missing, not found, or not associated with the AWS account.	400
InvalidManifestFieldException	One or more manifest fields was invalid. Please correct and resubmit.	400
InvalidParameterException	Description describing invalid parameter.	400
MalformedManifestException	Your manifest is not well-formed.	400
MissingCustomsException	One or more required customs parameters was missing from the manifest.	400
MissingManifestFieldException	One or more required fields were missing from the manifest file. Please correct and resubmit.	400
MissingParameterException	One or more required parameters was missing from the request.	400
MultipleRegionsException	Your manifest file contained buckets from multiple regions. A job is restricted to buckets from one region. Please correct and resubmit.	400

Code	Description	HTTP Code
NoSuchBucketException	The specified bucket does not exist. Create the specified bucket or change the manifest's *bucket*, *exportBucket*, or *logBucket* field to a bucket that the account, as specified by the credentials file, has write permissions to.	400
OptInRequired	The AWS account needs a subscription to the AWS Import/Export service.	403
UnableToCancelJobIdException	AWS Import/Export cannot cancel the job.	409
UnknownOperationException	The specified operation was invalid.	403

Job and Device Status Codes

Topics

Each AWS Import/Export job has both a device progress status and job progress status. The device location status provides details regarding the location of the storage device you mailed to AWS. The job progress code reflects the status of the data transfer, for example, when your storage device arrives it's status is `Pending`, when we are transferring the data the status is `InProgress`, and when the data transfer completes successfully the status is `Success`.

Device Location Status Codes

The following table describes device location status codes.

Code	Description
AtAWS	Your device is at AWS.
NotReceived	AWS has not received your device.
Returned	Your device has been returned. The *SHIPPER* tracking number is *TRACKING_NUMBER*.

Job Progress Status Codes

The following table describes job progress status codes.

Code	Description
Canceled	The job was canceled.
Complete	The data loaded successfully. The log was saved to *LOG_KEY* in the bucket *LOG_BUCKET*.
DeviceError	The job was aborted. Your device would not function properly.
Expired	The job expired.

Code	Description
InProgress	The data on your device is currently being loaded.
InvalidSignature	The job was aborted. The SIGNATURE file was invalid.
MissingDevice	The job was aborted. Your device, power supply, or device connectors were missing.
MissingSignature	The job was aborted. The SIGNATURE file was missing.
Pending	The specified job has not started.
SuccessWithErrors	The data load completed with *NUMBER* of errors. See *LOG_KEY* in the bucket *LOG_BUCKET*. for more details.
UnsupportedDeviceType	The job was aborted. AWS Import/Export requires a USB2 or eSATA connector type, or a 2.5" or 3.5" SATA hard drive.
UnsupportedFileFormat	The job was aborted. For supported file types, see For more information, see Guidelines and Limitations (p. 130).
UnsupportedPowerSupply	The job was aborted. AWS Import/Export on US Standard buckets requires a 120 volt 60Hz, compatible power supply. EU buckets require a 230 volt, 50 Hertz compatible power supply that uses a Euro plug (C connector or D connector) or UK plug (BS1363). Singapore buckets require 230 volts, 50 Hertz compatible power supply.

AWS Import/Export Log Status Codes

The following table describes AWS Import/Export status codes.

Code	Status	Description
AccessDenied	403 Forbidden	Access Denied. Your IAM account does not have read permission on the object.
DirectoryFull	302 Remapped	The current directory is full. Files can no longer be written to this location. Your file was saved to the recovery directory.
FilenameTooLong	302 Remapped	The requested bucket + prefix + key path exceeded the max file name length of the target file system.
FileTooLarge	413 Request Entity Too Large	The requested object was larger than the max file size of the target file system.
Ignored	103	File passed the manifest ignored filter.
InvalidFilename	302 Remapped	The requested key name is invalid on the target file system. Your file was saved to the recovery directory.
InvalidSubstitution	400	The specified *substitutions* manifest option resulted in an invalid object name.
OK	200	Success.

Code	Status	Description
ReadError	508 Device Error	Unable to read the specified file from disk.
UnableToWrite	508 Device Error	Unable to write data to your storage device.

Guidelines and Limitations

The following section provides guidelines and describes limitations when using the AWS Import/Export service. If any of these limitations are important to your use case, contact AWS Support Center with details.

