



Reference

# AWS Marketplace Catalog API



# AWS Marketplace Catalog API: Reference

Copyright © 2024 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon's trademarks and trade dress may not be used in connection with any product or service that is not Amazon's, in any manner that is likely to cause confusion among customers, or in any manner that disparages or discredits Amazon. All other trademarks not owned by Amazon are the property of their respective owners, who may or may not be affiliated with, connected to, or sponsored by Amazon.

# Table of Contents

<b>AWS Marketplace Catalog API .....</b>	<b>1</b>
Supported AWS Regions .....	1
Catalog API entities .....	1
General structure .....	2
Type versioning .....	2
Identifier .....	2
Facets .....	3
Entity type .....	4
Working with change sets .....	4
Making multiple change requests simultaneously .....	6
Invoking multiple change types in one change set .....	7
Working with the Details attribute (Legacy) .....	10
Using DescribeEntity .....	12
API access control .....	17
Allowing actions with AWS managed policies .....	18
Allowing actions on all resources .....	19
Allowing actions on specific resources .....	20
Allowing actions with specific ChangeType condition key .....	21
Allowing actions with specific aws:ResourceTag condition key .....	23
Creating a custom IAM role .....	25
Managing tags on resources .....	28
Managing tags when requesting changes to resources .....	31
Granting permission to manage tags on resources .....	32
Granting permission to manage tags on resources only when those resources have specific tags .....	33
Granting permission to create entities and change sets only with tags .....	34
Service quotas .....	35
<b>Working with seller products .....</b>	<b>37</b>
Create a product .....	38
Update product details .....	41
Add pricing dimensions .....	50
Update pricing dimensions .....	56
Restrict pricing dimensions .....	60
Update targeting configuration .....	64

---

Update product visibility .....	67
Publish a product .....	72
Find your product ID .....	74
Find a product based on product title .....	75
Find a product based on last modified date .....	76
Find a product based on product visibility .....	77
Find a product based on product title, last modified date, and product visibility .....	78
Get additional details about a product .....	80
Change set status and errors .....	80
Working with single AMI products .....	88
Add a new version .....	89
Update version information .....	96
Restrict a version .....	100
Update future AWS Region support .....	102
Add a supported AWS Region .....	105
Restrict an AWS Region .....	108
Add a new instance type .....	110
Restrict an instance type .....	113
Working with container-based products .....	117
Add a new version .....	118
Update the visibility for an Amazon EKS add-on .....	133
Create repositories and resources .....	137
Update version information .....	139
Restrict a version .....	150
Working with SaaS products .....	152
Add delivery options .....	153
Update delivery options .....	158
Update delivery option visibility .....	164
<b>Working with offers .....</b>	<b>168</b>
Create a new offer .....	169
Create a replacement offer .....	173
Update offer information .....	175
Update targeting configuration .....	179
Update refund policy .....	183
Update legal resources .....	186
Update pricing .....	190

---

Update the discoverability of the offer .....	208
Define the expiration date of agreements created using the offer .....	211
Update payment schedule details .....	217
Modify renewal options .....	221
Publish an offer .....	224
Describe existing offer details .....	227
<b>Working with Resale Authorizations .....</b>	<b>232</b>
Resale Authorization prerequisites .....	233
Create a new Resale Authorization .....	233
Update buyer targeting .....	238
Update availability .....	241
Update the validity of a future dated agreement .....	244
Update legal resources .....	247
Update pricing .....	252
Update payment schedule .....	261
Update Resale Authorization details .....	265
Restrict a Resale Authorization .....	269
Release a Resale Authorization and make it visible to a Channel Partner .....	271
Describe an existing Resale Authorization .....	274
<b>Working with CPPOs .....</b>	<b>279</b>
CPPO prerequisites .....	280
Create a CPPO .....	280
Create a channel partner private replacement offer .....	286
Update markup .....	289
Update targeting configuration .....	292
Update legal resources .....	296
Update the discoverability of the CPPO .....	300
Define the expiration date of agreements .....	302
Update pricing .....	308
Update payment schedule details .....	319
Publish the CPPO .....	323
Define an existing CPPO .....	325
<b>Working with a private marketplace .....</b>	<b>330</b>
Creating a private marketplace .....	331
Changing the branding of a private marketplace experience .....	335
Enabling or disabling a private marketplace experience .....	338

---

Enabling or disabling user requests .....	339
Getting a list of products in a private marketplace experience .....	340
Adding or removing products from a private marketplace .....	343
Finding products .....	344
Working with private marketplaces for AWS Organizations .....	344
Associating principals to experiences .....	346
Archiving and reactivating a private marketplace experience .....	348
Errors in the private marketplace API .....	350
Entity types defined by private marketplace .....	353
<b>Working with AWS RAM to share resources .....</b>	<b>354</b>
Prerequisites for sharing AWS Marketplace entities .....	354
Sharing an AWS Marketplace entity .....	354
Differences between sharing an entity through AWS RAM and the AWS Marketplace Catalog API .....	356
Attach read-only policy to your resource .....	357
Attach read and write resource policy to your resource .....	358
View resource policy set on your resource .....	359
Delete resource policy on your resource .....	360
View all resources owned by you and shared with you .....	360
<b>Logging and notifications .....</b>	<b>362</b>
Logging with AWS CloudTrail .....	362
AWS Marketplace Catalog API information in CloudTrail .....	362
Understanding AWS Marketplace catalog log file entries .....	363
Amazon EventBridge notifications .....	364
<b>AWS Marketplace Discovery API .....</b>	<b>366</b>
Getting access to the Discovery API .....	366
Logging Discovery API calls with CloudTrail .....	367
Discovery API information in CloudTrail .....	367
Understanding Discovery API log file entries .....	368
Release notes for AWS Marketplace Discovery API .....	369
Discovery API release notes for 2022 .....	370
<b>Actions .....</b>	<b>371</b>
CancelChangeSet .....	372
Request Syntax .....	372
URI Request Parameters .....	372
Request Body .....	372

---

Response Syntax .....	372
Response Elements .....	373
Errors .....	373
See Also .....	374
DeleteResourcePolicy .....	376
Request Syntax .....	376
URI Request Parameters .....	376
Request Body .....	376
Response Syntax .....	376
Response Elements .....	376
Errors .....	376
See Also .....	377
DescribeChangeSet .....	379
Request Syntax .....	379
URI Request Parameters .....	379
Request Body .....	379
Response Syntax .....	379
Response Elements .....	380
Errors .....	383
See Also .....	384
DescribeEntity .....	385
Request Syntax .....	385
URI Request Parameters .....	385
Request Body .....	385
Response Syntax .....	385
Response Elements .....	386
Errors .....	387
See Also .....	388
GetResourcePolicy .....	390
Request Syntax .....	390
URI Request Parameters .....	390
Request Body .....	390
Response Syntax .....	390
Response Elements .....	390
Errors .....	391
See Also .....	392

---

ListChangeSets .....	393
Request Syntax .....	393
URI Request Parameters .....	393
Request Body .....	393
Response Syntax .....	395
Response Elements .....	395
Errors .....	396
See Also .....	396
ListEntities .....	398
Request Syntax .....	398
URI Request Parameters .....	398
Request Body .....	398
Response Syntax .....	401
Response Elements .....	402
Errors .....	403
See Also .....	404
ListTagsForResource .....	405
Request Syntax .....	405
URI Request Parameters .....	405
Request Body .....	405
Response Syntax .....	405
Response Elements .....	406
Errors .....	406
See Also .....	407
PutResourcePolicy .....	409
Request Syntax .....	409
URI Request Parameters .....	409
Request Body .....	409
Response Syntax .....	410
Response Elements .....	410
Errors .....	410
See Also .....	411
StartChangeSet .....	412
Request Syntax .....	412
URI Request Parameters .....	413
Request Body .....	413

Response Syntax .....	415
Response Elements .....	415
Errors .....	415
Examples .....	417
See Also .....	417
TagResource .....	419
Request Syntax .....	419
URI Request Parameters .....	419
Request Body .....	419
Response Syntax .....	420
Response Elements .....	420
Errors .....	420
See Also .....	421
UntagResource .....	422
Request Syntax .....	422
URI Request Parameters .....	422
Request Body .....	422
Response Syntax .....	423
Response Elements .....	423
Errors .....	423
See Also .....	424
<b>Data Types .....</b>	<b>425</b>
AmiProductEntityIdFilter .....	428
Contents .....	428
See Also .....	428
AmiProductFilters .....	429
Contents .....	429
See Also .....	429
AmiProductLastModifiedDateFilter .....	431
Contents .....	431
See Also .....	431
AmiProductLastModifiedDateFilterDateRange .....	432
Contents .....	432
See Also .....	432
AmiProductSort .....	434
Contents .....	434

---

See Also .....	434
AmiProductSummary .....	435
Contents .....	435
See Also .....	435
AmiProductTitleFilter .....	436
Contents .....	436
See Also .....	436
AmiProductVisibilityFilter .....	438
Contents .....	438
See Also .....	438
Change .....	439
Contents .....	439
See Also .....	440
ChangeSetSummaryListItem .....	441
Contents .....	441
See Also .....	443
ChangeSummary .....	444
Contents .....	444
See Also .....	445
ContainerProductEntityIdFilter .....	446
Contents .....	446
See Also .....	446
ContainerProductFilters .....	447
Contents .....	447
See Also .....	447
ContainerProductLastModifiedDateFilter .....	449
Contents .....	449
See Also .....	449
ContainerProductLastModifiedDateFilterDateRange .....	450
Contents .....	450
See Also .....	450
ContainerProductSort .....	452
Contents .....	452
See Also .....	452
ContainerProductSummary .....	453
Contents .....	453

---

See Also .....	453
ContainerProductTitleFilter .....	454
Contents .....	454
See Also .....	454
ContainerProductVisibilityFilter .....	456
Contents .....	456
See Also .....	456
DataProductEntityIdFilter .....	457
Contents .....	457
See Also .....	457
DataProductFilters .....	458
Contents .....	458
See Also .....	458
DataProductLastModifiedDateFilter .....	460
Contents .....	460
See Also .....	460
DataProductLastModifiedDateFilterDateRange .....	461
Contents .....	461
See Also .....	461
DataProductSort .....	463
Contents .....	463
See Also .....	463
DataProductSummary .....	464
Contents .....	464
See Also .....	464
DataProductTitleFilter .....	465
Contents .....	465
See Also .....	465
DataProductVisibilityFilter .....	467
Contents .....	467
See Also .....	467
Entity .....	468
Contents .....	468
See Also .....	468
EntitySummary .....	470
Contents .....	470

---

See Also .....	472
EntityTypeFilters .....	474
Contents .....	474
See Also .....	475
EntityTypeSort .....	476
Contents .....	476
See Also .....	477
ErrorDetail .....	478
Contents .....	478
See Also .....	478
Filter .....	479
Contents .....	479
See Also .....	480
OfferAvailabilityEndDateFilter .....	481
Contents .....	481
See Also .....	481
OfferAvailabilityEndDateFilterDateRange .....	482
Contents .....	482
See Also .....	482
OfferBuyerAccountsFilter .....	484
Contents .....	484
See Also .....	484
OfferEntityIdFilter .....	485
Contents .....	485
See Also .....	485
OfferFilters .....	486
Contents .....	486
See Also .....	488
OfferLastModifiedDateFilter .....	489
Contents .....	489
See Also .....	489
OfferLastModifiedDateFilterDateRange .....	490
Contents .....	490
See Also .....	490
OfferNameFilter .....	492
Contents .....	492

See Also .....	492
OfferProductIdFilter .....	494
Contents .....	494
See Also .....	494
OfferReleaseDateFilter .....	495
Contents .....	495
See Also .....	495
OfferReleaseDateFilterDateRange .....	496
Contents .....	496
See Also .....	496
OfferResaleAuthorizationIdFilter .....	498
Contents .....	498
See Also .....	498
OfferSort .....	499
Contents .....	499
See Also .....	499
OfferStateFilter .....	500
Contents .....	500
See Also .....	500
OfferSummary .....	501
Contents .....	501
See Also .....	503
OfferTargetingFilter .....	504
Contents .....	504
See Also .....	504
ResaleAuthorizationAvailabilityEndDateFilter .....	505
Contents .....	505
See Also .....	505
ResaleAuthorizationAvailabilityEndDateFilterDateRange .....	506
Contents .....	506
See Also .....	506
ResaleAuthorizationCreatedDateFilter .....	508
Contents .....	508
See Also .....	508
ResaleAuthorizationCreatedDateFilterDateRange .....	509
Contents .....	509

---

See Also .....	509
ResaleAuthorizationEntityIdFilter .....	511
Contents .....	511
See Also .....	511
ResaleAuthorizationFilters .....	512
Contents .....	512
See Also .....	514
ResaleAuthorizationLastModifiedDateFilter .....	515
Contents .....	515
See Also .....	515
ResaleAuthorizationLastModifiedDateFilterDateRange .....	516
Contents .....	516
See Also .....	516
ResaleAuthorizationManufacturerAccountIdFilter .....	518
Contents .....	518
See Also .....	518
ResaleAuthorizationManufacturerLegalNameFilter .....	520
Contents .....	520
See Also .....	520
ResaleAuthorizationNameFilter .....	522
Contents .....	522
See Also .....	522
ResaleAuthorizationOfferExtendedStatusFilter .....	524
Contents .....	524
See Also .....	524
ResaleAuthorizationProductIdFilter .....	525
Contents .....	525
See Also .....	525
ResaleAuthorizationProductNameFilter .....	527
Contents .....	527
See Also .....	527
ResaleAuthorizationResellerAccountIdFilter .....	529
Contents .....	529
See Also .....	529
ResaleAuthorizationResellerLegalNameFilter .....	531
Contents .....	531

---

See Also .....	531
ResaleAuthorizationSort .....	533
Contents .....	533
See Also .....	533
ResaleAuthorizationStatusFilter .....	534
Contents .....	534
See Also .....	534
ResaleAuthorizationSummary .....	535
Contents .....	535
See Also .....	538
SaaSProductEntityIdFilter .....	539
Contents .....	539
See Also .....	539
SaaSProductFilters .....	540
Contents .....	540
See Also .....	540
SaaSProductLastModifiedDateFilter .....	542
Contents .....	542
See Also .....	542
SaaSProductLastModifiedDateFilterDateRange .....	543
Contents .....	543
See Also .....	543
SaaSProductSort .....	545
Contents .....	545
See Also .....	545
SaaSProductSummary .....	546
Contents .....	546
See Also .....	546
SaaSProductTitleFilter .....	547
Contents .....	547
See Also .....	547
SaaSProductVisibilityFilter .....	549
Contents .....	549
See Also .....	549
Sort .....	550
Contents .....	550

---

See Also .....	550
Tag .....	552
Contents .....	552
See Also .....	552
<b>Common Parameters .....</b>	<b>553</b>
<b>Common Errors .....</b>	<b>556</b>
<b>Document history .....</b>	<b>558</b>

# 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 in 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 API actions described in [Actions](#), or you can use an AWS SDK to access an API that's tailored to the programming language or platform that you're using. For more information about AWS application development, see [Getting Started with AWS](#). For more information about using AWS SDKs, 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 doesn't require that you 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](#)). 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).

## Entity type

The entity type defines what the entity represents. An entity can be a seller product in AWS Marketplace or a private marketplace experience. For more information, see [Working with seller products](#) and [Working with a private marketplace](#).

## Working with change sets

When using the Catalog API, requests are created and updated through entities and completed by using change requests. Every change specifies the entity to be changed, the type of change to be performed, and details of the change. The type of change to be performed is called a `ChangeType`. A collection of `ChangeTypes` is called a `ChangeSet`.

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. For more information, see [Working with seller products](#) and [Working with a private marketplace](#).
- `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.
  - `CANCELLED` – 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.

The following is an example of the DescribeChangeSet response.

```
{
  "ChangeSet":
  [
    {
      "ChangeName": "myChangeName",
      "ChangeType": "UpdateInformation",
      "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\" ]}",
      "DetailsDocument":
      {
        "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"
        ]
      },
      "Entity":
      {
        "Identifier": "example1-abcd-1234-5ef6-7890abcdef12",
        "Type": "AmiProduct@1.0"
      },
      "ErrorDetailList":
      []
    }
  ],
  "ChangeSetArn": "arn:aws:aws-marketplace:[exampleARN]",
  "ChangeSetId": "example123456789012abcdef",
  "ChangeSetName": "myChangeSetName",
  "EndTime": "2023-03-03T00:00:00Z",
  "FailureCode": null,
  "FailureDescription": null,
}
```

```
"StartTime": "2023-03-02T00:00:00Z",  
"Status": "SUCCEEDED"  
}
```

### Note

When polling or working with change sets programmatically, you must adhere to the service limits. For more information, see [Service quotas](#).

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

Within a **single change set**, you can bundle all change types and they are run together. Catalog API is built to make multiple changes simultaneously to provide the best performance. Sellers and Channel Partners can invoke changes with multiple `ChangeTypes` bundled into a `ChangeSet`. You can invoke multiple changes on single or different entities in the same `ChangeSet`. Catalog API evaluates which order the changes need to be applied and makes those changes.

However, if the requests are made as **separate change sets**, AWS Marketplace can't initiate 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.

## Invoking multiple change types in one change set

You can use the Catalog API to combine and chain up to 20 changes in one `StartChangeSet` request targeting one or multiple different entities.

A typical use case is to create a `SaaSProduct@1.0` draft product, an `Offer@1.0` draft offer, and also filling in the metadata information of the product and offer. This is done by including the following four change types in one change set:

- `CreateProduct` on `SaaSProduct@1.0`

Specify the `ChangeName` parameter. Then, the product created in this change type can be referenced in the same change set by subsequent changes.

For example, `CreateProductChange`.

- `UpdateInformation` on the `SaaSProduct@1.0` created in the same change set

In the `Entity.Identifier` field, you can refer to the product created by `CreateProduct` change type using the change name in this format:

```
#{ChangeName}.Entity.Identifier
```

For example, `#{CreateProductChange}.Entity.Identifier`.

- `CreateOffer` on `Offer@1.0` tied to the `SaaSProduct@1.0` created in the same change set

Specify the `ChangeName` parameter. Then, the product created in this change type can be referenced in the same change set by subsequent changes. For example, `CreateOfferChange`.

For the `ProductId` parameter in the payload of `CreateOffer` change type, you can also refer to the SaaS product created in `CreateProduct` change type by using `#{ChangeName}.Entity.Identifier` syntax.

For example, `{"ProductId": "$CreateProductChange.Entity.Identifier"}`.

- `UpdateInformation` on the `Offer@1.0` created in the same change set

In the `Entity.Identifier` field, you can refer to the offer created by the `CreateOffer` change type using the change name in this format:

`#{ChangeName}.Entity.Identifier`

For example, `$CreateOfferChange.Entity.Identifier`.

The following is an example of a combined change set.

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

{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "CreateProduct",
      "Entity": {
        "Type": "SaaSProduct@1.0"
      },
      "ChangeName": "CreateProductChange",
      "DetailsDocument": {}
    },
    {
      "ChangeType": "UpdateInformation",
      "Entity": {
        "Type": "SaaSProduct@1.0",
        "Identifier": "$CreateProductChange.Entity.Identifier"
      },
      "DetailsDocument": {
        "ProductTitle": "My product title",
        "ShortDescription": "My product short description",
        "LongDescription": "My product long description",
        "Sku": "123example456",
      }
    }
  ]
}
```

```

    "Highlights": [
      "Highlight 1",
      "Highlight 2"
    ],
    "LogoUrl": "https://example-logos.s3.amazonaws.com/
c23fcEXAMPLEe1ad2c0d435cb4f944",
    "SupportDescription": "My product support description",
    "SearchKeywords": [
      "Sample product"
    ],
    "Categories": [
      "Security",
      "Streaming solutions"
    ]
  }
},
{
  "ChangeType": "CreateOffer",
  "Entity": {
    "Type": "Offer@1.0"
  },
  "ChangeName": "CreateOfferChange",
  "DetailsDocument": {
    "ProductId": "$CreateProductChange.Entity.Identifier"
  }
},
{
  "ChangeType": "UpdateInformation",
  "Entity": {
    "Type": "Offer@1.0",
    "Identifier": "$CreateOfferChange.Entity.Identifier"
  },
  "DetailsDocument": {
    "Name": "My offer created together with SaaSProduct",
    "Description": "My offer created together with SaaSProduct in the same Catalog
API change set"
  }
}
]
}

```

## Working with the Details attribute (Legacy)

### Note

This section describes the legacy Details attribute in your change request, which requires additional formatting for your change details. We recommend using the alternative DetailsDocument attribute. It doesn't require additional formatting and the change details don't need to be changed. For examples of the DetailsDocument attribute, see [Working with seller products](#) and [Working with a private marketplace](#).

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\": [ \"111122223333.dkr.ecr.us-east-1.amazonaws.com/some-seller-prefix/my-repo-1:some-tag\" ], \"Compatibility\": { \"Platform\": \"Linux\" } }, { \"Id\": \"example4-2222-cccc-2222-cccccccccc\", \"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\":
\ "111122223333.dkr.ecr.us-east-1.amazonaws.com/some-seller-prefix/my-repo-1\", \ "Type\":
\ "ECR\"]] }",
  "DetailsDocument":
  {
    "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":
            [
              "111122223333.dkr.ecr.us-east-1.amazonaws.com/some-seller-prefix/my-
repo-1:some-tag"
            ],
            "Compatibility":
            {
              "Platform": "Linux"
            }
          }
        ],
        "DeliveryOptions":
        [
          {
            "Id": "example4-2222-cccc-2222-cccccccccccc",
            "Type": "ElasticContainerRegistry",
            "SourceId": "example3-1111-bbbb-5ef6-7890abcdef12",
            "Title": "New delivery option 1",
            "ShortDescription": "Delivery option 1",

```

```
        "isRecommended": false,
        "Compatibility":
        {
            "AWServices":
            [
                "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": "111122223333.dkr.ecr.us-east-1.amazonaws.com/some-seller-prefix/my-
repo-1",
  "Type": "ECR"
}
]
```

### Note

The `DetailsDocument` attribute contains the entity details as a JSON object. The legacy `Details` attribute contains the same JSON object as a string.

## API access control

You can use the AWS Marketplace Catalog API to manage [a seller product in AWS Marketplace](#) or an [experience in a private marketplace](#). However, first make sure your user or role can access the API functionality that you want to call.

Use AWS Identity and Access Management (IAM) to create users and roles and assign policies that grant limited permissions to end users. The policies define the actions that the user or role can take on your resources through the AWS Marketplace Catalog API.

