AWS Marketplace Catalog API

Reference

Table of Contents

AWS Marketplace Catalog API

The AWS Marketplace Catalog API service provides an API interface to manage AWS Marketplace for your AWS organization or AWS account. For approved sellers, you can manage your products programmatically, including the self-service publishing capabilities on the AWS Marketplace Management Portal. For private marketplace administrators, you can manage your private marketplace programmatically.

With Catalog API actions, you can view and update your existing product programmatically. You can automate your product update process by integrating the AWS Marketplace Catalog API with your AWS Marketplace product build or deployment pipelines. You can also create your own applications on top of the Catalog API to manage your products on AWS Marketplace. You can manage the products that users in your AWS account or AWS organization can see and purchase through your private marketplace.

The AWS Marketplace Catalog API service provides standard AWS API functionality. You can directly use the REST APIs described here (see Getting Started with AWS to learn more about AWS application development), or you can use one of the AWS SDKs to access an API that's tailored to the programming language or platform that you're using. For more information, see AWS SDKs.

Supported AWS Regions

You can access the AWS Marketplace Catalog API from the US East (N. Virginia) AWS Region with the following endpoint.

```
catalog.marketplace.us-east-1.amazonaws.com
```

Catalog API entities

AWS Marketplace entities are containers of data which serve different business purposes, such as a product or offer. Entities are categorized by types. Each entity type encapsulates data related to a specific business domain (for example, a product or a seller account).

To simplify this paradigm, entities are designed with some level of commonality in their structures. As a result, introducing a new business domain does not require you to learn a completely new structure.

General structure

The general structure of any entity is:

- A named type with a version.
- An identifier for the specific instance of the type.
- One or more facets that include the attributes of the entity.

Type versioning

Every named type has a type and version associated with it, for example, *Entity*`Product@1.0`. The *type* (*Entity*Product) represents the classification of the content. The *version* (1.0) represents the structure of *Entity*Product.

The version gives you details about the structure of the entity. The following describes when a version will be changed.

- Existing entities won't be restructured without changing the version. Additions of optional new fields will result in a minor version update.
- Any feature that fundamentally changes the structure of a type leads to a major version update. Examples include:
  - Removing a field
  - Renaming a field (different name for the same semantic)
  - Changing the semantic of an existing field (for example, changing the expected type)
- A major version update can retain a subset of facets from the previous version.
- Users are provided notifications and documentation for new versions.

Identifier

Each entity represents a unique *thing* within a business domain. To identify the unique thing, we use an identifier associating an `EntityId` with a `RevisionId`, for example, *`prod-ad8EXAMPLE651`*@*`3`*. In this example, the `EntityId` is prod-ad8EXAMPLE651 and the `RevisionId` is 3. Every successful change request to the entity will update the revision.

The following are important details about the Identifier.

- Each entity is uniquely identified by its `EntityId`, which is the key to globally distinguish one entity from another.
- Each published revision of an entity has a `RevisionId`. The `revisionId`, along with the `EntityId`, distinguish one published revision from another.
- AWS Marketplace generates `EntityIds` and `RevisionIds`.

You can use the `DescribeEntity` action to find the details and the Identifier with the most recent `revisionId`.

The `RevisionId` is an optional part of requests to `StartChangeSet` (see Working with change sets (p. 3)). If you include a `RevisionId`, then the request to `StartChangeSet` will fail with a `ValidationException` if the `RevisionId` is not the latest revision of the entity. This allows you to implement optimistic locking in your application.

> **Note**
> When you include a `RevisionId` that is not the latest revision, the `ValidationException` message includes the latest `RevisionId`.

If you omit the `RevisionId`, the request is performed on the latest revision of the entity automatically.

> **Warning**
> Two requests to change the same object could result with one request overwriting the changes of the other request, as the second request rewrites data changed by the first request. Using `RevisionIds` in your requests prevents this issue by not allowing a change to an earlier revision to overwrite the current revision.

Facets

A facet is a logical grouping of attributes. An entity usually includes several facets which represent different aspects of the entity. The attributes within a facet have the following properties:

- Each attribute has a unique name within the scope of the container it belongs to.
- Attributes can be of a simple type (string, integer, or floating number).

- Attributes can be of a complex type (container/structure or array).

Product entity

A software product you own and list on AWS Marketplace is represented by a product entity. Product entities have different types. Regardless of type, product entities have some common facets in addition to product type specific facets. The example below is an example of common facets: `Description`, `PromotionalResources`, `RegionAvailability`, and `SupportInformation`. `Details` is an example of a product type specific facet.

```
{
  "Details": "{\"Description\":{}, \"PromotionalResources\":{}, \"RegionAvailability\":{},
 \"SupportInformation\":{}}",
  "EntityArn": "arn:aws:aws-marketplace:us-east-1:0123456789012:AWSMarketplace/Entity-Type/
prod-9EXAMPLE01234,
  "EntityIdentifier": "prod-9EXAMPLE01234@23",
  "EntityType": "Entity-Type@1.0",
  "LastModifiedDate": "2019-07-31T21:59:39Z"
}
```

The Details facet is a string that includes JSON. For more information, see Working with the `Details` attribute (p. 4).

Working with change sets

When using the Catalog API, you perform many actions by creating change requests. You do this by creating and working with change sets. A change set is a list of change entities, each of which represents a request for a change in AWS Marketplace. You can make changes to the products that you provide (as a seller), or to the private marketplace that you maintain (as a buyer). Each change has an entity to change (for example, a specific product), a change type to indicate the kind of change to make (for example, updating product information), and details of the exact change to make (for example, the new product details).

There are four actions that allow you to work with change sets:

- `StartChangeSet` – Requests a set of changes. The changes are added to a queue and processed.
- `DescribeChangeSet` – Gets the details of a set of changes, including the status of the request. The statuses include:
  - `PREPARING` – Getting ready to apply the changes.
  - `APPLYING` – In the process of making the requested changes.
  - `SUCCEEDED` – Request was completed successfully.
  - `CANCELED` – Request was canceled by the user.
  - `FAILED` – Request was completed unsuccessfully. Further details are available in the response.
- `ListChangeSets` – Gets a list of the change sets that are currently in process.
- `CancelChangeSet` – Requests a change set be canceled. Changes can only be canceled while in the `PREPARING` status.

A typical workflow is to request a change with `StartChangeSet`, and then use the returned `ChangeSetId` to poll the `DescribeChangeSet` action until the change is complete.

> **Note**
> When polling or working with change sets programmatically, you must adhere to the Service limits (p. 11).

After your change is complete, you can use `ListEntities` to find the entity that you created or modified (and its associated `EntityID`). You can then use `DescribeEntity` with the `EntityID` to get details about it.

For more information about working with change requests in the console for sellers, see Creating a change request in the *AWS Marketplace Seller Guide*.

Making multiple change requests simultaneously

You can make multiple change requests at the same time, either in a single `ChangeSet`, or as separate change requests that overlap in time.

Within a **single change set**, you can bundle all change types and they are executed together.

However, if the requests are made as **separate change sets**, AWS Marketplace can't execute conflicting change requests on the same product. In these cases, AWS Marketplace returns a `ResourceInUseException` error.

- For modifying AMI and container products, most changes can be made without error, with the following exceptions.
  - If two requests are the same `ChangeType` on the same product, the second request returns an error.
  - If one request is to update the version information, and the other request is to restrict or add a version, then the second request returns an error.
  - If a request is `PREPARING`, another request can be made on the same product. However, a change that is currently `APPLYING` may block other requests, returning an error.
- For other product types and private marketplaces, you can only have a single request for a product at a time. If a different request to update the same product is made while a first request is ongoing, the second returns an error.
- If there is a request for any product that is pending with the AWS Marketplace Seller Operations team, then any other requests on that product return an error.

If you receive a `ResourceInUseException` error for a change request, you can retry the request later. Depending on the state of the ongoing request, you can also cancel the first request, to allow the resubmitted second request to complete sooner.

Working with the `Details` attribute

The `Details` attribute of the `StartChangeSet` operation is a string value. Its contents are JSON objects. To put a JSON object into a string attribute, you must convert the object to a single-line string by escaping all JSON control characters, and removing line breaks.

For example, if you are using the `StartChangeSet` operation with `UpdateProcurementPolicy` to disable requests from users in your private marketplace, make a request like the following.

```
POST /StartChangeSet HTTP/1.1
Content-type: application/json

{
   "Catalog": "AWSMarketplace",
   "ChangeSet": [
      {
         "ChangeType": "UpdateProcurementPolicy",
         "Details": "<string>",
         "Entity": {
            "Type": "Experience@1.0",
```

```
            "Identifier" : "exp-1234example@5"
         }
      }
   ]
}
```

In this case, the JSON object that you use for the `Details` attribute looks like the following (before conversion to a string).

```
{
    "Configuration": {
        "PolicyResourceRequests": "Deny"
    }
}
```

But the `Details` attribute requires a string, not JSON. After converting this JSON object to a single line string, it looks like the following.

```
"{\"Configuration\" : {\"PolicyResourceRequests\" : \"Deny\"}}"
```

With this string, you can create the full change set request, as follows.

```
POST /StartChangeSet HTTP/1.1
Content-type: application/json

{
   "Catalog": "AWSMarketplace",
   "ChangeSet": [
      {
         "ChangeType": "UpdateProcurementPolicy",
         "Details": "{\"Configuration\" : {\"PolicyResourceRequests\" : \"Deny\"}}",
         "Entity": {
            "Type": "Experience@1.0",
            "Identifier" : "exp-1234example@5"
         }
      }
   ]
}
```

Generally, examples in this API reference show the JSON object already converted to a string. In some cases, more complicated samples with new lines are included to enhance understanding.

Automate converting JSON to a string

Converting a JSON object to a string can be automated using tools such as jq, a lightweight command-line JSON processor. The following example shows using jq to convert a JSON object to a string that can be used in the `Details` attribute.

```
DETAILS_JSON='{
  "ProductTitle": "My Product Title",
  "ShortDescription": "My product short description.",
  "LongDescription": "My product long description."
}';

DETAILS_JSON_STRING="$(echo "${DETAILS_JSON}" | jq 'tostring';)";
```

If you echo "${DETAILS_JSON_STRING}", the result is the following string with JSON properly escaped:
`{\"ProductTitle\":\"My Product\",\"ShortDescription\":\"My product short description.\",\"LongDescription\":\"My product long description.\"}`

Using `DescribeEntity` to get information about your entities

You can programmatically get information about your existing entities, including products and private marketplace, through the Catalog API.

The `ListEntities` action returns a list of entities. Then, you can use the `DescribeEntity` action to get details about an individual entity. This can be directly useful, for example, to catalog the products you sell. It can also be useful when updating entities, because you can get the current state of the entity before updating just the parts that you want to update.

The following example shows using `ListEntities` to get a list of container products, and then using `DescribeEntity` to get information about one of the specific products.

```
POST /ListEntities HTTP/1.1
Content-type: application/json

{
  "Catalog": "AWSMarketplace",
  "EntityType": "ContainerProduct"
}
```

For the entity type, you must use the entity type without the version. It returns all entities of that type (and doesn't filter on version).

Here is a sample of the response to the `ListEntities` action.

```
{
    "EntitySummaryList": [
        {
            "Name": "Container Product 1",
            "EntityType": "ContainerProduct",
            "EntityId": "example1-abcd-1234-5ef6-7890abcdef12",
            "EntityArn": "arn:aws:aws-marketplace:[exampleARN]",
            "LastModifiedDate": "2021-03-01T00:00:00Z",
            "Visibility": "Public"
        },
        {
            "Name": "Container Product 2",
            "EntityType": "ContainerProduct",
            "EntityId": "example2-abcd-1234-5ef6-7890abcdef12",
            "EntityArn": "arn:aws:aws-marketplace:[exampleARN]",
            "LastModifiedDate": "2021-03-02T00:00:00Z",
            "Visibility": "Public"
        }
    ],
    "NextToken": "exampleabcdef12345..."
}
```

To get the details of one of these products, use the `DescribeEntity` action. The following example shows how to get details about the first product returned above.

```
GET /DescribeEntity?catalog=AWSMarketplace&entityId=example1-abcd-1234-5ef6-7890abcdef12
 HTTP/1.1
```

The following shows the response to `DescribeEntity`.

```
{
  "EntityType": "ContainerProduct@1.0",
  "EntityIdentifier": "example1-abcd-1234-5ef6-7890abcdef12@9",
  "EntityArn": "arn:aws:aws-marketplace:[exampleARN]",
  "LastModifiedDate": "2021-03-02T20:19:14Z",
  "Details": "{\"Versions\":[{\"Id\":\"example2-0000-aaaa-5ef6-7890abcdef12\",
\"ReleaseNotes\":\"My release notes\",\"UpgradeInstructions\":\"N/A\",\"VersionTitle\":
\"1.0\",\"CreationDate\":\"2021-03-02T00:00:00.000Z\",\"Sources\":[{\"Type\":\"DockerImages
\",\"Id\":\"example3-1111-bbbb-5ef6-7890abcdef12\",\"Images\":[\"709825985650.dkr.ecr.us-
east-1.amazonaws.com/some-seller-prefix/my-repo-1:some-tag\"],\"Compatibility\":
{\"Platform\":\"Linux\"}}],\"DeliveryOptions\":[{\"Id\":\"example4-2222-cccc-2222-
ccccccccccc\",\"Type\":\"ElasticContainerRegistry\",\"SourceId\":\"example3-1111-
bbbb-5ef6-7890abcdef12\",\"Title\":\"New delivery option 1\",\"ShortDescription
\":\"Delivery option 1\",\"isRecommended\":false,\"Compatibility\":{\"AWSServices
\":[\"ECS\",\"EKS\"]},\"Instructions\":{\"Usage\":\"test\"},\"Recommendations\":
{\"AdditionalArtifacts\":[]},\"Visibility\":\"Limited\"}]}],\"Description\":{\"Highlights
\":[\"Some highlight\"],\"LongDescription\":\"Description of my product\",\"ProductCode
\":\"123456789012abcdef1234567\",\"Manufacturer\":null,\"Visibility\":\"Limited
\",\"AssociatedProducts\":null,\"Sku\":null,\"SearchKeywords\":[\"some keyword\"],
\"ProductTitle\":\"Container Product 1\",\"ShortDescription\":\"Description of my
 product\",\"Categories\":[\"Operating Systems\"]},\"PromotionalResources\":{\"LogoUrl
\":\"https://awsmp-logos.s3.amazonaws.com/PLACEHOLDER_Logo_for_Containers_products.png
\",\"AdditionalResources\":[],\"Videos\":[]},\"SupportInformation\":{\"Description
\":\"Description of support information.\",\"Resources\":[]},\"RegionAvailability\":
{\"Regions\":[\"ap-south-1\",\"eu-west-3\",\"eu-north-1\",\"eu-west-2\",\"eu-west-1\",
\"ap-northeast-2\",\"ap-northeast-1\",\"me-south-1\",\"ca-central-1\",\"sa-east-1\",
\"ap-east-1\",\"ap-southeast-1\",\"ap-southeast-2\",\"eu-central-1\",\"us-east-1\",\"us-
east-2\",\"us-west-1\",\"us-west-2\"],\"FutureRegionSupport\":null},\"Repositories\":
[{\"Url\":\"709825985650.dkr.ecr.us-east-1.amazonaws.com/some-seller-prefix/my-repo-1\",
\"Type\":\"ECR\"}]}"
}
```


The `Details` attribute returned is a string that contains JSON. For more information about the `Details` attribute, see Working with the `Details` attribute (p. 4). For this example, here is a copy of the `Details` with newlines added for readability.


```
  "Details": "{
    \"Versions\": [
      {
        \"Id\": \"example2-0000-aaaa-5ef6-7890abcdef12\",
        \"ReleaseNotes\": \"My release notes\",
        \"UpgradeInstructions\": \"N/A\",
        \"VersionTitle\": \"1.0\",
        \"CreationDate\": \"2021-03-02T00:00:00.000Z\",
        \"Sources\": [
          {
            \"Type\": \"DockerImages\",
            \"Id\": \"example3-1111-bbbb-5ef6-7890abcdef12\",
            \"Images\": [
              \"709825985650.dkr.ecr.us-east-1.amazonaws.com/some-seller-prefix/my-
repo-1:some-tag\"
            ],
            \"Compatibility\": {
              \"Platform\": \"Linux\"
            }
          }
        ],
        \"DeliveryOptions\": [
          {
            \"Id\": \"example4-2222-cccc-2222-ccccccccccc\",
            \"Type\": \"ElasticContainerRegistry\",
            \"SourceId\": \"example3-1111-bbbb-5ef6-7890abcdef12\",
            \"Title\": \"New delivery option 1\",
            \"ShortDescription\": \"Delivery option 1\",
```

```
            \"isRecommended\": false,
            \"Compatibility\": {
              \"AWSServices\": [
                \"ECS\",
                \"EKS\"
              ]
            },
            \"Instructions\": {
              \"Usage\": \"test\"
            },
            \"Recommendations\": {
              \"AdditionalArtifacts\": []
            },
            \"Visibility\": \"Limited\"
          }
        ]
      }
    ],
    \"Description\": {
      \"Highlights\": [
        \"Some highlight\"
      ],
      \"LongDescription\": \"Description of my product\",
      \"ProductCode\": \"123456789012abcdef1234567\",
      \"Manufacturer\": null,
      \"Visibility\": \"Limited\",
      \"AssociatedProducts\": null,
      \"Sku\": null,
      \"SearchKeywords\": [
        \"some keyword\"
      ],
      \"ProductTitle\": \"Container Product 1\",
      \"ShortDescription\": \"Description of my product\",
      \"Categories\": [
        \"Operating Systems\"
      ]
    },
    \"PromotionalResources\": {
      \"LogoUrl\": \"https://awsmp-logos.s3.amazonaws.com/
PLACEHOLDER_Logo_for_Containers_products.png\",
      \"AdditionalResources\": [],
      \"Videos\": []
    },
    \"SupportInformation\": {
      \"Description\": \"Description of support information.\",
      \"Resources\": []
    },
    \"RegionAvailability\": {
      \"Regions\": [
        \"ap-south-1\",
        \"eu-west-3\",
        \"eu-north-1\",
        \"eu-west-2\",
        \"eu-west-1\",
        \"ap-northeast-2\",
        \"ap-northeast-1\",
        \"me-south-1\",
        \"ca-central-1\",
        \"sa-east-1\",
        \"ap-east-1\",
        \"ap-southeast-1\",
        \"ap-southeast-2\",
        \"eu-central-1\",
        \"us-east-1\",
        \"us-east-2\",
        \"us-west-1\",
```

```
        \"us-west-2\"
      ],
      \"FutureRegionSupport\": null
    },
    \"Repositories\": [
      {
        \"Url\": \"709825985650.dkr.ecr.us-east-1.amazonaws.com/some-seller-prefix/my-
repo-1\",
        \"Type\": \"ECR\"
      }
    ]
  }"
```

API access control

Before you can use the AWS Marketplace Catalog API, your account must have access to the functionality you want to call through the API.