- To connect your storage device to one of our AWS Import/Export stations, your device must meet the following requirements:
  - The device must be compatible with Red Hat Linux and warranted by the manufacturer to support eSATA, USB 3.0, or USB 2.0 interface on Linux.
  - Maximum device size is 14 inches/35 centimeters high by 19 inches/48 centimeters wide by 36 inches/91 centimeters deep (8Us in a standard 19 inch/48 centimeter rack).
  - Maximum weight is 50 pounds/22.5 kilograms.
  - Maximum device capacity is 16 TB.
  - Power requirements vary by region. For more information, go to the Selecting Your Storage Device section on the AWS Import/Export Product Details page.
- You can create up to 50 jobs per day. If you need to create more than 50 jobs, contact your account manager or AWS Support Center.
- Ship only one device per package. AWS Import/Export will accept multiple devices in a single shipment only for specific devices, with special packaging requirements. For details, see Shipping Multiple Devices (p. 68).
- AWS Import/Export service does not ship to PO Boxes.
- Expedited shipping is available only for return shipments from U.S. data loading facilities to U.S. addresses.
- AWS Import/Export exports only the latest version from an Amazon S3 bucket that has versioning turned on.
- Amazon DevPay buckets and accounts are not supported. You must use your AWS account credentials and non-DevPay buckets when using AWS Import/Export.
- Devices must be healthy. Providing an unhealthy device, such as a device with bad sectors or unstable power, may result in a partial import or export. In extreme cases the import or export will be impossible to perform, and AWS Import/Export might destroy the device. For more information, see Failed Jobs (p. 75).
- Amazon S3 import jobs support the following file systems:
  - NTFS
  - EXT2
  - EXT3
  - EXT4

  - FAT32
  - HFSPlus (only partitions 2TB and smaller)
  - exFAT
- Import to Amazon S3 job file names must be written using UTF-8 character encoding. Any file with a name that is not a valid UTF-8 string is not imported.
- Amazon S3 export jobs support the following file systems:
  - NTFS
  - EXT4
- With an Amazon S3 export, some objects might be re-mapped during export to different filenames due to file system limitations. Some of the reasons for remapping filenames include:
  - The source Amazon S3 object has a non UTF-8 key name.
  - The Amazon S3 source has two or more object key names that point to the same file name (for instance, `folder1/file1` and `folder1/folder2/../file1`)
  - Too many files exist under one directory.

  You can find out whether files were remapped for your job by examining the export log uploaded to your log bucket.
- If your import job fails, your device will be erased and returned to you. If your export job fails, your device will be returned to you. For more information, see Failed Jobs (p. 75).

Document History

The following table describes the important changes to the documentation since the last release of *AWS Import/Export*.

- **API version:** 2010-06-03
- **Latest document update:** May 6, 2014

Change	Description	Date Changed
File system support for export	EXT2, EXT3, exFAT, and FAT32 are no longer supported for export from Amazon S3. For more information, see Guidelines and Limitations (p. 130).	May 6, 2014
AWS IAM Support	AWS Import/Export integrates with AWS Identity and Access Management (IAM), which allows you to control which operations a user can perform. For more information, see Using IAM with AWS Import/Export (p. 26).	February 17, 2014
Device erasure and TrueCrypt encryption	AWS Import/Export will erase all devices following import to Amazon S3, Amazon EBS, and Amazon Glacier. AWS will encrypt all devices using TrueCrypt prior to performing an export to Amazon S3. See Using TrueCrypt Encryption (p. 66). The `notificationEmail` field is required for job types. Added information about failed jobs. For more information, see Failed Jobs (p. 75).	January 31, 2014
exFAT file system support for import	The exFAT file system is supported for import jobs. For more information, see Guidelines and Limitations (p. 130).	July 24, 2013
Amazon Glacier Storage Class	If you want the lower cost storage provided by Amazon Glacier, but you want your individual files to be stored as separate objects, first import your files to Amazon S3, then use Amazon S3 Object Lifecycle Management to transition your files to the Amazon Glacier Storage Class. For more information, see Creating Amazon Glacier Import Jobs (p. 36)	July 24, 2013