For example, you can define roles such as engineering, marketing, and pricing. Then, you can add a user in your organization to the engineering role. In that role, they might be granted permissions to initiate a change request to publish a new version of your seller product. However, the engineering role doesn't allow the user to 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*.

You can use AWS managed policies, or you can create your own IAM policies to have more granular control than what's available in AWS managed policies. For details about these approaches, see the following topics.

## Topics

- [Allowing actions with AWS managed policies](#)
- [Allowing actions on all resources](#)
- [Allowing actions on specific resources](#)
- [Allowing actions with specific ChangeType condition key](#)
- [Allowing actions with specific aws:ResourceTag condition key](#)
- [Creating a custom IAM role](#)
- [Managing tags on resources](#)
- [Managing tags when requesting changes to resources](#)
- [Granting permission to manage tags on resources](#)
- [Granting permission to manage tags on resources only when those resources have specific tags](#)
- [Granting permission to create entities and change sets only with tags](#)

## Allowing actions with AWS managed policies

You can use policies that are managed by AWS to grant permissions to your user or role.

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 information, see [Controlling access to AWS Marketplace Management Portal](#), [Policies and permissions for AWS Marketplace sellers](#), and [AWS managed policies for AWS Marketplace sellers](#) in the *AWS Marketplace Seller Guide*.

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. For more information, see [Controlling access to AWS Marketplace subscriptions](#), [Creating a private marketplace administrator](#), and [AWS managed policies for AWS Marketplace buyers](#) in the *AWS Marketplace Buyer Guide*.

Alternatively, you can create your own IAM policies to have more granular control than is available in AWS managed policies. Use the following topics to create your own IAM policies.

## Allowing actions on all resources

Resources are objects that the actions can act upon. Not every resource type can be specified with every action. Some resource types work with only certain actions. For more information, see [Actions, resources, and condition keys for the AWS Marketplace Catalog](#) in the *Service Authorization Reference*.

There are two resource types in the Catalog API:

- **Entity** – An entity is a [seller product in AWS Marketplace](#) or an [experience in a private marketplace](#).
- **ChangeSet** – A change set is returned each time you use Catalog API to make changes to an entity. The change set describes the requested changes and its status. A change set can be canceled if the status is in the PREPARING state.

To allow a user or role the permission to make changes to all entities in an AWS account, you can add the following IAM policy. With this policy, the user or role can use the StartChangeSet action on all resources ("\*").

```
{
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "aws-marketplace:StartChangeSet"
      ],
      "Resource": "*"
    }
  ]
}
```

For information about all actions available for the Catalog API, see [Actions, resources, and condition keys for AWS Marketplace Catalog](#) in the *Service Authorization Reference*.

## Allowing actions on specific resources

### Note

Resource-level permissions and condition context keys for the `StartChangeSet` action are only supported when used with Catalog API. They are not supported when used with the [AWS Marketplace Management Portal](#).

Instead of allowing changes to all resources, you can use resource-level permissions to allow changes to specific resources.

For example, you can allow changes to a specific seller product in the AWS account instead of to all seller products. You do this by specifying the Amazon Resource Name (ARN) of the seller product in the Resource of the IAM policy.

### Note

To specify granular, resource-level permissions with actions that create new change sets, you need to also include a ChangeSet ARN to the list of resources. The ChangeSet ARN must include the wildcard (`/*`) to match any new change set ID that's created as shown.

```
{
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "aws-marketplace:StartChangeSet"
      ],
      "Resource": [
        "arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/AmiProduct/
example1-abcd-1234-5ef6-7890abcdef12",
        "arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/ChangeSet/*"
      ]
    }
  ]
}
```

For another example, you can allow changes to a specific experience in a private marketplace instead of to all experiences. You do this by specifying the ARN of the experience in the Resource of the IAM policy.

### Note

To specify granular, resource-level permissions with actions that create new change sets, you need to also include a ChangeSet ARN to the list of resources. The ChangeSet ARN must include the wildcard (/\*) to match any new change set ID that's created as shown.

```
{
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "aws-marketplace:StartChangeSet"
      ],
      "Resource": [
        "arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/Experience/exp-
example12345",
        "arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/ChangeSet/*"
      ]
    }
  ]
}
```

## Allowing actions with specific ChangeType condition key

### Note

Resource-level permissions and condition context keys for the StartChangeSet action are only supported when used with Catalog API. They are not supported when used with the [AWS Marketplace Management Portal](#).

The Catalog API action StartChangeSet has several different change types. You can allow access to only specific change types.

For example, you might only want to allow changes to the metadata of the seller product, such as the product title, and not allow adding new product versions. In this example, the change type `UpdateInformation` allows changing the metadata of a seller product, including the title. For more information about the different change types, see [Working with seller products](#) and [Working with a private marketplace](#) in the *AWS Marketplace Catalog API Reference*.

To limit the action to one or multiple change types, specify the `ChangeType` in the condition keys. In the following example IAM policy, the condition operator `StringEquals` specifies that the action is only allowed if the `ChangeType` matches `UpdateInformation`. For more information about condition keys, see [Condition operators](#) in the *AWS Identity and Access Management User Guide*.

### Note

To specify granular, resource-level permissions with actions that create new change sets, you need to also include a `ChangeSet` ARN to the list of resources. The `ChangeSet` ARN must include the wildcard (`/*`) to match any new change set ID that's created as shown.

```
{
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "aws-marketplace:StartChangeSet"
      ],
      "Resource": [
        "arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/AmiProduct/example1-abcd-1234-5ef6-7890abcdef12",
        "arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/ChangeSet/*"
      ],
      "Condition": {
        "StringEquals": {
          "catalog:ChangeType": "UpdateInformation"
        }
      }
    }
  ]
}
```

## Allowing actions with specific `aws:ResourceTag` condition key

### Note

Resource-level permissions and condition context keys for the `StartChangeSet` action are only supported when used with Catalog API. They are not supported when used with the [AWS Marketplace Management Portal](#).

You can allow actions on a group of entities without having to keep updating the policy and specifying a possibly growing list of entity ARNs. You can do this with resource tagging. Adding tags to resources allows you to control access to those resources based on their tags. For example, you might want to allow describing a group of seller products without specifying individual ARNs for each seller product.

For example, the following IAM policy allows the `DescribeEntity` action on any entity resource ("`*`") that has a tag key of `product-team` and tag value of `team-xyz`.

```
{
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "aws-marketplace:DescribeEntity"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceTag/product-team": "team-xyz"
        }
      }
    }
  ]
}
```

You can also allow describing and canceling change sets that were created with specific tags.

For example, the following IAM policy allows the `DescribeChangeSet` and `CancelChangeSet` actions on any change set resource ("`*`") that has a tag key of `product-team` and tag value of `team-xyz`.

```
{
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "aws-marketplace:DescribeChangeSet",
        "aws-marketplace:CancelChangeSet"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceTag/product-team": "team-xyz"
        }
      }
    }
  ]
}
```

Also, you can allow starting change sets on entities only when those entities have specific tags. For example, you can allow changes to seller products with specific tags.

For example, the following IAM policy allows the `StartChangeSet` action on any entity resource ("`*`") that has a tag key of `product-team` and tag value of `team-xyz`. In addition, the `TagResource` action is required so that when the change set is created, it's tagged with the same tag key and value.

### Note

If your policy to allow the `StartChangeSet` action includes a condition to match against specific tags, the same policy must also include the `TagResource` action. This is because the policy condition must match both the tag on the entity and the tag on the newly created change set resulting from the change request. Thus, it requires the user or role to also have the permission to tag the newly created change set. For an example of starting a change set and tagging the change set, see [the section called "Example: Adding tags to an entity and change set during creation"](#).

```
{
  "Statement": [
    {
```

```

    "Effect": "Allow",
    "Action": [
      "aws-marketplace:StartChangeSet",
      "aws-marketplace:TagResource"
    ],
    "Resource": "*",
    "Condition": {
      "StringEquals": {
        "aws:ResourceTag/product-team": "team-xyz"
      }
    }
  }
]
}

```

## Creating a custom IAM role

Customers who want to use a Resale Authorization ChangeType or a CPPO ChangeType need to create a custom AWS Identity and Access Management (IAM) role. This will support the creation of the Resale Authorization product lifecycle.

### To create a custom IAM role

1. Sign in to the IAM console (<https://console.aws.amazon.com/iam/>).
2. Under **Access management**, choose **Policies**.
3. Choose **Create policy**.
4. For **Step 1: Specify permissions**,
  - a. In the **Policy editor**, select the **JSON** button, and then add the following policy:

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowResaleAuthorizationShareActionsRAMCreate",
      "Effect": "Allow",
      "Action": [
        "ram:CreateResourceShare",
        "ram:AssociateResourceShare"
      ],
      "Resource": [

```

```

    "arn:aws:ram:*:*:*"
  ],
  "Condition": {
    "StringLikeIfExists": {
      "ram:ResourceArn": "arn:aws:aws-marketplace:*:*:AWSMarketplace/
ResaleAuthorization/*"
    },
    "StringEqualsIfExists": {
      "ram:RequestedResourceType": "aws-marketplace:Entity"
    }
  }
},
{
  "Sid": "AllowResaleAuthorizationShareActionsRAMAccept",
  "Effect": "Allow",
  "Action": [
    "ram:AcceptResourceShareInvitation",
    "ram:GetResourceShareInvitations",
    "ram:GetResourcePolicies",
    "ram:GetResourceShareAssociations"
  ],
  "Resource": [
    "arn:aws:ram:*:*:*"
  ]
},
{
  "Sid": "AllowResaleAuthorizationShareActionsMarketplace",
  "Effect": "Allow",
  "Action": [
    "aws-marketplace:PutResourcePolicy",
    "aws-marketplace:GetResourcePolicy",
    "aws-marketplace:DescribeEntity"
  ],
  "Resource": "arn:aws:aws-marketplace:*:*:AWSMarketplace/ResaleAuthorization/
*"
}
]
}

```

b. Choose **Next**.

5. For **Step 2: Review and create**,

- a. For **Policy details**, enter **FullResaleAuthorizationAccess** under **Policy name** and enter an optional **Description**.
- b. Review the **Permissions defined in this policy**.
- c. For **Add tags**, add tags (optional).
- d. Choose **Create policy**.

You have created the FullResaleAuthorizationAccess policy.

6. Under **Access management**, choose **Roles**.
7. Choose **Create role**.
8. For **Step 1: Select trusted entity**,
  - a. For **Trusted entity type**, choose **Custom trust policy**.
  - b. Copy and paste the following custom trust policy into the JSON editor.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "resale-authorization.marketplace.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

- c. Choose **Next**.
9. For **Step 2: Add permissions**,
  - a. Enter **FullResaleAuthorizationAccess** in the search bar.
  - b. Select the **FullResaleAuthorizationAccess** permission policy and then choose **Next**.
10. For **Step 3: Name, review, and create**,
  1. For **Role details**, enter **FullResaleAuthorizationAccess** as the **Role name** and enter an optional **Description**.

2. Under **Step 1: Select trusted entities**, ensure that the policy name you choose is attached to the role.
3. Under **Step 2: Add permissions**, review the **Policy name**.
4. Under **Step 3: Add tags**, add tags (optional).
5. Choose **Create role**.

You have created the FullResaleAuthorizationAccess role.

## Managing tags on resources

You can add, list, and remove tags from existing entities or change sets.

### Add tags to resources

To add tags to an entity or change set, use the TagResource API action.

#### Request

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

{
  "Catalog": "AWSMarketplace",
  "ResourceArn": "string",
  "Tags": [
    {
      "Key": "string",
      "Value": "string"
    }
    ...
  ]
}
```

Request parameters include:

- **Catalog** (String) – (Required) Must be AWSMarketplace.
- **ResourceArn** (String) – (Required) ARN of the change set or entity. A change set describes changes you make with Catalog API. An entity can be a [seller product in AWS Marketplace](#) or an [experience in a private marketplace](#).

- **Tags (Array of objects) – (Required)** A list of objects specifying each tag key and value. Number of objects allowed: 1–50.
  - **Key (String) – (Required)** Name of the tag.
    - **Regex pattern** – `^[a-zA-Z0-9_./+=\-\@]*$`
    - **Character length** – 1–128
  - **Value (String) – (Required)** Value of the tag.
    - **Regex pattern** – `^[a-zA-Z0-9_./+=\-\@]*$`
    - **Character length** – 0–256

## Response

```
{}
```

## Remove tags from resources

To remove a tag or list of tags from an entity or change set, use the `UntagResource` API action.

## Request

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

{
  "Catalog": "AWSMarketplace",
  "ResourceArn": "string",
  "TagKeys": [
    "string"
    ...
  ]
}
```

Request parameters include:

- **Catalog (String) – (Required)** Must be `AWSMarketplace`.
- **ResourceArn (String) – (Required)** ARN of the change set or entity. A change set describes changes you make with Catalog API. An entity can be a [seller product in AWS Marketplace](#) or an [experience in a private marketplace](#).

- **Tags** (Array of objects) – (Required) A list of key names of tags to be removed. Number of strings allowed: 0–256.

## Response

```
{}
```

## List all tags on a resource

To list all tags that have been added to and not yet removed from a change set or entity, use the `ListTagsForResource` API action.

## Request

```
POST /ListTagsForResource HTTP/1.1
```

```
Content-type: application/json
```

```
{
  "Catalog": "AWSMarketplace",
  "ResourceArn": "string"
}
```

Request parameters include:

- **Catalog** (String) – (Required) Must be `AWSMarketplace`.
- **ResourceArn** (String) - (Required) ARN of the change set or entity. A change set describes changes you make with Catalog API. An entity can be a [seller product in AWS Marketplace](#) or an [experience in a private marketplace](#).

## Response

```
{
  "ResourceArn": "string",
  "Tags": [
    {
      "Key": "string",
      "Value": "string"
    }
    ...
  ]
}
```

```
}
```

## Managing tags when requesting changes to resources

You can add tags when entities or change sets are created.

### Example: Adding tags to a change set when creating a change set

The following is an example of a `StartChangeSet` request that updates the product metadata for a seller product. This request adds a tag to the change set that's created with this request by including the tag in the `ChangeSetTags` property.

```
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 updated title\"}"
    }
  ],
  "ChangeSetTags": [
    {
      "Key": "product-team",
      "Value": "team-xyz"
    }
  ]
}
```

For more information about managing seller products, see [Working with seller products](#) in the *AWS Marketplace Catalog API Reference*.

### Example: Adding tags to an entity and change set during creation

The following is an example of a `StartChangeSet` request that creates a private marketplace experience entity. The request adds tags to both the entity resource and change set resource

created with this request by including the tags in the `EntityTags` and `ChangeSetTags` properties. With these tags, the permission policy of a user or role can be specified to only allow describing or canceling the change set this request creates or only allow creating further change sets on the entity this request creates. For more information, see [Granting permission to create entities and change sets only with tags](#).

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

{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "CreateExperience",
      "Entity": {
        "Type": "Experience@1.0"
      },
      "Details": "{\"Name\": \"ExamplePrivateMarketplace\"}",
      "EntityTags": [
        {
          "Key": "product-team",
          "Value": "team-xyz"
        }
      ]
    }
  ],
  "ChangeSetTags": [
    {
      "Key": "product-team",
      "Value": "team-xyz"
    }
  ]
}
```

For more information about managing a private marketplace, see [Working with a private marketplace](#) in the *AWS Marketplace Catalog API Reference*.

## Granting permission to manage tags on resources

To allow a user or role to add, remove, and list tags on all entities or change sets, they need the following IAM policy.

```
{
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "aws-marketplace:TagResource",
        "aws-marketplace:UntagResource",
        "aws-marketplace:ListTagsForResource"
      ],
      "Resource": "*"
    }
  ]
}
```

## Granting permission to manage tags on resources only when those resources have specific tags

You can allow a user or role to add, remove, and list tags on entities or change sets that have specific tags. The following IAM policy allows those actions on any entity resource ("\*") that has a tag key of `product-team` and tag value of `team-xyz`.

```
{
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "aws-marketplace:TagResource",
        "aws-marketplace:UntagResource",
        "aws-marketplace:ListTagsForResource"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceTag/product-team": "team-xyz"
        }
      }
    }
  ]
}
```

## Granting permission to create entities and change sets only with tags

### Note

Resource-level permissions and condition context keys for the `StartChangeSet` action are only supported when used with Catalog API. They are not supported when used with the [AWS Marketplace Management Portal](#).

You can enforce tagging when entities or change sets are created. Add the following policy to allow the `StartChangeSet` and the `TagResource` actions, with a condition specifying the tag key matches `product-team` and the tag value matches `team-xyz`. This policy condition must match both the tag on the newly created entity and the tag on the newly created change set resulting from the creation request. For an example of tagging an entity on creation, see [the section called "Example: Adding tags to an entity and change set during creation"](#).

For existing entities, this policy also enforces tagging change sets when requesting changes to those entities. This also requires that the existing entity has this existing tag. This is because the policy condition must match both the tag on the existing entity and the newly created change set resulting from the change request. For an example of adding tags to change requests, see [the section called "Example: Adding tags to a change set when creating a change set"](#).

```
{
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "aws-marketplace:StartChangeSet",
        "aws-marketplace:TagResource"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceTag/product-team": "team-xyz"
        }
      }
    }
  ]
}
```

## Service quotas

The AWS Marketplace Catalog API has the following quotas.

### Request quotas

API operations	Request rate (per AWS account)
ListEntities	10 per second
DescribeEntity	20 per second
StartChangeSet	5 per second
ListChangeSets	5 per second
DescribeChangeSet	10 per second
CancelChangeSet	5 per second
TagResource	5 per second
UntagResource	5 per second
ListTagsForResource	5 per second
PutResourcePolicy	5 per second
GetResourcePolicy	5 per second
DeleteResourcePolicy	5 per second

### Account quotas

Quota	Description
Maximum number of open StartChangeSet requests per account	250
Maximum number of Offers created or updated concurrently per account	20

## Request history retention quotas

Description	Quota
Retention period for change requests. This applies after the end time of each change request.	90 days

## Working with seller products

You can use the AWS Marketplace Catalog API to automate tasks for working with seller products. This includes the ability to create, update, view, list, and sort products. This enables you to automate the management of your products including self-service publishing capabilities on the AWS Marketplace Management Portal.

A *product* is a unit or resource that sellers intend to sell in AWS Marketplace, often referred to as a base product. A base product is not complete for buyer consumption until product information, deployment attributes, and billing information are added.

A *product* describes the product information, software deployment attributes, and billing mechanism of the listing that a seller intends to sell. The *product* must be paired with an *offer* to become a transactable unit that can be sold and be used by buyers in AWS Marketplace.

You can also use the AWS Marketplace Catalog API to [create offers](#), [Resale Authorizations](#), or [channel partner private offers \(CPPOs\)](#).

Each product type has a different product entity. The following product types and entities are supported:

Product type	Entity
Amazon Machine Image (AMI) products	AmiProduct@1.0
Container products	ContainerProduct@1.0
Software as a service (SaaS) products	SaaSProduct@1.0

### Note

Single-AMI with AWS CloudFormation product types, AWS Data Exchange data products, and professional services products are not supported.

This chapter assumes that you have access to the API and have completed any seller prerequisites, as described in the [API access control](#) topic.

See the following resources:

- To understand the basics of using the AWS Marketplace Catalog API, see [AWS Marketplace Catalog API](#).
- For end-to-end labs with working code examples, see [Manage products with API](#) in the *AWS Marketplace seller workshop*.
- For code examples of API requests, see [Python](#) and [Java](#) examples in *AWS Samples* on GitHub.

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

## Topics

- [Create a product](#)
- [Update product details](#)
- [Add pricing dimensions](#)
- [Update pricing dimensions](#)
- [Restrict pricing dimensions](#)
- [Update targeting configuration](#)
- [Update product visibility](#)
- [Publish a product](#)
- [Find your product ID](#)
- [Change set status and errors](#)
- [Working with single AMI products](#)
- [Working with container-based products](#)
- [Working with SaaS products](#)

## Create a product

### Note

This change type is only needed when you intend to create a brand new product entity in the AWS Marketplace catalog. It is not needed when updating existing products.

You can use the Catalog API to create an AMI, container, or SaaS product document with identifiers (product code and product ID) in AWS Marketplace.

You create a product in Draft state by calling the StartChangeSet API operation with the CreateProduct change type.

If your request is processed successfully, then AWS Marketplace Catalog API generates a product in Draft state for you. This is an incomplete product and isn't visible to buyers in AWS Marketplace.

You then use Update change types to complete the create product process: [UpdateInformation](#), [UpdateDimensions](#), [UpdateTargeting](#), and [UpdateVisibility](#).

After the product is completed, you can use the [ReleaseProduct](#) change type to complete the product creation process, and then release the offer. This process validates the entire product and moves the product to the Limited state.

#### Note

For more information about creating a product using the AWS Marketplace Management Portal, see the following topics in the *AWS Marketplace Seller Guide*:

- [Create your single-AMI product](#)

You cannot update the AMI for the version. If you need to update the AMI, create a new version instead.

- [Creating a container product](#)
- [Creating a SaaS product](#)

If you use the AWS Marketplace Management Portal to create a product, the product will be in the Staging state.

To create a product in Draft state, call the StartChangeSet API operation with the CreateProduct change type, as shown in the following example.

### Request Syntax

```
POST /StartChangeSet HTTP/1.1
```

```
Content-type: application/json

{
  "Catalog": "AWSMarketplace",
  "ChangeSet":
  [
    {
      "ChangeType": "CreateProduct",
      "Entity":
      {
        "Type": "SaaSProduct@1.0"
      },
      "DetailsDocument": {}
    }
  ]
}
```

Provide information for the fields to add the `CreateProduct` change type. This change type can take in `ProductTitle` attribute, subject to the same restrictions as that sent into `UpdateInformation` change type.

- **Entity** (object) (required) – The named type of entity being created.
  - **Type** (string) (required) – The Type is based on the delivery method (product type) that your product will be using: `AmiProduct@1.0`, `ContainerProduct@1.0`, or `SaaSProduct@1.0`.
- **DetailsDocument** (object) (required) – It may be empty.
  - **ProductTitle** (optional) – The title for your product, max length is 72 characters. Note that you can also later set or update the product title via the `UpdateInformation` change type.

## Response Syntax

A change set is created for your request. The response to this request gives you the `ChangeSetId` and `ChangeSetArn` 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.

You can check the status of the request through the AWS Marketplace Management Portal, or directly through Catalog API using the [DescribeChangeSet](#) API operation.

When the request is complete (if the Status is SUCCEEDED), a new ProductId is generated.

## Synchronous Validations

The following schema validations are specific to CreateProduct actions in the AWS Marketplace Catalog API. These validations are performed when you call StartChangeSet. If the request doesn't meet the following requirements, it will fail with an HTTP response.