You must create AWS Identity and Access Management (IAM) users, roles, and policies before you can use the AWS Marketplace Catalog API.

Use AWS Identity and Access Management to create IAM roles and assign policies that grant limited permissions to end users. The policies define the actions the role can take on your product entities through the AWS Marketplace Catalog API. For example, you can define roles such as engineering, marketing, and pricing. A user in your organization who has been added to the engineering role might be granted permissions to initiate a change request to publish a new version but cannot list all change sets.

> **Note**
> To sell products on AWS Marketplace, your AWS account must be set up as a seller account. For more details about becoming an AWS Marketplace seller, see Getting started as a seller in the *AWS Marketplace Seller Guide*.

Set up IAM permissions

You can use policies that are managed by AWS to grant IAM permissions to your users.

To manage a private marketplace, you can use the `AWSPrivateMarketplaceAdminFullAccess` IAM managed policy which has full access to create and edit the private marketplace for your account or AWS organization.

To work with products that you sell on AWS Marketplace, you can use the `AWSMarketplaceSellerFullAccess` IAM managed policy which has full access to the AWS Marketplace Catalog API in addition to its other permissions. You can grant read-only access for the Catalog API with the `AWSMarketplaceSellerProductsReadOnly` policy.

For more details about these policies, their permissions, and other IAM managed policies, sign into the IAM console at https://console.aws.amazon.com/iam/, choose **Policies**, and enter *marketplace* in the **Search** field.

You can also create your own policies or limit the scope of managed policies to a subset of functionality available in the AWS Marketplace Catalog API. The following is a list of the actions that you can use in your IAM policies for scoping permissions to the AWS Marketplace Catalog API:

- `aws-marketplace:ListChangeSets`
- `aws-marketplace:DescribeChangeSet`
- `aws-marketplace:StartChangeSet`
- `aws-marketplace:CancelChangeSet`

- `aws-marketplace:ListEntities`
- `aws-marketplace:DescribeEntity`

For more information about using policies in AWS Marketplace, see the following topics:

- For buyers who want to manage a private marketplace, see Creating a private marketplace IT administrator, and Controlling access to AWS Marketplace subscriptions.
- For sellers who want to manage the products they sell, see Controlling access to AWS Marketplace Management Portal.

Condition keys

The AWS Marketplace Catalog API also supports condition keys for the `StartChangeSet` action, allowing you to tune IAM policies for each change type. For example, if an IAM user has the policy attached to their user, then they can only perform `StartChangeSet` when the change type name is `ExampleChangeTypeName`.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "aws-marketplace:StartChangeSet",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "catalog:ChangeType": [ "ExampleChangeTypeName" ]
                }
            }
        }
    ]
}
```

Note
Condition keys are supported when used with the AWS Marketplace Catalog API only. Using condition keys will not allow the user to use the AWS Marketplace Management Portal to make changes to products. For users to use the AWS Marketplace Management Portal, create a policy without a condition key on the `StartChangeSet` action.

Specifying resources in policy

The AWS Marketplace Catalog API supports specifying resources for the `StartChangeSet` action, allowing you to tune IAM policies for specific products. For example, if an IAM user has the policy attached to their user, then they can only perform `StartChangeSet` for the specified products.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "aws-marketplace:StartChangeSet",
            "Resource": [
                "arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/AmiProduct/
example1-abcd-1234-5ef6",
                "arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/AmiProduct/
example2-abcd-1234-5ef6"
            ]
```

```
        }
    ]
}
```

Note
Resource-level permissions are supported when used with the AWS Marketplace Catalog API only. Using resource-level permissions will not allow the user to use the AWS Marketplace Management Portal to make changes to products. For users to use the AWS Marketplace Management Portal, create a policy without a resource-level permission on the `StartChangeSet` action.

Service limits

The AWS Marketplace Catalog API has the following limits.

Request limits

API operations	Request rate (per AWS account)
ListEntities	10 per second
DescribeEntity	10 per second
StartChangeSet	5 per second
ListChangeSets	5 per second
DescribeChangeSet	10 per second
CancelChangeSet	5 per second

Account limits

Limit	Description
Maximum number of open `StartChangeSet` requests per account	250

Request history retention limits

Description	Limit
Retention period for change requests. This applies after the end time of each change request.	60 days

Working with seller products

You can use the AWS Marketplace Catalog API to manage products that you sell in AWS Marketplace.

The following topics describe how to use the Catalog API to perform actions on your single-AMI and container-based products.

> **Note**
> This chapter assumes that you have access to the API and have completed any seller prerequisites, as described in the API access control (p. 9) topic.
> To understand the basics of using the AWS Marketplace Catalog API, see AWS Marketplace Catalog API (p. 1).

Topics

Finding your product ID (server products)

You must get the product ID for your product before you can modify it with AWS Marketplace Catalog API. There are two ways to find the product ID for server products:

- Open the AWS Marketplace Management Portal and sign in with your seller account. From the **Products** menu, select **Server products**, then choose the product you are interested in. The product ID is listed in the **Product Summary** section.
- Use the `ListEntities` action with the `EntityType` **`AmiProduct`** or **`ContainerProduct`** to get a list of products, including their product IDs, via the Catalog API. `ListEntities` requires that you do not include the version of the entity type (for example, `AmiProduct@1.0`).

> **Note**
> The product ID is only available after your product has been published and is visible to at least yourself in AWS Marketplace. When you first create your product, it can take several days to be reviewed and fully created. During this time, it will not have a product ID available.

Change set status and errors

Making changes to seller products in the AWS Marketplace Catalog API involves creating change sets that describe the changes you want to make, and then using the `StartChangeSet` action to start the changes. The changes from the request can take minutes to hours or longer to complete, depending on the request. The response to this request looks like the following.

```
{
    "ChangeSetId": "example123456789012abcdef",
    "ChangeSetArn": "arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/
ChangeSet/example123456789012abcdef"
}
```

The change request is added to a queue and processed, including scanning the files and information to ensure that it meets the AWS Marketplace guidelines for products. Depending on the change requests, this process can take a few minutes to days. You can check the status of the request through the AWS

Marketplace Management Portal, or in the Catalog API with the `DescribeChangeSet` action. For more information about change sets, see Working with change sets (p. 3).

To check the status of your request, use the `DescribeChangeSet` action.

```
POST /DescribeChangeSet HTTP/1.1
Content-type: application/json

{
   "Catalog": "AWSMarketplace",
   "ChangeSetID": "example123456789012abcdef"
}
```

The result of this call looks like the following (in this case, for adding a new version to a container product).

```
{
    "ChangeSetId": "example123456789012abcdef",
    "ChangeSetArn": "arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/
ChangeSet/example123456789012abcdef",
    "ChangeSetName": "Submitted by 123456789012",
    "StartTime": "2020-10-27T22:21:26Z",
    "EndTime": "2020-10-27T22:32:19Z",
    "Status": "SUCCEEDED",
    "ChangeSet": [
        {
            "ChangeType": "AddDeliveryOptions",
            "Entity": {
                "Type": "ContainerProduct@1.0",
                "Identifier": "example-1234-abcd-56ef-abcdef12345678@4"
            },
            "ErrorDetailList": []
        }
    ]
}
```

The `Status` field shows the current status of the request, in this case, `SUCCEEDED`.

If there are failures, the result can include two types of errors. For most errors, the error message is included directly. However, errors found while scanning the product for security vulnerabilities instead include a URL to a file that lists all of the errors found, in the `ErrorMessage` field. Errors found while scanning have the `ErrorCode` "`SCAN_ERROR`".

```
{
    "ChangeSetId": "example123456789012abcdef",
    "ChangeSetArn": "arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/
ChangeSet/example123456789012abcdef",
    "ChangeSetName": "Submitted by 123456789012",
    "StartTime": "2020-10-27T22:21:26Z",
    "EndTime": "2020-10-27T22:32:19Z",
    "Status": "FAILED",
    "FailureDescription": "Change set preparation has failed. For details see
 'ErrorDetailList'.",
    "ChangeSet": [
        {
            "ChangeType": "AddDeliveryOptions",
            "Entity": {
                "Type": "ContainerProduct@1.0",
                "Identifier": "example-1234-abcd-56ef-abcdef12345678@4"
            },
            "ErrorDetailList": [
                {
```

```
                "ErrorCode": "DUPLICATE_VERSION_TITLE",
                "ErrorMessage": "The version title must be different from any other version
 titles of this product."
                },
                {
                "ErrorCode": "SCAN_ERROR",
                "ErrorMessage": "https://123sample456.cloudfront.net/example-1234-
abcd-5678-abcdef12345678/1234abcdef567890"
                }
            ]
        }
    ]
}
```

In this example, there is one error directly reported (`DUPLICATE_VERSION_TITLE`). The other error has a file with error messages (a single `SCAN_ERROR` can have multiple found errors in the file that is linked).

Note
The link returned in the `ErrorMessage` is valid for 60 days.

Working with single-AMI products

You can use the AWS Marketplace Catalog API to automate tasks for working with single Amazon Machine Image (AMI-based) products.

With the Catalog API, you can automate updating your existing AMI-based products. You can perform the following actions through the API:

- Update product details
- Add a new version
- Update version information
- Restrict a version

As a prerequisite for updating AMI-based products, you must have one or more existing AMI-based products, and you should be familiar with working with the AWS Marketplace Catalog API (p. 1).

Note
For details about creating an AMI product through the AWS Marketplace Management Portal, see Single-AMI products in the *AWS Marketplace Seller Guide*.
For a walk-through showing how to automate updating your AMI-based product, you can also refer to the video, Automating updates to your product listings in AWS Marketplace with Catalog API (5:08).

Updating product details for an AMI-based product

If you already have an AMI-based product in AWS Marketplace, you can use the Catalog API to modify the product details.

You update product details by calling `StartChangeSet` with the `UpdateInformation` change type and the details that you want to change, as shown in the following example.

```
POST /StartChangeSet HTTP/1.1
Content-type: application/json

{
"Catalog": "AWSMarketplace",
"ChangeSet": [
  {
```

```
    "ChangeType":"UpdateInformation",
    "Entity":{
      "Identifier":"example1-abcd-1234-5ef6-7890abcdef12",
      "Type":"AmiProduct@1.0"
     },
     "Details": "{
        \"ProductTitle\": \"My Product Title\",
        \"ShortDescription\": \"My product short description.\",
        \"LongDescription\": \"My product longer description.\",
        \"Sku\": \"123example456\",
        \"SupportDescription\": \"Need help? Contact our experts at support@example.com \n
\nYour purchase includes 24x7 support.\",
        \"Categories\": [
          \"Operating Systems\",
          \"Network Infrastructure\",
          \"Application Development\"
        ]
      }"
    }
  ]
}
```

Note
This example has line wraps added for readability and will not work as-is. The `Details` attribute is a string, and it should be converted from a JSON object to a string in your call to `StartChangeSet`. In this case, the string is `"{ \"ProductTitle\": \"My Product Title\", \"ShortDescription\": \"My product short description.\", \"LongDescription\": \"My product longer description.\", \"Sku\": \"123example456\", \"SupportDescription\": \"Need help? Contact our experts at support@example.com \n\nYour purchase includes 24x7 support.\", \"Categories\": [\"Operating Systems\", \"Network Infrastructure\", \"Application Development\"] }"`. For more information, see Working with the `Details` attribute (p. 4).

The following is information about the input fields you provide for adding the `UpdateInformation` change type:

- **Entity (object)** – Your AMI-based product. The `Identifier` is your product ID, and the type is always `AmiProduct@1.0`. For information about using the Identifier, see Identifier (p. 2).
- **Details (string)** – Details of the request. It includes all the information you want to update for your product. Each field is optional, but you must include at least one change to update. This field is a string field. The string is JSON, but it must be formatted properly for a single-line string field, including escaping characters (such as quotation marks) that cannot directly be in a string. For more information, see Working with the `Details` attribute (p. 4).
  - **ProductTitle (string)** – Name of the product to be displayed to buyers.
  - **ShortDescription (string)** – Description of key aspects of the product to be displayed to buyers. Typically 2–3 sentences.
  - **LongDescription (string)** – Longer description of your product to be displayed to buyers. Typically 1–3 paragraphs.
  - **Sku (string)** – A free-form string for you to define as a reference for your own use.
  - **LogoUrl (string)** – A URL to an image in a publicly accessible Amazon Simple Storage Service (Amazon S3) bucket. For more details about image formats, see Company and product logo requirements in the *AWS Marketplace Seller Guide*.
  - **VideoUrls (array of strings)** – A list of URLs to publicly available, externally hosted videos to be provided as a reference to buyers in your product information.
  - **Highlights (array of strings)** – A list of short callouts for key product features.
  - **AdditionalResources (array of structures)** – List of references to additional resources to learn about your product. Each reference is made up of a text name and a URL:

  - **Text (string)** – The name or title of the resource.
  - **Url (string)** – A URL to a resource that is helpful for a buyer to understand your product.
- **SupportDescription (string)** – Details about your support offerings for your product.
- **Categories (array of strings)** – A list of AWS Marketplace defined product categories that describe your product. For a list, see Product categories in the *AWS Marketplace Seller Guide*.
- **SearchKeywords (array of strings)** – A list of additional keywords for your product for the search experience. Seller name, product name, and product categories are automatically included in search keywords and do not need to be repeated here.

A change set is created for your request. The response to this request gives you the ID for the change set and looks like the following.

```
{
    "ChangeSetId": "example123456789012abcdef",
    "ChangeSetArn": "arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/
ChangeSet/example123456789012abcdef"
}
```

The change request is added to a queue and processed, including scanning the files and information to ensure that it meets the AWS Marketplace guidelines for products. This process can take a few minutes to hours. You can check the status of the request through the AWS Marketplace Management Portal, or in the Catalog API with the `DescribeChangeSet` action. For more information about change sets, see Working with change sets (p. 3). For more information about errors in seller product change sets, see Change set status and errors (p. 12).

Adding a new version to an AMI-based product

You can use the Catalog API to add a new version to an existing AMI-based product in AWS Marketplace. For more information about adding new AMI versions to your product using the AWS Marketplace Management Portal, see Adding a new version in the *AWS Marketplace Seller Guide*.

You add a new version in the Catalog API by calling `StartChangeSet` with the `AddDeliveryOptions` change type for single-AMI products, as shown in the following example.

> **Note**
> For single-AMI products, a version is made up of a single delivery option, which is the AMI that you are making available. In the Catalog API, working with delivery options for single-AMI products modifies versions of your product.

```
POST /StartChangeSet HTTP/1.1
Content-type: application/json

{
"Catalog": "AWSMarketplace",
"ChangeSet": [
  {
    "ChangeType":"AddDeliveryOptions",
    "Entity":{
      "Identifier":"example1-abcd-1234-5ef6-7890abcdef12",
      "Type":"AmiProduct@1.0"
     },
      "Details": "{
        \"Version\": {
          \"VersionTitle\": \"*My new title*\",
          \"ReleaseNotes\": \"*My new Release notes*\"
        },
        \"DeliveryOptions\": [
```

```
          {
            \"Details\": {
              \"AmiDeliveryOptionDetails\": {
                \"AmiSource\": {
                  \"AmiId\": \"ami-1234567890abcdef\",
                  \"AccessRoleArn\": \"arn:aws:iam::12345678901:role/
AwsMarketplaceAmiIngestion\",
                  \"UserName\": \"ec2-user\",
                  \"OperatingSystemName\": \"AMAZONLINUX\",
                  \"OperatingSystemVersion\": \"Amazon Linux 2 AMI 2.0.20210126.0 x86_64
 HVM gp2\"
                },
                \"UsageInstructions\": \"Easy to use AMI\",
                \"RecommendedInstanceType\": \"m4.xlarge\",
                \"SecurityGroups\": [
                  {
                    \"IpProtocol\": \"tcp\",
                    \"FromPort\": 443,
                    \"ToPort\": 443,
                    \"IpRanges\": [
                      \"0.0.0.0/0\"
                    ]
                  }
                ]
              }
            }
          }
        ]
      }"
    }
  ]
}
```

Note
This example has line wraps added for readability and will not work as-is. The `Details` attribute is a string, and it should be converted from a JSON object to a string in your call to `StartChangeSet`. In this case, the string is `"{ \"Version\": { \"VersionTitle\": \"*My new title*\", \"ReleaseNotes\": \"*My new Release notes*\" }, \"DeliveryOptions\": [{\"Details\": { \"AmiDeliveryOptionDetails\": { \"AmiSource\": { \"AmiId\": \"ami-1234567890abcdef\", \"AccessRoleArn \": \"arn:aws:iam::12345678901:role/AwsMarketplaceAmiIngestion\", \"UserName\": \"ec2-user\", \"OperatingSystemName\": \"AMAZONLINUX \", \"OperatingSystemVersion\": \"Amazon Linux 2 AMI 2.0.20210126.0 x86_64 HVM gp2\" }, \"UsageInstructions\": \"Easy to use AMI\", \"RecommendedInstanceType\": \"m4.xlarge\", \"SecurityGroups\": [{ \"IpProtocol\": \"tcp\", \"FromPort\": 443, \"ToPort\": 443, \"IpRanges\": [\"0.0.0.0/0\"] }] } } }] }"`. For more information, see Working with the `Details` attribute (p. 4).

The following is information about the input fields you provide for adding the `AddDeliveryOptions` change type. For more information about these fields, see Adding a new version in the AWS Marketplace Seller Guide.

- **Entity (object)** – Your AMI-based product. The `Identifier` is your product ID, and the type is always `AmiProduct@1.0`. For information about using the Identifier, see Identifier (p. 2).
- **Details (string)** – Details of the request. It includes all the information about the new version of your AMI-based product. This field is a string field. The string is JSON, but it must be formatted properly for a single-line string field, including escaping characters (such as quotation marks) that cannot directly be in a string. For more information, see Working with the `Details` attribute (p. 4).
  - **Version (object)** – Details about the software version you are adding. Made up of a title and release notes.

- **VersionTitle (string)** – Unique name of the version. Displayed to end users in product details page and configuration pages for the product in AWS Marketplace.
- **ReleaseNotes (string)** – Notes for buyers to tell them about changes from one version to the next.