Change	Description	Date Changed
Device encryption	Added support for encrypting devices using TrueCrypt for Amazon S3 import and export. See Using TrueCrypt Encryption (p. 66).	May 22, 2013
Shipping multiple devices.	Added instructions for shipping multiple devices in the same package. See Shipping Multiple Devices (p. 68).	January 1, 2013
Added regions for import to EBS.	AWS Import/Export has expanded support for importing data to Amazon Elastic Block Store (Amazon EBS) to include the EU West (Dublin) and Asia Pacific (Singapore) regions.	December 31, 2012
Reformat summary of Import/Export types	The summary of Import/Export types has been reformatted to improve readability on Kindle devices.	August 20, 2012
Add summary of Import/Export types	The AWS Import/Export Concepts section includes a table summarizing the source, target, and results for each of the AWS Import/Export job types. For more information, see Job (p. 2).	August 20, 2012
Add support for Amazon Glacier	AWS Import/Export now supports importing data to Amazon Glacier vaults. For more information, see Create Your First Amazon Glacier Import Job (p. 17).	August 20, 2012
Consolidate guides.	The *AWS Import/Export API Reference* is now part of the AWS Import/Export Developer Guide.	July 7, 2011
API version correction.	The correct, current API version is 2010-06-03.	July 7, 2011
New instructions for locating your Amazon EBS Snapshot.	After a successful upload of data to an Amazon EBS Snapshot, you can find the data in the AWS Management Console for Amazon EC2. See Your Amazon EBS Snapshot in the AWS Management Console (p. 35).	July 7, 2011
New Getting Started instructions for importing data to Amazon Elastic Block Store (Amazon EBS).	AWS Import/Export now supports importing data to Amazon EBS snapshots. Amazon EBS snapshots can be converted into volumes for use with Amazon Elastic Compute Cloud (EC2). For more information, see Create Your First Amazon EBS Import Job (p. 13).	July 7, 2011
New instructions for preparing an import job to Amazon EBS.	An import job to Amazon EBS follows a different process than an import job to Amazon S3. For more information, see Creating Amazon EBS Import Jobs (p. 32).	July 7, 2011
New manifest file options for an import job to Amazon EBS.	An import job to Amazon EBS has its own unique manifest file options. For more information, see Import to Amazon EBS Manifest File Options (p. 91).	July 7, 2011

Change	Description	Date Changed
New import manifest options for adding metadata	AWS Import/Export has new manifest options for users to better manage their data loads into Amazon S3. For import jobs, you can assign user defined metadata to all objects and a metadata value to store the last modified time stamp for each imported file. See the *diskTimestampMetadataKey* and *staticMetadata* options in Import to Amazon S3 Manifest File Options (p. 84).	March 22, 2011
A new common manifest option for modifying object keys and file names	Modify uploaded object keys and downloaded filenames with the *substitutions* option. This option allows you to specify rules for naming the keys when importing data to Amazon S3, and filenames when exporting data to your device. See the *substitutions* option in Common Manifest File Options (p. 76)	March 22, 2011
AWS Import/Export US West (Northern California) region support	AWS Import/Export now supports importing and exporting data into and out of Amazon S3 buckets in the US West (Northern California) region.	February 1, 2011
AWS Import/Export Asia Pacific (Singapore) region support	AWS Import/Export now supports importing and exporting data into and out of Amazon S3 buckets in the Asia Pacific (Singapore) region.	December 28, 2010
Support for large objects	AWS Import/Export now supports importing and exporting objects up to 5 TB in size.	December 9, 2010
AWS Import/Export becomes a web service	AWS Import/Export is now a web service. In previous versions of this product, you used email to create and manage jobs. Those email commands and procedures are being deprecated and will stop working on December 31, 2010. This guide no longer contains those commands.	June 9, 2010
AWS Import/Export command line interface	In addition to the new web service API, AWS Import/Export now has a command line interface. For more information, go to AWS Import/Export Command Line Quick Reference Card.	June 9, 2010
New support for internal SATA and 4 TB device capacity limit	AWS Import/Export now supports internal SATA hard drives for data loads in the Amazon S3 Standard and EU (Ireland) Regions. Additionally, you can now send portable storage devices with capacities up to 4 TB, reducing the number of devices required for large data loads.	March 5, 2010
International support for AWS Import/Export	Now, you can send and receive storage devices to and from most international locations for loading data to and from US Standard Region buckets. For more information, see Shipping Your Storage Device (p. 63).	December 9, 2009
Support for EU (Ireland) buckets	Now, you can import and export data to and from EU (Ireland) buckets. For more information, see AWS Shipping Addresses (p. 68).	December 9, 2009

Appendices

This AWS Import/Export guide appendix include the following sections.

Topics

Appendix A: Other Tools and Services

In addition to the AWS Import/Export Web Service Tool and the AWS SDK, there are a number of third party tools and services available to work with AWS Import/Export. For more information, go to http://aws.amazon.com/importexport/tools/.