Input field	Validation rule	HTTP code
ProductTitle (string)	Max length: 72	400

## Asynchronous Errors

The following errors are specific to CreateProduct 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](#).

Error code	Error message
INVALID_INPUT	Inappropriate content '{InappropriateContent}' found in ProductTitle field. Provide ProductTitle with no inappropriate content.

## Update product details

If you already have a product in AWS Marketplace, you can use the Catalog API to update the product details for an AMI, container, or SaaS product.

### Note

For more information about updating the product details using the AWS Marketplace Management Portal, see the following topics in the *AWS Marketplace Seller Guide*:

- AMI-based product: [Update product information](#)
- Container-based product: [Creating or updating product information for your container product](#)
- SaaS-based product: [Update product information](#)

To update product details, call the `StartChangeSet` API operation with the `UpdateInformation` change type and the details that you want to change, as shown in the following example.

## Request Syntax

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

{
  "Catalog": "AWSMarketplace",
  "ChangeSet":
  [
    {
      "ChangeType": "UpdateInformation",
      "Entity":
      {
        "Identifier": "prod-example12345",
        "Type": "AmiProduct@1.0"
      },
      "DetailsDocument":
      {
        "ProductTitle": "My Product Title",
        "ShortDescription": "My product short description.",
        "LongDescription": "My product longer description.",
        "Sku": "123example456",
        "LogoUrl": "https://awsmp-logos.s3.amazonaws.com/
ca60b754fe05a24257176cdbf31c4e0d",
        "VideoUrls":
        [
          "https://example.com/my-video"
        ],
        "Highlights":
        [
          "123example45"
        ]
      }
    }
  ]
}
```

```

    ],
    "AdditionalResources":
    [
      {
        "Text": "123example456",
        "Url": "https://example.com/some-link"
      }
    ],
    "SupportDescription": "Need help? Contact our experts at support@example.com \n
\nYour purchase includes 24x7 support.",
    "Categories":
    [
      "Operating Systems",
      "Network Infrastructure",
      "Application Development"
    ],
    "SearchKeywords":
    [
      "123example456"
    ]
  }
]
}

```

Provide information for the fields to add the `UpdateInformation` change type:

- **Entity** (object) (required) – The named type of entity being created.
  - **Identifier** (string) (required) – Your product ID. For more information, see [Identifier](#).
  - **Type** (string) (required) – The Type is based on the delivery method (product type) that your product will be using: `AmiProduct@1.0`, `ContainerProduct@1.0`, or `SaaSProduct@1.0`.
- **DetailsDocument** (object) – The details of the request including the information that you want to update for your product. Each field is optional, but you must include at least one change to update.
  - **ProductTitle** (string) – The name of the product to be displayed to buyers.
  - **ShortDescription** (string) – The description of key aspects of the product to be displayed to buyers. This is usually 2–3 sentences.
  - **LongDescription** (string) – The longer description of your product to be displayed to buyers. This is usually 1–3 paragraphs.

- **Sku** (string or null) – The free-form string that you define as a reference for your own use. Use `null` to unset this field.
- **LogoUrl** (string) – The URL to an image in a publicly accessible Amazon Simple Storage Service (Amazon S3) bucket. For more information, see [Company and product logo requirements](#).
- **VideoUrls** (array of strings) – The list of URLs to publicly available, externally hosted videos to be provided as a reference to buyers in your product information.

 **Note**

Currently, AWS Marketplace supports one URL in the array.

- **Highlights** (array of strings) – The list of short callouts for key product features.
- **AdditionalResources** (array of structures) – The 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) – The URL to a resource that might be helpful for a buyer to understand your product.
- **SupportDescription** (string) – The details about your support offering for your product.
- **Categories** (array of strings) – The list of AWS Marketplace defined product categories that describe your product. For more information, see [Product categories](#) in the *AWS Marketplace Buyer Guide*.
- **SearchKeywords** (array of strings) – The list of keywords for your product to enhance the search experience. Seller name, product name, and product categories are automatically included in search keywords and don't need to be repeated here.

 **Note**

When you are initially populating product information (metadata) for a Draft product, you will need to supply all of the following in the `DetailsDocument` object of `UpdateInformation` change type: `ProductTitle`, `ShortDescription`, `LongDescription`, `LogoUrl`, `Highlights`, `AdditionalResources`, `SupportDescription`, `Categories`, and `SearchKeywords`.

The `ProductTitle` can be omitted if it has already been provided during `CreateProduct` change type. However, when you are updating existing fields on the product, you can

include only the attributes that need to be changed in the `DetailsDocument` object of the `UpdateInformation` change type.

## Response Syntax

A change set is created for your request. The response to this request gives you the `ChangeSetId` and `ChangeSetArn` 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.

You can check the status of the request through the AWS Marketplace Management Portal, or directly through Catalog API using the [DescribeChangeSet](#) API operation.

## Synchronous Validations

The following schema validations are specific to `UpdateInformation` actions in the AWS Marketplace Catalog API. These validations are performed when you call `StartChangeSet`. If the request doesn't meet the following requirements, it will fail with an HTTP response.

Input field	Validation rule	HTTP code
String (general)	No control characters "\\x00-\\x08\\ \\x0B-\\x1F"	400
ProductTitle (string)	Max length: 72 Required	400
ShortDescription (string)	Max length: 1000 Required	400
LongDescription (string)	Max length: 5000	400

Input field	Validation rule	HTTP code
	Required	
Skus (array of strings)	Max length: 100  Optional	400
LogoUrl (string)	URL pattern:  ^https://(www\.)?[-a-zA-Z0-9@.] {1,256}\.[a-zA-Z0-9()]{2,63}\b([-a- zA-Z0-9@+./]*)  Required	400
VideoUrls (array of strings)	URL pattern:  https://(www\.)?[-a-zA-Z0-9@.] {1,256}\.[a-zA-Z0-9()]{2,63}\b([-a- zA-Z0-9@_+./\])  Optional	400
Highlights (array of strings)	Required: Min 1 - Max 3	400
AdditionalResources (array of structures)	Max length: 500  Optional	400
SupportDescription (string)	Max length: 2000  Required	400
Categories (array of strings)	Required: Min 1 - Max 3  Required	400
SearchKeywords (array of strings)	Required: Min 1 - Max 3  Required	400

## Asynchronous Errors

The following errors are specific to UpdateInformation actions in the AWS Marketplace Catalog API. These errors are returned when you call DescribeChangeSet after a change set is processing. For more information about using DescribeChangeSet to get the status of a change request, see [Working with change sets](#).

Error code	Error message
MISSING_DATA	No data provided to perform an update. Provide data for at least 1 field of the product.
INVALID_INPUT	Provide LogoUrl.
INVALID_INPUT	Provide ProductTitle.
INVALID_INPUT	Provide ShortDescription.
INVALID_INPUT	Provide LongDescription.
INVALID_INPUT	Provide SupportDescription.
INVALID_INPUT	Provide at least one search keyword.
INVALID_INPUT	Provide at least one highlight.
INVALID_INPUT	Provide between 1 and 3 product categories.
INVALID_INPUT	Inappropriate content '{InappropriateContent}' found in ProductTitle field. Provide ProductTitle with no inappropriate content.
INVALID_INPUT	Inappropriate content '{InappropriateContent}' found in ShortDescription field. Provide ShortDescription with no inappropriate content.
INVALID_INPUT	Inappropriate content '{InappropriateContent}' found in LongDescription field. Provide

Error code	Error message
	LongDescription with no inappropriate content.
INVALID_INPUT	Inappropriate content '{InappropriateContent}' found in SupportDescription field. Provide SupportDescription with no inappropriate content.
INVALID_INPUT	Invalid ProductTitle field. Remove spaces before trademark symbol.
INVALID_INPUT	Invalid ShortDescription field. Remove spaces before trademark symbol.
INVALID_INPUT	Invalid LongDescription field. Remove spaces before trademark symbol.
INVALID_INPUT	Invalid SupportDescription field. Remove spaces before trademark symbol.
INVALID_INPUT	Invalid ProductTitle field. Remove unsupported characters [UnsupportedCharacters].
INVALID_INPUT	Invalid ShortDescription field. Remove unsupported characters [UnsupportedCharacters].
INVALID_INPUT	Invalid LongDescription field. Remove unsupported characters [UnsupportedCharacters].
INVALID_INPUT	Invalid SupportDescription field. Remove unsupported characters [UnsupportedCharacters].
INVALID_INPUT	Search keywords must be no more than 250 combined characters.

Error code	Error message
INVALID_INPUT	The input for this change type could not be read. Submit a properly formatted input.
INVALID_ADDITIONAL_RESOURCES	Invalid URLs in AdditionalResources: [InvalidAdditionalResourcesUrls] Provide valid URLs.
INVALID_CATEGORY_NAMES	Provide valid category names supported by AWS Marketplace.
InvalidImageProperties	Validation errors found: The file is not image type. Supported image types: [png jpg gif].
EXPLICIT_CONTENT	Explicit content: '{ExplicitContent}' detected. Provide media with no explicit content.
INVALID_MEDIA	Invalid URL: {MediaUrl} Provide a new URL for media stored in S3.
INVALID_MEDIA	Invalid URL: {MediaUrl} Provide a valid URL that does not exceed 2048 characters.
INVALID_MEDIA	Location provided not accessible: {MediaUrl} Provide an accessible URL for media stored in S3.
INVALID_MEDIA	There was an issue copying the media from S3. Image size exceeds 5 MB. Provide an image that is under 5 MB.
INVALID_MEDIA	There was an issue copying the media from S3. Video size exceeds 100 MB. Provide a video that is under 100 MB.
INVALID_MEDIA	Malware detected in media. Please resubmit media without malware.

# Add pricing dimensions

You can use the Catalog API to add billable dimensions in order to charge users for an AMI, container, or SaaS product in AWS Marketplace.

A *pricing dimension* is a unit of measure that sellers define for charging buyers. Sellers must set up this information to bill buyers for using the product, whether it's a usage-based or contract-based pricing model. The type of dimension is dependent on the intended product's pricing model. For more information about product pricing, see [AMI product pricing](#), [Container products pricing](#), and [SaaS product pricing](#) in the AWS Marketplace Seller Guide.

## Note

For more information about adding pricing dimensions using the AWS Marketplace Management Portal, see the following topics in the *AWS Marketplace Seller Guide*:

- AMI-based product: [Update pricing](#)
- Container-based product: [Adding a pricing dimension](#)
- SaaS-based product: [Add pricing dimensions](#)

To add pricing dimensions, call the `StartChangeSet` API operation with the `AddDimensions` change type, as shown in the following example.

## Note

After submitting the first `AddDimensions` change type with dimensions specifying a type of pricing model (usage, contract, and contract with consumption), you must work with the AWS Marketplace Seller Operations team. They will assist you in adding a dimension with types that are outside of the initially set pricing model.

## Request Syntax

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

{
```

```
"Catalog": "AWSMarketplace",
"ChangeSet":
[
  {
    "ChangeType": "AddDimensions",
    "DetailsDocument":
    [
      {
        "Description": "Description of the dimension",
        "Key": "uniqueapikey",
        "Unit": "HostHrs",
        "Name": "First Dimension",
        "Types":
        [
          "ExternallyMetered"
        ]
      }
    ],
    "Entity":
    {
      "Identifier": "prod-example12345",
      "Type": "SaaSProduct@1.0"
    }
  }
]
}
```

Provide information for the fields to add the `AddDimensions` change type.

- **DetailsDocument** (object) (required) – Details of the request.
  - **Description** (string) (required) – Full details of the dimension that will be the long description on the buyer’s viewing page.
  - **Key** (string) (required) – Enter in the facet that will be used for defining the rates in the offer. Also, enter the dimensions published to the AWS Marketplace Metering Service (MMS) if the dimension can’t be metered externally. After the dimension is created, this can’t be changed.
  - **Unit** (string) (required) – The unit type for the dimension. Possible units are Users, Hosts, GB, MB, TB, Gbps, Mbps, Requests, Units, UserHrs, UnitHrs, Units, HostHrs, TierHrs, and TaskHrs.
  - **Name** (string) (required) – The display name for the dimension on the website and customer's bill.

- **Types** (array of strings) (required) (also known as **Tags**) – These indicate whether the dimension covers metering, entitlement, or support for external metering. This is not changeable after the dimension is created.
- **Metered** – Indicates that Commerce Platform usage types should be created to allow metering to occur for this dimension.
- **ExternallyMetered** – Indicates that AWS Marketplace Metering Service (MMS) dimensions should be created during publishing to allow sellers to meter through the AWS SDK.
- **Entitled** – Indicates that entitlements can be granted for the dimension during the product or offer publishing.

### Valid Pricing Dimension Types Combinations

Pricing Dimension Type	Product
[Metered]	AMI
[Metered, ExternallyMetered]	SaaS, AMI/Flexible Consumption Pricing (FCP)  When <b>ExternallyMetered</b> appears, <b>Metered</b> is masked/inferred.
[Entitled]	SaaS Contracts  The <b>Entitled</b> tag grants rights to use a software/service set start and end dates for the usage. Also, to grant rights to have usage discount for AMI annual products. Each entitlement is identified by a <b>Dimension Key</b> in AWS Marketplace Entitlement Service for creating or updating the entitlements.
[Metered, ExternallyMetered, Entitled]	Contracts with Consumption Pricing, where dimensions can be prepaid or metered are a combination of both <b>[ExternallyMetered]</b> and <b>[Entitled]</b> .

- **Entity** (object) (required) – The named type of entity being created.

- **Identifier** (string) (required) – Your product ID. For more information, see [Identifier](#).
- **Type** (string) (required) – The Type is based on the delivery method (product type) that your product will be using: `AmiProduct@1.0` or `SaaSProduct@1.0`.

## Response Syntax

A change set is created for your request. The response to this request gives you the `ChangeSetId` and `ChangeSetArn` 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 includes validating information to ensure it meets the AWS Marketplace guidelines. The validation process can take anywhere from a few minutes to a few hours.

You can check the status of the request through the AWS Marketplace Management Portal, or directly through Catalog API using the [DescribeChangeSet](#) API operation.

## Synchronous Validations

The following schema validations are specific to `AddDimensions` actions in the AWS Marketplace Catalog API. These validations are performed when you call `StartChangeSet`. If the request doesn't meet the following requirements, it will fail with an HTTP response.

Input field	Validation rule	HTTP code
Description	Max length: 1000	422
	Required	
Key	Max length: 100	422
	Pattern: <code>[A-Za-z0-9_-.]+</code>	
	Required	

Input field	Validation rule	HTTP code
Dimension Unit	Max length: 20 Required	422
Name	Max length: 500 Required	422
Type (tag)	Required: Min 1 - Max 3  Inputs: Entitled, Metered, ExternallyMetered  Required	422

## Asynchronous Errors

The following errors are specific to AddDimensions actions in the AWS Marketplace Catalog API. These errors are returned when you call DescribeChangeSet after a change set is processing. For more information about using DescribeChangeSet to get the status of a change request, see [Working with change sets](#).

Error code	Error message
MISSING_DATA	No data provided to perform an update. Provide data for at least 1 dimension.
INVALID_DIMENSION	Provide no more than 24 dimensions.
INVALID_DIMENSION	Can't add duplicate dimensions.
INVALID_DIMENSION	Dimension can't be added in current state '%s'. States that support dimension updates are %s.
INVALID_DIMENSION	Can't add dimension. The field '%s' has duplicate values '%s' in other dimensions.

Error code	Error message
INVALID_DIMENSION	Provide non-empty fields (Key, Unit, Name, Types) for each dimension.
IINVALID_TYPE	Remove invalid type '%s'. Valid types are ["Metered", "Entitled", "ExternallyMetered"].
INVALID_UNIT	Remove invalid Unit '%s'. Valid units are ["GB", "Gbps", "HostHrs", "Hosts", "MB", "Mbps", "Requests", "TaskHrs", "TB", "TierHrs", "UnitHrs", "Units", "UserHrs", "Users"].
INVALID_INPUT	Inappropriate content '%s' found in %s field. Provide %s with no inappropriate content.
INVALID_INPUT	Invalid '%s' field. Remove spaces before trademark symbol.
INVALID_INPUT	Invalid '%s' field. Remove unsupported characters %s.
INVALID_DIMENSION	Remove invalid dimension type combination %s. Allowed values are %s.
INVALID_DIMENSION	Remove invalid dimension key '%s' for Metered dimension.
INVALID_DIMENSION	Dimension named '%s' for productCode '%s' did not pass AWS Marketplace Metering Service validation %s.
INVALID_DIMENSION	Dimension named '%s' for productCode '%s' has no metering record present in Metering Service. The product has either never been launched for testing or is misconfigured and does not make the appropriate calls to the AWS Marketplace Metering Service.

# Update pricing dimensions

You can use the Catalog API to update existing pricing dimensions of an AMI, container, or SaaS product in AWS Marketplace.

Each dimension is uniquely identified by the dimension key and dimension types to perform the update. Updating a dimension doesn't affect any active offer or customers that the original dimension had created.

## Note

For more information about updating pricing dimensions using the AWS Marketplace Management Portal, see the following topics in the *AWS Marketplace Seller Guide*:

- AMI-based product: [Update pricing](#)
- Container-based product: [Updating dimension information](#)
- SaaS-based product: [Update pricing dimensions](#)

To update pricing dimensions, call the `StartChangeSet` API operation with the `UpdateDimensions` change type, as shown in the following example.

## Request Syntax

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

{
  "Catalog": "AWSMarketplace",
  "ChangeSet":
  [
    {
      "ChangeType": "UpdateDimensions",
      "DetailsDocument":
      [
        {
          "Key": "uniqueapikey",
          "Types":
          [
            "ExternallyMetered"
```

```

    ],
    "Name": "First Dimension",
    "Description": "Description of the dimension"
  }
],
"Entity":
{
  "Identifier": "prod-example12345",
  "Type": "SaaSProduct@1.0"
}
]
}

```

Provide information for the fields to add the `UpdatedDimensions` change type:

- **DetailsDocument** (array of objects) (required) – Details of the request.
- **Key** (string) (required) – Provide key of existing dimension from the product to change description and name on. For `UpdatedDimension`, this field is only for identifying the dimension to be changed.
- **Types** (array of strings) (required) (also known as **Tags**) – These indicate whether the dimension covers metering, entitlement, or support for external metering. This is not changeable after the dimension is created.
- **ExternallyMetered** – Indicates that AWS Marketplace Metering Service (MMS) dimensions should be created during publishing to allow sellers to meter through the AWS SDK.
- **Entitled** – Indicates that entitlements can be granted for the dimension during product/offer publishing.

### Valid Pricing Dimension Types Combinations

Pricing Dimension Type	Product
[ <b>ExternallyMetered</b> ]	SaaS, AMI/Flexible Consumption Pricing (FCP)  When <b>ExternallyMetered</b> appears, <b>Metered</b> is masked/inferred.
[ <b>Entitled</b> ]	SaaS Contracts

Pricing Dimension Type	Product
	The <b>Entitled</b> tag grants rights to use a software/service set start and end dates for the usage. Also, to grant rights to have usage discount for AMI annual products. Each entitlement is identified by a <b>Dimension Key</b> in AWS Marketplace Entitlement Service for creating or updating the entitlements.
<b>[ExternallyMetered, Entitled]</b>	Contracts with Consumption Pricing, where dimensions can be prepaid or metered are a combination of both <b>[ExternallyMetered]</b> and <b>[Entitled]</b> .

- **Description** (string) (optional) – Full description of the dimension that will be the long description on the buyer's viewing page.
- **Name** (string) optional – Display name for the dimension on the website and customer's bill.
- **Entity** (object) (required) – The named type of entity being created.
- **Identifier** (string) (required) – Your product ID. For more information, see [Identifier](#).
- **Type** (string) (required) – The Type is based on the delivery method (product type) that your product will be using: `AmiProduct@1.0` or `SaaSProduct@1.0`.

## Response Syntax

A change set is created for your request. The response to this request gives you the `ChangeSetId` and `ChangeSetArn` 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 included validating information to ensure it meets the AWS Marketplace guidelines. The validation process can take anywhere from a few minutes to a few hours.

You can check the status of the request through the AWS Marketplace Management Portal, or directly through Catalog API using the [DescribeChangeSet](#) API operation.

## Synchronous Validations

The following schema validations are specific to UpdateDimensions actions in the AWS Marketplace Catalog API. These validations are performed when you call StartChangeSet. If the request doesn't meet the following requirements, it will fail with an HTTP response.

Input field	Validation rule	HTTP code
Description	Max length: 1000 Required	400
Key	Max length: 100 Pattern: [A-Za-z0-9_-.]+\$ Required	400
Name	Max length: 5 Required	400
Types (tag)	Required: Min 1 - Max 3 Inputs: Entitled, Metered, ExternallyMetered Required	422

## Asynchronous Errors

The following errors are specific to UpdateDimensions actions in the AWS Marketplace Catalog API. These errors are returned when you call DescribeChangeSet after a change set is

processing. For more information about using `DescribeChangeSet` to get the status of a change request, see [Working with change sets](#).

Error code	Error message
INVALID_INPUT	Invalid '%s' field. Remove spaces before trademark symbol.
INVALID_INPUT	Invalid '%s' field. Remove unsupported characters %s.
INVALID_DIMENSION	Provide non-empty fields (Key, Types, Name and/or Description ) for each dimension.
INVALID_DIMENSION	Cannot update dimension. The field Name has duplicate values '%s' in other dimensions.
INVALID_DIMENSION	Cannot update same dimension with key '%s' and types '%s' multiple times in the same request.
INVALID_DIMENSION	Cannot restrict dimension. The dimension key '%s' with types '%s' does not exist.
INVALID_DIMENSION	Cannot update dimension. The dimension key '%s' is Metered.

## Restrict pricing dimensions

You can use the Catalog API to restrict existing pricing dimensions of an AMI or SaaS product in AWS Marketplace.

Each dimension is uniquely identified by the dimension key and dimension types to perform the update. Restricting a dimension doesn't affect any active offer or customers that the original dimension had created.

To restrict pricing dimensions, call the `StartChangeSet` API operation with the `RestrictDimensions` change type, as shown in the following example.

## Request Syntax

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

{
  "Catalog": "AWSMarketplace",
  "ChangeSet":
  [
    {
      "ChangeType": "RestrictDimensions",
      "DetailsDocument":
      [
        {
          "Key": "uniqueapikey",
          "Types":
          [
            "ExternallyMetered"
          ]
        }
      ],
      "Entity":
      {
        "Identifier": "prod-example12345",
        "Type": "SaaSProduct@1.0"
      }
    }
  ]
}
```

Provide information for the fields to add the `RestrictDimensions` change type:

- **DetailsDocument** (array of objects) (required) – Details of the request.
  - **Key** (string) (required) – Provide key of existing dimension from the product to change description and name on. For `RestrictDimensions`, this field is only for identifying the dimension to be changed.
  - **Types** (array of strings) (required) (also known as **Tags**) – These indicate whether the dimension covers metering, entitlement, or support for external metering. This is not changeable after the dimension is created.
    - **Metered** – Indicates that Commerce Platform usage types should be created to allow metering to take place for this dimension.