- **DeliveryOptions (array)** – List of `DeliveryOption` objects, including the details of each:
  - **Details (object)** – Holds the details of an AMI delivery option. Note that this nested details object does *not* need to be double-escaped.
    - **AmiDeliveryOptionDetails (object)** – The details of one AMI delivery option.
      - **AmiSource (object)** – Details about the AMI to be used for the added version.
        - **AmiId (string)** – ID for the source AMI, located in the AWS Region where the API is being called (currently must always be US East (N. Virginia) because that is the only Region where the Catalog API is available). Must belong to the caller account.
        - **AccessRoleArn (string)** – IAM role Amazon Resource Name (ARN) used by AWS Marketplace to access the provided AMI. For details about creating and using this ARN, see Giving AWS Marketplace access to your AMI in the *AWS Marketplace Seller Guide*.
        - **UserName (string)** – Login user name to access the operating system (OS) in the AMI. Typically `ec2-user` for Linux AMIs or `Administrator` for Windows.
        - **ScanningPort (integer)** – SSH or RDP port used to access the OS. Used for scanning the provided AMI for security vulnerabilities. Defaults to 22.
        - **OperatingSystemName (string)** – Name of the operating system displayed to buyers.
        - **OperatingSystemVersion (string)** – Operating system version string displayed to buyers.
      - **UsageInstructions (string)** – Instructions for using the AMI, or a link to more information about the AMI.
      - **AccessEndpointUrl (object)** – Used to create a path to access the AMI after it is used.
        - **Port (string)** – The port number used to access the service running on the AMI.
        - **Protocol (string)** – The protocol (`http` or `https`) used to access the service running on the AMI.
        - **RelativePath (string)** – The path from the web root to access the service running on the AMI (for example *`/index.html`*).
      - **RecommendedInstanceType (string)** – The instance type that is recommended to run the service with the AMI and is the default for 1-click installs of your service. For a list of instance types, see Instance types in the *Amazon Elastic Compute Cloud User Guide for Linux Instances*.
      - **SecurityGroups (array of objects)** – A list of objects representing ingress rules for the automatically created groups for the version.
        - **FromPort (integer)** – The source port.
        - **IpProtocol (string)** – The protocol to use (`tcp` or `udp`).
        - **IpRanges (array of strings)** – IP ranges to allow, in CIDR format (in the form *xxx.xxx.xxx.xxx/nn*, for example, *`192.0.2.0/24`*).
        - **ToPort (integer)** – The destination port.

A change set is created for your request. The response to this request gives you the ID for the change set and looks like the following.

```
{
    "ChangeSetId": "example123456789012abcdef",
    "ChangeSetArn": "arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/
ChangeSet/example123456789012abcdef"
}
```

The change request is added to a queue and processed, including scanning the files and information to ensure that it meets the AWS Marketplace guidelines for AMI products. This process can take a few minutes to hours. You can check the status of the request through the AWS Marketplace Management

Portal, or in the Catalog API with the `DescribeChangeSet` action. For more information about change sets, see Working with change sets (p. 3). For more information about errors in seller product change sets, see Change set status and errors (p. 12).

When the request is complete, the version is added, and any existing subscribers will receive an email message telling them about the new version. For more information about the process of adding a new version, see Adding a new version in the *AWS Marketplace Seller Guide*.

Updating version information for an AMI-based product

You can use the Catalog API to update the details of an existing version of your AMI-based product in AWS Marketplace. For more information about updating version information using the AWS Marketplace Management Portal, see Updating version information in the *AWS Marketplace Seller Guide*. You cannot update the AMI for the version. If you need to update the AMI, create a new version instead.

You add a new version in the Catalog API by calling `StartChangeSet` with the `UpdateDeliveryOptions` change type, as shown in the following example.

```
POST /StartChangeSet HTTP/1.1
Content-type: application/json

{
"Catalog": "AWSMarketplace",
"ChangeSet": [
  {
    "ChangeType":"UpdateDeliveryOptions",
    "Entity":{
      "Identifier":"example1-abcd-1234-5ef6-7890abcdef12",
      "Type":"AmiProduct@1.0"
     },
      "Details": "{
        \"Version\": {
          \"ReleaseNotes\": \"*My new Release notes*\"
        },
        \"DeliveryOptions\": [
          {
            \"Id\": \"example1-2222-cccc-2222-ccccccccccc\",
            \"Details\": {
              \"AmiDeliveryOptionDetails\": {
                \"UsageInstructions\": \"Easy to use AMI\"
              }
            }
          }
        ]
      }"
    }
  ]
}
```

Note
This example has line wraps added for readability and will not work as-is. The `Details` attribute is a string, and it should be converted from a JSON object to a string in your call to `StartChangeSet`. In this case, the string is `"{ \"Version\": { \"ReleaseNotes\": \"*My new Release notes*\" }, \"DeliveryOptions \": [ { \"Id\": \"example1-2222-cccc-2222-ccccccccccc\", \"Details\": { \"AmiDeliveryOptionDetails\": { \"UsageInstructions\": \"Easy to use AMI\" } } } ] }"`. For more information, see Working with the `Details` attribute (p. 4).

The following is information about the input fields you provide for adding the `UpdateDeliveryOptions` change type. For more information about these fields, see Updating version information in the AWS Marketplace Seller Guide.

- **Entity (object)** – Your AMI-based product. The `Identifier` is your product ID, and the type is always `AmiProduct@1.0`. For information about using the Identifier, see Identifier (p. 2).
- **Details (string)** – Details of the request. It includes any information about the version of your AMI-based product that you would like to update. This field is a string field. The string is JSON, but it must be formatted properly for a single-line string field, including escaping characters (such as quotation marks) that cannot directly be in a string. For more information, see Working with the `Details` attribute (p. 4). The included fields are all optional, but you must include at least one field to update.
  - **Version (object)** – Details about the software version.
    - **ReleaseNotes (string)** – Notes for buyers to tell them about changes from one version to the next.
  - **DeliveryOptions (array of objects)** – List of `DeliveryOption` objects, including the details of each:
    - **Id (string)** – Unique identifier for the `DeliveryOption` (you can get the unique identifier for the `DeliveryOption` by calling the `DescribeEntity` action on the product you are updating).
    - **Details (object)** – Holds the details of an AMI delivery option. Note that this nested details object does *not* need to be double-escaped.
      - **AmiDeliveryOptionDetails (object)** – The details of one AMI delivery option.
        - **UsageInstructions (string)** – Instructions for using the AMI, or a link to more information about the AMI.
        - **AccessEndpointUrl (object)** – Used to create a path to access the AMI after it is used.
          - **Port (string)** – The port number used to access the service running on the AMI.
          - **Protocol (string)** – The protocol (`http` or `https`) used to access the service running on the AMI.
          - **RelativePath (string)** – The path from the web root to access the service running on the AMI (for example *`/index.html`*).
        - **RecommendedInstanceType (string)** – The instance type that is recommended to run the service with the AMI and is the default for 1-click installs of your service.
        - **SecurityGroups (array of objects)** – A list of objects representing ingress rules for the automatically created groups for the version:
          - **FromPort (integer)** – The source port.
          - **IpProtocol (string)** – The protocol to use (`tcp` or `idp`).
          - **IpRanges (array of strings)** – IP ranges to allow, in CIDR format (in the form *xxx.xxx.xxx.xxx/nn*, for example, *`192.0.2.0/24`*).
          - **ToPort (integer)** – The destination port.

A change set is created for your request. The response to this request gives you the ID for the change set and looks like the following.

```
{
    "ChangeSetId": "example123456789012abcdef",
    "ChangeSetArn": "arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/
ChangeSet/example123456789012abcdef"
}
```

The change request is added to a queue and processed, including scanning the information to ensure that it meets the AWS Marketplace guidelines for AMI products. This process can take a few minutes to hours. You can check the status of the request through the AWS Marketplace Management Portal, or in the Catalog API with the `DescribeChangeSet` action. For more information about change sets, see Working with change sets (p. 3). For more information about errors in seller product change sets, see Change set status and errors (p. 12).

Restricting a version of an AMI-based product

You can use the Catalog API to restrict a version of your AMI-based product in AWS Marketplace. This prevents new buyers from being able to use that version. There must always be at least one unrestricted version of a product available, so you cannot restrict the last publicly available version for a product. For more information about restricting AMI versions in AWS Marketplace via the AWS Marketplace Management Portal, see Restricting a version in the *AWS Marketplace Seller Guide*.

You restrict a version in the Catalog API by calling `StartChangeSet` with the `RestrictDeliveryOptions` change type, as shown in the following example.

> **Note**
> Restricted versions are not shown to new buyers, but they are still available for existing customers.

```
POST /StartChangeSet HTTP/1.1
Content-type: application/json

{
"Catalog": "AWSMarketplace",
"ChangeSet": [
  {
    "ChangeType":"RestrictDeliveryOptions",
    "Entity":{
      "Identifier":"example1-abcd-1234-5ef6-7890abcdef12",
      "Type":"AmiProduct@1.0"
     },
      "Details": "{ \"DeliveryOptionIds\": [ \"example1-2222-cccc-2222-ccccccccccc\" ] }"
    }
  ]
}
```

The following is information about the input fields you provide for adding the `RestrictDeliveryOptions` change type:

- **Entity (object)** – Your AMI-based product. The `Identifier` is your product ID, and the type is always `AmiProduct@1.0`. For information about using the Identifier, see Identifier (p. 2).
- **Details (string)** – Details of the request. It includes IDs for the versions of your AMI-based product that you would like to restrict. This field is a string field. The string is JSON, but it must be formatted properly for a single-line string field, including escaping characters (such as quotation marks) that cannot directly be in a string. For more information, see Working with the `Details` attribute (p. 4).
  - **DeliveryOptionIds (array of objects)** – List of `DeliveryOption` IDs for the versions that you want to restrict. You can get the unique identifier for the `DeliveryOption` by calling the `DescribeEntity` action on the version you are restricting.

A change set is created for your request. The response to this request gives you the ID for the change set and looks like the following.

```
{
    "ChangeSetId": "example123456789012abcdef",
    "ChangeSetArn": "arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/
ChangeSet/example123456789012abcdef"
}
```

The change request is added to a queue and processed. This process can take a few minutes to hours. You can check the status of the request through the AWS Marketplace Management Portal, or in the Catalog API with the `DescribeChangeSet` action. For more information about change sets, see

Working with change sets (p. 3). For more information about errors in seller product change sets, see Change set status and errors (p. 12).

Errors in the AMI products API

The following errors are specific to the AMI product actions in the AWS Marketplace Catalog API. These errors are returned when you call `DescribeChangeSet` after a change set is processing. For more details about using `DescribeChangeSet` to get the status of a change request, see Working with change sets (p. 3).

Change type	Error code	Error message
AddDeliveryOptions	INVALID_PRODUCT	Use an existing limited or public product.
AddDeliveryOptions	DUPLICATE_VERSION_TITLE	The version title must be different from any other version titles of this product.
AddDeliveryOptions	INVALID_VERSION_TITLE	Remove spaces before the trademark symbol.
AddDeliveryOptions	INVALID_VERSION_TITLE	Remove unsupported characters: [x, y, z]
AddDeliveryOptions	INVALID_VERSION_TITLE	Remove spaces from the beginning of the version title.
AddDeliveryOptions	INVALID_VERSION_TITLE	Provide version title with fewer than [x] characters.
AddDeliveryOptions	INVALID_RELEASE_NOTES	Remove spaces before the trademark symbol.
AddDeliveryOptions	INVALID_RELEASE_NOTES	Remove unsupported characters: [x, y, z]
AddDeliveryOptions	INVALID_RELEASE_NOTES	Remove spaces from the beginning of release notes.
AddDeliveryOptions	INVALID_RELEASE_NOTES	Provide release notes with fewer than (x) characters.
AddDeliveryOptions	INVALID_USAGE_INSTRUCTIONS	Remove spaces before the trademark symbol.
AddDeliveryOptions	INVALID_USAGE_INSTRUCTIONS	Remove unsupported characters: [x, y, z]
AddDeliveryOptions	INVALID_USAGE_INSTRUCTIONS	Remove spaces from the beginning of release notes.
AddDeliveryOptions	INVALID_USAGE_INSTRUCTIONS	Provide usage instructions with fewer than (x) characters.
AddDeliveryOptions	RECOMMENDED_INSTANCE_TYPE_NOT_AVAILABLE	Provide an existing, available instance type.
AddDeliveryOptions	INVALID_RECOMMENDED_INSTANCE_TYPE	Provide a valid instance type.

Change type	Error code	Error message
AddDeliveryOptions	INVALID_SECURITY_GROUP	Security group ports must be between 1 and [max].
AddDeliveryOptions	INVALID_SECURITY_GROUP	Provide a value for CIDR IP ranges.
AddDeliveryOptions	INVALID_SECURITY_GROUP	Provide security group start port that is not greater than end port.
AddDeliveryOptions	INVALID_SECURITY_GROUP_PROTOCOL	Security group protocol must either be 'tcp' or 'udp'.
AddDeliveryOptions	INVALID_CIDR_IP	Provide standard CIDR IP range in form '0.0.0.0/0'.
AddDeliveryOptions	INVALID_ACCESS_ENDPOINT_PORT	Provide endpoint port less than [x].
AddDeliveryOptions	INVALID_ACCESS_ENDPOINT_PORT	Provide endpoint port between 1 and [max].
AddDeliveryOptions	INVALID_ACCESS_ENDPOINT_PORT	Provide endpoint port.
AddDeliveryOptions	INVALID_ACCESS_ENDPOINT_RELATIVE_PATH	Remove spaces in the relative path.
AddDeliveryOptions	INVALID_ACCESS_ENDPOINT_RELATIVE_PATH	Remove preceding '/' from relative path.
AddDeliveryOptions	INCOMPATIBLE_OPERATING_SYSTEM	Provide operating system name and version that is compatible with instance types: [x]
AddDeliveryOptions	INCOMPATIBLE_OPERATING_SYSTEM_NAME	Provide name with fewer than (x) characters.
AddDeliveryOptions	INCOMPATIBLE_OPERATING_SYSTEM_NAME	Provide operating system name that is supported.
AddDeliveryOptions	INCOMPATIBLE_OPERATING_SYSTEM_VERSION	Provide version with fewer than (x) characters.
AddDeliveryOptions	INVALID_SCANNING_PORT	Provide scanning port between 1 and [max].
AddDeliveryOptions	INVALID_AMI_ID	Provide valid AMI ID.
AddDeliveryOptions	EXISTING_AMI_PRODUCT_CODE	Remove product code attached to image X.
AddDeliveryOptions	INVALID_AMI_ARCHITECTURE	Provide new AMI with architecture [x].
AddDeliveryOptions	INVALID_AMI_VIRTUALIZATION_TYPE	Provide new AMI with virtualization type [x].
AddDeliveryOptions	INVALID_AMI_VIRTUALIZATION_TYPE	Provide expected [z] volume on image [x].

Change type	Error code	Error message
AddDeliveryOptions	INCOMPATIBLE_AMI	Provide new AMI as architecture [x] on [y] is not supported by following instance types: [z]
AddDeliveryOptions	INCOMPATIBLE_AMI	Provide new AMI as virtualization type [x] on [y] is not supported by following instance types: [z]
AddDeliveryOptions	INCOMPATIBLE_AMI	Enable ENA support for image x because following instance types require ENA support: [y]
AddDeliveryOptions	ASSET_NOT_FOUND	Check if [ami-id] exists in us-east-1 Region of [account-id] AWS account and the `AccessARN` provided [ARN] has permissions to share this AMI with AWS Marketplace.
AddDeliveryOptions	ASSET_ACCESS_EXCEPTION	Unable to copy AMI [x] into AWS Marketplace account.
AddDeliveryOptions	SCAN_ERROR	Fix security vulnerability [y] on Image [x].
RestrictDeliveryOptions	INVALID_PRODUCT	Use an existing public product.
RestrictDeliveryOptions	MISSING_DELIVERY_OPTION_IDS	Provide at least one delivery option ID.
RestrictDeliveryOptions	INVALID_DELIVERY_OPTION_IDS	Provide delivery option IDs that can be found in the product. IDs not found: [x]
RestrictDeliveryOptions	INVALID_DELIVERY_OPTION	Provide delivery option IDs that are in a public state. IDs not in public state: [x]
RestrictDeliveryOptions	ALL_DELIVERY_OPTIONS_RESTRICTED	Provide fewer delivery options to restrict as at least one must remain in public state.
UpdateDeliveryOptions	INVALID_PRODUCT	Use an existing limited or public product.
UpdateDeliveryOptions	MISSING_DELIVERY_OPTION_IDS	Provide at least one delivery option ID.
UpdateDeliveryOptions	INVALID_DELIVERY_OPTION_IDS	Provide delivery option IDs that can be found in the product. IDs not found: [x]
UpdateDeliveryOptions	INVALID_DELIVERY_OPTIONS	Provide delivery option IDs that belong to the same version.

Change type	Error code	Error message
UpdateInformation	MISSING_UPDATES	Nothing to update. Provide updated information for product.
UpdateInformation	MISSING_LOGO_URL	Provide URL for logo stored in S3.
UpdateInformation	LOGO_COPY_FAILURE	There was an issue copying the logo from S3. Provide a new URL for logo stored in S3.
UpdateInformation	INVALID_CATEGORY_NAMES	Provide valid category names supported by AWS Marketplace.
UpdateInformation	INVALID_PRODUCT_TITLE	Remove spaces before trademark symbol.
UpdateInformation	INVALID_PRODUCT_TITLE	Remove unsupported characters [x, y, z].
UpdateInformation	INVALID_SHORT_DESCRIPTION	Remove spaces before trademark symbol.
UpdateInformation	INVALID_SHORT_DESCRIPTION	Remove unsupported characters [x, y, z].
UpdateInformation	INVALID_LONG_DESCRIPTION	Remove spaces before trademark symbol.
UpdateInformation	INVALID_LONG_DESCRIPTION	Remove unsupported characters [x, y, z].
UpdateInformation	INVALID_SUPPORT_DESCRIPTION	Remove spaces before trademark symbol.
UpdateInformation	INVALID_SUPPORT_DESCRIPTION	Remove unsupported characters [x, y, z].
UpdateInformation	INVALID_HIGHLIGHTS	Remove spaces before trademark symbol.
UpdateInformation	INVALID_HIGHLIGHTS	Remove unsupported characters [x, y, z].
UpdateInformation	INVALID_ADDITIONAL_RESOURCES	Remove spaces before trademark symbol.
UpdateInformation	INVALID_ADDITIONAL_RESOURCES	Remove unsupported characters [x, y, z].
UpdateInformation	INVALID_PRODUCT_CATEGORIES	Provide between 1 and 3 product categories.
UpdateInformation	INVALID_IMAGE_PROPERTIES	Provide an image that meets the product logo requirements. (link)

Working with container-based products

You can use the AWS Marketplace Catalog API to automate tasks for working with container-based products.