Appendix B: Examples

Topics

Import Examples

This section walks you through creating three sample AWS Import/Export jobs and shows the manifest file's corresponding signature and log files. In these examples, the following files are at the root directory of the storage device. The resulting log files illustrate how manifest fields effect the data load.

> **Note**
> These examples do not apply to Amazon Glacier. Amazon Glacier imports the entire contents of a device to a single archive.

- /README
- /NOTICE.txt

- /images/wookie1.jpg
- /images/chewie.gif
- /images/chewie.tif
- /images/resources/chewie.psd

The following sections show different ways to create AWS Import/Export jobs.

Example – TSTD1

The TSTD1 example demonstrates the default behavior using a manifest file with the minimum configuration options. The example files associated with job TSTD1 are:

- `manifest-TSTD1.txt`
- `import-log-TSTD1.csv`

The following is the contents of `manifest-TSTD1.txt` file.

```
manifestVersion: 2.0
bucket: myawsbucket
deviceId: 3QD0T87T
eraseDevice: Yes
notificationEmail: john.doe@example.com;jane.roe@example.com
returnAddress:
   name: Jane Roe
   company: Example Corp.
   street1: 123 Any Street
   city: Anytown
   stateOrProvince: WA
   postalCode: 91011-1111
   phoneNumber: 206-555-1111
   country: USA
```

Important
Your return address must be a physical street address in the United States that UPS can deliver to.

TSTD1 Log File

After AWS Import/Export processes this job, the service saves the resulting log file to `http://s3.amazonaws.com/myawsbucket/import-log-TSTD1`. The contents of the log file are shown in the following table.

DateTime	File	Status	Code	Key	MD5	Bytes	Content-Type
Tue 03 Feb 2009 21:57:43 GMT	SIGNATURE	103	Ignored			90	text/plain

DateTime	File	Status	Code	Key	MD5	Bytes	Content-Type
Tue 03 Feb 2009 21:57:43 GMT	Notice.txt	200	OK	myawsbucket/ Notice.txt	f60fe317bc4 97b1204b327094 959eb64	13	text/plain
Tue 03 Feb 2009 21:57:43 GMT	README.txt	200	OK	myawsbucket/ README.txt	d5a46fa2259 6d2464eff10bab b19000e	13	text/plain
Tue 03 Feb 2009 21:57:43 GMT	images /wookie1.jpg	200	OK	myawsbucket/ images/ wookie1.jpb	ccac0d031603 ff1c2994e07703 a64845	447	image/jpeg
Tue 03 Feb 2009 21:57:43 GMT	images/ chewie.GIF	200	OK	myawsbucket/ images/ chewie.GIF	5fc22dc594e 0c5929c090b958 5ba9e70	848	image/gif
Tue 03 Feb 2009 21:57:43 GMT	images/ resources/ chewie.psd	200	OK	myawsbucket/ images/ resources/ chewie.psd	130b64d171 e9bbc2a8dbe117 f189796c	524	application/ octet-stream
Tue 03 Feb 2009 21:57:43 GMT	images/ chewie.TIF	200	OK	myawsbucket/ images/ chewie.TIF	823d17b0 5321b968e01f917 4e3dcace7	738	image/tiff

Example – TSTD2

The TSTD2 example demonstrates the behavior associated with adding the following manifest fields: *`acl`*, *`prefix`*, and *`logPrefix`*, to the manifest file used in TSTD1. The example files associated with job TSTD2 are:

- `manifest-TSTD2.txt`
- `import-log-TSTD2.csv`

The following text shows the contents of `manifest-TSTD2.txt`.

```
manifestVersion: 2.0
bucket: myawsbucket
deviceId: eQD0T87T
eraseDevice: Yes
notificationEmail: john.doe@example.com;jane.doe@example.com
returnAddress:
   name: Jane Roe
```

```
   company: Example Corp.
   street1: 123 Any Street
   city: Anytown
   stateOrProvince: WA
   postalCode: 91011-1111
   phoneNumber: 206-555-1111
   country: USA
acl: public-read
prefix: images/
logPrefix: loadlogs-
```

TSTD2 Log File

After AWS Import/Export processes this job, the service saves the resulting log file to `http://s3.amazonaws.com/myawsbucket/import-log-TSTD2`. The contents of the log file are shown in the following table.