- **ExternallyMetered** – Indicates that AWS Marketplace Metering Service (MMS) dimensions should be created during publishing to allow sellers to meter through the AWS SDK.
- **Entitled** – Indicates that entitlements can be granted for the dimension during product/offer publishing.
- **Valid Pricing Dimension Types Combinations:**
  - **[Metered]** – AMI Products
  - **[ExternallyMetered]** – SaaS, AMI/FCP (Flexible Consumption Pricing)
  - **[Entitled]** – SaaS Contracts, ProServe Products. Entitled tag grants rights to use a software/service, sets start and end dates for the usage. Also, to grant rights to have usage discount for AMI annual products. Each entitlement is identified by a Dimension Key in AWS Marketplace Entitlement Service for creating/updating the entitlements.
  - Contracts with Consumption Pricing, where dimensions can be prepaid or metered are a combination of both **[ExternallyMetered]** and **[Entitled]**
- **Entity** (object) (required) – The named type of entity being created.
  - **Identifier** (string) (required) – Your product ID. For more information, see [Identifier](#).
  - **Type** (string) (required) – The Type is based on the delivery method (product type) that your product will be using: `AmiProduct@1.0` or `SaaSProduct@1.0`.

## Response Syntax

A change set is created for your request. The response to this request gives you the `ChangeSetId` and `ChangeSetArn` 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 included validating information to ensure it meets the AWS Marketplace guidelines. The validation process can take anywhere from a few minutes to a few hours.

You can check the status of the request through the AWS Marketplace Management Portal, or directly through Catalog API using the [DescribeChangeSet](#) API operation.

## Synchronous Validations

The following schema validations are specific to `RestrictDimensions` actions in the AWS Marketplace Catalog API. These validations are performed when you call `StartChangeSet`. If the request doesn't meet the following requirements, it will fail with an HTTP response.

Input field	Validation rule	HTTP code
Description	Max length: 1000 Required	400
Key	Max length: 100 Pattern: [A-Za-z0-9_-.]+\$ Required	400
Name	Max length: 5 Required	400
Types (tag)	Required: Min 1 - Max 3 Inputs: Entitled, Metered, ExternallyMetered Required	422

## Asynchronous Errors

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

Error code	Error message
INVALID_INPUT	Invalid '%s' field. Remove spaces before trademark symbol.

Error code	Error message
INVALID_INPUT	Invalid '%s' field. Remove unsupported characters %s.
INVALID_DIMENSION	Provide non-empty fields (Key, Types, Name and/or Description ) for each dimension.
INVALID_DIMENSION	Cannot update dimension. The field Name has duplicate values '%s' in other dimensions.
INVALID_DIMENSION	Cannot update same dimension with key '%s' and types '%s' multiple times in the same request.
INVALID_DIMENSION	Cannot restrict dimension. The dimension key '%s' with types '%s' does not exist.
INVALID_DIMENSION	Cannot update dimension. The dimension key '%s' is Metered.

## Update targeting configuration

You can use the Catalog API to add AWS account IDs that are allowed to view the AMI, container, or SaaS product in AWS Marketplace before it's moved to a Public state by calling the `UpdateVisibility` change type.

### Note

For more information about adding AWS account IDs using the AWS Marketplace Management Portal, see the following topics in the *AWS Marketplace Seller Guide*:

- AMI-based product: [Update the allowlist \(preview accounts\)](#)
- Container-based product: [Updating the allowlist of AWS account IDs](#)
- SaaS-based product: [Updating the allowlist of AWS account IDs](#)

To add AWS account IDs that are allowed to view the AMI, container, or SaaS product, call the `StartChangeSet` API operation with the `UpdateTargeting` change type, as shown in the following example.

## Request Syntax

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

{
  "Catalog": "AWSMarketplace",
  "ChangeSet":
  [
    {
      "ChangeType": "UpdateTargeting",
      "Entity":
      {
        "Type": "SaaSProduct@1.0",
        "Identifier": "prod-example12345"
      },
      "DetailsDocument":
      {
        "PositiveTargeting":
        {
          "BuyerAccounts":
          [
            "1112223334444"
          ]
        }
      }
    }
  ]
}
```

Provide information for the fields to add the `UpdateTargeting` change type.

- **Entity** (object) (required) – The named type of entity being created.
- **Identifier** (string) (required) – Your product ID. For more information, see [Identifier](#).
- **Type** (string) (required) – The Type is based on the delivery method (product type) that your product will be using: `AmiProduct@1.0` or `SaaSProduct@1.0`.
- **DetailsDocument** (object) (required) – The details required to run the `ChangeSet`.

- **PositiveTargeting** (object) (optional) – Positive targeting defines the criteria which any buyer's profile should fulfill in order to be allowed to access the offer. This field is optional, but at least one targeting option should be provided when this field is present.
- **BuyerAccounts** (array of strings) (optional) – List as an option to allow targeting based on AWS accounts (also known as, Private Offer). If the intention is to not target the offer to an AWS account, this field should be omitted.

## Response Syntax

A change set is created for your request. The response to this request gives you the `ChangeSetId` and `ChangeSetArn` 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 included validating information to ensure it meets the AWS Marketplace guidelines. The validation process can take anywhere from a few minutes to a few hours.

You can check the status of the request through the AWS Marketplace Management Portal, or directly through Catalog API using the [DescribeChangeSet](#) API operation.

When the request is complete (if the `Status` is `SUCCEEDED`), a new `ProductId` is generated.

## Synchronous Validations

The following schema validations are specific to `UpdateTargeting` actions in the AWS Marketplace Catalog API. These validations are performed when you call `StartChangeSet`. If the request doesn't meet the following requirements, it will fail with an HTTP response.

Type of targeting	Valid current visibility states	BuyerAccounts (input)	Check
Positive	Public, Limited, or Draft	Array of 12-digit AWS account ID strings.	The input must be different from the

Type of targeting	Valid current visibility states	BuyerAccounts (input)	Check
		Min size: 0.  Max size: 5000.	current document targeted accounts.  Input must be in valid AWS accounts.

## Asynchronous Errors

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

Error code	Error message
INVALID_PRODUCT_VISIBILITY	Use an existing Public, Limited or Draft product.
INVALID_AWS_ACCOUNT_IDS	Provide valid AWS account IDs. AWS accounts not found: [x, y, z].

## Update product visibility

You can use the Catalog API to update the visibility (also known as lifecycle state) of an AMI, container, or SaaS product in AWS Marketplace.

### Note

For more information updating product visibility using the AWS Marketplace Management Portal, see the following topics in the *AWS Marketplace Seller Guide*:

- AMI-based product: [Update product visibility](#)
- Container-based product: [Updating product visibility](#)

- SaaS-based product: [Update product visibility](#)

Allowed target lifecycle states are `Limited`, `Public`, or `Restricted`.

### Limited

The state where the product is complete and has successfully completed the `ReleaseProduct` `ChangeType`. Sellers can view details of the product in this state. The product is not public. However, sellers can target specific buyers to allow to preview the product.

### Public

The state where the product is visible in AWS Marketplace. Buyers can view and subscribe to the product.

### Restricted

The state where the product is no longer visible to the public and doesn't accept new subscribers. Existing subscribers can continue using this product until their subscription expires.

#### Note

The `UpdateVisibility` change type requires a manual review from the AWS Marketplace Seller Operations team, which results in a longer execution time. If all change types in a change set succeed or fail, a manual rejection in `UpdateVisibility` will cause a failure of the whole change set. If you want other change types in a change set to be applied sooner and without manual review, you can call `UpdateVisibility` separately in its own change set.

To update the visibility of your product, call the `StartChangeSet` API operation with the `UpdateVisibility` change type, as shown in the following example.

### Request Syntax

For when `TargetVisibility` is `Public` or `Limited`.

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

```
{
  "Catalog": "AWSMarketplace",
  "ChangeSet":
  [
    {
      "ChangeType": "UpdateVisibility",
      "Entity":
      {
        "Type": "SaaSProduct@1.0",
        "Identifier": "prod-example12345"
      },
      "DetailsDocument":
      {
        "TargetVisibility": "Public"
      }
    }
  ]
}
```

For when TargetVisibility is Restricted.

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

{
  "Catalog": "AWSMarketplace",
  "ChangeSet":
  [
    {
      "ChangeType": "UpdateVisibility",
      "Entity":
      {
        "Type": "SaaSProduct@1.0",
        "Identifier": "prod-example12345"
      },
      "DetailsDocument":
      {
        "TargetVisibility": "Restricted",
        "ReplacementProductId": "prod-example54321"
      }
    }
  ]
}
```

Provide information for the fields to add the `UpdateVisibility` change type.

- **Entity** (object) (required) – The named type of entity being created.
  - **Identifier** (string) (required) – Your product ID. For more information, see [Identifier](#).
  - **Type** (string) (required) – The Type is based on the delivery method (product type) that your product will be using: `AmiProduct@1.0` or `SaaSProduct@1.0`.
- **DetailsDocument** (object) (required) – The details required to run the `ChangeSet`.
  - **TargetVisibility** – The intended new visibility of the product.

Possible values: `Public`, `Limited`, and `Restricted`

- **ReplacementProductId** (string) (optional) – Replacement product ID for the product to be `Restricted`. Used to notify current subscribers about the product restriction.

Only accepts `Restricted` for `TargetVisibility`.

## Synchronous Validations

The following schema validations are specific to `UpdateVisibility` actions in the AWS Marketplace Catalog API. These validations are performed when you call `StartChangeSet`. If the request doesn't meet the following requirements, it will fail with an HTTP response.

Type of targeting	Valid current states	ReplacementProductId (input)	Validation checks
Public	Limited and Restricted	Not allowed	Valid current state
Limited	Public and Restricted	Not allowed	Valid current state
Restricted	Public and Limited	String (Optional)	ReplacementProductId must belong to an existing Limited or Public product.

After triggering this change type, it can take up to 37 days to complete. This includes the time the AWS Marketplace Seller Operations Team needs to review, audit, and approve. When

restricting a product, you have 24 hours to change your mind, by calling `CancelChangeSet`, before the AWS Marketplace Seller Operations Team begins auditing. For more information, see [CancelChangeSet](#).

## Response Syntax

A change set is created for your request. The response to this request gives you the `ChangeSetId` and `ChangeSetArn` 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 included validating information to ensure it meets the AWS Marketplace guidelines. The validation process can take anywhere from a few minutes to a few hours.

You can check the status of the request through the AWS Marketplace Management Portal, or directly through Catalog API using the [DescribeChangeSet](#) API operation.

When the request is complete (if the `Status` is `SUCCEEDED`), a new `ProductId` is generated.

## Asynchronous Errors

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

Error code	Error message
INVALID_PRODUCT_STATE	Use an existing <code>Public</code> , <code>Limited</code> , or <code>Restricted</code> product.
INVALID_TARGET_VISIBILITY	Provide a valid target visibility state: <code>Public</code> , <code>Limited</code> , or <code>Restricted</code> .

Error code	Error message
EMPTY_TARGET_VISIBILITY	Provide a valid target visibility state: Public, Limited, or Restricted .
INVALID_REPLACEMENT_PRODUCT_ID	Use an existing Public or Limited product as replacement.
INVALID_REPLACEMENT_PRODUCT_ID	Replacement product ID is only valid when restricting a product.
AUDIT_ERROR	Varies based on MCO manual review.
MISSING_SELLER_PROFILE_INFORMATION	Before you can update your product to Public, you must add a public profile to your seller account.

## Publish a product

You can use the Catalog API to publish a Draft AMI, container, or SaaS product into Limited state in AWS Marketplace.

### Note

For `AmiProduct@1.0` and `SaaSProduct@1.0`, the `ReleaseProduct` change type must be accompanied by `ReleaseOffer` change type on the corresponding draft public `Offer@1.0` entity created for this product.

To publish a product, call the `StartChangeSet` API operation with the `ReleaseProduct` change type, as shown in the following example.

### Request Syntax

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

{
```

```
"Catalog": "AWSMarketplace",
"ChangeSet":
[
  {
    "ChangeType": "ReleaseProduct",
    "Entity":
    {
      "Type": "SaaSProduct@1.0",
      "Identifier": "prod-example12345"
    },
    "DetailsDocument": {}
  }
]
}
```

Provide information for the fields to add to the ReleaseProduct change type. This change type does not take any parameter payload.

- **Entity** (object) (required) – The named type of entity being created.
  - **Identifier** (string) (required) – Your product ID. For more information, see [Identifier](#).
  - **Type** (string) (required) – The Type is based on the delivery method (product type) that your product will be using: AmiProduct@1.0, SaaSProduct@1.0, ContainerProduct@1.0.
- **DetailsDocument** (object) (required) - Must be an empty object. The change type ReleaseProduct doesn't accept any details.

## Response Syntax

A change set is created for your request. The response to this request gives you the ChangeSetId and ChangeSetArn 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 included validating information to ensure it meets the AWS Marketplace guidelines. The validation process can take anywhere from a few minutes to a few hours.

You can check the status of the request through the AWS Marketplace Management Portal, or directly through Catalog API using the [DescribeChangeSet](#) API operation.

When the request is complete (if the Status is SUCCEEDED), a new ProductId is generated.

## Asynchronous Errors

The following errors are specific to ReleaseProduct actions in the AWS Marketplace Catalog API. These errors are returned when you call DescribeChangeSet after a change set is processing. For more information about using DescribeChangeSet to get the status of a change request, see [Working with change sets](#).

Error code	Error message
VALIDATION_FAILED	Provide Description information.
VALIDATION_FAILED	Provide Versions information.
VALIDATION_FAILED	Provide Dimensions information.
VALIDATION_FAILED	Provide Description PromotionalResources SupportInformation information.

## Find your product ID

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**, **SaaSProduct**, or **DataProduct** 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.

The following topics explain how to find a product by filtering on entity id, product title, last modified date, or visibility.

**Topics**

- [Find a product based on product title](#)
- [Find a product based on last modified date](#)
- [Find a product based on product visibility](#)
- [Find a product based on product title, last modified date, and product visibility](#)
- [Get additional details about a product](#)

**Find a product based on product title****Request**

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

{
  "Catalog": "AWSMarketplace",
  "EntityType": "AmiProduct",
  "MaxResults": 10,
  "EntityTypeFilters": {
    "AmiProductFilters": {
      "ProductTitle": {
        "WildcardValue": "XYZ"
      }
    }
  }
}
```

## Response

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

{
  "EntitySummaryList": [
    {
      "EntityArn": "arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/
AmiProduct/example-abcd-1234",
      "EntityId": "example1-abcd-1234-5ef6-7890abcdef12@1",
      "EntityType": "AmiProduct",
      "LastModifiedDate": "2018-02-27T13:45:22Z",
      "AmiProductSummary": {
        "ProductTitle": "ABC-XYZ-123",
        "Visibility": "Public"
      }
    }
  ],
  "NextToken": ""
}
```

## Find a product based on last modified date

### Request

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

{
  "Catalog": "AWSMarketplace",
  "EntityType": "AmiProduct",
  "MaxResults": 10,
  "EntityTypeFilters": {
    "AmiProductFilters": {
      "LastModifiedDate": {
        "DateRange": {
          "BeforeValue": "2018-03-27T13:45:22Z",
          "AfterValue": "2018-01-27T13:45:22Z"
        }
      }
    }
  }
}
```

```
}  
}
```

## Response

```
HTTP/1.1 200  
Content-type: application/json  
  
{  
  "EntitySummaryList": [  
    {  
      "EntityArn": "arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/  
AmiProduct/example-abcd-1234",  
      "EntityId": "example1-abcd-1234-5ef6-7890abcdef12@1",  
      "EntityType": "AmiProduct",  
      "LastModifiedDate": "2018-02-27T13:45:22Z",  
      "AmiProductSummary": {  
        "ProductTitle": "ABC-XYZ-123",  
        "Visibility": "Public"  
      }  
    }  
  ],  
  "NextToken": ""  
}
```

## Find a product based on product visibility

### Request

```
POST /ListEntities HTTP/1.1  
Content-Type: application/json  
  
{  
  "Catalog": "AWSMarketplace",  
  "EntityType": "AmiProduct",  
  "MaxResults": 10,  
  "EntityTypeFilters": {  
    "AmiProductFilters": {  
      "Visibility": {  
        "ValueList": [  
          "Public"  
        ]  
      }  
    }  
  }  
}
```

```
    ]
  }
}
```

## Response

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

{
  "EntitySummaryList": [
    {
      "EntityArn": "arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/
AmiProduct/example-abcd-1234",
      "EntityId": "example1-abcd-1234-5ef6-7890abcdef12@1",
      "EntityType": "AmiProduct",
      "LastModifiedDate": "2018-02-27T13:45:22Z",
      "AmiProductSummary": {
        "ProductTitle": "ABC-XYZ-123",
        "Visibility": "Public"
      }
    }
  ],
  "NextToken": ""
}
```

## Find a product based on product title, last modified date, and product visibility

### Request

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

{
  "Catalog": "AWSMarketplace",
  "EntityType": "AmiProduct",
  "MaxResults": 10,
  "EntityTypeFilters": {
```

```
    "AmiProductFilters": {
      "LastModifiedDate": {
        "DateRange": {
          "BeforeValue": "2018-03-27T13:45:22Z",
          "AfterValue": "2018-01-27T13:45:22Z"
        }
      },
      "Visibility": {
        "ValueList": [
          "Public"
        ]
      },
      "ProductTitle": {
        "ValueList": [
          "ABC-XYZ-123"
        ]
      }
    }
  }
}
```

## Response

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

{
  "EntitySummaryList": [
    {
      "EntityArn": "arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/AmiProduct/example-abcd-1234",
      "EntityId": "example1-abcd-1234-5ef6-7890abcdef12@1",
      "EntityType": "AmiProduct",
      "LastModifiedDate": "2018-02-27T13:45:22Z",
      "AmiProductSummary": {
        "ProductTitle": "ABC-XYZ-123",
        "Visibility": "Public"
      }
    }
  ],
  "NextToken": ""
}
```

## Get additional details about a product

You can get additional details about the product using the entity id with the DescribeEntity action.

### Request

```
GET /DescribeEntity?catalog=AWSMarketplace&entityId=example-abcd-1234 HTTP/1.1
```

### Response

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

{
  "DetailsDocument": {
    "ProductTitle": "ABC-XYZ-123",
    "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"
    ]
  }
  "EntityArn": "arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/
AmiProduct/example-abcd-1234",
  "EntityId": "example1-abcd-1234-5ef6-7890abcdef12@1",
  "EntityType": "AmiProduct",
  "LastModifiedDate": "2018-02-27T13:45:22Z",
}
```

## 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](#).

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"
  },
  "Details": "{\\"Version\\": {\\"VersionTitle\\": \\"1.1\\",\\"ReleaseNotes\\": \\"Minor
bug fix\\"},\\"DeliveryOptions\\": [{\\"DeliveryOptionTitle\\": \\"EKSDelivery\\",\\"Details
\\": {\\"EcrDeliveryOptionDetails\\": {\\"ContainerImages\\": [\\"111122223333.dkr.ecr.us-
east-1.amazonaws.com/sellername/reponame1:1.1\\"],\\"DeploymentResources\\": [{\\"Name\\":
\\"HelmDeploymentTemplate\\",\\"Url\\": \\"111122223333.dkr.ecr.us-east-1.amazonaws.com/
sellername/reponame2:mychart1.1\\"}],\\"CompatibleServices\\": [\\"EKS\\"],\\"Description
\\": \\"Sample Description\\",\\"UsageInstructions\\":\\"helm pull 111122223333.dkr.ecr.us-
east-1.amazonaws.com/sellername/reponame2:mychart1.1\\"}}},{\\"DeliveryOptionTitle
\\": \\"HelmChartDeliveryOption\\",\\"Details\\": {\\"HelmDeliveryOptionDetails\\":
{\\"CompatibleServices\\": [\\"EKS\\", \\"EKS-Anywhere\\"],\\"ContainerImages\\":
[\\"111122223333.dkr.ecr.us-east-1.amazonaws.com/sellername/reponame1:1.1\\"],
\\"HelmChartUri\\": \\"111122223333.dkr.ecr.us-east-1.amazonaws.com/sellername/
reponame1:helmchart1.1\\",\\"Description\\": \\"Helm chart description\\",
\\"UsageInstructions\\": \\"Usage instructions\\",\\"QuickLaunchEnabled\\": true,
\\"MarketplaceServiceAccountName\\": \\"Service account name\\",\\"ReleaseName\\": \\"Optional
release name\\",\\"Namespace\\": \\"Optional Kubernetes namespace\\",\\"OverrideParameters
\\": [{\\"Key\\": \\"HelmKeyName1\\",\\"DefaultValue\\": \\"${AWSMP_LICENSE_SECRET}\\",
\\"Metadata\\": {\\"Label\\": \\"AWS CloudFormation template field label\\",\\"Description\\":
\\"AWS CloudFormation template field description\\",\\"Obfuscate\\": false}},{\\"Key\\":
\\"HelmKeyName2\\",\\"DefaultValue\\": \\"${AWSMP_SERVICE_ACCOUNT}\\",\\"Metadata\\": {\\"Label
\\": \\"AWS CloudFormation template field label\\",\\"Description\\": \\"AWS CloudFormation
template field description\\",\\"Obfuscate\\": false}}]}]}]}]}]}"}",
  "DetailsDocument":
  {
    "Version":
    {
      "VersionTitle": "1.1",
      "ReleaseNotes": "Minor bug fix"
    },
    "DeliveryOptions":
    [
      {
        "DeliveryOptionTitle": "EKSDelivery",
        "Details":
        {
          "EcrDeliveryOptionDetails":
          {
            "ContainerImages":
            [
              "111122223333.dkr.ecr.us-east-1.amazonaws.com/sellername/
reponame1:1.1"
            ],
          }
        }
      }
    ]
  }

```