With the Catalog API, you can automate updating your existing container-based products. You can perform the following actions through the API:

- Update product information
- Add a new version
- Add repositories for your container images and other resources
- Update a version
- Restrict a version

As a prerequisite for updating container-based products, you must have one or more existing container-based products, and you should be familiar with working with the AWS Marketplace Catalog API (p. 1).

> **Note**
> For details about creating a container-based product through the AWS Marketplace Management Portal, see Getting started with container products in the *AWS Marketplace Seller Guide*.

Updating product information for a container-based product

If you already have a container-based product in AWS Marketplace, you can use the Catalog API to modify the product information.

You update product information by calling `StartChangeSet` with the `UpdateInformation` change type and the details that you want to change, as shown in the following example.

```
POST /StartChangeSet HTTP/1.1
Content-type: application/json

{
"Catalog": "AWSMarketplace",
"ChangeSet": [
  {
    "ChangeType":"UpdateInformation",
    "Entity":{
      "Identifier":"example1-abcd-1234-5ef6-7890abcdef12",
      "Type":"ContainerProduct@1.0"
     },
     "Details": "{
        \"ProductTitle\": \"My Product Title\",
        \"ShortDescription\": \"My product short description.\",
        \"LongDescription\": \"My product longer description.\",
        \"Sku\": \"123example456\",
        \"SupportDescription\": \"Need help? Contact our experts at support@example.com \n
\nYour purchase includes 24x7 support.\",
        \"Categories\": [
          \"Operating Systems\",
          \"Network Infrastructure\",
          \"Application Development\"
        ]
      }"
    }
```

```
  ]
}
```

Note
This example has line wraps added for readability and will not work as-is. The `Details` attribute is a string, and it should be converted from a JSON object to a string in your call to `StartChangeSet`. In this case, the string is `"{ \"ProductTitle\": \"My Product Title\", \"ShortDescription\": \"My product short description.\", \"LongDescription\": \"My product longer description.\", \"Sku\": \"123example456\", \"SupportDescription\": \"Need help? Contact our experts at support@example.com \n\nYour purchase includes 24x7 support. \", \"Categories\": [\"Operating Systems\", \"Network Infrastructure \", \"Application Development\"] }"`. For more information, see Working with the `Details` attribute (p. 4).

The following is information about the input fields you provide for adding the `UpdateInformation` change type:

- **Entity (object)** – The container-based product that you want to update. The `Identifier` is your product ID, and the type is always `ContainerProduct@1.0`. For information about using the Identifier, see Identifier (p. 2).
- **Details (string)** – Details of the request. It includes all the information you want to update for your product. Each field is optional, but you must include at least one change to update. This field is a string field. The string is JSON, but it must be formatted properly for a single-line string field, including escaping characters (such as quotation marks) that cannot directly be in a string. For more information, see Working with the `Details` attribute (p. 4).
  - **ProductTitle (string)** – Name of the product to be displayed to buyers.
  - **ShortDescription (string)** – Description of key aspects of the product to be displayed to buyers. Typically 2–3 sentences.
  - **LongDescription (string)** – Longer description of your product to be displayed to buyers. Typically 1–3 paragraphs.
  - **Sku (string)** – A free-form string for you to define as a reference for your own use.
  - **LogoUrl (string)** – A URL to an image in a publicly accessible Amazon S3 bucket. For more details about image formats, see Company and product logo requirements in the *AWS Marketplace Seller Guide*.
  - **VideoUrls (array of strings)** – A list of URLs to publicly available, externally hosted videos to be provided as a reference to buyers in your product information.
  - **Highlights (array of strings)** – A list of short callouts for key product features.
  - **AdditionalResources (array of objects** – List of references to additional resources to learn about your product. Each reference is made up of a text name and a URL:
    - **Text (string)** – The name or title of the resource.
    - **Url (string)** – A URL to a resource that is helpful for a buyer to understand your product.
  - **SupportDescription (string)** – Details about your support offerings for your product.
  - **Categories (array of strings)** – A list of AWS Marketplace defined product categories that describe your product.
  - **SearchKeywords (array of strings)** – A list of additional keywords for your product for the search experience. Seller name, product name, and product categories are automatically included in search keywords and do not need to be repeated here.

A change set is created for your request. The response to this request gives you the ID for the change set and looks like the following.

```
{
```

```
    "ChangeSetId": "example123456789012abcdef",
    "ChangeSetArn": "arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/
ChangeSet/example123456789012abcdef"
}
```

The change request is added to a queue and processed, including scanning the files and information to ensure that it meets the AWS Marketplace guidelines for products. This process can take a few minutes to hours. You can check the status of the request through the AWS Marketplace Management Portal, or in the Catalog API with the `DescribeChangeSet` action. For more information about change sets, see Working with change sets (p. 3). For more information about errors in seller product change sets, see Change set status and errors (p. 12).

Adding a new version to a container-based product

If you already have a container-based product in AWS Marketplace, you can use the Catalog API to add a new version. This requires that you have already created repositories in AWS Marketplace for each container image or artifact that is part of your product, and that you can copy them from your local Docker and Helm files.

> **Note**
> For details about creating a container-based product, see Getting started with container products.
> For details about adding a new version, including creating repositories and building Docker and Helm files into those repositories, by using the AWS Marketplace Management Portal, see Add a new version of your product in the *AWS Marketplace Seller Guide*.
> If you have not already created new repositories, you can create them using the Catalog API, see Creating repositories and resources for a container-based product (p. 30).

You add a new version by calling `StartChangeSet` with the `AddDeliveryOptions` change type, as shown in the following example.

> **Note**
> A version of a container-based product is made up of one or more delivery options. For example, you might have two delivery options, one that works with a noSQL database, and another that works with MySQL, so that your users can choose how they want to work with your product. You create the version of your product and add multiple delivery options in a single request with `AddDeliveryOptions`.

```
POST /StartChangeSet HTTP/1.1
Content-type: application/json

{
"Catalog": "AWSMarketplace",
"ChangeSet": [
  {
    "ChangeType":"AddDeliveryOptions",
    "Entity":{
      "Identifier":"example1-abcd-1234-5ef6-7890abcdef12",
      "Type":"ContainerProduct@1.0"
     },
     "Details": "{
       \"Version\": {
         \"VersionTitle\": \"1.1\",
         \"ReleaseNotes\": \"Minor bug fix\"},
       \"DeliveryOptions\": [{
         \"DeliveryOptionTitle\": \"EKSDelivery\",
         \"Details\": {
           \"EcrDeliveryOptionDetails\" : {
             \"ContainerImages\": [
               \"709825985650.dkr.ecr.us-east-1.amazonaws.com/sellername/reponame1:1.1\"],
```

```
            \"DeploymentResources\": [{
              \"Name\": \"HelmDeploymentTemplate\",
              \"Url\": \"709825985650.dkr.ecr.us-
east-1.amazonaws.com/sellername/reponame2:mychart1.1\"}],
            \"CompatibleServices\": [\"EKS\"],
            \"Description\": \"Sample Description\",
            \"UsageInstructions\":\"helm pull 709825985650.dkr.ecr.us-
east-1.amazonaws.com/sellername/reponame2:mychart1.1\"
          }
        }
      }]
    }"
  }]
}
```

Note
This example has line wraps added for readability and will not work as-is. The `Details` attribute is a string, and it should be converted from a JSON object to a string in your call to `StartChangeSet`. In this case, the string is `"{\"Version\": {\"VersionTitle\": \"1.1\",\"ReleaseNotes\": \"Minor bug fix\"},\"DeliveryOptions\": [{\"DeliveryOptionTitle\": \"EKSDelivery\", \"Details\": { \"EcrDeliveryOptionDetails\" : {\"ContainerImages\": [\"709825985650.dkr.ecr.us-east-1.amazonaws.com/sellername/reponame1:1.1\"],\"DeploymentResources\": [{\"Name\": \"HelmDeploymentTemplate\",\"Url\": \"709825985650.dkr.ecr.us-east-1.amazonaws.com/sellername/reponame2:mychart1.1\"}], \"CompatibleServices\": [\"EKS\"],\"Description\": \"Sample Description\",\"UsageInstructions\":\"helm pull 709825985650.dkr.ecr.us-east-1.amazonaws.com/sellername/reponame2:mychart1.1\"}}}]}"`. For more information, see Working with the `Details` attribute (p. 4).

The following is information about the input fields you must provide for adding the `AddDeliveryOptions` change type for container-based products:

- **Entity (object)** – Your container-based product. The `Identifier` is your product ID, and the type is always `ContainerProduct@1.0`. For information about using the Identifier, see Identifier (p. 2).
- **Details (string)** – Details of the request. It includes all the information about the version that you are adding. This field is a string field. The string is itself JSON, but it must be formatted properly for a string field, including escaping characters (such as quotation marks) that cannot directly be in a string. For more information, see Working with the `Details` attribute (p. 4).
  - **Version (object)** – Details about the version that you are adding to your product.
    - **VersionTitle (string)** – The title of the version that you are creating. Typically this is a description of the version, like **`Version 1.1`** or simply **`1.1`**. Your buyers will be able to choose the version to deploy from a list of version titles.
    - **ReleaseNotes (string)** – The detailed notes about this version. Must be less than 30,000 characters.
  - **DeliveryOptions (array of objects)** – An array of delivery options, where each is a method of delivery for your product version. For example, if you have one delivery option for Amazon Elastic Container Service (Amazon ECS) and another for Amazon Elastic Kubernetes Service (Amazon EKS), you will need to have two delivery options.
    - **DeliveryOptionTitle (string)** – A short description that helps your buyer to choose between your delivery options.
    - **Details (string)** – The resources used for this delivery option. This is a details field within the details field. You do not need to doubly escape characters in this field.
      - **ContainerImages (array of strings)** – An array of container image URLs used by this version. The path will be the repository that you have uploaded the image to, with the tag for the

image used by this version. The list must include all needed images, even images that have not changed from previous versions. See the next section for information about creating repositories using the Catalog API.

- **DeploymentResources (array of objects)** – An array of other resources needed for the version, such as Helm charts. Each resource includes a `Name` to describe it, and a `URL` that points at the resource.

- **CompatibleServices (array of strings)** – An array of services that the release is compatible with. Valid options are `ECS` and `EKS`.
- **Description (string)** – A longer description of the delivery option to give details to your buyer. You can also include a link to more instructions provided elsewhere.
- **UsageInstructions (string)** – Provide instructions about the usage for this delivery option. Can be up to 4,000 characters.

A change set is created for your request. The response to this request gives you the ID for the change set and looks like the following.

```
{
    "ChangeSetId": "example123456789012abcdef",
    "ChangeSetArn": "arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/
ChangeSet/example123456789012abcdef"
}
```

The change request is added to a queue and processed, including scanning the container images and other information to ensure that it meets the AWS Marketplace guidelines for container products. This process can take a few minutes to hours, depending on the number and size of your containers. You can check the status of the request through the AWS Marketplace Management Portal, or in the Catalog API with the `DescribeChangeSet` action. For more information about change sets, see Working with change sets (p. 3). For more information about errors in seller product change sets, see Change set status and errors (p. 12).

Creating repositories and resources for a container-based product

To create a new version of a container-based product, you must have the resources for the version available in AWS Marketplace repositories. You create the repositories and then push (upload) the Docker (and Helm) resources into the repositories. To learn how to create the repositories through the AWS Marketplace Management Portal, see Add a new version of your product in the *AWS Marketplace Seller Guide*.

To create new repositories with the Catalog API, call `StartChangeSet` with the `AddRepositories` change type, as shown in the following example.

```
POST /StartChangeSet HTTP/1.1
Content-type: application/json

{
"Catalog": "AWSMarketplace",
"ChangeSet": [
  {
    "ChangeType":"AddRepositories",
    "Entity":{
      "Identifier":"example1-abcd-1234-5ef6-7890abcdef12",
      "Type":"ContainerProduct@1.0"
     },
      "Details": "{
```

```
        \"Repositories\": [
          {
            \"RepositoryName\": \"new-repo-1\",
            \"RepositoryType\": \"ECR\"
          },
          {
            \"RepositoryName\": \"new-repo-2\",
            \"RepositoryType\": \"ECR\"
          }
        ]
      }"
    }
  ]
}
```

Note
This example has line wraps added for readability and will not work as-is. The `Details` attribute is a string, and it should be converted from a JSON object to a string in your call to `StartChangeSet`. In this case, the string is `"{ \"Repositories\": [ { \"RepositoryName\": \"new-repo-1\", \"RepositoryType\": \"ECR\" }, { \"RepositoryName\": \"new-repo-2\", \"RepositoryType\": \"ECR\" } ] }"`. For more information, see Working with the `Details` attribute (p. 4).

The following is information about the input fields you provide for adding the `AddRepositories` change type. For more information about creating repositories, see Adding a new version in the *AWS Marketplace Seller Guide*.

- **Entity (structure)** – The container-based product you are creating repositories for. The `Identifier` is your product ID, and the type is always `ContainerProduct@1.0`. For information about using the Identifier, see Identifier (p. 2).
- **Details (string)** – Details of the request. It includes the information about the repositories that you want to create. This field is a string field. The string is JSON, but it must be formatted properly for a single-line string field, including escaping characters (such as quotation marks) that cannot directly be in a string. For more information, see Working with the `Details` attribute (p. 4). The included fields are all required.
  - **Repositories (array of structures)** – A list of repository objects. Each repository object includes a name and type.
    - **RepositoryName (string)** – The name of the repository to create.
    - **RepositoryType (string)** – The type of the repository to create. The only allowed value is `ECR`.

Note
You can only have 50 repositories per product, although you can add multiple resources (and versions of resources) to a single repository by giving them different tags when you push them.

After you have created one or more repositories for your resources, you add your resources to the repositories. For general information about how to push resources to repositories, see Pushing an image in the *Amazon Elastic Container Registry User Guide*. For instructions about how to get the specific push commands needed for one of your repositories, see Adding a new version in the *AWS Marketplace Seller Guide*.

Updating version information for a container-based product

You can use the Catalog API to update the details of an existing version of your container-based product in AWS Marketplace.

Note
When a product is publicly available, you cannot update the version title, container images, delivery option title, or deployment resources for the version. If you need to update these aspects of a product, create a new version instead.

You update an existing version of your container-based product in the Catalog API by calling `StartChangeSet` with the `UpdateDeliveryOptions` change type, as shown in the following example. This updates the detail information for the delivery options that you specify, as well as the associated version. You must include at least one delivery option.

```
POST /StartChangeSet HTTP/1.1
Content-type: application/json

{
"Catalog": "AWSMarketplace",
"ChangeSet": [
  {
    "ChangeType":"UpdateDeliveryOptions",
    "Entity":{
      "Identifier":"example1-abcd-1234-5ef6-7890abcdef12",
      "Type":"ContainerProduct@1.0"
     },
      "Details": "{
        \"Version\": {
          \"ReleaseNotes\": \"New release notes\",
          \"VersionTitle\": \"Version 1.2\"
        },
        \"DeliveryOptions\": [
          {
            \"Id\": \"example4-2222-cccc-2222-ccccccccccc\",
            \"Details\": {
              \"EcrDeliveryOptionDetails\": {
                \"DeliveryOptionTitle\": \"New Delivery Option Title\",
                \"Description\": \"New description\",
                \"UsageInstructions\": \"New usage instructions\",
                \"CompatibleServices\": [
                  \"EKS\"
                ]
              }
            }
          }
        ]
      }"
    }
  ]
}
```

Note
This example has line wraps added for readability and will not work as-is. The `Details` attribute is a string, and it should be converted from a JSON object to a string in your call to `StartChangeSet`. In this case, the string is `"{ \"Version\": { \"ReleaseNotes \": \"New release notes\", \"VersionTitle\": \"Version 1.2\" }, \"DeliveryOptions\": [{ \"Id\": \"example4-2222-cccc-2222-ccccccccccc \", \"Details\": { \"EcrDeliveryOptionDetails\": { \"DeliveryOptionTitle \": \"New Delivery Option Title\", \"Description\": \"New description\", \"UsageInstructions\": \"New usage instructions\", \"CompatibleServices \": [\"EKS\"] } } }] }"`. For more information, see Working with the `Details` attribute (p. 4).

The following is information about the input fields you provide for adding the `UpdateDeliveryOptions` change type for container-based products. For more information about these fields, see Adding a new version in the AWS Marketplace Seller Guide.

- **Entity (object)** – Your container-based product. The `Identifier` is your product ID, and the type is always `ContainerProduct@1.0`. For information about using the Identifier, see Identifier (p. 2).
- **Details (string)** – Details of the request. It includes any information about the version of your container-based product that you would like to update. This field is a string field. The string is JSON, but it must be formatted properly for a single-line string field, including escaping characters (such as quotation marks) that cannot directly be in a string. For more information, see Working with the `Details` attribute (p. 4). The included fields are all optional, but you must include at least one field to update.
  - **Version (object)** – Details about the software version.
    - **VersionTitle (string)** – The title of the version that you are creating. Typically this is a description of the version, such as **`Version 1.1`** or simply **`1.1`**. Your buyers will be able to choose the version to deploy from a list of all version titles.

      This property can't be updated if the product is already published publicly.
    - **ReleaseNotes (string)** – Notes for buyers to tell them about changes from one version to the next.
  - **DeliveryOptions (list of objects)** – List of `DeliveryOption` objects, including the details of each:
    - Id (string) – Unique identifier for the `DeliveryOption` (you can get the unique identifier for the `DeliveryOption` by calling the `DescribeEntity` action on the product you are updating).
    - **Details (string)** – Holds the details of a delivery option. Note that this nested details object does *not* need to be double-escaped.
      - **EcrDeliveryOptionDetails (object)** – The details of the delivery option.
        - **DeliveryOptionTitle (string)** – A short description that allows your buyer to choose between your delivery options.

          This property can't be updated if the product is already published publicly.
        - **ContainerImages (array of strings)** – An array of container image URLs used by this version. The path will be the repository that you have uploaded the image to, with the tag for the image used by this version. If this field is included, the list must include all needed images, even images that are not changing.