DateTime	File	Status	Code	Key	MD5	Bytes	Content-Type
Tue 03 Feb 200921:57:43 GMT	SIGNATURE	103	Ignored			90	
Tue 03 Feb 200921:57:43 GMT	Notice.txt	200	OK	myawsbucket/images/Notice.txt	f60fe317bc497b12004b327094959eb64	13	text/plain
Tue 03 Feb 200921:57:43 GMT	README.txt	200	OK	myawsbucket/images/README.txt	d5a46fa22596d2464eff10babb19000e	13	text/plain
Tue 03 Feb 200921:57:43 GMT	images/wookie1.jpg	200	OK	myawsbucket/images/images/wookie1.jpg	ccac0d031603ff1c94e07703a64845	444	image/jpeg
Tue 03 Feb 200921:57:43 GMT	images/chewie.GIF	200	OK	myawsbucket/images/images/chewie.GIF	5f22dc594e0c592929c090b9585ba9e70	846	image/gif
Tue 03 Feb 200921:57:43 GMT	images/resources/chewie.psd	200	OK	myawsbucket/images/images/resources-chewie.psd	130b64d171e9bbc28dbe117f189796c	590	application/octet-stream
Tue 03 Feb 200921:57:43 GMT	images/chewie.TIF	200	OK	myawsbucket/images/images/chewie.TIF	823d17b05321b968e01f9174e3dcace7	734	image/tiff

Example – TSTD3

The TSTD3 example load demonstrates the behavior of the following manifest fields: *`acl`*, *`contentTypes`*, *`prefix`*, and *`ignore`*. The example files associated with job TSTD3 are:

- `manifest-TSTD3.txt`
- `import-log-TSTD3.csv`

The following text shows the contents of `manifest-TSTD3.txt` file.

```
manifestVersion: 2.0
bucket: myawsbucket
deviceId: 3QD0T87T
eraseDevice: Yes
notificationEmail: john.doe@example.com;jane.doe@example.com
returnAddress:
   name: Jane Roe
   company: Example Corp.
   street1: 123 Any Street
   city: Anytown
   stateOrProvince: WA
   postalCode: 91011-1111
   phoneNumber: 206-555-1111
   country: USA
acl: public-read
prefix: dropoff
contentTypes:
  jpg: application/octet-stream
  gif: application/octet-stream
  tif: application/octet-stream
  txt: text/html
ignore:
    - \.psd$
```

TSTD3 Log File

After AWS Import/Export processes this job, the service saves the resulting log file to `http://s3.amazonaws.com/myawsbucket/import-log-TSTD3`. The contents of the log file are shown in the following table.

DateTime	File	Status	Code	Key	MD5	Bytes	Content-Type
Tue 03 Feb 200921:57:43 GMT	SIGNATURE	103	Ignored			90	
Tue 03 Feb 200921:57:43 GMT	Notice.txt	200	OK	myawsbucket/ dropoffNotice .txt	386317bc4 97b1202709 4959eb64	13	text/html
Tue 03 Feb 200921:57:43 GMT	README.txt	200	OK	myawsbucket/ dropoffREADME. txt	4686fa225 96d2464e0b abb19000e	13	text/html
Tue 03 Feb 200921:57:43 GMT	images/ wookie1.jpg	200	OK	myawsbucket/ dropoffimages/ wookie1.jpg	586d0316 03ff1c299770 3a64845	449	application/ octet-stream
Tue 03 Feb 200921:57:43 GMT	images/ chewie.GIF	200	OK	myawsbucket/ dropoffimages/ chewie.GIF	89022dc5 94e0c5c090b 585ba9e70	846	image/gif
Tue 03 Feb 200921:57:43 GMT	images/ resources/ chewie.psd	108	Ignored			590	

DateTime	File	Status	Code	Key	MD5	Bytes	Content-Type
Tue 03 Feb 200921:57:43 GMT	images/ chewie.TIF	200	OK	myawsbucket/ dropoffimages/ chewie.TIF	8237897-21 b968e0174 e3dcace7	734	image/tiff

Note
The *contentTypes* field is case sensitive, so `chewie.TIF` and `chewie.GIF` were not picked up by the `gif` or `tif` *contentTypes* filter.
The *prefix* field, *dropoff*, did not have a trailing forward slash, (/), so the key name did not have a delimiter between *dropoff* and the loaded file path.

Export Examples

Topics

This section walks through example AWS Import/Export jobs and shows the manifest file's corresponding `SIGNATURE` and log files.

Export Files

The following keys are referenced as part of each export job.