```

        "DeploymentResources":
        [
            {
                "Name": "HelmDeploymentTemplate",
                "Url": "111122223333.dkr.ecr.us-east-1.amazonaws.com/sellername/
reponame2:mychart1.1"
            }
        ],
        "CompatibleServices":
        [
            "EKS"
        ],
        "Description": "Sample Description",
        "UsageInstructions": "helm pull 111122223333.dkr.ecr.us-
east-1.amazonaws.com/sellername/reponame2:mychart1.1"
    }
},
{
    "DeliveryOptionTitle": "HelmChartDeliveryOption",
    "Details":
    {
        "HelmDeliveryOptionDetails":
        {
            "CompatibleServices":
            [
                "EKS",
                "EKS-Anywhere"
            ],
            "ContainerImages":
            [
                "111122223333.dkr.ecr.us-east-1.amazonaws.com/sellername/
reponame1:1.1"
            ],
            "HelmChartUri": "111122223333.dkr.ecr.us-east-1.amazonaws.com/
sellername/reponame1:helmchart1.1",
            "Description": "Helm chart description",
            "UsageInstructions": "Usage instructions",
            "QuickLaunchEnabled": true,
            "MarketplaceServiceAccountName": "Service account name",
            "ReleaseName": "Optional release name",
            "Namespace": "Optional Kubernetes namespace",
            "OverrideParameters":
            [

```



```

"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"
    },
    "Details": "{\"Version\": {\"VersionTitle\": \"1.1\", \"ReleaseNotes\": \"Minor
bug fix\"}, \"DeliveryOptions\": [{\"DeliveryOptionTitle\": \"EKSDelivery\", \"Details
\": {\"EcrDeliveryOptionDetails\": {\"ContainerImages\": [\"111122223333.dkr.ecr.us-
east-1.amazonaws.com/sellername/reponame1:1.1\"], \"DeploymentResources\": [{\"Name\":
\"HelmDeploymentTemplate\", \"Url\": \"111122223333.dkr.ecr.us-east-1.amazonaws.com/
sellername/reponame2:mychart1.1\"}], \"CompatibleServices\": [\"EKS\"], \"Description
\": \"Sample Description\", \"UsageInstructions\": \"helm pull 111122223333.dkr.ecr.us-
east-1.amazonaws.com/sellername/reponame2:mychart1.1\"}}], {\"DeliveryOptionTitle
\": \"HelmChartDeliveryOption\", \"Details\": {\"HelmDeliveryOptionDetails\":
{\"CompatibleServices\": [\"EKS\", \"EKS-Anywhere\"], \"ContainerImages\":
[\"111122223333.dkr.ecr.us-east-1.amazonaws.com/sellername/reponame1:1.1\"],
\"HelmChartUri\": \"111122223333.dkr.ecr.us-east-1.amazonaws.com/sellername/
reponame1:helmchart1.1\", \"Description\": \"Helm chart description\",
\"UsageInstructions\": \"Usage instructions\", \"QuickLaunchEnabled\": true,
\"MarketplaceServiceAccountName\": \"Service account name\", \"ReleaseName\": \"Optional
release name\", \"Namespace\": \"Optional Kubernetes namespace\", \"OverrideParameters
\": [{\"Key\": \"HelmKeyName1\", \"DefaultValue\": \"${AWSMP_LICENSE_SECRET}\"},
\"Metadata\": {\"Label\": \"AWS CloudFormation template field label\", \"Description\":
\"AWS CloudFormation template field description\", \"Obfuscate\": false}}, {\"Key\":
\"HelmKeyName2\", \"DefaultValue\": \"${AWSMP_SERVICE_ACCOUNT}\", \"Metadata\": {\"Label
\": \"AWS CloudFormation template field label\", \"Description\": \"AWS CloudFormation
template field description\", \"Obfuscate\": false}}]}]}\"},
    "DetailsDocument":
    {
      "Version":
      {
        "VersionTitle": "1.1",

```

```

    "ReleaseNotes": "Minor bug fix"
  },
  "DeliveryOptions":
  [
    {
      "DeliveryOptionTitle": "EKSDelivery",
      "Details":
      {
        "EcrDeliveryOptionDetails":
        {
          "ContainerImages":
          [
            "111122223333.dkr.ecr.us-east-1.amazonaws.com/sellername/
reponame1:1.1"
          ],
          "DeploymentResources":
          [
            {
              "Name": "HelmDeploymentTemplate",
              "Url": "111122223333.dkr.ecr.us-east-1.amazonaws.com/sellername/
reponame2:mychart1.1"
            }
          ],
          "CompatibleServices":
          [
            "EKS"
          ],
          "Description": "Sample Description",
          "UsageInstructions": "helm pull 111122223333.dkr.ecr.us-
east-1.amazonaws.com/sellername/reponame2:mychart1.1"
        }
      }
    },
    {
      "DeliveryOptionTitle": "HelmChartDeliveryOption",
      "Details":
      {
        "HelmDeliveryOptionDetails":
        {
          "CompatibleServices":
          [
            "EKS",
            "EKS-Anywhere"
          ],

```



```
        "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.

For information about creating a single AMI-based product using the Catalog API, see [Create a product](#).

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

### Topics

- [Add a new version](#)
- [Update version information](#)
- [Restrict a version](#)
- [Update future AWS Region support](#)

- [Add a supported AWS Region](#)
- [Restrict an AWS Region](#)
- [Add a new instance type](#)
- [Restrict an instance type](#)

### Note

You can also create a single AMI-based product using the AWS Marketplace Management Portal. For more information, 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).

## Add a new version

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*.

To add a new version, call the `StartChangeSet` API operation with the `AddDeliveryOptions` change type for single-AMI products, as shown in the following example. To test your API call without actually creating a new version, set the `Intent` parameter to `VALIDATE`. For more information, see [Intent](#).

### 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.

## Request Syntax

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

{
```

```

"Catalog": "AWSMarketplace",
"ChangeSet":
[
  {
    "ChangeType": "AddDeliveryOptions",
    "Entity":
    {
      "Identifier": "example1-abcd-1234-5ef6-7890abcdef12@1",
      "Type": "AmiProduct@1.0"
    },
    "DetailsDocument":
    {
      "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"
      ]
    }
  ]
}
],
"Intent": "APPLY"
}

```

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) (required) – Your AMI-based product.
  - **Identifier** (string) (required) – Your product ID. For more information, see [Identifier](#).
  - **Type** (string) (required) – The Type is based on the delivery method (product type) that your product will be using: `AmiProduct@1.0`.
- **DetailsDocument** (object) – Details of the request. It includes all the information about the new version of your AMI-based product.
- **Versio**n (object) – Details about the software version you are adding. Made up of a title and release notes.
  - **Versio**nTitle (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.

- **AmId** (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.

## Response Syntax

A change set is created for your request. The response to this request gives you the `ChangeSetId` and `ChangeSetArn` 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 included validating information to ensure it meets the AWS Marketplace guidelines. The validation process can take anywhere from a few minutes to a few hours.

You can check the status of the request through the AWS Marketplace Management Portal, or directly through Catalog API using the [DescribeChangeSet](#) API operation.

For more information about errors in seller product change sets, see [Change set status and errors](#).

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*.

## Asynchronous Errors

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

Error code	Error message
INVALID_PRODUCT	Use an existing limited or public product.
DUPLICATE_VERSION_TITLE	The version title must be different from any other version titles of this product.
INVALID_VERSION_TITLE	Remove spaces before the trademark symbol.
INVALID_VERSION_TITLE	Remove unsupported characters: [x, y, z]

Error code	Error message
INVALID_VERSION_TITLE	Remove spaces from the beginning of the version title.
INVALID_VERSION_TITLE	Provide version title with fewer than [x] characters.
INVALID_RELEASE_NOTES	Remove spaces before the trademark symbol.
INVALID_RELEASE_NOTES	Remove unsupported characters: [x, y, z]
INVALID_RELEASE_NOTES	Remove spaces from the beginning of release notes.
INVALID_RELEASE_NOTES	Provide release notes with fewer than (x) characters.
INVALID_USAGE_INSTRUCTIONS	Remove spaces before the trademark symbol.
INVALID_USAGE_INSTRUCTIONS	Remove unsupported characters: [x, y, z]
INVALID_USAGE_INSTRUCTIONS	Remove spaces from the beginning of release notes.
INVALID_USAGE_INSTRUCTIONS	Provide usage instructions with fewer than (x) characters.
RECOMMENDED_INSTANCE_TYPE_NOT_AVAILABLE	Provide an existing, available instance type.
INVALID_RECOMMENDED_INSTANCE_TYPE	Provide a valid instance type.
INVALID_SECURITY_GROUP	Security group ports must be between 1 and [max].
INVALID_SECURITY_GROUP	Provide a value for CIDR IP ranges.
INVALID_SECURITY_GROUP	Provide security group start port that is not greater than end port.

Error code	Error message
INVALID_SECURITY_GROUP_PROTOCOL	Security group protocol must either be 'tcp' or 'udp'.
INVALID_CIDR_IP	Provide standard CIDR IP range in form '0.0.0.0/0'.
INVALID_ACCESS_ENDPOINT_PORT	Provide endpoint port less than [x].
INVALID_ACCESS_ENDPOINT_PORT	Provide endpoint port between 1 and [max].
INVALID_ACCESS_ENDPOINT_PORT	Provide endpoint port.
INVALID_ACCESS_ENDPOINT_RELATIVE_PATH	Remove spaces in the relative path.
INVALID_ACCESS_ENDPOINT_RELATIVE_PATH	Remove preceding '/' from relative path.
INCOMPATIBLE_OPERATING_SYSTEM	Provide operating system name and version that is compatible with instance types: [x]
INCOMPATIBLE_OPERATING_SYSTEM_NAME	Provide name with fewer than (x) characters.
INCOMPATIBLE_OPERATING_SYSTEM_NAME	Provide operating system name that is supported.
INCOMPATIBLE_OPERATING_SYSTEM_VERSION	Provide version with fewer than (x) characters.
INVALID_SCANNING_PORT	Provide scanning port between 1 and [max].
INVALID_AMI_ID	Provide valid AMI ID.
EXISTING_AMI_PRODUCT_CODE	Remove product code attached to image X.
INVALID_AMI_ARCHITECTURE	Provide new AMI with architecture [x].
INVALID_AMI_VIRTUALIZATION_TYPE	Provide new AMI with virtualization type [x].
INVALID_AMI_VIRTUALIZATION_TYPE	Provide expected [z] volume on image [x].

Error code	Error message
INCOMPATIBLE_AMI	Provide new AMI as architecture [x] on [y] is not supported by following instance types: [z]
INCOMPATIBLE_AMI	Provide new AMI as virtualization type [x] on [y] is not supported by following instance types: [z]
INCOMPATIBLE_AMI	Enable ENA support for image x because following instance types require ENA support: [y]
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.
ASSET_ACCESS_EXCEPTION	Unable to copy AMI [x] into AWS Marketplace account.
SCAN_ERROR	Fix security vulnerability [y] on Image [x].

## Update version information

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

### Note

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.

To add a new version, call the StartChangeSet API operation with the UpdateDeliveryOptions change type, as shown in the following example.

## Request Syntax

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

{
  "Catalog": "AWSMarketplace",
  "ChangeSet":
  [
    {
      "ChangeType": "UpdateDeliveryOptions",
      "Entity":
      {
        "Identifier": "example1-abcd-1234-5ef6-7890abcdef12@1",
        "Type": "AmiProduct@1.0"
      },
      "DetailsDocument":
      {
        "Version":
        {
          "ReleaseNotes": "*My new Release notes*"
        },
        "DeliveryOptions":
        [
          {
            "Id": "example1-2222-cccc-2222-cccccccccccc",
            "Details":
            {
              "AmiDeliveryOptionDetails":
              {
                "UsageInstructions": "Easy to use AMI"
              }
            }
          }
        ]
      }
    }
  ]
}
```

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) (required) – Your AMI-based product.
  - **Identifier** (string) (required) – Your product ID. For more information, see [Identifier](#).
  - **Type** (string) (required) – The Type is based on the delivery method (product type) that your product will be using: `AmiProduct@1.0`.
- **DetailsDocument** (object) – Details of the request. It includes any information about the version of your AMI-based product that you would like to update. 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 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.

## Response Syntax

A change set is created for your request. The response to this request gives you the `ChangeSetId` and `ChangeSetArn` 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 included validating information to ensure it meets the AWS Marketplace guidelines. The validation process can take anywhere from a few minutes to a few hours.

You can check the status of the request through the AWS Marketplace Management Portal, or directly through Catalog API using the [DescribeChangeSet](#) API operation.

For more information about change sets, see [Working with change sets](#). For more information about errors in seller product change sets, see [Change set status and errors](#).

## Asynchronous Errors

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

Error code	Error message
INVALID_PRODUCT	Use an existing limited or public product.
MISSING_DELIVERY_OPTION_IDS	Provide at least one delivery option ID.

Error code	Error message
INVALID_DELIVERY_OPTION_IDS	Provide delivery option IDs that can be found in the product. IDs not found: [x]
INVALID_DELIVERY_OPTIONS	Provide delivery option IDs that belong to the same version.

## Restrict a version

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.

### Note

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*.

To restrict a version, call the `StartChangeSet` API operation with the `RestrictDeliveryOptions` change type, as shown in the following example.

### Note

All subscribers can use the current version regardless of the restriction status. AWS Marketplace guidelines require that you continue to offer support to existing buyers for 90 days after restricting the version. Your AMI will be marked as deprecated after the version is restricted. For more information, see [Deprecate an AMI](#) in the *Amazon Elastic Compute Cloud User Guide for Windows Instances*.

## Request Syntax

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

```
{
  "Catalog": "AWSMarketplace",
  "ChangeSet":
  [
    {
      "ChangeType": "RestrictDeliveryOptions",
      "Entity":
      {
        "Identifier": "example1-abcd-1234-5ef6-7890abcdef12@1",
        "Type": "AmiProduct@1.0"
      },
      "DetailsDocument":
      {
        "DeliveryOptionIds":
        [
          "example1-2222-cccc-2222-cccccccccccc"
        ]
      }
    }
  ]
}
```

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

- **Entity** (object) (required) – Your AMI-based product.
  - **Identifier** (string) (required) – Your product ID. For more information, see [Identifier](#).
  - **Type** (string) (required) – The Type is based on the delivery method (product type) that your product will be using: `AmiProduct@1.0`.
- **DetailsDocument** (object) – Details of the request. It includes IDs for the versions of your AMI-based product that you would like to restrict.
  - **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.

## Response Syntax

A change set is created for your request. The response to this request gives you the `ChangeSetId` and `ChangeSetArn` 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 included validating information to ensure it meets the AWS Marketplace guidelines. The validation process can take anywhere from a few minutes to a few hours.

You can check the status of the request through the AWS Marketplace Management Portal, or directly through Catalog API using the [DescribeChangeSet](#) API operation.

### Asynchronous Errors

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

Error code	Error message
INVALID_PRODUCT	Use an existing public product.
MISSING_DELIVERY_OPTION_IDS	Provide at least one delivery option ID.
INVALID_DELIVERY_OPTION_IDS	Provide delivery option IDs that can be found in the product. IDs not found: [x]
INVALID_DELIVERY_OPTION	Provide delivery option IDs that are in a public state. IDs not in public state: [x]
ALL_DELIVERY_OPTIONS_RESTRICTED	Provide fewer delivery options to restrict as at least one must remain in public state.

## Update future AWS Region support

You can use the Catalog API to change future AWS Region support preferences for your AMI-based product in AWS Marketplace.

**Note**

For more information about changing future Region support using the AWS Marketplace Management Portal, see [Update support for future AWS Regions](#) in the *AWS Marketplace Seller Guide*.

**Note**

The `UpdateFutureRegionSupport` change type is only available on `AmiProduct@1.0`.

To change future AWS Region support preferences, call the `StartChangeSet` API operation with the `UpdateFutureRegionSupport` change type, as shown in the following example.

**Request Syntax**

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

{
  "Catalog": "AWSMarketplace",
  "ChangeSet":
  [
    {
      "ChangeType": "UpdateFutureRegionSupport",
      "Entity":
      {
        "Identifier": "prod-example12345",
        "Type": "AmiProduct@1.0"
      },
      "DetailsDocument":
      {
        "FutureRegionSupport":
        {
          "SupportedRegions":
          [
            "All"
          ]
        }
      }
    }
  ]
}
```

```
    }  
  ]  
}
```

Provide information for the fields to add the `UpdateFutureRegionSupport` change type:

- **Entity** (object) (required) – Your AMI-based product.
  - **Identifier** (string) (required) – Your product ID. For more information, see [Identifier](#).
  - **Type** (string) (required) – The Type is based on the delivery method (product type) that your product will be using: `AmiProduct@1.0`.
- **DetailsDocument** (object) (required) – The details required to execute the `ChangeSet`.
  - **FutureRegionSupport** – Object
    - **SupportedRegions** – Single-element array of strings

Element supported values: one of ["All", "US", "None"]

## Response Syntax

A change set is created for your request. The response to this request gives you the `ChangeSetId` and `ChangeSetArn` 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 included validating information to ensure it meets the AWS Marketplace guidelines. The validation process can take anywhere from a few minutes to a few hours.

You can check the status of the request through the AWS Marketplace Management Portal, or directly through Catalog API using the [DescribeChangeSet](#) API operation.

## Asynchronous Errors

The following errors are specific to `UpdateFutureRegionSupport` actions in the AWS Marketplace Catalog API. These errors are returned when you call `DescribeChangeSet` after a

change set is processing. For more information about using `DescribeChangeSet` to get the status of a change request, see [Working with change sets](#).

Error code	Error message
TOO_MANY_REGIONS	Currently, only 1 value is supported for <code>FutureRegionSupport</code> : All, US, or None
INVALID_REGIONS	Requested Regions [a, b, c] are invalid or unavailable. Only supported values are [x, y, z].
INVALID_INPUT	<code>SupportedRegions</code> can't be empty.

## Add a supported AWS Region

You can use the Catalog API to add new supported AWS Regions for your AMI-based product in AWS Marketplace.

### Note

For more information about adding new supported Regions using the AWS Marketplace Management Portal, see [Add an AWS Region](#) in the *AWS Marketplace Seller Guide*.

### Note

The `AddRegions` change type is only available on `AmiProduct@1.0`.

To add new supported Regions, call the `StartChangeSet` API operation with the `AddRegions` change type, as shown in the following example.

### Request Syntax

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

```
{
  "Catalog": "AWSMarketplace",
  "ChangeSet":
  [
    {
      "ChangeType": "AddRegions",
      "Entity":
      {
        "Identifier": "prod-example12345",
        "Type": "AmiProduct@1.0"
      },
      "DetailsDocument":
      {
        "Regions":
        [
          "us-east-1",
          "ap-northeast-2"
        ]
      }
    }
  ]
}
```

Provide information for the fields to add the AddRegions change type.

- **Entity** (object) (required) – Your AMI-based product.
- **Identifier** (string) (required) – Your product ID. For more information, see [Identifier](#).
- **Type** (string) (required) – The Type is based on the delivery method (product type) that your product will be using: AmiProduct@1.0.
- **Example**

<caption>**DetailsDocument** (object) (required) – The details required to execute the ChangeSet.</caption>

- **Regions:** Array of strings

Element supported values: Valid AWS Region code strings.

For example, ["us-east- 1"].

## Response Syntax

Add a supported AWS Region

A change set is created for your request. The response to this request gives you the `ChangeSetId` and `ChangeSetArn` 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 included validating information to ensure it meets the AWS Marketplace guidelines. The validation process can take anywhere from a few minutes to a few hours.

You can check the status of the request through the AWS Marketplace Management Portal, or directly through Catalog API using the [DescribeChangeSet](#) API operation.

- If the change set execution status is `SUCCEEDED`: A new `Entity Identifier` (or `EntityId`) is generated. You can use the [DescribeEntity](#) API operation on the product entity to check the result.
- If the change set execution status is `CLIENT_ERROR`: The `DescribeChangeSet` response gives the details of the error, as well as corresponding actions to take to fix the error.

## Asynchronous Errors

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

Error code	Error message
INVALID_REGIONS	Requested regions [a, b, c] are invalid or unavailable. Only supported values are [x, y, z].
INVALID_INPUT	Regions can't be empty.

## Restrict an AWS Region

You can use the Catalog API to restrict previously supported AWS Regions for your AMI-based product in AWS Marketplace.

### Note

For more information about restricting previously supported Regions using the AWS Marketplace Management Portal, see [Restrict an AWS Region](#) in the *AWS Marketplace Seller Guide*.

### Note

The RestrictRegions change type is only available on AmiProduct@1.0.

To restrict previously supported Regions, call the StartChangeSet API operation with the RestrictRegions change type, as shown in the following example.

### Request Syntax

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

{
  "Catalog": "AWSMarketplace",
  "ChangeSet":
  [
    {
      "ChangeType": "RestrictRegions",
      "Entity":
      {
        "Identifier": "prod-example12345",
        "Type": "AmiProduct@1.0"
      },
      "DetailsDocument":
      {
        "Regions":
        [
```

```
        "us-east-1",
        "ap-northeast-2"
    ]
}
]
```

Provide information for the fields to add the AddRegions change type.

- **Entity** (object) (required) – Your AMI-based product.
  - **Identifier** (string) (required) – Your product ID. For more information, see [Identifier](#).
  - **Type** (string) (required) – The Type is based on the delivery method (product type) that your product will be using: AmiProduct@1.0.
- **DetailsDocument** (object) (required) – The details required to execute the ChangeSet, in this case InstanceTypes.
  - **Regions** – Array of strings

Element supported values: Valid AWS Region code strings. For example, ["us-east- 1"].

## Response Syntax

A change set is created for your request. The response to this request gives you the ChangeSetId and ChangeSetArn 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 included validating information to ensure it meets the AWS Marketplace guidelines. The validation process can take anywhere from a few minutes to a few hours.

You can check the status of the request through the AWS Marketplace Management Portal, or directly through Catalog API using the [DescribeChangeSet](#) API operation.

- If the change set execution status is SUCCEEDED – A new Entity Identifier (or EntityId) is generated. You can use DescribeEntity on the product entity to check the result. For more information, see [DescribeEntity](#).
- If the change set execution status is CLIENT\_ERROR: The DescribeChangeSet response gives the details of the error, as well as corresponding actions to take to fix the error.

## Add a new instance type

You can use the Catalog API to add new instance types for your AMI-based product in AWS Marketplace.

### Note

For more information about adding instance types using the AWS Marketplace Management Portal, see [Add an instance](#) in the *AWS Marketplace Seller Guide*.

AddInstanceTypes will add new instance types to existing products and newly created products when creating a product. The change type will update all versions in product document with a new instance type.

### Note

The AddInstanceTypes change type is only available on AmiProduct@1.0.

When adding a restricted instance type, the instance type can be removed from the restricted list and added to the available instance type list. This gives sellers more control to change their product restriction. The instance type list is interchangeable and not a permanent restricted status for a product.

For internally metered products, sellers need to call separate change types AddDimensions and UpdatePricingTerms to update pricing for the instance type.