          This property can't be updated if the product is already published publicly.
        - **DeploymentResources (array of objects)** – An array of other deployment resources needed for the version, such as links to Helm charts or other documentation. Each resource includes a name to describe it and a URL that points at the resource. On the launch page for your version, this displays as a list of links.

          This property can't be updated if the product is already published publicly.
          - **Name (string)** – The text of the hyperlink that is shown to the buyer.
          - **Url (string)** – The URL of the hyperlink shown to the buyer.
        - **CompatibleServices (array of strings)** – A list of services that the release is compatible with. Valid options are `ECS` and `EKS`.
        - **Description (string)** – A longer description of the delivery option to give details to your buyer. You can also include a link to more instructions hosted elsewhere.
        - **UsageInstructions (string)** – Provide instructions on how to deploy and use your product. You can also add a link to usage instructions hosted elsewhere. Can be up to 4,000 characters.

A change set is created for your request. The response to this request gives you the ID for the change set and looks like the following.

```
{
    "ChangeSetId": "example123456789012abcdef",
    "ChangeSetArn": "arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/
ChangeSet/example123456789012abcdef"
```

```
}
```

The change request is added to a queue and processed, including scanning the information to ensure that it meets the AWS Marketplace guidelines for container products. This process can take a few minutes to hours. You can check the status of the request through the AWS Marketplace Management Portal, or in the Catalog API with the `DescribeChangeSet` action. For more information about change sets, see Working with change sets (p. 3). For more information about errors in seller product change sets, see Change set status and errors (p. 12).

Restricting a version of a container-based product

You can use the Catalog API to restrict a version of your container-based product in AWS Marketplace. This prevents new buyers from being able to use that version. There must be at least one publicly available version in a product. You cannot restrict the only remaining publicly available version for a product.

You restrict a version in the Catalog API by calling `StartChangeSet` with the `RestrictDeliveryOptions` change type, as shown in the following example.

> **Note**
> Restricting one or more, but not all, delivery options from a version will remove those options from being available to your buyers. Restricting all delivery options for a version will remove that version from the AWS Marketplace catalog.
> Restricted versions are still available for existing customers.

```
POST /StartChangeSet HTTP/1.1
Content-type: application/json

{
"Catalog": "AWSMarketplace",
"ChangeSet": [
  {
    "ChangeType":"RestrictDeliveryOptions",
    "Entity":{
      "Identifier":"example1-abcd-1234-5ef6-7890abcdef12",
      "Type":"ContainerProduct@1.0"
     },
      "Details": "{ \"DeliveryOptionIds\": [ \"Id\": \"example1-2222-cccc-2222-
ccccccccccc\" ] }"
    }
  ]
}
```

The following is information about the input fields you provide for adding the `RestrictDeliveryOptions` change type:

- **Entity (object)** – Your container-based product. The `Identifier` is your product ID, and the type is always `ContainerProduct@1.0`. For information about using the Identifier, see Identifier (p. 2).
- **Details (string)** – Details of the request. It includes IDs for the delivery options of your container-based product that you would like to restrict. This field is a string field. The string is JSON, but it must be formatted properly for a single-line string field, including escaping characters (such as quotation marks) that cannot directly be in a string. For more information, see Working with the `Details` attribute (p. 4).
  - **DeliveryOptionIds (array of strings)** – List of `DeliveryOption` IDs for the versions that you want to restrict. You can get the unique identifier for the `DeliveryOption` by calling the `DescribeEntity` action on the product you are restricting.

A change set is created for your request. The response to this request gives you the ID for the change set and looks like the following.

```
{
    "ChangeSetId": "example123456789012abcdef",
    "ChangeSetArn": "arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/
ChangeSet/example123456789012abcdef"
}
```

The change request is added to a queue and processed. This process can take a few minutes to hours. You can check the status of the request through the AWS Marketplace Management Portal, or in the Catalog API with the `DescribeChangeSet` action. For more information about change sets, see Working with change sets (p. 3). For more information about errors in seller product change sets, see Change set status and errors (p. 12).

Errors in the container products API

The following errors are specific to the container product actions in the AWS Marketplace Catalog API. These errors are returned when you call `DescribeChangeSet` after a change set is processing. For more details about using `DescribeChangeSet` to get the status of a change request, see Working with change sets (p. 3).

Change type	Error code	Error message
AddDeliveryOptions	INCOMPATIBLE_PRODUCT_STATUS	Use an existing limited or public product.
AddDeliveryOptions	INCOMPATIBLE_SERVICES	Provide a valid list of compatible services.
AddDeliveryOptions	NO_SERVICE_SPECIFIED	Provide at least 1 compatible service.
AddDeliveryOptions	DUPLICATE_COMPATIBLE_AWS_SERVICES	Provide unique list of compatible services.
AddDeliveryOptions	INVALID_VERSION_TITLE	Remove spaces before the trademark symbol.
AddDeliveryOptions	INVALID_VERSION_TITLE	Remove the following unsupported characters: [x, y, z]
AddDeliveryOptions	INVALID_VERSION_TITLE	Remove spaces from the beginning of the version title.
AddDeliveryOptions	INVALID_VERSION_TITLE	Provide version title with fewer than [x] characters.
AddDeliveryOptions	DUPLICATE_VERSION_TITLE	The version title must be different from any other version titles of this product.
AddDeliveryOptions	INVALID_RELEASE_NOTES	Remove spaces before the trademark symbol.
AddDeliveryOptions	INVALID_RELEASE_NOTES	Remove unsupported characters: [x, y, z]

Change type	Error code	Error message
AddDeliveryOptions	INVALID_RELEASE_NOTES	Remove spaces from the beginning of release notes.
AddDeliveryOptions	INVALID_RELEASE_NOTES	Provide release notes with fewer than (x) characters.
AddDeliveryOptions	INVALID_USAGE_INSTRUCTIONS	Remove spaces before the trademark symbol.
AddDeliveryOptions	INVALID_USAGE_INSTRUCTIONS	Remove unsupported characters: [x, y, z]
AddDeliveryOptions	INVALID_USAGE_INSTRUCTIONS	Provide usage instructions with fewer than (x) characters.
AddDeliveryOptions	INVALID_USAGE_INSTRUCTIONS	Provide usage instructions.
AddDeliveryOptions	MISSING_CONTAINER_IMAGES	Provide at least 1 container image.
AddDeliveryOptions	TOO_MANY_CONTAINER_IMAGES	Provide fewer than 50 container images.
AddDeliveryOptions	DUPLICATE_CONTAINER_IMAGES	Provide a unique list of container images.
AddDeliveryOptions	INVALID_CONTAINER_IMAGES	Provide a valid URI for the container image.
AddDeliveryOptions	INVALID_CONTAINER_IMAGE_URI	Provide a valid URI for the container image.
AddDeliveryOptions	INVALID_CONTAINER_IMAGE_TAG	Avoid using 'latest' tag.
AddDeliveryOptions	DUPLICATE_DELIVERY_OPTION_TITLES	Provide unique delivery option title.
AddDeliveryOptions	INVALID_DELIVERY_OPTION_TITLES	Delivery option title already exists, retry with a different title.
AddDeliveryOptions	INVALID_FULFILLMENT_OPTION_TITLE	Provide delivery option title with fewer than (x) characters.
AddDeliveryOptions	DUPLICATE_DELIVERY_OPTION_TITLES	Provide unique delivery option title.
AddDeliveryOptions		Provided Details is not valid.
AddDeliveryOptions	EMPTY_RESOURCE_NAME	Provide resource name.
AddDeliveryOptions	EMPTY_RESOURCE_URL	Provide resource URL.
AddDeliveryOptions	INVALID_RESOURCE_NAME	Provide resource name with fewer than 256 characters.
AddDeliveryOptions	INVALID_RESOURCE_URL	Provide resource URL with fewer than 256 characters.

Change type	Error code	Error message
AddDeliveryOptions	INVALID_SHORT_DESCRIPTION	Provide a short description with fewer than 1,000 characters.
AddDeliveryOptions	INVALID_SHORT_DESCRIPTION	Provide short description.
AddDeliveryOptions	SCAN_ERROR	Fix security vulnerability ""[y]"" on Image ""[x]"".
AddDeliveryOptions	IMAGE_NOT_FOUND	Provide a valid public image URI.
AddDeliveryOptions	INVALID_ARN	Provide a valid ARN for image access.
AddDeliveryOptions	IMAGE_INACCESSIBLE	Provide a valid ARN for image access.
RestrictDeliveryOptions	INCOMPATIBLE_PRODUCT_STATUS	Use a public product.
RestrictDeliveryOptions	MISSING_DELIVERY_OPTION_IDS	Provide delivery option from existing list of IDs.
RestrictDeliveryOptions	INVALID_DELIVERY_OPTIONS_STATUS	Provide delivery options in public state.
RestrictDeliveryOptions	EMPTY_DELIVERY_OPTION_IDS	Provide non-empty list of delivery option IDs.
RestrictDeliveryOptions	INVALID_MINIMUM_PUBLIC_DELIVERY_OPTIONS	Can not restrict all delivery option IDs.
RestrictDeliveryOptions	DUPLICATE_DELIVERY_OPTION_IDS	Provide unique delivery option IDs.
UpdateDeliveryOptions	INCOMPATIBLE_PRODUCT_STATUS	Use an existing limited or public product.
UpdateDeliveryOptions	INCOMPATIBLE_SERVICES	Provide a valid list of compatible services.
UpdateDeliveryOptions	NO_SERVICE_SPECIFIED	Provide at least 1 compatible service.
UpdateDeliveryOptions	DUPLICATE_COMPATIBLE_AWS_SERVICES	Provide unique list of compatible services.
UpdateDeliveryOptions	INVALID_VERSION_TITLE	Remove spaces before the trademark symbol.
UpdateDeliveryOptions	INVALID_VERSION_TITLE	Remove the following unsupported characters: [x, y, z]
UpdateDeliveryOptions	INVALID_VERSION_TITLE	Remove spaces from the beginning of the version title.
UpdateDeliveryOptions	INVALID_VERSION_TITLE	Provide version title with fewer than [x] characters.

Change type	Error code	Error message
UpdateDeliveryOptions	DUPLICATE_VERSION_TITLE	The version title must be different from any other version titles of this product.
UpdateDeliveryOptions	INVALID_RELEASE_NOTES	Remove spaces before the trademark symbol.
UpdateDeliveryOptions	INVALID_RELEASE_NOTES	Remove unsupported characters: [x, y, z]
UpdateDeliveryOptions	INVALID_RELEASE_NOTES	Remove spaces from the beginning of release notes.
UpdateDeliveryOptions	INVALID_RELEASE_NOTES	Provide release notes with fewer than (x) characters.
UpdateDeliveryOptions	INVALID_USAGE_INSTRUCTIONS	Remove spaces before the trademark symbol.
UpdateDeliveryOptions	INVALID_USAGE_INSTRUCTIONS	Remove unsupported characters: [x, y, z]
UpdateDeliveryOptions	INVALID_USAGE_INSTRUCTIONS	Provide usage instructions with fewer than (x) characters.
UpdateDeliveryOptions	INVALID_USAGE_INSTRUCTIONS	Provide usage instructions.
UpdateDeliveryOptions	MISSING_CONTAINER_IMAGES	Provide at least 1 container image.
UpdateDeliveryOptions	TOO_MANY_CONTAINER_IMAGES	Provide fewer than 50 container images.
UpdateDeliveryOptions	DUPLICATE_CONTAINER_IMAGES	Provide a unique list of container images.
UpdateDeliveryOptions	INVALID_CONTAINER_IMAGES	Provide a valid URI for the container image.
UpdateDeliveryOptions	INVALID_CONTAINER_IMAGE_URI	Provide a valid URI for the container image.
UpdateDeliveryOptions	INVALID_CONTAINER_IMAGE_TAG	Avoid using 'latest' tag.
UpdateDeliveryOptions	MISSING_DELIVERY_OPTION_IDS	Provide delivery option from existing list of Ids.
UpdateDeliveryOptions	EMPTY_DELIVERY_OPTION_IDS	Provide non-empty list of delivery option IDs.
UpdateDeliveryOptions	DUPLICATE_DELIVERY_OPTION_IDS	Provide unique delivery option IDs.
UpdateDeliveryOptions	DUPLICATE_DELIVERY_OPTION_TITLES	Provide unique delivery option title.
UpdateDeliveryOptions	INVALID_DELIVERY_OPTION_TITLES	Delivery option title already exists, retry with a different title.

Change type	Error code	Error message
UpdateDeliveryOptions	INVALID_FULFILLMENT_OPTION_TITLE	Provide delivery option title with fewer than (x) characters.
UpdateDeliveryOptions	DUPLICATE_DELIVERY_OPTION_TITLES	Provide unique delivery option title.
UpdateDeliveryOptions	EMPTY_RESOURCE_NAME	Provide resource name.
UpdateDeliveryOptions	EMPTY_RESOURCE_URL	Provide resource URL.
UpdateDeliveryOptions	INVALID_RESOURCE_NAME	Provide resource name with fewer than 256 characters.
UpdateDeliveryOptions	INVALID_RESOURCE_URL	Provide resource URL with fewer than 256 characters.
UpdateDeliveryOptions	INVALID_SHORT_DESCRIPTION	Provide a short description with fewer than 1,000 characters.
UpdateDeliveryOptions	INVALID_SHORT_DESCRIPTION	Provide short description.
UpdateDeliveryOptions	SCAN_ERROR	Fix security vulnerability ""[y]"" on Image ""[x]"".
UpdateDeliveryOptions	FIELD_NOT_ALLOWED_TO_CHANGE	Field [x] cannot be changed.
UpdateDeliveryOptions	INVALID_DELIVERY_OPTIONS_STATUS	Provide delivery options in public or limited state.
UpdateDeliveryOptions	NO_CHANGE_FOUND	Provide at least 1 change.
UpdateDeliveryOptions	MULTIPLE_VERSION_UPDATE	Provide delivery option IDs from the same version.
AddRepositories	INVALID_ECR_REPOSITORY_NAME	Provide repository name in the format: 'nginx-web-app'
AddRepositories	DUPLICATE_ECR_REPOSITORY_NAME	The repository name must be unique.
AddRepositories	MISSING_REPOSITORY_INFORMATION	Provide at least 1 repository name.
AddRepositories	INVALID_ECR_REPOSITORY_NAME	Maximum character length 256 reached. Character length count is inclusive of the seller namespace.
UpdateInformation	MISSING_UPDATES	Nothing to update. Provide updated information for product.
UpdateInformation	MISSING_LOGO_URL	Provide URL for logo stored in S3.
UpdateInformation	LOGO_COPY_FAILURE	There was an issue copying the logo from S3. Provide a new URL for logo stored in S3.

Change type	Error code	Error message
UpdateInformation	INVALID_CATEGORY_NAMES	Provide valid category names supported by AWS Marketplace.
UpdateInformation	INVALID_PRODUCT_TITLE	Remove spaces before trademark symbol.
UpdateInformation	INVALID_PRODUCT_TITLE	Remove unsupported characters [x, y, z]
UpdateInformation	INVALID_SHORT_DESCRIPTION	Remove spaces before trademark symbol.
UpdateInformation	INVALID_SHORT_DESCRIPTION	Remove unsupported characters [x, y, z]
UpdateInformation	INVALID_LONG_DESCRIPTION	Remove spaces before trademark symbol.
UpdateInformation	INVALID_LONG_DESCRIPTION	Remove unsupported characters [x, y, z].
UpdateInformation	INVALID_SUPPORT_DESCRIPTION	Remove spaces before trademark symbol.
UpdateInformation	INVALID_SUPPORT_DESCRIPTION	Remove unsupported characters [x, y, z].
UpdateInformation	INVALID_HIGHLIGHTS	Remove spaces before trademark symbol.
UpdateInformation	INVALID_HIGHLIGHTS	Remove unsupported characters [x, y, z].
UpdateInformation	INVALID_ADDITIONAL_RESOURCES	Remove spaces before trademark symbol.
UpdateInformation	INVALID_ADDITIONAL_RESOURCES	Remove unsupported characters [x, y, z].
UpdateInformation	INVALID_PRODUCT_CATEGORIES	Provide between 1 and 3 product categories.
UpdateInformation	INVALID_IMAGE_PROPERTIES	Provide an image that meets the product logo requirements. (link)

Working with a private marketplace

You can use the AWS Marketplace Catalog API to manage a *private marketplace* for your AWS account or organization.

For more information about private marketplaces, see Private marketplaces in the *AWS Marketplace Buyer Guide*.

The following topics describe how to use the Catalog API to perform actions on your private marketplace.

Topics

Creating a private marketplace

A private marketplace for an AWS account can be thought of as a list of products that users are allowed to procure in that account, and branding for the marketplace. In an organization with multiple accounts, the accounts can be grouped together and share experiences. For example, you could have one set of products that all accounts in the organization are allowed to procure, or you could have a different list of products for each account in the organization. Each list of approved products and branding is called a procurement *experience*. In the AWS Marketplace Catalog API, four entities represent an experience. At the highest level is an `Experience` entity, which contains two child entities. A `ProcurementPolicy` represents the products that have been allowed and denied in your private marketplace. You can also create a `BrandingSettings` entity to define how your private marketplace looks to your users. You must also associate one or more `Audience` entities, which define the set of AWS accounts that the experience applies to.

The steps to create a procurement experience are as follows:

1. Create the `Experience` entity.
2. Create a `ProcurementPolicy` entity to store the list of products that are allowed or denied for the experience.
3. (Optional) Create a `BrandingSettings` entity to customize the look of your marketplace experience.
4. Associate accounts with your experience. For single AWS accounts, this is just the one account.
5. Enable the experience.

> **Note**
> If your account is part of an organization in AWS Organizations, see Working with private marketplaces for AWS Organizations (p. 48).

Create the `Experience` entity

To create the `Experience` entity, use the `StartChangeSet` action with the `CreateExperience` value for the `ChangeType` parameter to request that the experience be created by AWS Marketplace. See the following code example.

```
POST /StartChangeSet HTTP/1.1
Content-type: application/json

{
   "Catalog": "AWSMarketplace",
   "ChangeSet": [
      {
         "ChangeType": "CreateExperience",
         "Details": "{\"Name\": \"ExamplePrivateMarketplace\"}",
         "Entity": {
            "Type": "Experience@1.0"
         }
      }
   ],
   "ChangeSetName": "Create Private Marketplace Example"
}
```

In this action, `Entity` is a template for the entity that you want to create. It is assigned an `EntityId` when it is created. `ChangeSetName` identifies the change to help you find it later.