- /myawsbucket/README
- /myawsbucket /NOTICE.txt
- /myawsbucket /images/wookie1.jpg
- /myawsbucket /images/chewie.gif
- /myawsbucket /images/chewie.tif
- /myawsbucket /images/resources/chewie.psd
- /myawsbucket//backup/images/chewie.gif
- /lostbucket/bin/lost.rhtml
- /lostbucket/html/directory.html
- /lostbucket/images/hurley.jpg
- /lostbucket/images/john.jpg
- /lostbucket/images/kate.jpg
- /lostbucket/images/sawyer.jpg
- /lostbucket/index.html

Example – TSTDA

The TSTDA example upload demonstrates the default behavior using a manifest file with the minimum configuration options. The example files associated with job TSTDA are:

- `manifest-TSTDA.txt`
- `export-log-TSTDA.csv`

The following is the contents of `manifest-TSTDA.txt` file.

```
manifestVersion: 1.2

deviceId: 532404500021
logBucket: myawsbucket
fileSystem: FAT32
operations:
    - exportBucket: myawsbucket
returnAddress:
   name: Jane Roe
   company: Example Corp.
   street1: 123 Any Street
   city: Anytown
   stateOrProvince: WA
   postalCode: 91011-1111
   phoneNumber: 206-555-1111
   country: USA
```

TSTDA Log File

After AWS Import/Export processes this job, the service saves the resulting log file to `http://s3.amazonaws.com/myawsbucket/export-log-TSTDA`. The contents of the log file are shown in the following table.

DateTime	File	Status	Code	Key	MD5	Bytes	Content-Type
Tue 03 Feb 200921:57:43 GMT	myawsbucket/ README.txt	200	OK	myawsbucket/ README.txt	d5a46fa2259 6d2464eff10 babb19000e	13	text/html
Tue 03 Feb 200921:57:43 GMT	myawsbucket/ NOTICE.txt	200	OK	myawsbucket/ NOTICE.txt	f60fe317bc4 97b1204b327 094959eb64	13	text/html
Tue 03 Feb 200921:57:43 GMT	myawsbucket/ images/ wookie1.jpg	200	OK	myawsbucket/ images/ wookie1.jpg	ccac0d031603 ff1c2994e07 703a64845	447	image/jpeg
Tue 03 Feb 200921:57:43 GMT	myawsbucket/ images/ chewie.gif	200	OK	myawsbucket/ images/ chewie.gif	5fc22dc594e 0c5929c090 b9585ba9e70	848	image/gif
Tue 03 Feb 200921:57:43 GMT	myawsbucket/ images/ resources/chewie.psd	200	OK	myawsbucket /images/ resources/ chewie.psd	130b64d171 e9bbc2a8dbe1 17f189796c	524	application/ octet-stream
Tue 03 Feb 200921:57:43 GMT	EXPORT-RECOVERY /0000/0000	302	Remapped	myawsbucket// backup/ images/ chewie.gif	823d17b0 5321b968e01f9 174e3dcace7	738	image/gif

Example – TSTDB

The TSTDB example upload demonstrates the default behavior using a manifest file with the following configuration options: multiple buckets, *`targetDirectory`*, *`prefix`*, *`beginMarker`*, and *`endMarker`*. The example files associated with job TSTDB are:

- `manifest-TSTDB.txt`
- `export-log-TSTDB.csv`

The following is the contents of `manifest-TSTDB.txt` file.

```
manifestVersion: 2.0
deviceId: 532404500021
logBucket: myawsbucket
logPrefix: logs/
fileSystem: NTFS
operations:
    - exportBucket: myawsbucket
      prefix: images
      targetDirectory: starwars/image-backup
    - exportBucket: lostbucket
      beginMarker: html/
      endMarker: images/kate.jpg
returnAddress:
   name: Jane Roe
   company: Example Corp.
   street1: 123 Any Street
   city: Anytown
   stateOrProvince: WA
   postalCode: 91011-1111
   phoneNumber: 206-555-1111
   country: USA
```

TSTDB Log File

After AWS Import/Export processes this job, the service saves the resulting log file to `http://s3.amazonaws.com/myawsbucket/export-log-TSTDB`. The contents of the log file are shown in the following table.