To add new instance types, call the StartChangeSet API operation with the RestrictRegions change type, as shown in the following example.

## Request Syntax

Only AddInstanceTypes change type is shown below. Although internally metered AMI sellers are required to call AddInstanceTypes and UpdatePricingTerms change types for their AMI.

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

{
  "Catalog": "AWSMarketplace",
  "ChangeSet":
  [
    {
      "ChangeType": "AddInstanceTypes",
      "Entity":
      {
        "Identifier": "prod-example12345",
        "Type": "AmiProduct@1.0"
      },
      "DetailsDocument":
      {
        "InstanceTypes":
        [
          "m1.medium"
        ]
      }
    }
  ]
}
```

Provide information for the fields to add the AddInstanceTypes change type:

- **Entity** (object) (required) – Your AMI-based product.
  - **Identifier** (string) (required) – Your product ID. For more information, see [Identifier](#).
  - **Type** (string) (required) – The Type is based on the delivery method (product type) that your product will be using: AmiProduct@1.0.
- **DetailsDocument** (object) (required) – The details required to execute the ChangeSet, in this case InstanceTypes.
  - **InstanceTypes** (array of strings) (required) – List of InstanceTypes to add to the product. These instances will be added to the existing InstanceTypes.

## Response Syntax

A change set is created for your request. The response to this request gives you the `ChangeSetId` and `ChangeSetArn` 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 included validating information to ensure it meets the AWS Marketplace guidelines. The validation process can take anywhere from a few minutes to a few hours.

You can check the status of the request through the AWS Marketplace Management Portal, or directly through Catalog API using the [DescribeChangeSet](#) API operation.

## Synchronous Validations

The following schema validations are specific to `AddInstanceTypes` actions in the AWS Marketplace Catalog API. These validations are performed when you call `StartChangeSet`. If the request doesn't meet the following requirements, it will fail with an HTTP response.

Input field	Validation rule	HTTP
<code>InstanceTypes</code>	Required	422
<code>InstanceTypes</code>	Must not be empty	422
<code>InstanceTypes</code>	Entries must be between 1 to 24 characters long. Must match <code>^[A-Za-z0-9_-.]+\$</code>	422
<code>InstanceTypes</code>	Entries must be unique	422
<code>InstanceTypes</code>	Must not be more than 1500 entries	422
An unknown property	No additional properties are allowed	422

## Asynchronous Errors

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

Error code	Error message
INVALID_INSTANCE_TYPES	The following instance types are not valid: [x]
INVALID_INSTANCE_VIRTUALIZATION	Virtualization of the product is [x]. It is not supported by the following instance types: [x]
INVALID_AMI_ARCHITECTURE	CPU architecture of the product is '%s!' + "It is not supported by the following instance types: [x]
INCOMPATIBLE_OPERATING_SYSTEM	The instance types are incompatible with the OS defined in the product. Provide instance types that are compatible with the OS defined in the product.
INVALID_PRODUCT_TYPE	Use an existing single AMI product.
INVALID_ENA_SETTING	The product does not support ENA. ENA support is required by the following instance types: [x]
INVALID_DIMENSIONS	No internally metered dimensions found for instance types: [x]
UPDATE_PRICING_REQUIRED	UpdatePricingTerms change type is required when internally metered dimensions are available on the product.

## Restrict an instance type

You can use the Catalog API to limit or restrict the instance types available for your AMI-based product in AWS Marketplace.

**Note**

For more information about limiting or restrict the instance types available using the AWS Marketplace Management Portal, see [Restrict an instance](#) in the *AWS Marketplace Seller Guide*.

Existing subscribers won't be impacted by this change and they are able to use the restricted instance types. However, no new buyers will be able to use restricted instance types. To stop current instance types subscriptions (once instance types are restricted), you must contact the AWS Marketplace Seller Operations Team.

`RestrictInstanceTypes` restricts instance types to all the versions in the product document. In the `AddInstanceTypes` change type, you are updating all versions of the product. You won't be able to restrict the recommended instance types. The recommended instance type is at the version level, so it's possible that the seller won't be able to restrict multiple instance types.

For an internally metered product, you need to call separate change types when calling `RestrictDimensions`. This prevents new offers being created for the restricted instance types.

**Note**

The `RestrictInstanceTypes` change type is only available on `AmiProduct@1.0`.

To limit or restrict the instance types available for your AMI-based product, call the `StartChangeSet` API operation with the `RestrictInstanceTypes` change type, as shown in the following example.

**Request Syntax**

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

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

```
    "ChangeType": "RestrictInstanceTypes",
    "Entity":
    {
      "Identifier": "prod-example12345",
      "Type": "AmiProduct@1.0"
    },
    "DetailsDocument":
    {
      "InstanceTypes":
      [
        "m1.medium"
      ]
    }
  ]
}
```

Provide information for the fields to add the `RestrictInstanceTypes` change type.

- **Entity** (object) (required) – Your AMI-based product.
  - **Identifier** (string) (required) – Your product ID. For more information, see [Identifier](#).
  - **Type** (string) (required) – The Type is based on the delivery method (product type) that your product will be using: `AmiProduct@1.0`.
- **DetailsDocument** (object) (required) – The details required to execute the `ChangeSet`, in this case `InstanceTypes`.
  - **InstanceTypes** (array of strings) (required) – List of `InstanceTypes` to restrict to the product. These instances are added to the current (or if there are no existing instance types, it will add) to restricted `InstanceTypes`.

## Response Syntax

A change set is created for your request. The response to this request gives you the `ChangeSetId` and `ChangeSetArn` 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 included validating information to ensure it meets the AWS Marketplace guidelines. The validation process can take anywhere from a few minutes to a few hours.

You can check the status of the request through the AWS Marketplace Management Portal, or directly through Catalog API using the [DescribeChangeSet](#) API operation.

## Synchronous Validations

The following schema validations are specific to `RestrictInstanceTypes` actions in the AWS Marketplace Catalog API. These validations are performed when you call `StartChangeSet`. If the request doesn't meet the following requirements, it will fail with an HTTP response.

Input field	Validation rule	HTTP
InstanceTypes	Required	422
InstanceTypes	Must not be empty	422
InstanceTypes	Entries must be between 1 to 24 characters long. Must match <code>^[A-Za-z0-9_-.]+\$</code>	422
InstanceTypes	Entries must be unique	422
InstanceTypes	Must not be more than 1500 entries	422
An unknown property	No additional properties are allowed	422

## Asynchronous Errors

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

Error code	Error message
INVALID_INSTANCE_TYPES	The following instance types are not valid: [x]

Error code	Error message
INVALID_PRODUCT_TYPE	Use an existing single AMI product.
DUPLICATE_INSTANCE_TYPE	Provide instance types with no duplicates.
UNAVAILABLE_INSTANCE_TYPE	Provide an available instance type.
RECOMMENDED_INSTANCE_TYPE_RESTRICTED	The following instance types cannot be restricted. Recommended instance type must be changed to a different one before being restricted. Delivery Options Id [X] Instance Type[X]
DIMENSIONS_NOT_RESTRICTED	Restrict dimensions before restricting internally metered instance types: [x]
REGION_NO_INSTANCES	Your restricted instance types would cause product launch failure in region: X. Consider restricting fewer instances.

## Working with container-based products

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

For information about creating a container-based product using the Catalog API, see [Create a product](#).

The following topics describe how to use the Catalog API to perform actions on your container-based products:

### Topics

- [Add a new version](#)
- [Update the visibility for an Amazon EKS add-on](#)
- [Create repositories and resources](#)
- [Update version information](#)
- [Restrict a version](#)

## Add a new version

If you already have a container-based product in AWS Marketplace, you can use the AWS Marketplace 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 using the AWS Marketplace Management Portal, see [Getting started with container products](#) in the *AWS Marketplace Seller Guide*.

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 [Create repositories and resources](#).

To add a new version, call the `StartChangeSet` API operation 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`.

## Container Image Delivery Request Syntax

```
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"
    },
    "DetailsDocument":
    {
      "Version":
      {
        "VersionTitle": "1.1",
        "ReleaseNotes": "Minor bug fix"
      },
      "DeliveryOptions":
      [
        {
          "DeliveryOptionTitle": "EKS Container image only delivery option",
          "Details":
          {
            "EcrDeliveryOptionDetails":
            {
              "ContainerImages":
              [
                "111122223333.dkr.ecr.us-east-1.amazonaws.com/sellername/
reponame1:1.1"
              ],
              "DeploymentResources":
              [
                {
                  "Name": "HelmDeploymentTemplate",
                  "Url": "111122223333.dkr.ecr.us-east-1.amazonaws.com/sellername/
reponame2:mychart1.1"
                }
              ],
              "CompatibleServices":
              [
                "EKS"
              ],
              "Description": "Sample Description",
              "UsageInstructions": "helm pull 111122223333.dkr.ecr.us-
east-1.amazonaws.com/sellername/reponame2:mychart1.1"
            }
          }
        }
      ]
    }
  }
]
```

```

    }
  }
]
}
]
}
}

```

## Helm Chart Delivery Request Syntax

```

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"
      },
      "DetailsDocument":
      {
        "Version":
        {
          "VersionTitle": "1.1",
          "ReleaseNotes": "Minor bug fix"
        },
        "DeliveryOptions":
        [
          {
            "DeliveryOptionTitle": "HelmChartDeliveryOption",
            "Details":
            {
              "HelmDeliveryOptionDetails":
              {
                "CompatibleServices":
                [
                  "EKS",

```



```
]
}
```

## Amazon EKS Add-On Delivery Request Syntax

```
POST /StartChangeSet HTTP/1.1
Content-type: application/json
{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "AddDeliveryOptions",
      "Entity": {
        "Type": "ContainerProduct@1.0",
        "Identifier": "$CreateProductChange.Entity.Identifier"
      },
      "DetailsDocument": {
        "Version": {
          "VersionTitle": "1.1",
          "ReleaseNotes": "New Addon Release"
        },
        "DeliveryOptions": [
          {
            "DeliveryOptionTitle": "AWS Marketplace Test AddOn from CAPI 1",
            "Visibility": "Limited",
            "Details": {
              "EksAddOnDeliveryOptionDetails": {
                "ContainerImages": [
                  "111122223333.dkr.ecr.us-east-1.amazonaws.com/test-seller/canary-
test-repo-product-6:mongo"
                ],
                "HelmChartUri": "111122223333.dkr.ecr.us-east-1.amazonaws.com/rocket/
rocket-product-helm:1.0",
                "Description": "Description for delivery option provided by ISV",
                "UsageInstructions": "Usage instructions with launch instructions",
                "AddOnName": "aws-mp-test",
                "AddOnVersion": "1.2.1",
                "AddOnType": "networking",
                "CompatibleKubernetesVersions": [
                  "1.25",
                  "1.26"
                ],
                "SupportedArchitectures": [
```



- **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** (object) – 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.
  - **AddOnName** – A unique add-on name that buyers will see in the Amazon EKS catalog. This name will add a prefix later using `SellerAlias`. For example, where `isv-alias_` is the ISV provided add-on name.
  - **AddOnType** – The type of add-on chosen from a list of supported values from Amazon EKS: `Gitops | monitoring | logging | cert-management | policy-management | cost-management | autoscaling | storage | kubernetes-management | service-mesh | etcd-backup | ingress-service-type | load-balancer | local-registry | networking | Security | backup | ingress-controller | observability`
  - **AddOnVersion** – A semantic version so that buyer can choose a specific version of AddOn they need to install or upgrade.
  - **CompatibleKubernetesVersions** – The Amazon EKS Kubernetes versions that this software is compatible with.
  - **CompatibleServices** (array of strings) – An array of services that the release is compatible with. Valid options are ECS and EKS.
  - **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.
  - **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.
  - **EcrDeliveryOptionDetails** – **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.
  - **EnvironmentOverrideParameters** – List of system parameters to be used by the add-on. Some of the ISV provided AddOn (HelmChart) might require configurations

with information derived from the Amazon EKS execution environment state (/system information). For example, EksClusterRegion, EKSClusterName, and others. You can avoid additional actions from Buyer by dynamically substituting these values at Amazon EKS AddOn launch. Amazon EKS System already supports automatic substitutions of system param for addons. AWS Marketplace ISV experience can be extended to collect this params which would require substitution.

The generic system information to be substituted can be indicated by providing a AWS Marketplace specified constant following convention similar to Helm substitution. The supported values are `${AWS_REGION}` and `${AWS_EKS_CLUSTER_NAME}`.

```
"EnvironmentOverrideParameters" : [ {
  "Key" : "my-field.region"
  "Value" : "${AWS_REGION}"
},
{
  "Key" : "my-second-field"
  "Value" : "${AWS_EKS_CLUSTER_NAME}"
},
```

- `HelmDeliveryOptionDetails` - **HelmChartUri (string)** – The URL to the Helm chart hosted in Amazon ECR that the buyer will install to launch the software.
- `HelmDeliveryOptionDetails` - **QuickLaunchEnabled (boolean)** – A boolean to determine if buyers can use QuickLaunch to launch the software. For more information about QuickLaunch, see [QuickLaunch in AWS Marketplace](#).
- `HelmDeliveryOptionDetails` - **MarketplaceServiceAccountName (string)** – *Optional* – The name of the Kubernetes service account. The service account will be used to connect to AWS Identity and Access Management (IAM) for permissions to call AWS services.
- `HelmDeliveryOptionDetails` - **ReleaseName (string)** – *Optional* – The name for the Helm release provided to the `helm install` command that buyers use to launch the software. If not included, Helm will provide an automatically generated release name for you.
- `HelmDeliveryOptionDetails` - **Namespace (string)** – *Optional* – The Kubernetes namespace where the Helm chart will be installed.

- `HelmDeliveryOptionDetails` - **OverrideParameters (array of objects)** – Parameters that will be used in the Helm commands that launch the application. Buyers can override the default values.

 **Note**

For Amazon EKS Anywhere products, provide at least 1 override parameter for the license secret. Provide `DefaultValue` of `"${AWSMP_LICENSE_SECRET}"`. For paid products, provide at least 1 override parameter for service account configuration. Provide `DefaultValue` of `"${AWSMP_SERVICE_ACCOUNT}"`.

- `Key` (string) – The key for the parameter in dot notation (`override.example.key`).
- `DefaultValue` (string) – The default value for this override parameter.
- `Metadata` (array of objects)– *Required if `QuickLaunchEnabled` is set to **true*** – An array of objects that include details about the override parameter, including AWS CloudFormation template information.
  - `Label` (string) – The name of the field in the AWS CloudFormation stack creation form that buyers use during QuickLaunch.
  - `Description` (string) – The description of the field in the AWS CloudFormation stack creation form that buyers use during QuickLaunch.
  - `Obfuscate` (boolean) – A boolean to determine if sensitive information such as secrets and passwords are masked in AWS CloudFormation consoles, commands, and APIs.
- `Namespace` – The ISV provided namespace for add-on installation.
- `SupportedArchitectures` – The list of supported architectures, like `amd64` and `arm64`.
- `UsageInstructions` (string) – Provide instructions about the usage for this delivery option. Can be up to 4,000 characters.

## Response Syntax

A change set is created for your request. The response to this request gives you the `ChangeSetId` and `ChangeSetArn` 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 directly through Catalog API using the [DescribeChangeSet](#) API operation.

For more information about change sets, see [Working with change sets](#). For more information about errors in seller product change sets, see [Change set status and errors](#).

### Asynchronous Errors

The following errors are specific to AddDeliveryOptions actions in the AWS Marketplace Catalog API. These errors are returned when you call DescribeChangeSet after a change set is processing. For more information about using DescribeChangeSet to get the status of a change request, see [Working with change sets](#).

Error code	Error message
INCOMPATIBLE_PRODUCT_STATUS	Use an existing limited or public product.
INCOMPATIBLE_SERVICES	Provide a valid list of compatible services.
NO_SERVICE_SPECIFIED	Provide at least 1 compatible service.
DUPLICATE_COMPATIBLE_AWS_SERVICES	Provide unique list of compatible services.
INVALID_VERSION_TITLE	Remove spaces before the trademark symbol.
INVALID_VERSION_TITLE	Remove the following unsupported characters: [x, y, z]

Error code	Error message
INVALID_VERSION_TITLE	Remove spaces from the beginning of the version title.
INVALID_VERSION_TITLE	Provide version title with fewer than [x] characters.
DUPLICATE_VERSION_TITLE	The version title must be different from any other version titles of this product.
INVALID_RELEASE_NOTES	Remove spaces before the trademark symbol.
INVALID_RELEASE_NOTES	Remove unsupported characters: [x, y, z]
INVALID_RELEASE_NOTES	Remove spaces from the beginning of release notes.
INVALID_RELEASE_NOTES	Provide release notes with fewer than (x) characters.
INVALID_USAGE_INSTRUCTIONS	Remove spaces before the trademark symbol.
INVALID_USAGE_INSTRUCTIONS	Remove unsupported characters: [x, y, z]
INVALID_USAGE_INSTRUCTIONS	Provide usage instructions with fewer than (x) characters.
INVALID_USAGE_INSTRUCTIONS	Provide usage instructions.
MISSING_CONTAINER_IMAGES	Provide at least 1 container image.
NO_LICENSE_SECRET_KEYS	For Amazon EKS Anywhere products, provide 1 override parameter for license secret. Needs <code>DefaultValue</code> of <code>"\${AWSMP_LICENSE_SECRET}"</code> , see example in section.
TOO_MANY_CONTAINER_IMAGES	Provide fewer than 50 container images.
DUPLICATE_CONTAINER_IMAGES	Provide a unique list of container images.

Error code	Error message
INVALID_CONTAINER_IMAGES	Provide a valid URI for the container image.
INVALID_CONTAINER_IMAGE_URI	Provide a valid URI for the container image.
INVALID_CONTAINER_IMAGE_TAG	Avoid using 'latest' tag.
DUPLICATE_DELIVERY_OPTION_TITLES	Provide unique delivery option title.
INVALID_DELIVERY_OPTION_TITLES	Delivery option title already exists, retry with a different title.
INVALID_FULFILLMENT_OPTION_TITLE	Provide delivery option title with fewer than (x) characters.
DUPLICATE_DELIVERY_OPTION_TITLES	Provide unique delivery option title.
NO_SERVICE_ACCOUNT_CONFIGURATION	For paid products, provide 1 override parameter for service account configuration. Needs DefaultValue of "\${AWSMP_SERVICE_ACCOUNT}" , see example in section.
INVALID_DETAILS	Provided Details is not valid.
EMPTY_RESOURCE_NAME	Provide resource name.
EMPTY_RESOURCE_URL	Provide resource URL.
INVALID_RESOURCE_NAME	Provide resource name with fewer than 256 characters.
INVALID_RESOURCE_URL	Provide resource URL with fewer than 256 characters.
INVALID_SHORT_DESCRIPTION	Provide a short description with fewer than 1,000 characters.
INVALID_SHORT_DESCRIPTION	Provide short description.

Error code	Error message
SCAN_ERROR	Fix security vulnerability ""[y]"" on Image ""[x]"".
IMAGE_NOT_FOUND	Provide a valid public image URI.
INVALID_ARN	Provide a valid ARN for image access.
IMAGE_INACCESSIBLE	Provide a valid ARN for image access.
DUPLICATE_ADDON_NAME	The AddOn name you provided is already in use by a different product. Provide a different name.
DUPLICATE_ADDON_VERSION	The AddOn version you provided is already in use for the specified AddOn. Provide a different version.
INVALID_ADDON_TYPE	Provide an add-on type from the following supported Amazon EKS add-on types: %s.
INVALID_KUBERNETES_VERSION	Provide a valid list of supported Kubernetes versions from the current Amazon EKS supported versions: %s
DUPLICATE_KUBERNETES_VERSIONS	Provide a unique list of supported Kubernetes versions.
INVALID_ARCHITECTURE	Provide a valid list of supported architectures from the current Amazon EKS supported architectures: [amd64, arm64]
DUPLICATE_SUPPORTED_ARCHITECTURES	Provide a unique list of supported architectures.
INVALID_VISIBILITY_STATE	This state isn't supported for the EksAddOn delivery option. Provide a valid visibility state from the following allowed values: Limited.

Error code	Error message
INVALID_ENVIRONMENT_OVERRIDE_PARAMETER_VALUE	Provide a valid environment override parameter value from the following list of supported values: [\${AWS_REGION}, \${AWS_EKS_CLUSTER_NAME}]
DUPLICATE_ENVIRONMENT_OVERRIDE_PARAMETER_KEY	Remove duplicate keys from environment override parameters.
TOO_MANY_EKS_ADDON_DELIVERY_OPTIONS	Provide only one Amazon EKS add-on delivery option for the version.
INCOMPATIBLE_ADDON_NAME	The add-on names don't match. Reuse the existing add-on name from the public add-on version or previous add-on versions of this product. Only one add-on name is supported for each product.
INCOMPATIBLE_ADDON_TYPE	The add-on types don't match. Reuse the existing add-on type from the public add-on version or previous add-on versions of this product. Only one add-on type is supported for each product.
INCOMPATIBLE_ADDON_NAMESPACE	The add-on namespaces don't match. Reuse the existing add-on namespace from the public add-on version or previous add-on versions of this product. Only one add-on namespace is supported for each product.
INVALID_HELM_CHART_URI	Provide a Helm chart URI that follows the SemVer2 format. For example, 1.5.2
INCOMPATIBLE_HELM_OBJECTS(INVALID_HELM_OBJECTS)	Provide a Helm chart without using the following unsupported Helm Objects: <unsupported-objects>.

Error code	Error message
INVALID_DEPENDENT_HELM_CHARTS	Provide a Helm chart that contains the following dependent charts directly in the parent chart directory and not externally sourced: <invalid-subcharts>.
INVALID_HELM_SENSITIVE_CONFIG	Provide an advanced configuration schema without sensitive information or secrets. Keywords: <sensitive-parameters-identified>
INVALID_HELM_UNDECLARED_IMAGES	Provide the following Helm chart images within the delivery option of the request: <list-of-images>.
INVALID_HELM_CHART_IMAGES	Provide a Helm chart containing images within repositories created via the AddRepositories change type. External images: <images-identified>.
INVALID_HELM_LINT	Provide a Helm chart that successfully passes Helm lint.
INVALID_HELM_TEMPLATE	Provide a Helm chart that successfully passes Helm template.
INVALID_HELM_CHART	Provide a Helm chart that adheres to AWS Marketplace guidance identified in <a href="#">Helm Charts bulleted list</a> in the <i>AWS Marketplace Seller Guide</i> .
INVALID_ADDON_NAME	Provide an AddOn name that follows the following regex pattern: xx
INVALID_ADDON_NAMESPACE	Provide a namespace that starts with a letter or digit, and then a combination of letters, digits, and hyphens. For example, namespace, namespace-test.

Error code	Error message
INVALID_ADDON_NAME_PATTERN	Provide an add-on name that starts with a letter or digit, and then a combination of letters, digits, and hyphens. For example, test-addon, eksaddon
INVALID_ADDON_VERSION_PATTERN	Provide an add-on version using the following pattern: "<major>.<minor>.<patch>" (for example, 1.2.3, 0.1.2, 0.1.1)
EMPTY_DELIVERY_OPTION_IDS	Provide a list of delivery option IDs.

## Update the visibility for an Amazon EKS add-on

You can use the Catalog API to update visibility for an Amazon EKS add-on delivery option of your product version in AWS Marketplace. Container and Helm delivery options for your container product are automatically created with 'Public' visibility status.

### Note

The ability to update visibility of your product version is supported only for the Amazon EKS add-on delivery option from the listed versions. If your product isn't 'Public' already, submit a request to publish the product with 'Public' visibility status by using the AWS Marketplace Management Portal.

By default, when you create a version with the Amazon EKS add-on delivery option, it's published in 'Limited' status. A 'Limited' status means the product isn't publicly available across all the Regions for your buyers to use and deploy in an Amazon EKS cluster. You can update the visibility of the delivery option from 'Limited' to 'Public' by calling the `StartChangeSet` API operation with the `UpdateDeliveryOptionsVisibility` change type. Specify the `DeliveryOptions Id` from your product version that corresponds to the Amazon EKS add-on delivery option.

### Request Syntax

```
{
```

```
"Catalog": "AWSMarketplace",
"ChangeSet":
[
  {
    "ChangeType": "UpdateDeliveryOptionsVisibility",
    "Entity":
    {
      "Identifier": "prod-example12345",
      "Type": "ContainerProduct@1.0"
    },
    "DetailsDocument":
    {
      "DeliveryOptions":
      [
        {
          "Id": "do-1234567891234567891234",
          "TargetVisibility": "Public"
        }
      ]
    }
  }
]
```

To add the `UpdateDeliveryOptionsVisibility` change type, provide information for the following fields :

- **Entity (object) (required)** – Your container-based product.
  - **Identifier (string) (required)** – Your product ID. For more information, see [Identifier](#).
  - **Type (string) (required)** – The Type is based on the delivery method (product type) that your product uses: `ContainerProduct@1.0`.
- **DetailsDocument (object)** – Details of the request, including the information about the repositories that you want to create. The following fields are all required.
  - **DeliveryOptions (list of objects)** – List of `DeliveryOption` objects, including the details of each:
    - **Id (string)** – Unique identifier for the `DeliveryOption`. (To get the unique identifier for the `DeliveryOption`, call the `DescribeEntity` action on the product that you're updating.
    - **TargetVisibility** – The intended new visibility of the product.

## Response Syntax

A change set is created for your request. The response to this request gives you the `ChangeSetId` and `ChangeSetArn` for the change set.