The response looks like the following.

```
{
   "ChangeSetArn": "arn:...:AWSMarketplace/ChangeSet/abcd1234example5678frjzkz",
   "ChangeSetId": "abcd1234example5678frjzkz"
}
```

The response includes a `ChangeSetId` that you can use to get the status of your change request as it is processed with `DescribeChangeSet`. You can also use `ListEntities` to find your `Experience` entity without the `ChangeSetId`. For more information about change sets, see Working with change sets (p. 3).

A newly created `Experience` entity doesn't have a procurement policy by default. It is also created with default settings for branding. For more information about branding settings, including how to customize them, see Changing the branding of a private marketplace experience (p. 43).

Create a `ProcurementPolicy` entity

You must create a `ProcurementPolicy` entity. By default, a new `Experience` entity is disabled, so you can create the procurement policy before enabling it.

> **Note**
> An `Experience` entity with *no* procurement policy (null) allows all products to be procured in your private marketplace. An `Experience` entity with an *empty* procurement policy has no products available to users to procure.

To allow and deny products in your private marketplace, you must create the procurement policy. To do this, you again call `StartChangeSet`, but this time with the `ChangeType` of `CreateProcurementPolicy`. The following code example creates an empty procurement policy.

```
POST /StartChangeSet HTTP/1.1
Content-type: application/json

{
   "Catalog": "AWSMarketplace",
```

```
   "ChangeSet": [
      {
         "ChangeType": "CreateProcurementPolicy",
         "Details": "{\"Name\": \"ExampleProcurementPolicy\"}",
         "Entity": {
            "Type": "Experience@1.0",
            "Identifier" : "exp-1234example@1"
         }
      }
   ]
}
```

The `Entity` you provide in this action is the `Experience` entity that you want the procurement policy created within, so you must include the identifier for the entity that you created earlier. Use `ListEntities` to find the `Experience` entity. You can also return the identifier by using `DescribeChangeSet` with the change set identifier from the `CreateExperience` action

> **Note**
> This example shows the identifier with a revision of `1`. For more information about revisions for identifiers, see Identifier (p. 2).

You can again use `DescribeChangeSet` on the `CreateProcurementPolicy` change to follow the processing of your request.

> **Note**
> The names you give the `Experience` and `ProcurementPolicy` objects do not appear in AWS Marketplace. The names are only for your ease of finding the entities in the API.

After you have created the procurement policy, your private marketplace displays in the AWS Management Console. (You can go to the private marketplace page to see it.) After you have completed these steps, your private marketplace will be disabled, have default branding, have an empty procurement policy, and will not be associated with any accounts in your organization. You can update the branding and add any products that you want in it, associate the experience with one or more accounts, and then enable your private marketplace.

The following sections describing managing your private marketplace with the AWS Marketplace Catalog API.

Changing the branding of a private marketplace experience

You can customize the look of your private marketplace for your users. Without customization, your private marketplace will have the default branding settings, which are described below. Aspects of branding that you can change in a private marketplace include the following:

- `Title` – The name displayed for your private marketplace. This is the same as the **Name** field in the private marketplace **Profile settings** screen. If you set the `Title` to **`Example`**, then the text displayed is **Example Private Marketplace**. The default is **Private Marketplace**.
- `Information` – The paragraph displayed under the name in your private marketplace. This is the same as the **Description** field in **Profile settings**. The default is no information, in which case a general description of private marketplaces is displayed.
- `ThemeColor` – The color displayed in the banner of your private marketplace. This is a color in RGB hexadecimal format. This value is the same as the **Theme color** field in **Profile settings**. The default value is `#232F3E`.
- `LogoUrl` – The URL that points to an image file to be used as the logo on your private marketplace. The URL must be publicly available (for example, a signed Amazon S3 URL). The file must be either

a .png or .svg file and be under 500kb. If necessary, the image file will be resized to a maximum height of 30 pixels and a maximum width of 100 pixels. This is the same value as the **Logo Select** in **Profile Settings**. The default is to not show a logo.

To set these values, you must first create a `BrandingSettings` entity with the `CreateBrandingSettings` change type. You can then request an `UpdateBrandingSettings` change to set or change the branding. You only need to create a `BrandingSettings` object once. To create this object, call `StartChangeSet` with the `CreateBrandingSettings` change type, as shown in the following code example.

```
POST /StartChangeSet HTTP/1.1
Content-type: application/json

{
   "Catalog": "AWSMarketplace",
   "ChangeSet": [
      {
         "ChangeType": "CreateBrandingSettings",
         "Details": "{ \"Name\": \"ExampleBrandingSettingsName\"}",
         "Entity": {
            "Type": "Experience@1.0",
            "Identifier" : "exp-1234example@2"
         }
      }
   ]
}
```

This examples modifies the `Experience` entity by adding the `BrandingSettings` object to it. The revision of the entity identifier has incremented to `2`. For more information about revisions for identifiers, see Identifier (p. 2).

Note
You can specify all the details of the branding settings in the call to create the branding settings entity. The details facet is the same for `CreateBrandingSettings` and `UpdateBrandingSettings`.

You modify the settings by calling `StartChangeSet` with the `UpdateBrandingSettings` change type. The settings are part of the `Configuration` of the `Details` object.

```
POST /StartChangeSet HTTP/1.1
Content-type: application/json

{
   "Catalog": "AWSMarketplace",
   "ChangeSet": [
      {
         "ChangeType": "UpdateBrandingSettings",
         "Details": "{
             \"Name\": \"ExampleBrandingSettingsName\",
             \"Description\": \"Example description\",
             \"Configuration\": {
                 \"Title\": \"ExampleName\",
                 \"Information\": \"Example description.\",
                 \"ThemeColor\" : \"#0e7f74\",
                 \"LogoUrl\" : \"https://example.com/path/mylogo.png\"
             }
         }",
         "Entity": {
            "Type": "Experience@1.0",
            "Identifier" : "exp-1234example@3"
         }
```

```
      }
   ]
}
```

Note
This example has line wraps added for readability and will not work as-is. The `Details` attribute is a string, and it should be converted from a JSON object to a string in your call to `StartChangeSet`. In this case, the string would be `"{\"Name\": \"ExampleBrandingSettingsName\",\"Description\": \"Example description \",\"Configuration\": {\"Title\": \"ExampleName\",\"Information\": \"Example description.\",\"ThemeColor\" : \"#0e7f74\",\"LogoUrl\" : \"https://example.com/path/mylogo.png\"}}"`. For more information, see Working with the `Details` attribute (p. 4).

Note
The URL for the logo is used to make a copy during the update change. After the change is complete, if you remove or change the URL at that path, it will not affect your private marketplace unless you again request `UpdateBrandingSettings`.

Enabling or disabling a private marketplace experience

When a private marketplace is enabled (and has a procurement policy), users in associated accounts can only purchase products that you have approved. When no private marketplace experience is enabled for an account, users can purchase products across the full AWS Marketplace catalog.

To enable a private marketplace, use the `StartChangeRequest` action with the `UpdateExperience` change type.

```
POST /StartChangeSet HTTP/1.1
Content-type: application/json

{
   "Catalog": "AWSMarketplace",
   "ChangeSet": [
      {
         "ChangeType": "UpdateExperience",
         "Details": "{\"Status\": \"Enabled\"}",
         "Entity": {
            "Type": "Experience@1.0",
            "Identifier": "exp-1234example@4"
         }
      }
   ]
}
```

Similarly, you can use the same action and `ChangeType`, but change the `Status` in `Details` to `Disabled` to disable a private marketplace.

Note
Disabling a private marketplace keeps your list of both allowed and denied products, as well as customizations, such as branding. When a private marketplace is disabled, users no longer see the private marketplace (although they may still be governed by the management account experience). If there are no private marketplace experiences enabled for an account, then all restrictions are removed, and users are able to procure any products in the public AWS Marketplace.

Enabling or disabling user requests

Users in your organization can view the full public AWS Marketplace, but they can only subscribe to the products that you have allowed. By default, they can request that a product that is not in the private marketplace be added to it. These requests show up in the private marketplace administrator page (https://aws.amazon.com/marketplace/privatemarketplace/admin/), where you can decide whether to accept or deny the request (and whether to block further requests for the same product). You cannot see or respond to the requests by using the Catalog API.

You can enable or disable the ability for users to create requests for your private marketplace experience. Use `StartChangeSet` with the `UpdateProcurementPolicy` change type. The ability to make requests is disabled in the following code example.

```
POST /StartChangeSet HTTP/1.1
Content-type: application/json

{
   "Catalog": "AWSMarketplace",
   "ChangeSet": [
      {
         "ChangeType": "UpdateProcurementPolicy",
         "Details": "{\"Configuration\" : {\"PolicyResourceRequests\" : \"Deny\"}}",
         "Entity": {
            "Type": "Experience@1.0",
            "Identifier" : "exp-1234example@5"
         }
      }
   ]
}
```

To enable the change request capability for users, use `Allow` instead of `Deny` in `PolicyResourceRequests`.

To learn how to get the current status of this setting, see the next section, Getting a list of products in a private marketplace experience (p. 46).

Getting a list of products in a private marketplace experience

The products allowed (and denied) in a private marketplace are part of the procurement policy in the `Experience` entity. To get the details about the procurement policies in a private marketplace, you first get the procurement policy identifier from the `Experience` entity, and then call `DescribeEntity` with that identifier.

To get the procurement policy identifier, use `DescribeEntity` on the `Experience` entity that you are interested in, as shown in the following command.

```
GET /DescribeEntity?catalog=AWSMarketplace&entityId=exp-example01
```

Following is an example response (new lines added for readability).

```
{
    "Details": "{\"Name\":\"New Private Marketplace\",
          \"Status\":\"Enabled\",
          \"ProcurementPolicies\":[\"procpolicy-123example456\"],
          \"BrandingSettings\":[\"brandsettings-456example123\"]}",
```

```
    "EntityArn": "arn:<...>:AWSMarketplace/Experience/exp-example-01",
    "EntityIdentifier": "exp-example01@6",
    "EntityType": "Experience@1.0",
    "LastModifiedDate": "2021-01-13T20:31:36Z"
}
```

You can use the returned `EntityId` for the procurement policy to get the details, as shown in the following command.

```
GET /DescribeEntity?catalog=AWSMarketplace&entityId=procpolicy-123example456
```

This returns the full details of the policy, including both allowed and denied products. Following is an example response (new lines added for readability).

```
{
    "Details": "{\"Name\":\"ExampleProcurementPolicy\",
         \"Statements\":[
            {\"Effect\":\"Allow\",\"Resources\":[
               {\"Type\":\"Product\",\"Ids\":[\"example1-1234-abcd-5678-90abcdef1234\"]},
               {\"Type\":\"Product\",\"Ids\":[\"example2-2345-bcde-6789-01bcdea2345\"]}]},
            {\"Effect\":\"Deny\",\"Resources\":[
               {\"Type\":\"Product\",\"Ids\":[\"example3-3456-
cdef-7890-12defabc5678\"]}]}],
         \"Configuration\":{\"PolicyResourceRequests\":\"Allow\"}}",
    "EntityArn": "arn:<...>AWSMarketplace/ProcurementPolicy/procpolicy-123example456",
    "EntityIdentifier": "procpolicy-123example456@4",
    "EntityType": "ProcurementPolicy@1.0",
    "LastModifiedDate": "2020-10-01T12:00:00Z"
}
```

In this sample, the procurement policy has two allowed products and one denied product. The policy allows user resource requests.

Adding or removing products from a private marketplace

By default, a private marketplace does not have any approved products in it. Use change requests to add or remove a product. To add a product, use the `AllowProductProcurement` change type. To remove a product, use the `DenyProductProcurement` change type.

The following code example shows the `AllowProductProcurement` change type with the `StartChangeSet` action to add a product to a private marketplace.

```
POST /StartChangeSet HTTP/1.1
Content-type: application/json

{
   "Catalog": "AWSMarketplace",
   "ChangeSet": [
      {
         "ChangeType": "AllowProductProcurement",
         "Details": "{\"Products\":[{\"Ids\":[\"example-1234-abcd-5678-90abcded1234\"],
\"Notes\":\"Useful product\"}]}",
         "Entity": {
            "Identifier": "exp-1234example@6",
            "Type": "Experience@1.0"
         }
      }
```

```
    ]
}
```

You add the product to the `Experience` entity for a private marketplace by using `AllowProductProcurement`. The syntax to remove a product from a private marketplace is identical, with the exception that you use the `DenyProductProcurement` `ChangeType` instead of `AllowProductProcurement`. The products are added to the allow (or deny) list of the `ProcurementPolicy` entity that is contained by your `Experience` entity.

> **Note**
> The list of products in the `Details` of your change is an array of `Ids`, so you can add (or remove) multiple products with one call by including a list of product identifiers. The limit is 50 products in a single request.
> The `Notes` field for the list of `Ids` is not required. However, you can use it to record why a decision to allow or deny a set of products was made.

Finding products

By getting the details of your procurement policy, you can find the product IDs for the products that are already in a private marketplace. However, the AWS Marketplace Catalog API does not provide a way to find the product IDs for other products. There are two ways to get product IDs to use with the Catalog API service:

- **Public marketplace** – After you find a product in the public marketplace, choose **Continue to Subscribe** to see a details page about the product (it will not subscribe you to the product). The URL will include the product ID as a parameter. For example, in the URL `https://aws.amazon.com/marketplace/fulfillment?productId=ab1234cd-1234-abcd-5678-90abcdef1234&ref_=dtl_psb_continue`, *`ab1234cd-1234-abcd-5678-90abcdef1234`* is the product ID.
- **AWS Marketplace Discovery API** – Programmatically, you can access the full list of products in the AWS Marketplace by using the Discovery API. The Discovery API is a private API. You must request access to be able to use it. For more information, see Getting access to the Discovery API (p. 55).

Working with private marketplaces for AWS Organizations

Whether you are working with a private marketplace for your account or your organization, you use the same API. However, there are differences when working within your organization:

- You must create the first `Experience` of your private marketplace from within the management account. This creates the `Experience` entity and enables the private marketplace feature for your organization.
- After the first `Experience` is created, it is automatically shared, allowing you to edit and update it from any account in the organization, as long as you have the correct permissions. You can also create additional `Experience` entities from any account.
- When listing objects in a private marketplace from another account (that has been automatically shared across the organization), you must specifically request them with the `SharedWithMe` filter. This applies to both `ListEntities` and `ListChangeSets` actions.

For example, when you call `ListEntities` from an account that isn't the management account, it only returns entities created by or owned by that account, by default. To list `Experience` objects in your own account, call `ListEntities` as shown in the following code example.

```
POST /ListEntities HTTP/1.1
Content-Type: application/json

{
    "Catalog":"AWSMarketplace",
    "EntityType":"Experience"
}
```

However, to list the entities that have been shared with you, you must add a `FilterList` with a `Scope` of `SharedWithMe`, as shown in the following code example. As a result, AWS Marketplace searches outside of your own account to find entities that are shared with you.

```
POST /ListEntities HTTP/1.1
Content-Type: application/json

{"Catalog":"AWSMarketplace",
"EntityType":"Experience",
 "FilterList":
  [{
     "Name": "Scope",
     "ValueList":
      ["SharedWithMe"]
      }]}
```

In this case, only entities outside of your account (the ones for your organization) are returned.

Similarly, to call `ListChangeSets`, you must set the scope, as shown in the following code example.

```
POST /ListChangeSets HTTP/1.1
Content-Type: application/json

{"Catalog":"AWSMarketplace",
 "FilterList":
  [{
     "Name": "Scope",
     "ValueList":
      ["SharedWithMe"]
      }]}
```

This returns change sets that apply to a shared private marketplace for your organization.

Associating accounts to experiences

A private marketplace experience must have one or more accounts associated with it in order to have any effects in your organization. For a single AWS account, you must associate the account with the experience to use the private marketplace. In an organization, you can have multiple experiences apply to different accounts.

> **Note**
> The experience that is associated with the management account is the default for all other accounts in the organization. Associating a linked account with a different experience directly sets a different experience for that account.

To associate an account to an experience use the `AssociateAudience` change type with the `StartChangeSet` action, as shown in the following code example.

```
POST /StartChangeSet HTTP/1.1
Content-type: application/json
```

```
{
   "Catalog": "AWSMarketplace",
    "ChangeSet" : [
        {
            "Entity": {
                "Type": "Experience@1.0",
                "Identifier": "exp-example01@1"
            },
            "ChangeType": "AssociateAudience",
            "Details": "{\"Name\": \"AudienceName\",\"Description\" : \"Audience example.
\",\"Principals\" : [\"012345678901\"]}"
        }
    ],
    "ChangeSetName": "Set Audience for experience 01"
}
```

The *audience* is the list of *principals* that are associated with the `Experience`. A principal is an AWS account, defined by its ID. `Principals` is a list, so you can include multiple accounts to be associated with the experience. After the first call, subsequent calls to the `AssociateAudience` change type will add principals to the association for the experience.

You can also remove accounts from an experience. Use the `DisassociateAudience` change type to do this, as shown in the following code example.

```
POST /StartChangeSet HTTP/1.1
Content-type: application/json

{
   "Catalog": "AWSMarketplace",
    "ChangeSet" : [
        {
            "Entity": {
                "Type": "Experience@1.0",
                "Identifier": "exp-example01@02"
            },
            "ChangeType": "DisassociateAudience",
            "Details": "{\"Principals\" : [\"012345678901\"]}"
        }
    ],
    "ChangeSetName": "Disassociate audience example"
}
```

Note
An account can only be directly associated with one experience. To move an account from being directly associated with one experience to another experience, you must disassociate it from the initial experience, then associate it with the second.

Errors in the private marketplace API

The following errors are specific to the private marketplace actions in the AWS Marketplace Catalog API.

Change type	Error code	Error message	Description
Errors returned directly by the StartChangeSet action			
All	422	Document not valid JSON format	Invalid JSON input used, check your syntax.