DateTime	File	Status	Code	Key	MD5	Bytes	Content-Type
Tue 03 Feb 200921:57:43 GMT	starwars/ image-backup/ images/ wookie1.jpg	200	OK	myawsbucket /images/wookie1.jpg	d5a46fa2259 6d246f10bab b19000e	13	image/ jpeg
Tue 03 Feb 200921:57:43 GMT	starwars/ image-backup/ images/ chewie.gif	200	OK	myawsbucket /images/chewie.gif	f60fe317bc4 97b12047094 959eb64	13	image/gif
Tue 03 Feb 200921:57:43 GMT	starwars/ image-backup/ images/chewie.tif	200	OK	myawsbucket /images/chewie.tif	ccac0d031603 ff1c4e07703 a64845	447	image/ tiff

DateTime	File	Status	Code	Key	MD5	Bytes	Content-Type
Tue 03 Feb 200921:57:43 GMT	starwars/ image-backup/ images/ resources/ chewie.psd	200	OK	myawsbucket /images/ resources/ chewie.psd	5fc22dc594e 0c5929cb958 5ba9e70	848	appli cation/ octet-stream
Tue 03 Feb 200921:57:43 GMT	lostbucket/ html/ directory.html	200	OK	lostbucket/html/ directory.html	130b64d171 e9bbc2ae117 f189796c	524	text/html
Tue 03 Feb 200921:57:43 GMT	lostbucket/ images/ hurley.jpg	200	OK	lostbucket/ images/hurley.jpg	823d17b0 5321b968f917 4e3dcace7	738	image/ jpeg
Tue 03 Feb 200921:57:43 GMT	lostbucket/ images/ john.jpg	200	OK	lostbucket/ images/john.jpg	823d17b0 5321b968e91 74e3dcace7	615	image/ jpeg
Tue 03 Feb 200921:57:43 GMT	lostbucket/ images/ kate.jpg	200	OK	lostbucket/ images/kate.jpg	5fc22dc594 e0c5929c90b 9585ba9e70	345	image/ jpeg

Appendix C: File Extension to Mime Types

The following table is derived from the Internet Assigned Numbers Authorities MIME Media Types web site. For more information, go to http://www.iana.org/assignments/media-types. AWS Import/Export does a case insensitive lookup against this table. File types not included in this table use the Amazon S3 default, Content-Type application/octet-stream.

Attribute	Description
ai	application/postscript
aif	audio/x-aiff
aifc	audio/x-aiff
aiff	audio/x-aiff
asc	text/plain
asf	video/x-ms-asf
asx	video/x-ms-asf
au	audio/basic
avi	video/x-msvideo
bcpio	application/x-bcpio
bin	application/octet-stream
bmp	image/bmp
bz2	application/x-bzip2

Attribute	Description
cdf	application/x-netcdf
chrt	application/x-kchart
class	application/octet-stream
cpio	application/x-cpio
cpt	application/mac-compactpro
csh	application/x-csh
css	text/css
dcr	application/x-director
dir	application/x-director
djv	image/vnd.djvu
djvu	image/vnd.djvu
dll	application/octet-stream
dms	application/octet-stream
doc	application/msword
dvi	application/x-dvi
dxr	application/x-director
eps	application/postscript
etx	text/x-setext
exe	application/octet-stream
ez	application/andrew-inset
flv	video/x-flv
gif	image/gif
gtar	application/x-gtar
gz	application/x-gzip
hdf	application/x-hdf
hqx	application/mac-binhex40
htm	text/html
html	text/html
ice	x-conference/x-cooltalk
ief	image/ief
iges	model/iges
igs	model/iges

Attribute	Description
img	application/octet-stream
iso	application/octet-stream
jad	text/vnd.sun.j2me.app-descriptor
jar	application/x-java-archive
jnlp	application/x-java-jnlp-file
jpe	image/jpeg
jpeg	image/jpeg
jpg	image/jpeg
js	application/x-javascript
kar	audio/midi
kil	application/x-killustrator
kpr	application/x-kpresenter
kpt	application/x-kpresenter
ksp	application/x-kspread
kwd	application/x-kword
kwt	application/x-kword
latex	application/x-latex
lha	application/octet-stream
lzh	application/octet-stream
m3u	audio/x-mpegurl
man	application/x-troff-man
me	application/x-troff-me
mesh	model/mesh
mid	audio/midi
midi	audio/midi
mif	application/vnd.mif
mov	video/quicktime
movie	video/x-sgi-movie
mp2	audio/mpeg
mp3	audio/mpeg
mpe	video/mpeg
mpeg	video/mpeg