```
{
  "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 through the AWS Marketplace Catalog API by using the [DescribeChangeSet](#) API operation.

For more information about change sets, see [Working with change sets](#). For more information about errors in seller product change sets, see [Change set status and errors](#).

## Asynchronous Errors

The following table shows errors that are specific to `AddDeliveryOptions` actions in the AWS Marketplace Catalog API. These errors are returned when you call `DescribeChangeSet` after a change set is processing. For more information about using `DescribeChangeSet` to get the status of a change request, see [Working with change sets](#).

Error code	Error message
EMPTY_DELIVERY_OPTION_IDS	Provide a list of delivery option IDs.
INVALID_VISIBILITY_STATE	The <code>TargetVisibility</code> option you provided is not supported. Please try again with an allowed option. The allowed option(s) are: <code>Public</code>

Error code	Error message
INVALID_DELIVERY_OPTION_IDS	You provided invalid delivery option details. Provide delivery option IDs that can be found in the product. IDs not found: [x]
DUPLICATE_DELIVERY_OPTION_IDS	Provide unique delivery option IDs.
AUDIT_ERROR	You haven't completed independent software vendor (ISV) testing for all compatible Amazon EKS cluster versions for your Amazon EKS add-on version(s). You must complete testing before we can release the delivery option(s).
INVALID_DELIVERY_OPTION_TYPE	The delivery option type you provided is not valid. Ensure that your delivery option is of type: EksAddOn and try again.
INCOMPATIBLE_HELM_OBJECTS	Provide a Helm chart without unsupported Helm Objects: Unsupported Helm objects are as follows: all Release objects (except .Name and .Namespace), Helm hooks, and lookup functions.
INCOMPATIBLE_ADDON_NAME	The add-on names don't match. Reuse the existing add-on name from the public add-on version or previous add-on versions of this product. Only one add-on is supported for each product.
NCOMPATIBLE_ADDON_TYPE	The add-on types don't match. Reuse the existing add-on type from the public add-on version or previous add-on versions of this product. Only one add-on is supported for each product.

Error code	Error message
INCOMPATIBLE_ADDON_NAMESPACE	The add-on namespace don't match. Reuse the existing add-on namespace from the public add-on version or previous add-on versions of this product. Only one add-on is supported for each product.

## Create repositories and resources

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, call `StartChangeSet` with the `AddRepositories` change type, as shown in the following example.

### Request Syntax

```
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"
      },
      "DetailsDocument":
      {
        "Repositories":
        [
          {
```

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

Provide information for the fields to add the `AddRepositories` change type:

For more information about creating repositories, see [Adding a new version](#) in the *AWS Marketplace Seller Guide*.

- **Entity** (object) (required) – Your container-based product.
  - **Identifier** (string) (required) – Your product ID. For more information, see [Identifier](#).
  - **Type** (string) (required) – The Type is based on the delivery method (product type) that your product will be using: `ContainerProduct@1.0`.
- **DetailsDocument** (object) – Details of the request. It includes the information about the repositories that you want to create. 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*.

## Asynchronous Errors

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

Error code	Error message
INVALID_ECR_REPOSITORY_NAME	Provide repository name in the format: 'nginx-web-app'
DUPLICATE_ECR_REPOSITORY_NAME	The repository name must be unique.
MISSING_REPOSITORY_INFORMATION	Provide at least 1 repository name.
INVALID_ECR_REPOSITORY_NAME	Maximum character length 256 reached. Character length count is inclusive of the seller namespace.

## Update version information

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.

To update an existing version of your container-based product, call the `StartChangeSet` API operation 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.

### Container Image Delivery Request Syntax

```
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"
      },
      "DetailsDocument": {
        "Version": {
          "ReleaseNotes": "New release notes",
          "VersionTitle": "Version 1.2"
        },
        "DeliveryOptions": [
          {
            "Id": "example4-2222-cccc-2222-cccccccccccc",
            "Details": {
              "EcrDeliveryOptionDetails": {
                "DeliveryOptionTitle": "New Delivery Option Title",
                "Description": "New description",
                "UsageInstructions": "New usage instructions",
                "CompatibleServices": [
                  "EKS"
                ]
              }
            }
          }
        ]
      }
    }
  ]
}
```

## Helm Chart Delivery Request Syntax

```

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"
      },
      "DetailsDocument":{
        "Version":{
          "ReleaseNotes":"New release notes",
          "VersionTitle":"Version 1.2"
        },
        "DeliveryOptions":[
          {
            "Id":"example5-2222-cccc-2222-cccccccccccc",
            "Details":{
              "HelmDeliveryOptionDetails":{
                "DeliveryOptionTitle":"New Delivery Option Title",
                "ContainerImages":[
                  "111122223333.dkr.ecr.us-east-1.amazonaws.com/sellername/
imagename:1.0"
                ],
                "HelmChartUri":"111122223333.dkr.ecr.us-east-1.amazonaws.com/
sellername/helmname:1.0",
                "CompatibleServices":[
                  "EKS-Anywhere"
                ],
                "Description":"New description",
                "UsageInstructions":"New usage instructions",
                "MarketplaceServiceAccountName":"new-service-account-name",
                "ReleaseName":"new-release-name",
                "Namespace":"new-cluster-namespace",
                "QuickLaunchEnabled":true,
                "OverrideParameters":[
                  {
                    "Key":"new.parameter.key",
                    "DefaultValue":"New parameter default value",

```

```

        "Metadata":{
            "Label":"New metadata label",
            "Description":"New metadata description",
            "Obfuscate":false
        }
    ]
}

```

## Amazon EKS Add-On Delivery Request Syntax

```

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"
      },
      "DetailsDocument":{
        "Version":{
          "ReleaseNotes":"New release notes",
          "VersionTitle":"Version 1.2"
        },
        "DeliveryOptions":[
          {
            "Id":"example4-2222-cccc-2222-cccccccccccc",
            "Details":{
              "EksAddOnDeliveryOptionDetails":{
                "ContainerImages":[
                  "709825985650.dkr.ecr.us-east-1.amazonaws.com/test-seller/
canary-test-repo-product-6:mongo"

```



- **Type** (string) (required) – The Type is based on the delivery method (product type) that your product will be using: `ContainerProduct@1.0`.
- **DetailsDocument** (object) – Details of the request. It includes any information about the version of your container-based product that you would like to update. 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** (object) – 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 container image 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.
- **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 a delivery option. Note that this nested details object does *not* need to be double-escaped.
- **HelmDeliveryOptionDetails** (object) – The details of the Helm chart 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. 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.
- **HelmChartUri** (string) – The URL to the Helm chart hosted in Amazon ECR that the buyer will install to launch the software.
- **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.

- **MarketplaceServiceAccountName** (string) – The name of the Kubernetes service account. The service account will be used to connect to AWS Identity and Access Management for permissions to call AWS services.
- **ReleaseName** (string) – The name for the Helm release provided to the `helm install` command that buyers use to launch the software.
- **Namespace** (string) – The Kubernetes namespace where the Helm chart will be installed.
- **QuickLaunchEnabled** (boolean) – A boolean to determine if buyers can use QuickLaunch to launch the software. For more information about QuickLaunch, see [QuickLaunch in AWS Marketplace](#).
- **OverrideParameters** (array of objects) – Parameters that will be used in the Helm commands that launch the application. Buyers can override the default values.
  - **Key** (string)– The key for the parameter in dot notation (`override.example.key`).
  - **DefaultValue** (string) – The default value for this override parameter.
  - **Metadata** (array of objects) – *Only required if QuickLaunchEnabled is set to **true*** – An array of objects that include details about the override parameter, including AWS CloudFormation template information.
    - **Label** (string) – The name of the field in the AWS CloudFormation stack creation form that buyers use during QuickLaunch.
    - **Description** (string) – The description of the field in the AWS CloudFormation stack creation form that buyers use during QuickLaunch.
    - **Obfuscate** (boolean) – A boolean to determine if sensitive information such as secrets and passwords are masked in AWS CloudFormation consoles, commands, and APIs.

## Response Syntax

A change set is created for your request. The response to this request gives you the `ChangeSetId` and `ChangeSetArn` 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 directly through Catalog API using the [DescribeChangeSet](#) API operation.

For more information about change sets, see [Working with change sets](#). For more information about errors in seller product change sets, see [Change set status and errors](#).

### Asynchronous Errors

The following errors are specific to UpdateDeliveryOptions actions in the AWS Marketplace Catalog API. These errors are returned when you call DescribeChangeSet after a change set is processing. For more information about using DescribeChangeSet to get the status of a change request, see [Working with change sets](#).

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

Error code	Error message
INVALID_RELEASE_NOTES	Remove spaces before the trademark symbol.
INVALID_RELEASE_NOTES	Remove unsupported characters: [x, y, z]
INVALID_RELEASE_NOTES	Remove spaces from the beginning of release notes.
INVALID_RELEASE_NOTES	Provide release notes with fewer than (x) characters.
INVALID_USAGE_INSTRUCTIONS	Remove spaces before the trademark symbol.
INVALID_USAGE_INSTRUCTIONS	Remove unsupported characters: [x, y, z]
INVALID_USAGE_INSTRUCTIONS	Provide usage instructions with fewer than (x) characters.
INVALID_USAGE_INSTRUCTIONS	Provide usage instructions.
MISSING_CONTAINER_IMAGES	Provide at least 1 container image.
TOO_MANY_CONTAINER_IMAGES	Provide fewer than 50 container images.
DUPLICATE_CONTAINER_IMAGES	Provide a unique list of container images.
INVALID_CONTAINER_IMAGES	Provide a valid URI for the container image.
INVALID_CONTAINER_IMAGE_URI	Provide a valid URI for the container image.
INVALID_CONTAINER_IMAGE_TAG	Avoid using 'latest' tag.
MISSING_DELIVERY_OPTION_IDS	Provide delivery option from existing list of Ids.
EMPTY_DELIVERY_OPTION_IDS	Provide non-empty list of delivery option IDs.
DUPLICATE_DELIVERY_OPTION_IDS	Provide unique delivery option IDs.
DUPLICATE_DELIVERY_OPTION_TITLES	Provide unique delivery option title.

Error code	Error message
INVALID_DELIVERY_OPTION_TITLES	Delivery option title already exists, retry with a different title.
INVALID_FULFILLMENT_OPTION_TITLE	Provide delivery option title with fewer than (x) characters.
DUPLICATE_DELIVERY_OPTION_TITLES	Provide unique delivery option title.
EMPTY_RESOURCE_NAME	Provide resource name.
EMPTY_RESOURCE_URL	Provide resource URL.
INVALID_RESOURCE_NAME	Provide resource name with fewer than 256 characters.
INVALID_RESOURCE_URL	Provide resource URL with fewer than 256 characters.
INVALID_SHORT_DESCRIPTION	Provide a short description with fewer than 1,000 characters.
INVALID_SHORT_DESCRIPTION	Provide short description.
NO_LICENSE_SECRET_KEYS	For Amazon EKS Anywhere products, provide 1 override parameter for license secret. Needs <code>DefaultValue</code> of <code>"\${AWSMP_LICENSE_SECRET}"</code> , see example in section.
NO_SERVICE_ACCOUNT_CONFIGURATION	For paid products, provide 1 override parameter for service account configuration. Needs <code>DefaultValue</code> of <code>"\${AWSMP_SERVICE_ACCOUNT}"</code> , see example in section.
SCAN_ERROR	Fix security vulnerability <code>""[y]""</code> on Image <code>""[x]""</code> .
FIELD_NOT_ALLOWED_TO_CHANGE	Field [x] cannot be changed.

Error code	Error message
INVALID_DELIVERY_OPTIONS_STATUS	Provide delivery options in public or limited state.
NO_CHANGE_FOUND	Provide at least 1 change.
MULTIPLE_VERSION_UPDATE	Provide delivery option IDs from the same version.

## Restrict a version

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.

To restrict a version, call the `StartChangeSet` API operation 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.

Restricting an Amazon EKS add-on is currently not supported through the Catalog API. Restricted versions are still available for existing customers.

## Request Syntax

```
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"
},
"DetailsDocument":
{
  "DeliveryOptionIds":
  [
    "example1-2222-cccc-2222-cccccccccccc"
  ]
}
]
}
```

Provide information for the fields to add the RestrictDeliveryOptions change type:

- **Entity** (object) (required) – Your container-based product.
  - **Identifier** (string) (required) – Your product ID. For more information, see [Identifier](#).
  - **Type** (string) (required) – The Type is based on the delivery method (product type) that your product will be using: ContainerProduct@1.0.
- **DetailsDocument** (object) – Details of the request. It includes IDs for the delivery options of your container-based product that you would like to restrict.
  - **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.

## Response Syntax

A change set is created for your request. The response to this request gives you the ChangeSetId and ChangeSetArn 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 directly through Catalog API using the [DescribeChangeSet](#) API operation.

For more information about change sets, see [Working with change sets](#). For more information about errors in seller product change sets, see [Change set status and errors](#).

## Asynchronous Errors

The following errors are specific to RestrictDeliveryOptions actions in the AWS Marketplace Catalog API. These errors are returned when you call DescribeChangeSet after a change set is processing. For more information about using DescribeChangeSet to get the status of a change request, see [Working with change sets](#).

Error code	Error message
INCOMPATIBLE_PRODUCT_STATUS	Use a public product.
MISSING_DELIVERY_OPTION_IDS	Provide delivery option from existing list of IDs.
INVALID_DELIVERY_OPTIONS_STATUS	Provide delivery options in public state.
EMPTY_DELIVERY_OPTION_IDS	Provide non-empty list of delivery option IDs.
INVALID_MINIMUM_PUBLIC_DELIVERY_OPTIONS	Cannot restrict all delivery option IDs.
DUPLICATE_DELIVERY_OPTION_IDS	Provide unique delivery option IDs.

## Working with SaaS products

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

For information about creating a SaaS-based product using the Catalog API, see [Create a product](#).

The following topics describe how to use the Catalog API to perform actions on your SaaS-based products:

## Topics

- [Add delivery options](#)
- [Update delivery options](#)
- [Update delivery option visibility](#)

## Add delivery options

You can use the Catalog API to add delivery options for a SaaS product in AWS Marketplace.

To add detailed information for delivery options, call the `StartChangeSet` API operation with the `AddDeliveryOptions` change type to add delivery details, as shown in the following example.

### Note

This is only supported for one delivery option: `SaaSUrlDeliveryOptionDetails`. It allows you to add the `FulfillmentUrl` to the SaaS product.

## Request Syntax

```
{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [{
    "ChangeType": "AddDeliveryOptions",
    "Entity": {
      "Identifier": "example1-abcd-1234-5ef6-7890abcdef12@1",
      "Type": "SaaSProduct@1.0"
    },
  },
  "Details": "{
    \"DeliveryOptions\": [{
      \"Details\": \"{
        \"SaaSUrlDeliveryOptionDetails\": {
          \"FulfillmentUrl\": \"url to seller's account registration/login page\",
          \"QuickLaunchEnabled\": true,
          \"LaunchUrl\": \"URL\",
          \"UsageInstructions\": \"Instructions\",

```



- **DeploymentTemplates** (array) – Deployment templates that customers can use to set up and configure the SaaS product and any related AWS resources.
- **Title** (string) – The display name of the deployment template.
- **Description** (string) – A description for what the deployment template contains.
- **IamPolicy** (string) – An IAM policy describing the permissions needed to deploy the template. Buyers can use this IAM policy to quickly deploy the template.
- **CloudFormationDetails** (object) – The details of a AWS CloudFormation template.
- **TemplateUrl** (string) – The URL for the deployment template.

 **Note**

For support obtaining your CloudFormation template URL, contact your AWS Marketplace business development partner or the [AWS Marketplace Seller Operations](#) team.

- **DefaultStackName** (string) – The default name used in CloudFormation when the seller creates the stack.
- **UsageInstructions** (string) – (Optional) Instructions for using this delivery option. Include documentation for manual steps for customers who won't use `DeploymentTemplates`.

## Response Syntax

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"
}
```

## Synchronous Validations

Error condition	Message	HTTP code
Invalid Fulfillment Url	Provide a valid fulfillment URL beginning with "https://".	422
Multiple URL Delivery Options	You provided more than one URL delivery option. Provide one URL delivery option.	422
Invalid Delivery Option Types	You provided invalid delivery option details. You must provide a valid parameter for SaaSUrlDeliveryOptionDetails.	422
Invalid Launch URL	Provide a valid launch URL beginning with "https://".	422
Missing Launch Url	Required parameter LaunchUrl is missing. You must provide a LaunchUrl .	422
Missing deployment templates	The deployment template is missing. Provide at least one deployment template.	422
Too many deployment templates	You cannot provide more than 20 deployment templates.	422
Invalid template URL	Quick Start URL is invalid. Provide deployment template URL that is published through AWS QuickStarts to Amazon S3. Invalid deployment templates URL: [x]	422
Invalid deployment template stack name	The deployment template stack name is invalid. Provide a valid stack name using only alphanumeric characters and hyphens. It must start with an alphabetical	422

Error condition	Message	HTTP code
	character and can't be longer than 128 characters.	
Duplicate deployment template title	You provided duplicate deployment template titles. Provide unique deployment template titles.	422
Duplicate deployment template URL	You provided duplicate deployment template urls. Provide unique deployment template titles.	422
Invalid deployment template type	The deployment template type is invalid. Provide a valid deployment template type. Supported values are ["CloudFormation@1.0"].	422
Invalid deployment template IAM policy	The deployment template IAM policy is invalid. Provide a valid IAM policy.	422
Invalid usage instructions	<ul style="list-style-type: none"> <li>• Images aren't supported by the usage instructions. Remove the image [x].</li> <li>• You provided a link to an invalid URL in the usage instructions: [x]. Provide a valid URL.</li> <li>• You provided a link with an unsupported URI scheme in the usage instructions. Use a supported scheme: ["http", "https", "tel", "mailto"].</li> </ul>	422

## Asynchronous Errors

Error code	Error message
DUPLICATE_DELIVERY_OPTIONS	You provided one or more delivery option types that already exist for this product. Provide a unique delivery option type or use <code>UpdateDeliveryOptions</code> if you intended to change an existing delivery option.
INVALID_FULFILLMENT_URL	The URL you provided returned HTTP status code [x]. Provide a fulfillment URL that renders with a 200.
INVALID_LAUNCH_URL	The URL you provided returned HTTP status code [x]. Provide a launch URL that renders with a 200.
INVALID_TEMPLATE_URL	Quick Start URL is invalid. Provide deployment template URL that is published through AWS QuickStarts to Amazon S3. Invalid deployment templates URL: [x]

## Update delivery options

You can use the Catalog API to update the delivery options for a SaaS product in AWS Marketplace.

To update the delivery options, call the `StartChangeSet` API operation with the `UpdateDeliveryOptions` change type, as shown in the following example.

### Note

This is only supported for one delivery option: `SaaSUrlDeliveryOptionDetails`. It allows you to update the `FulfillmentUrl`.

## Request Syntax

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

```

"ChangeSet": [{
  "ChangeType": "UpdateDeliveryOptions",
  "Entity": {
    "Identifier": "example1-abcd-1234-5ef6-7890abcdef12@1",
    "Type": "SaaSProduct@1.0"
  },
  "Details": "{
    \"DeliveryOptions\": [{
      \"Id\": \"do-1234567891234567891234\",
      \"Details\": \"{
        \"SaaSUrlDeliveryOptionDetails\": {
          \"FulfillmentUrl\": \"https://www.aws.amazon.com/marketplace/management\",
          \"LaunchUrl\": \"URL\",
          \"UsageInstructions\": \"Instructions\",
          \"DeploymentTemplates\": [
            {
              \"Title\": \"CloudFormation Template 123\",
              \"Description\": \"CloudFormation description\",
              \"IamPolicy\": \"Policy\": {
                \"TemplateUrl\": \"CloudFormation URL\",
                \"DefaultStackName\": \"Name\"
              }
            }
          ]
        }
      }
    ]
  }"
}]
}

```

Provide information for the fields to add the AddDeliveryOptions change type:

- **Entity** (object) (required) – Your SaaS-based product.
- **Identifier** (string) (required) – Your product ID. For more information, see [Identifier](#).
- **Type** (string) (required) – The Type is based on the delivery method (product type) that your product will be using: SaaSProduct@1.0.
- **Details** (string) – Details of the request. This field is a JSON string field. It must be formatted properly for a single-line string field, including escaping characters (such as quotation marks) that can't be in a string. For more information, see [Working with the Details attribute \(Legacy\)](#).
- **DeliveryOptions** (array) – Contains the details of the delivery options being updated.

- **Id** (string) – Unique identifier for the `DeliveryOption`. (Get the unique identifier for the `DeliveryOption` by calling the `DescribeEntity` action on the product you are updating.)
- **Details** (object) – Contains the `SaaSUrlDeliveryOptionDetails` of the delivery option to be updated.
  - **SaaSUrlDeliveryOptionDetails** (object) – Contains the `FulfillmentUrl` of a delivery option for SaaS product.
    - **FulfillmentUrl** (string) – The URL to be updated for the SaaS product.
    - **LaunchUrl** (string) – The URL to your SaaS product's landing page. This is required if `QuickLaunchEnabled` is set to `True`.
    - **UsageInstructions** (string) – Instructions for using this delivery option. Include documentation for manual steps for customers who won't use `DeploymentTemplates`.
  - **DeploymentTemplates** (array) – Deployment templates that customers can use to set up and configure the SaaS product and any related AWS resources.
    - **Title** (string) – The display name of the deployment template.
    - **Description** (string) – A description for what the deployment template contains.
    - **IamPolicy** (string) – An IAM policy describing the permissions needed to deploy the template. Buyers can use this IAM policy to quickly deploy the template.
    - **CloudFormationDetails** (object) – The details of a CloudFormation template.
      - **TemplateUrl** (string) – The URL for the deployment template.
      - **DefaultStackName** (string) – The default name used in AWS CloudFormation when the customer is creating the template.

## Response Syntax

```
{
  "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 includes validating information to ensure it meets the AWS Marketplace guidelines. The validation process can take anywhere from a

few minutes to a few hours. You can check the status of the request through the AWS Marketplace Management Portal, or in the Catalog API with the `DescribeChangeSet` action.