Change type	Error code	Error message	Description
AllowProductProcurement, DenyProductProcurement	422	Values in Ids array must be unique	You can't include the same product multiple times in a single change request.
AllowProductProcurement, DenyProductProcurement	422	Cumulative number of values in Ids array must be less than or equal to 50	You can allow or deny up to 50 products in a single change request.
Errors found by calling the DescribeChangeSet action			
CreateBrandingSettings, UpdateBrandingSettings	INVALID_URL	Image could not be fetched from the input URL	You must specify a valid, reachable URL for the logo field in `BrandingSettings`.
CreateBrandingSettings, UpdateBrandingSettings	INVALID_IMAGE	Image verification for type, content, or file size failed. Only .png and .svg file types with sizes less than or equal to 500KB are supported.	Your image file must match the logo requirements for branding settings.
AllowProductProcurement, DenyProductProcurement	ENTITY_NOT_FOUND	Procurement policy missing from Experience	You must create a `ProcurementPolicy` before allowing or denying products.
CreateProcurementPolicy	ENTITY_ALREADY_EXISTS	Procurement policy exists for Experience	You can only have a single procurement policy for a private marketplace.
UpdateProcurementPolicy	ENTITY_NOT_FOUND	Procurement policy missing from Experience	You must create a `ProcurementPolicy` before updating the procurement policy.
CreateBrandingSettings	ENTITY_ALREADY_EXISTS	Branding settings exists for Experience	You can only have a single branding settings for a private marketplace.
UpdateBrandingSettings	ENTITY_NOT_FOUND	Branding settings missing from Experience	You must create a `BrandingSettings` entity before updating the branding settings.
AssociateAudience	CALLER_NOT_AUTHORIZED	Caller not authorized to execute the action	You must have permissions to call the action. The accounts being added must be in the same organization.

Change type	Error code	Error message	Description
CreateExperience	CALLER_NOT_AUTHORIZED	Caller not authorized to create experience. Initial experience must be created from management account, if AWS Orgs is enabled.	You must have permissions to create an experience, and the initial experience for an organization must be created in the management account.
AssociateAudience	ENTITY_ALREADY_EXISTS	An experience is already associated with the account {accountId}. Disassociate previous experience before updating	You can only associate a single experience with an account. Disassociate the current experience before associating a new one.
AssociateAudience, DisassociateAudience	ENTITY_IN_USE	There is already a conflicting changing in progress for the selected account(s). Try again later	You can't change the association with an account while another change request to change the association is already in progress.

Logging AWS Marketplace Catalog API calls with CloudTrail

The AWS Marketplace Catalog API is integrated with AWS CloudTrail, a service that provides a record of actions taken by a user, role, or an AWS service. CloudTrail captures all calls to the Catalog API as events, including calls from the AWS Marketplace Management Portal.

If you create a trail, you can enable continuous delivery of CloudTrail events to an Amazon S3 bucket. If you don't configure a trail, you can still view the most recent events in the CloudTrail console in **Event history**. Using the information collected by CloudTrail, you can determine the request, the IP address from which the request was made, who made the request, when it was made, and additional details.

AWS Marketplace Catalog API information in CloudTrail

CloudTrail is enabled on your AWS account when you create the account. When activity occurs in the AWS Marketplace Catalog API, that activity is recorded in a CloudTrail event along with other AWS service events in **Event history**. You can view, search, and download recent events in your AWS account. For more information, see Viewing Events with CloudTrail Event History in the *AWS CloudTrail User Guide*.

For an ongoing record of events in your AWS account, create a trail. A trail enables CloudTrail to deliver log files to an Amazon S3 bucket. By default, when you create a trail in the console, the trail applies to all AWS Regions. The trail logs events from all AWS Regions in the AWS partition and delivers the log files to the Amazon S3 bucket that you specify. Additionally, you can configure other AWS services to further analyze and act upon the event data collected in CloudTrail logs. For more information, see:

- Overview for Creating a Trail
- CloudTrail Supported Services and Integrations
- Configuring Amazon SNS Notifications for CloudTrail
- Receiving CloudTrail Log Files from Multiple Regions
- Receiving CloudTrail Log Files from Multiple Accounts

All AWS Marketplace Catalog API actions are logged by CloudTrail and are documented in this API Reference. For example, calls to the `StartChangeSet`, `DescribeChangeSet`, and `ListChangeSets` API actions generate entries in the CloudTrail log files. Every event or log entry contains information about who generated the request. The identity information helps you determine the following:

- Whether the request was made with root or IAM user credentials.
- Whether the request was made with temporary security credentials for a role or federated user.
- Whether the request was made by another AWS service.

For more information, see CloudTrail userIdentity Element in the *AWS CloudTrail User Guide*.

Understanding AWS Marketplace catalog log file entries

A trail is a configuration that enables delivery of events as log files to an Amazon S3 bucket that you specify. CloudTrail log files contain one or more log entries. An event represents a single request from any source and includes information about the requested action, the date and time of the action, request parameters, and so on. CloudTrail log files are not an ordered stack trace of the public API calls, so they do not appear in any specific order.

Note
These examples have been formatted for improved readability. In a CloudTrail log file, all entries and events are concatenated into a single line. In addition, this example has been limited to a single AWS Marketplace Catalog API entry. In a real CloudTrail log file, you see entries and events from multiple AWS services.

The following example shows a AWS Marketplace Catalog API log entry that demonstrates the `ListEntities` action:


```
[
    {
        "eventVersion": "1.05",
        "userIdentity": {
            "type": "IAMUser",
            "principalId": "ABCDEFGHIJKLMNOP12345",
            "arn": "arn:aws:iam::123456789010:user/CloudTrailTestUser",
            "accountId": "123456789010",
            "accessKeyId": "ABCDEFGHIJKLMNOP1234",
            "userName": "CloudTrailTestUser"
        },
        "eventTime": "2019-10-17T21:49:23Z",
        "eventSource": "marketplacecatalog.amazonaws.com",
        "eventName": "ListEntities",
        "awsRegion": "us-east-1",
        "sourceIPAddress": "127.0.0.1",
        "userAgent": "PostmanRuntime/7.18.0",
        "requestParameters": {
            "catalog": "AWSMarketplace",
            "entityType": "EntityProduct",
            "sort": {
                "sortBy": "LastUpdateTimeInMillis",
                "sortOrder": "DESC"
            },
            "maxResults": 20
        },
        "responseElements": null,
        "requestID": "fEXAMPLE-cb3e-4e21-86fd-6b3EXAMPLEd1",
        "eventID": "7EXAMPLE-97d6-4139-91e3-01aEXAMPLE48",
        "readOnly": true,
        "eventType": "AwsApiCall",
        "recipientAccountId": "123456789010"
    }
]
```

Discovery API

Using the AWS Marketplace Catalog API service gives you access to manage the products that you create as a seller in AWS Marketplace, and to manage your private marketplace. However, you must use a separate API, the AWS Marketplace Discovery API, in the following scenarios:

- Get product information from AWS Marketplace
- Populate your private marketplace with products
- Create a custom view of AWS Marketplace for your customers where you show offerings from other sellers and offer value-added functionality

With the Discovery API, you can create browse and search functionality for the full AWS Marketplace catalog of products, with links to AWS Marketplace so customers can buy products. Or, you can access the full list of AWS Marketplace products to help populate your private marketplace.

Note
The AWS Marketplace Discovery API is a separate, private API from the AWS Marketplace Catalog API. You must request access to be able to use it. For more information, see the next section, Getting access to the Discovery API (p. 55).

Getting access to the Discovery API

The AWS Marketplace Discovery API can be a useful addition to the AWS Marketplace Catalog API. The Discovery API is a private API, only available to select customers. Calls to the Discovery API require an Integration ID that is provided to a you when you are approved for access.

To request access to the Discovery API, or get answers to other questions about the Discovery API, reach out to your existing AWS Marketplace contact. If you don't have a contact, or don't know who your contact is, you can reach out to the AWS Marketplace Seller Operations team.

After you have been approved for access, you will receive your Integration ID, as well as documentation for how to use the Discovery API.

Actions

The following actions are supported:

CancelChangeSet

Used to cancel an open change request. Must be sent before the status of the request changes to `APPLYING`, the final stage of completing your change request. You can describe a change during the 60-day request history retention period for API calls.

Request Syntax

```
PATCH /CancelChangeSet?catalog=Catalog&changeSetId=ChangeSetId HTTP/1.1
```

URI Request Parameters

The request uses the following URI parameters.

Catalog (p. 57)

Required. The catalog related to the request. Fixed value: `AWSMarketplace`.

Length Constraints: Minimum length of 1. Maximum length of 64.

Pattern: `^[a-zA-Z]+$`

Required: Yes

ChangeSetId (p. 57)

Required. The unique identifier of the `StartChangeSet` request that you want to cancel.

Length Constraints: Minimum length of 1. Maximum length of 255.

Pattern: `^[\w\-]+$`

Required: Yes

Request Body

The request does not have a request body.

Response Syntax

```
HTTP/1.1 200
Content-type: application/json

{
   "ChangeSetArn": "string",
   "ChangeSetId": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

ChangeSetArn (p. 57)

The ARN associated with the change set referenced in this request.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 2048.

Pattern: `^[a-zA-Z0-9:*/-]+$`

ChangeSetId (p. 57)

The unique identifier for the change set referenced in this request.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 255.

Pattern: `^[\w\-]+$`

Errors

For information about the errors that are common to all actions, see Common Errors (p. 93).

AccessDeniedException

Access is denied.

HTTP Status Code: 403

InternalServiceException

There was an internal service exception.

HTTP Status Code: 500

ResourceInUseException

The resource is currently in use.

HTTP Status Code: 423

ResourceNotFoundException

The specified resource wasn't found.

HTTP Status Code: 404

ThrottlingException

Too many requests.

HTTP Status Code: 429

ValidationException

An error occurred during validation.

HTTP Status Code: 422

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

DescribeChangeSet

Provides information about a given change set.

Request Syntax

```
GET /DescribeChangeSet?catalog=Catalog&changeSetId=ChangeSetId HTTP/1.1
```

URI Request Parameters

The request uses the following URI parameters.

Catalog (p. 60)

Required. The catalog related to the request. Fixed value: `AWSMarketplace`

Length Constraints: Minimum length of 1. Maximum length of 64.

Pattern: `^[a-zA-Z]+$`

Required: Yes

ChangeSetId (p. 60)

Required. The unique identifier for the `StartChangeSet` request that you want to describe the details for.

Length Constraints: Minimum length of 1. Maximum length of 255.

Pattern: `^[\w\-]+$`

Required: Yes

Request Body

The request does not have a request body.

Response Syntax

```
HTTP/1.1 200
Content-type: application/json

{
   "ChangeSet": [
      {
         "ChangeName": "string",
         "ChangeType": "string",
         "Details": "string",
         "Entity": {
            "Identifier": "string",
            "Type": "string"
         },
         "ErrorDetailList": [
            {
               "ErrorCode": "string",
```

```
                "ErrorMessage": "string"
            }
        ]
    }
  ],
  "ChangeSetArn": "string",
  "ChangeSetId": "string",
  "ChangeSetName": "string",
  "EndTime": "string",
  "FailureCode": "string",
  "FailureDescription": "string",
  "StartTime": "string",
  "Status": "string"
}
```


Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

ChangeSet (p. 60)

An array of `ChangeSummary` objects.

Type: Array of ChangeSummary (p. 83) objects

ChangeSetArn (p. 60)

The ARN associated with the unique identifier for the change set referenced in this request.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 2048.

Pattern: `^[a-zA-Z0-9:*/-]+$`

ChangeSetId (p. 60)

Required. The unique identifier for the change set referenced in this request.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 255.

Pattern: `^[\w\-]+$`

ChangeSetName (p. 60)

The optional name provided in the `StartChangeSet` request. If you do not provide a name, one is set by default.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 100.

Pattern: `^[\w\s+=.:@-]+$`

EndTime (p. 60)

The date and time, in ISO 8601 format (2018-02-27T13:45:22Z), the request transitioned to a terminal state. The change cannot transition to a different state. Null if the request is not in a terminal state.

Type: String

Length Constraints: Fixed length of 20.

Pattern: `^([\d]{4})\-(1[0-2]|0[1-9])\-(3[01]|0[1-9]|[12][\d])T(2[0-3]|[01][\d]):([0-5][\d]):([0-5][\d])Z$`

FailureCode (p. 60)

Returned if the change set is in `FAILED` status. Can be either `CLIENT_ERROR`, which means that there are issues with the request (see the `ErrorDetailList`), or `SERVER_FAULT`, which means that there is a problem in the system, and you should retry your request.

Type: String

Valid Values: `CLIENT_ERROR | SERVER_FAULT`

FailureDescription (p. 60)

Returned if there is a failure on the change set, but that failure is not related to any of the changes in the request.

Type: String

StartTime (p. 60)

The date and time, in ISO 8601 format (2018-02-27T13:45:22Z), the request started.

Type: String

Length Constraints: Fixed length of 20.

Pattern: `^([\d]{4})\-(1[0-2]|0[1-9])\-(3[01]|0[1-9]|[12][\d])T(2[0-3]|[01][\d]):([0-5][\d]):([0-5][\d])Z$`

Status (p. 60)

The status of the change request.

Type: String

Valid Values: `PREPARING | APPLYING | SUCCEEDED | CANCELLED | FAILED`

Errors

For information about the errors that are common to all actions, see Common Errors (p. 93).

AccessDeniedException

Access is denied.

HTTP Status Code: 403

InternalServiceException

There was an internal service exception.

HTTP Status Code: 500

ResourceNotFoundException

The specified resource wasn't found.

HTTP Status Code: 404

ThrottlingException

Too many requests.

HTTP Status Code: 429

ValidationException

An error occurred during validation.

HTTP Status Code: 422

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

DescribeEntity

Returns the metadata and content of the entity.

Request Syntax

```
GET /DescribeEntity?catalog=Catalog&entityId=EntityId HTTP/1.1
```

URI Request Parameters

The request uses the following URI parameters.

Catalog (p. 64)

Required. The catalog related to the request. Fixed value: `AWSMarketplace`

Length Constraints: Minimum length of 1. Maximum length of 64.

Pattern: `^[a-zA-Z]+$`

Required: Yes

EntityId (p. 64)

Required. The unique ID of the entity to describe.

Length Constraints: Minimum length of 1. Maximum length of 255.

Pattern: `^[\w\-]+$`

Required: Yes

Request Body

The request does not have a request body.

Response Syntax

```
HTTP/1.1 200
Content-type: application/json

{
   "Details": "string",
   "EntityArn": "string",
   "EntityIdentifier": "string",
   "EntityType": "string",
   "LastModifiedDate": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

Details (p. 64)

This stringified JSON object includes the details of the entity.

Type: String

Length Constraints: Minimum length of 2. Maximum length of 16384.

Pattern: `^[\s]*\{[\s\S]*\}[\s]*$`

EntityArn (p. 64)

The ARN associated to the unique identifier for the change set referenced in this request.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 2048.

Pattern: `^[a-zA-Z0-9:*/-]+$`

EntityIdentifier (p. 64)

The identifier of the entity, in the format of `EntityId@RevisionId`.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 255.

Pattern: `^[\w\-@]+$`

EntityType (p. 64)

The named type of the entity, in the format of `EntityType@Version`.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 255.

Pattern: `^[a-zA-Z]+$`

LastModifiedDate (p. 64)

The last modified date of the entity, in ISO 8601 format (2018-02-27T13:45:22Z).

Type: String

Errors

For information about the errors that are common to all actions, see Common Errors (p. 93).

AccessDeniedException

Access is denied.

HTTP Status Code: 403

InternalServiceException

There was an internal service exception.

HTTP Status Code: 500

ResourceNotFoundException

The specified resource wasn't found.

HTTP Status Code: 404

ResourceNotSupportedException

Currently, the specified resource is not supported.

HTTP Status Code: 415

ThrottlingException

Too many requests.

HTTP Status Code: 429

ValidationException

An error occurred during validation.

HTTP Status Code: 422

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

ListChangeSets

Returns the list of change sets owned by the account being used to make the call. You can filter this list by providing any combination of `entityId`, `ChangeSetName`, and status. If you provide more than one filter, the API operation applies a logical AND between the filters.

You can describe a change during the 60-day request history retention period for API calls.

Request Syntax

```
POST /ListChangeSets HTTP/1.1
Content-type: application/json

{
   "Catalog": "string",
   "FilterList": [
      {
         "Name": "string",
         "ValueList": [ "string" ]
      }
   ],
   "MaxResults": number,
   "NextToken": "string",
   "Sort": {
      "SortBy": "string",
      "SortOrder": "string"
   }
}
```

URI Request Parameters

The request does not use any URI parameters.

Request Body

The request accepts the following data in JSON format.

Catalog (p. 67)

The catalog related to the request. Fixed value: `AWSMarketplace`

Type: String

Length Constraints: Minimum length of 1. Maximum length of 64.

Pattern: `^[a-zA-Z]+$`

Required: Yes

FilterList (p. 67)

An array of filter objects.

Type: Array of Filter (p. 89) objects

Array Members: Minimum number of 1 item. Maximum number of 8 items.

Required: No

MaxResults (p. 67)

The maximum number of results returned by a single call. This value must be provided in the next call to retrieve the next set of results. By default, this value is 20.

Type: Integer

Valid Range: Minimum value of 1. Maximum value of 20.

Required: No

NextToken (p. 67)

The token value retrieved from a previous call to access the next page of results.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 2048.

Pattern: `^[\w+=.:@\-\/]$`

Required: No

Sort (p. 67)

An object that contains two attributes, `SortBy` and `SortOrder`.

Type: Sort (p. 90) object

Required: No

Response Syntax

```
HTTP/1.1 200
Content-type: application/json

{
   "ChangeSetSummaryList": [
      {
         "ChangeSetArn": "string",
         "ChangeSetId": "string",
         "ChangeSetName": "string",
         "EndTime": "string",
         "EntityIdList": [ "string" ],
         "FailureCode": "string",
         "StartTime": "string",
         "Status": "string"
      }
   ],
   "NextToken": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

ChangeSetSummaryList (p. 68)

Array of `ChangeSetSummaryListItem` objects.

Type: Array of ChangeSetSummaryListItem (p. 81) objects

NextToken (p. 68)

The value of the next token, if it exists. Null if there are no more results.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 2048.

Pattern: `^[\w+=.:@\-\/]$`

Errors

For information about the errors that are common to all actions, see Common Errors (p. 93).

AccessDeniedException

Access is denied.

HTTP Status Code: 403

InternalServiceException

There was an internal service exception.

HTTP Status Code: 500

ThrottlingException

Too many requests.

HTTP Status Code: 429

ValidationException

An error occurred during validation.

HTTP Status Code: 422

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

ListEntities

Provides the list of entities of a given type.

Request Syntax

```
POST /ListEntities HTTP/1.1
Content-type: application/json

{
   "Catalog": "string",
   "EntityType": "string",
   "FilterList": [
      {
         "Name": "string",
         "ValueList": [ "string" ]
      }
   ],
   "MaxResults": number,
   "NextToken": "string",
   "Sort": {
      "SortBy": "string",
      "SortOrder": "string"
   }
}
```

URI Request Parameters

The request does not use any URI parameters.

Request Body

The request accepts the following data in JSON format.

Catalog (p. 70)

The catalog related to the request. Fixed value: `AWSMarketplace`

Type: String

Length Constraints: Minimum length of 1. Maximum length of 64.

Pattern: `^[a-zA-Z]+$`

Required: Yes

EntityType (p. 70)

The type of entities to retrieve.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 255.

Pattern: `^[a-zA-Z]+$`

Required: Yes

FilterList (p. 70)

An array of filter objects. Each filter object contains two attributes, `filterName` and `filterValues`.

Type: Array of Filter (p. 89) objects

Array Members: Minimum number of 1 item. Maximum number of 8 items.

Required: No

MaxResults (p. 70)

Specifies the upper limit of the elements on a single page. If a value isn't provided, the default value is 20.

Type: Integer

Valid Range: Minimum value of 1. Maximum value of 20.

Required: No

NextToken (p. 70)

The value of the next token, if it exists. Null if there are no more results.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 2048.

Pattern: `^[\w+=.:@\-\/]$`

Required: No

Sort (p. 70)

An object that contains two attributes, `SortBy` and `SortOrder`.

Type: Sort (p. 90) object

Required: No

Response Syntax

```
HTTP/1.1 200
Content-type: application/json

{
   "EntitySummaryList": [
      {
         "EntityArn": "string",
         "EntityId": "string",
         "EntityType": "string",
         "LastModifiedDate": "string",
         "Name": "string",
         "Visibility": "string"
      }
   ],
   "NextToken": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

EntitySummaryList (p. 71)

Array of `EntitySummary` object.