Attribute	Description
mpg	video/mpeg
mpga	audio/mpeg
ms	application/x-troff-ms
msh	model/mesh
mxu	video/vnd.mpegurl
nc	application/x-netcdf
odb	application/vnd.oasis.opendocument.database
odc	application/vnd.oasis.opendocument.chart
odf	application/vnd.oasis.opendocument.formula
odg	application/vnd.oasis.opendocument.graphics
odi	application/vnd.oasis.opendocument.image
odm	application/vnd.oasis.opendocument.text-master
odp	application/vnd.oasis.opendocument.presentation
ods	application/vnd.oasis.opendocument.spreadsheet
odt	application/vnd.oasis.opendocument.text
ogg	application/ogg
otg	application/vnd.oasis.opendocument.graphics-template
oth	application/vnd.oasis.opendocument.text-web
otp	application/vnd.oasis.opendocument.presentation-template
ots	application/vnd.oasis.opendocument.spreadsheet-template
ott	application/vnd.oasis.opendocument.text-template
pbm	image/x-porinformaltable-bitmap
pdb	chemical/x-pdb
pdf	application/pdf
pgm	image/x-porinformaltable-graymap
pgn	application/x-chess-pgn
png	image/png
pnm	image/x-porinformaltable-anymap
ppm	image/x-porinformaltable-pixmap
ppt	application/vnd.ms-powerpoint
ps	application/postscript
qt	video/quicktime

Attribute	Description
ra	audio/x-realaudio
ram	audio/x-pn-realaudio
ras	image/x-cmu-raster
rgb	image/x-rgb
rm	audio/x-pn-realaudio
roff	application/x-troff
rpm	application/x-rpm
rtf	application/rtf
rtf	text/rtf
rtx	text/richtext
sgm	text/sgml
sgml	text/sgml
sh	application/x-sh
shar	application/x-shar
silo	model/mesh
sis	application/vnd.symbian.install
sit	application/x-stuffit
skd	application/x-koan
skm	application/x-koan
skp	application/x-koan
skt	application/x-koan
smi	application/smil
smil	application/smil
snd	audio/basic
so	application/octet-stream
spl	application/x-futuresplash
src	application/x-wais-source
stc	application/vnd.sun.xml.calc.template
std	application/vnd.sun.xml.draw.template
sti	application/vnd.sun.xml.impress.template
stw	application/vnd.sun.xml.writer.template
sv4cpio	application/x-sv4cpio

Attribute	Description
sv4crc	application/x-sv4crc
swf	application/x-shockwave-flash
sxc	application/vnd.sun.xml.calc
sxd	application/vnd.sun.xml.draw
sxg	application/vnd.sun.xml.writer.global
sxi	application/vnd.sun.xml.impress
sxm	application/vnd.sun.xml.math
sxw	application/vnd.sun.xml.writer
t	application/x-troff
tar	application/x-tar
tcl	application/x-tcl
tex	application/x-tex
texi	application/x-texinfo
texinfo	application/x-texinfo
tgz	application/x-gzip
tif	image/tiff
tiff	image/tiff
torrent	application/x-bittorrent
tr	application/x-troff
tsv	text/tab-separated-values
txt	text/plain
ustar	application/x-ustar
vcd	application/x-cdlink
vrml	model/vrml
wav	audio/x-wav
wax	audio/x-ms-wax
wbmp	image/vnd.wap.wbmp
wbxml	application/vnd.wap.wbxml
wm	video/x-ms-wm
wma	audio/x-ms-wma
wml	text/vnd.wap.wml
wmlc	application/vnd.wap.wmlc

Attribute	Description
wmls	text/vnd.wap.wmlscript
wmlsc	application/vnd.wap.wmlscriptc
wmv	video/x-ms-wmv
wmx	video/x-ms-wmx
wrl	model/vrml
wvx	video/x-ms-wvx
xbm	image/x-xbitmap
xht	application/xhtml+xml
xhtml	application/xhtml+xml
xls	application/vnd.ms-excel
xml	text/xml
xpm	image/x-xpixmap
xsl	text/xml
xwd	image/x-xwindowdump
xyz	chemical/x-xyz
zip	application/zip

Appendix D: AWS Import/Export Resources

The following table lists related resources that you'll find useful as you work with this service.

Resource	Description
API Reference (p. 103)	Explains the API operations.
AWS Import/Export Command Line Quick Reference Card	Explains the command line version of the Amazon Import/Export operations.
FAQ	Presents frequently asked questions and answers about AWS Import/Export.
Amazon S3 Discussion Forum	A community-based forum for developers to discuss technical questions related to Amazon S3 and AWS Import/Export.
AWS Support Center	The home page for AWS Technical Support, including access to our Developer Forums, Technical FAQs, Service Status page, and Premium Support.
Contact Us	A central contact point for inquiries concerning AWS billing, account, events, abuse etc.

Resource	Description
Legal	Detailed information about the terms of use, privacy policy and other topics.