Type: Array of EntitySummary (p. 86) objects

NextToken (p. 71)

The value of the next token if it exists. Null if there is no more result.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 2048.

Pattern: `^[\w+=.:@\-\/]$`

Errors

For information about the errors that are common to all actions, see Common Errors (p. 93).

AccessDeniedException

Access is denied.

HTTP Status Code: 403

InternalServiceException

There was an internal service exception.

HTTP Status Code: 500

ResourceNotFoundException

The specified resource wasn't found.

HTTP Status Code: 404

ThrottlingException

Too many requests.

HTTP Status Code: 429

ValidationException

An error occurred during validation.

HTTP Status Code: 422

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface

- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

StartChangeSet

This operation allows you to request changes for your entities. Within a single ChangeSet, you cannot start the same change type against the same entity multiple times. Additionally, when a ChangeSet is running, all the entities targeted by the different changes are locked until the ChangeSet has completed (either succeeded, cancelled, or failed). If you try to start a ChangeSet containing a change against an entity that is already locked, you will receive a `ResourceInUseException`.

For example, you cannot start the ChangeSet described in the example later in this topic, because it contains two changes to execute the same change type (`AddRevisions`) against the same entity (`entity-id@1`).

For more information about working with change sets, see Working with change sets.

Request Syntax

```
POST /StartChangeSet HTTP/1.1
Content-type: application/json

{
   "Catalog": "string",
   "ChangeSet": [
      {
         "ChangeName": "string",
         "ChangeType": "string",
         "Details": "string",
         "Entity": {
            "Identifier": "string",
            "Type": "string"
         }
      }
   ],
   "ChangeSetName": "string",
   "ClientRequestToken": "string"
}
```

URI Request Parameters

The request does not use any URI parameters.

Request Body

The request accepts the following data in JSON format.

Catalog (p. 74)

The catalog related to the request. Fixed value: `AWSMarketplace`

Type: String

Length Constraints: Minimum length of 1. Maximum length of 64.

Pattern: `^[a-zA-Z]+$`

Required: Yes

ChangeSet (p. 74)

Array of `change` object.

Type: Array of Change (p. 79) objects

Array Members: Minimum number of 1 item. Maximum number of 20 items.

Required: Yes

ChangeSetName (p. 74)

Optional case sensitive string of up to 100 ASCII characters. The change set name can be used to filter the list of change sets.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 100.

Pattern: `^[\w\s+=.:@-]+$`

Required: No

ClientRequestToken (p. 74)

A unique token to identify the request to ensure idempotency.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 36.

Pattern: `^[\w\-]+$`

Required: No

Response Syntax

```
HTTP/1.1 200
Content-type: application/json

{
   "ChangeSetArn": "string",
   "ChangeSetId": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

ChangeSetArn (p. 75)

The ARN associated to the unique identifier generated for the request.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 2048.

Pattern: `^[a-zA-Z0-9:*/-]+$`

ChangeSetId (p. 75)

Unique identifier generated for the request.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 255.

Pattern: `^[\w\-]+$`

Errors

For information about the errors that are common to all actions, see Common Errors (p. 93).

AccessDeniedException

Access is denied.

HTTP Status Code: 403

InternalServiceException

There was an internal service exception.

HTTP Status Code: 500

ResourceInUseException

The resource is currently in use.

HTTP Status Code: 423

ResourceNotFoundException

The specified resource wasn't found.

HTTP Status Code: 404

ServiceQuotaExceededException

The maximum number of open requests per account has been exceeded.

HTTP Status Code: 402

ThrottlingException

Too many requests.

HTTP Status Code: 429

ValidationException

An error occurred during validation.

HTTP Status Code: 422

Examples

Example

You cannot start this ChangeSet because it contains two changes to execute the same change type (`AddRevisions`) against the same entity (`entity-id@1`).

```
{
    "Catalog": "AWSMarketplace",
```

```
    "ChangeSetName": "Adding revisions to my test Data Product",
    "ChangeSet": [
        {
            "ChangeType": "AddRevisions",
            "Entity": {
                "Identifier": "entity-id@1",
                "Type": "DataProduct@1.0"
            },
            "Details": "{\"DataSetArn\": \"data-set-arn\", \"RevisionArns\": [\"revision-
arn\", \"revision-arn-2\"] }"
        },
        {
            "ChangeType": "AddRevisions",
            "Entity": {
                "Identifier": "entity-id@1",
                "Type": "DataProduct@1.0"
            },
            "Details": "{\"DataSetArn\": \"data-set-arn\", \"RevisionArns\": [\"revision-
arn3\"] }"
        }
    ]
}
```

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

Data Types

The AWS Marketplace Catalog Service API contains several data types that various actions use. This section describes each data type in detail.

Note
The order of each element in a data type structure is not guaranteed. Applications should not assume a particular order.

The following data types are supported:

Change

An object that contains the `ChangeType`, `Details`, and `Entity`.

Contents

ChangeName

Optional name for the change.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 72.

Pattern: `^[a-zA-Z]$`

Required: No

ChangeType

Change types are single string values that describe your intention for the change. Each change type is unique for each `EntityType` provided in the change's scope.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 255.

Pattern: `^[A-Z][\w]*$`

Required: Yes

Details

This object contains details specific to the change type of the requested change.

Type: String

Length Constraints: Minimum length of 2. Maximum length of 16384.

Pattern: `^[\s]*\{[\s\S]*\}[\s]*$`

Required: Yes

Entity

The entity to be changed.

Type: Entity (p. 85) object

Required: Yes

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for Ruby V3

ChangeSetSummaryListItem

A summary of a change set returned in a list of change sets when the `ListChangeSets` action is called.

Contents

ChangeSetArn

The ARN associated with the unique identifier for the change set referenced in this request.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 2048.

Pattern: `^[a-zA-Z0-9:*/-]+$`

Required: No

ChangeSetId

The unique identifier for a change set.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 255.

Pattern: `^[\w\-]+$`

Required: No

ChangeSetName

The non-unique name for the change set.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 100.

Pattern: `^[\w\s+=.:@-]+$`

Required: No

EndTime

The time, in ISO 8601 format (2018-02-27T13:45:22Z), when the change set was finished.

Type: String

Length Constraints: Fixed length of 20.

Pattern: `^([\d]{4})\-(1[0-2]|0[1-9])\-(3[01]|0[1-9]|[12][\d])T(2[0-3]|[01][\d]):([0-5][\d]):([0-5][\d])Z$`

Required: No

EntityIdList

This object is a list of entity IDs (string) that are a part of a change set. The entity ID list is a maximum of 20 entities. It must contain at least one entity.

Type: Array of strings

Length Constraints: Minimum length of 1. Maximum length of 255.

Pattern: `^[\w\-]+$`

Required: No

FailureCode

Returned if the change set is in `FAILED` status. Can be either `CLIENT_ERROR`, which means that there are issues with the request (see the `ErrorDetailList` of `DescribeChangeSet`), or `SERVER_FAULT`, which means that there is a problem in the system, and you should retry your request.

Type: String

Valid Values: `CLIENT_ERROR | SERVER_FAULT`

Required: No

StartTime

The time, in ISO 8601 format (2018-02-27T13:45:22Z), when the change set was started.

Type: String

Length Constraints: Fixed length of 20.

Pattern: `^([\d]{4})\-(1[0-2]|0[1-9])\-(3[01]|0[1-9]|[12][\d])T(2[0-3]|[01][\d]):([0-5][\d]):([0-5][\d])Z$`

Required: No

Status

The current status of the change set.

Type: String

Valid Values: `PREPARING | APPLYING | SUCCEEDED | CANCELLED | FAILED`

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for Ruby V3

ChangeSummary

This object is a container for common summary information about the change. The summary doesn't contain the whole change structure.

Contents

ChangeName

Optional name for the change.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 72.

Pattern: `^[a-zA-Z]$`

Required: No

ChangeType

The type of the change.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 255.

Pattern: `^[A-Z][\w]*$`

Required: No

Details

This object contains details specific to the change type of the requested change.

Type: String

Length Constraints: Minimum length of 2. Maximum length of 16384.

Pattern: `^[\s]*\{[\s\S]*\}[\s]*$`

Required: No

Entity

The entity to be changed.

Type: Entity (p. 85) object

Required: No

ErrorDetailList

An array of `ErrorDetail` objects associated with the change.

Type: Array of ErrorDetail (p. 88) objects

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for Ruby V3

Entity

An entity contains data that describes your product, its supported features, and how it can be used or launched by your customer.

Contents

Identifier

The identifier for the entity.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 255.

Pattern: `^[\w\-@]+$`

Required: No

Type

The type of entity.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 255.

Pattern: `^[a-zA-Z]+$`

Required: Yes

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for Ruby V3

EntitySummary

This object is a container for common summary information about the entity. The summary doesn't contain the whole entity structure, but it does contain information common across all entities.

Contents

EntityArn

The ARN associated with the unique identifier for the entity.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 2048.

Pattern: `^[a-zA-Z0-9:*/-]+$`

Required: No

EntityId

The unique identifier for the entity.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 255.

Pattern: `^[\w\-]+$`

Required: No

EntityType

The type of the entity.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 255.

Pattern: `^[a-zA-Z]+$`

Required: No

LastModifiedDate

The last time the entity was published, using ISO 8601 format (2018-02-27T13:45:22Z).

Type: String

Required: No

Name

The name for the entity. This value is not unique. It is defined by the seller.

Type: String

Required: No

Visibility

The visibility status of the entity to buyers. This value can be `Public` (everyone can view the entity), `Limited` (the entity is visible to limited accounts only), or `Restricted` (the entity was published and then unpublished and only existing buyers can view it).

Type: String

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for Ruby V3

ErrorDetail

Details about the error.

Contents

ErrorCode

The error code that identifies the type of error.

Type: String

Required: No

ErrorMessage

The message for the error.

Type: String

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for Ruby V3

Filter

A filter object, used to optionally filter results from calls to the `ListEntities` and `ListChangeSets` actions.

Contents

Name

For `ListEntities`, the supported value for this is an `EntityId`.

For `ListChangeSets`, the supported values are as follows:

Type: String

Length Constraints: Minimum length of 1. Maximum length of 255.

Pattern: `^[a-zA-Z]+$`

Required: No

ValueList

`ListEntities` - This is a list of unique `EntityIds`.

`ListChangeSets` - The supported filter names and associated `ValueLists` is as follows:

- `ChangeSetName` - The supported `ValueList` is a list of non-unique `ChangeSetNames`. These are defined when you call the `StartChangeSet` action.
- `Status` - The supported `ValueList` is a list of statuses for all change set requests.
- `EntityId` - The supported `ValueList` is a list of unique `EntityIds`.
- `BeforeStartTime` - The supported `ValueList` is a list of all change sets that started before the filter value.
- `AfterStartTime` - The supported `ValueList` is a list of all change sets that started after the filter value.
- `BeforeEndTime` - The supported `ValueList` is a list of all change sets that ended before the filter value.
- `AfterEndTime` - The supported `ValueList` is a list of all change sets that ended after the filter value.

Type: Array of strings

Array Members: Minimum number of 1 item. Maximum number of 10 items.

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for Ruby V3

Sort

An object that contains two attributes, `SortBy` and `SortOrder`.

Contents

SortBy

For `ListEntities`, supported attributes include `LastModifiedDate` (default), `Visibility`, `EntityId`, and `Name`.

For `ListChangeSets`, supported attributes include `StartTime` and `EndTime`.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 255.

Pattern: `^[a-zA-Z]+$`

Required: No

SortOrder

The sorting order. Can be `ASCENDING` or `DESCENDING`. The default value is `DESCENDING`.

Type: String

Valid Values: `ASCENDING | DESCENDING`

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for Ruby V3

Common Parameters

The following list contains the parameters that all actions use for signing Signature Version 4 requests with a query string. Any action-specific parameters are listed in the topic for that action. For more information about Signature Version 4, see Signature Version 4 Signing Process in the *Amazon Web Services General Reference*.

Action

The action to be performed.

Type: string

Required: Yes

Version

The API version that the request is written for, expressed in the format YYYY-MM-DD.

Type: string

Required: Yes

X-Amz-Algorithm

The hash algorithm that you used to create the request signature.

Condition: Specify this parameter when you include authentication information in a query string instead of in the HTTP authorization header.

Type: string

Valid Values: `AWS4-HMAC-SHA256`

Required: Conditional

X-Amz-Credential

The credential scope value, which is a string that includes your access key, the date, the region you are targeting, the service you are requesting, and a termination string ("aws4_request"). The value is expressed in the following format: *access_key*/*YYYYMMDD*/*region*/*service*/aws4_request.

For more information, see Task 2: Create a String to Sign for Signature Version 4 in the *Amazon Web Services General Reference*.

Condition: Specify this parameter when you include authentication information in a query string instead of in the HTTP authorization header.

Type: string

Required: Conditional

X-Amz-Date

The date that is used to create the signature. The format must be ISO 8601 basic format (YYYYMMDD'T'HHMMSS'Z'). For example, the following date time is a valid X-Amz-Date value: `20120325T120000Z`.

Condition: X-Amz-Date is optional for all requests; it can be used to override the date used for signing requests. If the Date header is specified in the ISO 8601 basic format, X-Amz-Date is

not required. When X-Amz-Date is used, it always overrides the value of the Date header. For more information, see Handling Dates in Signature Version 4 in the *Amazon Web Services General Reference*.

Type: string

Required: Conditional

X-Amz-Security-Token

The temporary security token that was obtained through a call to AWS Security Token Service (AWS STS). For a list of services that support temporary security credentials from AWS Security Token Service, go to AWS Services That Work with IAM in the *IAM User Guide*.

Condition: If you're using temporary security credentials from the AWS Security Token Service, you must include the security token.

Type: string

Required: Conditional

X-Amz-Signature

Specifies the hex-encoded signature that was calculated from the string to sign and the derived signing key.

Condition: Specify this parameter when you include authentication information in a query string instead of in the HTTP authorization header.

Type: string

Required: Conditional

X-Amz-SignedHeaders

Specifies all the HTTP headers that were included as part of the canonical request. For more information about specifying signed headers, see Task 1: Create a Canonical Request For Signature Version 4 in the *Amazon Web Services General Reference*.

Condition: Specify this parameter when you include authentication information in a query string instead of in the HTTP authorization header.

Type: string

Required: Conditional

Common Errors

This section lists the errors common to the API actions of all AWS services. For errors specific to an API action for this service, see the topic for that API action.

AccessDeniedException

You do not have sufficient access to perform this action.

HTTP Status Code: 400

IncompleteSignature

The request signature does not conform to AWS standards.

HTTP Status Code: 400

InternalFailure

The request processing has failed because of an unknown error, exception or failure.

HTTP Status Code: 500

InvalidAction

The action or operation requested is invalid. Verify that the action is typed correctly.

HTTP Status Code: 400

InvalidClientTokenId

The X.509 certificate or AWS access key ID provided does not exist in our records.

HTTP Status Code: 403

InvalidParameterCombination

Parameters that must not be used together were used together.

HTTP Status Code: 400

InvalidParameterValue

An invalid or out-of-range value was supplied for the input parameter.

HTTP Status Code: 400

InvalidQueryParameter

The AWS query string is malformed or does not adhere to AWS standards.

HTTP Status Code: 400

MalformedQueryString

The query string contains a syntax error.

HTTP Status Code: 404

MissingAction

The request is missing an action or a required parameter.

HTTP Status Code: 400

MissingAuthenticationToken

The request must contain either a valid (registered) AWS access key ID or X.509 certificate.

HTTP Status Code: 403

MissingParameter

A required parameter for the specified action is not supplied.

HTTP Status Code: 400

NotAuthorized

You do not have permission to perform this action.

HTTP Status Code: 400

OptInRequired

The AWS access key ID needs a subscription for the service.

HTTP Status Code: 403

RequestExpired

The request reached the service more than 15 minutes after the date stamp on the request or more than 15 minutes after the request expiration date (such as for pre-signed URLs), or the date stamp on the request is more than 15 minutes in the future.

HTTP Status Code: 400

ServiceUnavailable

The request has failed due to a temporary failure of the server.

HTTP Status Code: 503

ThrottlingException

The request was denied due to request throttling.

HTTP Status Code: 400

ValidationError

The input fails to satisfy the constraints specified by an AWS service.

HTTP Status Code: 400

Document history

The following table describes the documentation for this release of the *AWS Marketplace Catalog API Reference*.

update-history-change	update-history-description	update-history-date
Support for managing seller products (p. 95)	Added the ability to manage AMI and container products programmatically. See Working with seller products.	March 26, 2021
Support for managing private marketplaces (p. 95)	Added the ability to create and maintain private marketplaces for AWS Organizations programmatically. See Working with a private marketplace.	December 3, 2020
The AWS Marketplace Discovery API is now available (p. 95)	The Discovery API provides programmatic access to find products in the AWS Marketplace. For details, see Discovery API.	September 30, 2020
The AWS Marketplace Catalog API is now generally available (p. 95)	This service provides an API interface for approved providers to programmatically access the self-service publishing capabilities on the AWS Marketplace Management Portal.	November 12, 2019