## Synchronous Validations

Error condition	Message	HTTP code
Empty Delivery Option Ids	Provided Details is not valid. String " at /DeliveryOptions/0/Id does not match required schema regex, '^do-[a-zA-Z0-9]+\$'	422
Missing Delivery Option Ids	Provided Details is not valid. JSON at /DeliveryOptions/0 is missing required properties: ['Id'].	422
Duplicate Delivery Option Ids	Provide unique delivery option IDs.	422
Invalid Fulfillment Url	Provide a valid fulfillment URL beginning with "https://".	422
Invalid Delivery Option IDs	Provide delivery option IDs that can be found in the product. IDs not found: [x]	422
Multiple URL Delivery Options	You provided more than one URL delivery option. Provide one URL delivery option.	422
Missing delivery option ids	The delivery option ID is missing. Provide one or more valid delivery option IDs that you wish to update, or use <code>AddDeliveryOptions</code> if you intended to add a new delivery option.	422
Invalid Launch URL	Provide a valid launch URL beginning with "https://".	422

Error condition	Message	HTTP code
Missing Launch Url	Required parameter LaunchUrl is missing. You must provide a LaunchUrl .	422
Missing deployment templates	The deployment template is missing. Provide at least one deployment template.	422
Too many deployment templates	You cannot provide more than 20 deployment templates.	422
Invalid template URL	Quick Start URL is invalid. Provide deployment template URL that is published through AWS QuickStarts to Amazon S3. Invalid deployment templates URL: [x]	422
Invalid deployment template stack name	The deployment template stack name is invalid. Provide a valid stack name using only alphanumeric characters and hyphens. It must start with an alphabetical character and can't be longer than 128 characters.	422
Duplicate deployment template title	You provided duplicate deployment template titles. Provide unique deployment template titles.	422
Duplicate deployment template URL	You provided duplicate deployment template urls. Provide unique deployment template titles.	422

Error condition	Message	HTTP code
Invalid deployment template type	The deployment template type is invalid. Provide a valid deployment template type. Supported values are ["CloudFormation@1.0"].	422
Invalid deployment template IAM policy	The deployment template IAM policy is invalid. Provide a valid IAM policy.	422
Invalid usage instructions	<ul style="list-style-type: none"> <li>Images aren't supported by the usage instructions. Remove the image [x].</li> <li>You provided a link to an invalid URL in the usage instructions: [x]. Provide a valid URL.</li> <li>You provided a link with an unsupported URI scheme in the usage instructions. Use a supported scheme: ["http", "https", "tel", "mailto"].</li> </ul>	422

## Asynchronous Errors

Error code	Error message
INVALID_DELIVERY_OPTION_IDS	Provide delivery option IDs that can be found in the product. IDs not found: [x]
AUDIT_ERROR	AWS MP Catalog Audits List - CQ team
INVALID_FULFILLMENT_URL	The URL you provided returned HTTP status code [x]. Provide a fulfillment URL that renders with a 200.

Error code	Error message
INVALID_LAUNCH_URL	The URL you provided returned HTTP status code [x]. Provide a launch URL that renders with a 200.
INVALID_TEMPLATE_URL	Quick Start URL is invalid. Provide deployment template URL that is published through AWS QuickStarts to Amazon S3. Invalid deployment templates URL: [x]

## Update delivery option visibility

You can use the Catalog API to configure permissions so that only some users can change the visibility for a SaaS product in AWS Marketplace.

To configure permissions so that only some users can change the visibility for a SaaS product, call the `StartChangeSet` API operation with the `UpdateDeliveryOptionsVisibility` change type, as shown in the following example.

### Note

This is only supported for one delivery option: `SaaSUrlDeliveryOptionDetails`.

## Request Syntax

```
{
  "Catalog": "AWSMarketplace",
  "ChangeSet":
  [
    {
      "ChangeType": "UpdateDeliveryOptionsVisibility",
      "Entity":
      {
        "Identifier": "prod-example12345",
        "Type": "SaaSProduct@1.0"
      },
      "DetailsDocument":
```

```
{
  "DeliveryOptions":
  [
    {
      "Id": "do-1234567891234567891234",
      "TargetVisibility": "Public"
    },
    {
      "Id": "do-43210987654321",
      "TargetVisibility": "Limited",
      "Targeting":
      {
        "PositiveTargeting":
        {
          "BuyerAccounts":
          [
            "123456789012"
          ]
        }
      }
    }
  ]
}
```

Provide information for the fields to add the UpdateDeliveryOptionsVisibility change type:

- **Entity** (object) (required) – Your SaaS-based product.
  - **Identifier** (string) (required) – Your product ID. For more information, see [Identifier](#).
  - **Type** (string) (required) – The Type is based on the delivery method (product type) that your product will be using: SaaSProduct@1.0.
- **DetailsDocument** (object) – Details of the request.
- **DeliveryOptions** (array) – List of DeliveryOptions to be updated.
  - **Id** (string) – The delivery option id to be updated.
  - **TargetVisibility** (string) – The intended new visibility of the delivery option.

Possible values: Limited, Public, and Unavailable.

**Note**

There is always exactly one `Public` delivery option, and a maximum of one `Limited` delivery option.

- **Targeting** (object) *optional* – Targeting of the delivery option, used in conjunction with the `Limited` visibility status to be able to test the new delivery option before changing the visibility to `Public`.
- **PositiveTargeting** (object) – Specifying inclusive targeting.
  - **BuyerAccounts** (array of strings) – The list of buyer AWS account ids who will be able to use the new delivery option.

Min size: 0. Max size: 100.

**Response Syntax**

```
{
  "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 includes validating information to ensure that it meets the AWS Marketplace guidelines. The validation process can take anywhere from a few minutes to a few hours. You can check the status of the request through the AWS Marketplace Management Portal, or in the Catalog API with the `DescribeChangeSet` action.

**Synchronous Validations**

Error condition	Message	HTTP code
Missing delivery option ids	The delivery option ID is missing. Provide one or more valid delivery option IDs that you wish to update, or use <code>AddDeliveryOptions</code> if you intended to add a new delivery option.	422

Error condition	Message	HTTP code
Invalid visibility	You provided invalid option for TargetVisibility . Allowed options are: Limited, Public, Unavailable .	422
Invalid targeting	You provided invalid option for PositiveTargeting . You must provide a valid parameter for BuyerAccounts .	422
Missing Visibility and Targeting	You provided invalid delivery option visibility details. You must provide a valid parameter for at least one of TargetVisibility or Targeting .	422
Too many AWS account ids	You cannot provide more than 100 targeted buyer accounts.	422

## Asynchronous Errors

Error code	Error message
INVALID_DELIVERY_OPTION_IDS	You provided invalid delivery option details. Provide delivery option IDs that can be found in the product. IDs not found: [x]
INVALID_VISIBILITY	You provided more than one delivery option for the public state. Provide only one public delivery option.
INVALID_VISIBILITY	You didn't provide a public delivery option. Provide one public delivery option.
AUDIT_ERROR	Varies based on MCO manual review.

# Working with offers

You can use the AWS Marketplace Catalog API to automate tasks for working with offers.

While the *product* describes what is being sold in AWS Marketplace, the *offer* describes the terms and rules around how this product can be purchased and consumed. AWS Marketplace products can have multiple offers sold by different sellers. However, each AWS Marketplace offer can only be created for one product.

An *offer* has a collection of terms to be accepted for an agreement between two parties. The accepted terms from the offer are translated into an agreement as a proof of transaction.

There are two types of offers:

- **Private offers** are a purchasing program allowing sellers and buyers to negotiate custom prices. An end-user licensing agreement (EULA) is signed for software purchases in AWS Marketplace. This offer is only visible to the specified buyer.

For more information, see [Private offers](#) in the *AWS Marketplace Seller Guide*.

- **Public offers** are a globally discoverable purchasing program. Sellers can specify targeting based on programs and geographical location, which makes the offer accessible to a specific set of customers based on the given criteria.

See the following resources:

- For end-to-end labs with working code examples, see [Manage offers with API](#) in the *AWS Marketplace seller workshop*.
- For code examples of API requests, see [Python](#) and [Java](#) examples in *AWS Samples* on GitHub.
- For a video on creating private offers, see [Create a Private Offer Using the AWS Marketplace Catalog API](#) on YouTube.
- For a video on updating AMI instance pricing, see [Update AMI Product Pricing Using the AWS Marketplace Catalog API](#) on YouTube.

The following topics describe how to use the Catalog API to create and update offers:

## Topics

- [Create a new offer](#)

- [Create a replacement offer](#)
- [Update offer information](#)
- [Update targeting configuration](#)
- [Update refund policy](#)
- [Update legal resources](#)
- [Update pricing](#)
- [Update the discoverability of the offer](#)
- [Define the expiration date of agreements created using the offer](#)
- [Update payment schedule details](#)
- [Modify renewal options](#)
- [Publish an offer](#)
- [Describe existing offer details](#)
- [Working with Resale Authorizations](#)
- [Working with channel partner private offers](#)

## Create a new offer

You can use the Catalog API to create a new offer in AWS Marketplace.

If your request has been processed successfully, AWS Marketplace Catalog API creates an offer in Draft state. While in Draft, it's an incomplete offer and not visible to buyers in AWS Marketplace. You can use the Update change types to complete the offer.

After the offer is completed, use the [ReleaseOffer](#) change type to complete the offer creation process and release the offer. Releasing the offer validates the entire offer and makes it visible to buyers in AWS Marketplace.

To create a new offer, call the StartChangeSet API operation with the CreateOffer change type, as shown in the following example.

### Request Syntax

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

```
{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "CreateOffer",
      "Entity": {
        "Type": "Offer@1.0"
      },
      "DetailsDocument": {
        "ProductId": "prod-ad8EXAMPLE51",
        "Name": "Test Offer"
      }
    }
  ]
}
```

Provide information for the fields to add the `CreateOffer` change type:

- **Entity** (object) (required) – Your offer.
  - **Type** (string) (required) – The Type is always `Offer@1.0`.
- **DetailsDocument** (object) (required) – The JSON value of specifics of the request.
  - **ProductId** (string) (required) – The unique identifier of the product being offered.
  - **Name** (string) (optional) – The name associated with the offer for better readability to you and your customers. It is displayed as a part of the Agreement information as well.

## Response Syntax

A change set is created for your request. The response to this request gives you the `ChangeSetId` and `ChangeSetArn` 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.

You can check the status of the request through the AWS Marketplace Management Portal, or directly through Catalog API using the [DescribeChangeSet](#) API operation.

If the Status is SUCCEEDED, then a new OfferId is generated.

The response looks like the following.

```
{
  "ChangeSetId": "example123456789012abcdef",
  "ChangeSetArn": "arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/
ChangeSet/example123456789012abcdef",
  "ChangeSetName": "Submitted by 123456789012",
  "StartTime": "2021-05-27T22:21:26Z",
  "EndTime": "2021-05-27T22:32:19Z",
  "Status": "SUCCEEDED",
  "ChangeSet": [
    {
      "ChangeType": "CreateOffer",
      "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "offer-123456789"
      },
      "DetailsDocument": {
        "ProductId": "prod-ad8EXAMPLE51",
        "Name": "Test Offer"
      },
      "ErrorDetailList": []
    }
  ]
}
```

## Synchronous Validations

The following schema validations are specific to CreateOffer actions in the AWS Marketplace Catalog API. These validations are performed when you call StartChangeSet. If the request doesn't meet the following requirements, it will fail with an HTTP response.

Input field	Validation rule	HTTP code
ProductId	Required	422
	Length must be between 1 and 50 characters	

Input field	Validation rule	HTTP code
	Must not contain illegal characters (\, <, >)	
ProductId	Required User must be authorized to create offer for the given product	403
ProductId	Required Must be an existing product in the catalog or being created in the same change set	404
Name	Optional  Length must be between 1 and 150 characters  Must not contain illegal characters (\, <, >)	422

## Asynchronous Errors

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

Error code	Error message
INCOMPATIBLE_PRODUCT	Use an active product in <code>Limited</code> or <code>Public</code> state.
INCOMPATIBLE_PRODUCT	Managing offers for your chosen product type isn't currently supported in the AWS Marketplace Catalog API.

# Create a replacement offer

You can use the Catalog API to create a replacement offer (also known as an agreement-based offer) in AWS Marketplace.

If your request has been processed successfully, AWS Marketplace Catalog API will have an offer in `Draft` state generated for you, which is an incomplete offer and not visible to buyers on AWS Marketplace. You will use `Update` change types to complete the offer. After the offer is completed, you will use [ReleaseOffer](#) change type to complete offer creation process and release the offer, which will validate the entire offer and make your offer visible to buyers on AWS Marketplace. From there, the buyer has the option to accept the replacement offer or to continue to operate under the original agreement.

To create a replacement offer, call the `StartChangeSet` API operation with the `CreateReplacementOffer` change type and provide a pre-existing agreement id, as shown in the following example.

## Request Syntax

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

{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "CreateReplacementOffer",
      "Entity": {
        "Type": "Offer@1.0"
      },
      "DetailsDocument": {
        "AgreementId": "agmt-12345",
        "Name": "Offer name"
      }
    }
  ]
}
```

Provide information for the fields to add the `CreateReplacementOffer` change type:

- **Entity** (object) (required) – Your offer.

- **Type** (string) (required) – The Type is always `Offer@1.0`.
- **DetailsDocument** (object) (required) – The JSON value of specifics of the request.
- **AgreementId** (string) (required) – The unique identifier for the current agreement to be replaced.
- **Name** (string) (optional) – The name associated with the offer for better readability to you and your customers. It will be displayed as part of Agreement information as well.

## Response Syntax

A change set is created for your request. The response to this request gives you the `ChangeSetId` and `ChangeSetArn` 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.

You can check the status of the request through the AWS Marketplace Management Portal, or directly through Catalog API using the [DescribeChangeSet](#) API operation.

## Synchronous Validations

The following schema validations are specific to `CreateReplacementOffer` actions in the AWS Marketplace Catalog API. These validations are performed when you call `StartChangeSet`. If the request doesn't meet the following requirements, it will fail with an HTTP response.

Input field	Validation rule	HTTP Code
AgreementId	Required  Length must be between 1 and 64 characters	422
AgreementId	Required User must be authorized to create offer for the given agreement	403

Input field	Validation rule	HTTP Code
Name	Optional  Length must be between 1 and 150 characters  Must not contain invalid characters (\, <, >)	422

## Asynchronous Errors

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

Error code	Error message
INCOMPATIBLE_PRODUCT	Replacement offers aren't supported for the product.

## Update offer information

You can use the Catalog API to update the offer information in AWS Marketplace.

To update the offer information, call the `StartChangeSet` API operation with the `UpdateInformation` change type, as shown in the following example. All other information will remain unchanged.

### Request Syntax

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

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

```
"ChangeSet": [
  {
    "ChangeType": "UpdateInformation",
    "Entity": {
      "Type": "Offer@1.0",
      "Identifier": "offer-123456789"
    },
    "DetailsDocument": {
      "Name": "New offer name",
      "Description": "New offer description",
      "PreExistingAgreement": {
        "AcquisitionChannel": "External",
        "PricingModel": "Contract"
      }
    }
  }
]
```

Provide information for the fields to add the `UpdateInformation` change type:

- **Entity** (object) (required) – Your offer.
  - **Type** (string) (required) – The Type is always `Offer@1.0`.
  - **Identifier** (string) (required) – Your offer ID. For more information, see [Identifier](#).
- **DetailsDocument** (object) (required) – The JSON value of specifics of the request.
  - **Name** (string) (optional) – Name associated with the offer for better readability. It is displayed as part of agreement information.
  - **Description** (string) (optional) – A free-form text that is meant to be used only by you and will never be visible to buyers.
  - **PreExistingAgreement** (object) (optional) – Determines if this offer is a renewal for an existing agreement with an existing customer for the same underlying product. The existing agreement can be within or outside AWS Marketplace. AWS may audit and verify your offer is a renewal. If AWS is unable to verify your offer, then AWS may revoke the offer and entitlements from your customer.
    - **AcquisitionChannel** (string) (required) – Indicates if the existing agreement was signed outside AWS Marketplace or within AWS Marketplace.

Possible values: `External`, `AwsMarketplace`

- **PricingModel** (string) (required) – Indicates which pricing model the existing agreement uses.

Possible values: Contract, Usage, Byo1, Free

## Response Syntax

A change set is created for your request. The response to this request gives you the `ChangeSetId` and `ChangeSetArn` 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 includes validating information to ensure that it meets the AWS Marketplace guidelines. The validation process can take a few minutes.

You can check the status of the request through the AWS Marketplace Management Portal, or directly through Catalog API using the [DescribeChangeSet](#) API operation.

## Synchronous Validations

The following schema validations are specific to `UpdateInformation` actions in the AWS Marketplace Catalog API. These validations are performed when you call `StartChangeSet`. If the request doesn't meet the following requirements, it will fail with an HTTP response.

Input field	Validation rule	HTTP code
Properties	At least one of the following properties must be provided	422
Name	Optional Length must be between 1 and 150 characters	422

Input field	Validation rule	HTTP code
	Must not contain illegal characters (\, <, >)	
Description	Optional  Length must be between 1 and 255 characters	422
PreExistingAgreement	Optional  Can be null to remove PreExistingAgreement from offer	422
PreExistingAgreement.PricingModel	Required  Can be one of these values: [Byo1, Free, Usage, Contract]	422
PreExistingAgreement.AcquisitionChannel	Required  Can be one of these values: [AwsMarketplace , External]	422

## Asynchronous Errors

The following errors are specific to UpdateInformation actions in the AWS Marketplace Catalog API. These errors are returned when you call DescribeChangeSet after a change set is processing. For more information about using DescribeChangeSet to get the status of a change request, see [Working with change sets](#).

Error code	Error message
INCOMPATIBLE_PRE_EXISTING_AGREEMENT	PreExistingAgreement can't be changed after the offer is released.

# Update targeting configuration

You can use the Catalog API to update the targeting configuration of your offer in AWS Marketplace.

All existing targeting options that aren't included in the latest request and will be removed from the offer.

To update the targeting configuration of your offer, call the `StartChangeSet` API operation with the `UpdateTargeting` change type, as shown in the following example.

## Request Syntax

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

{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "UpdateTargeting",
      "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "offer-123456789"
      },
      "DetailsDocument": {
        "PositiveTargeting": {
          "CountryCodes": [
            "US",
            "CA"
          ],
          "BuyerAccounts": [
            "111122223333"
          ]
        },
        "NegativeTargeting": {
          "CountryCodes": [
            "XX"
          ]
        }
      }
    }
  ]
}
```

```
}
```

Provide information for the fields to add the `UpdateTargeting` change type:

- **Entity** (object) (required) – Your offer.
  - **Type** (string) (required) – The Type is always `Offer@1.0`.
  - **Identifier** (string) (required) – Your offer ID. For more information, see [Identifier](#).
- **DetailsDocument** (object) (required) – The JSON value of specifics of the request.
- **PositiveTargeting** (object) (optional) – Positive targeting defines the criteria which any buyer's profile should fulfill in order to be allowed to access the offer. This field is optional, but at least one targeting option should be provided when this field is present.
  - **CountryCodes** (array of strings) (optional) – List as option for allowing targeting based on country. If the intention isn't to target the offer to a country, this field should be omitted. If it's present, the list must contain at least one country code. Each element in this list should be a valid 2-letter country code, using this format: ISO 3166-1 alpha-2.
  - **BuyerAccounts** (array of strings) (optional) – List as an option to allow targeting based on AWS accounts (also known as Private Offer). If the intention is to not target the offer to an AWS account, this field should be omitted.
- **NegativeTargeting** (object) (optional) – Negative targeting defines the criteria which any customer's profile should fulfill to be restricted to access the offer. Although this field is optional, at least one targeting option should be provided when this field is present.
  - **CountryCodes** (array of strings) (required) – List as option for allowing targeting based on country. If the intention isn't to target the offer to a specific country, then this field should be omitted. If it's present, the list must contain at least one country code. Each element in this list should be a valid 2-letter country code using this format: ISO 3166-1 alpha-2.

## Response Syntax

A change set is created for your request. The response to this request gives you the `ChangeSetId` and `ChangeSetArn` 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 includes validating information to ensure that it meets the AWS Marketplace guidelines. The validation process can take a few minutes.

You can check the status of the request through the AWS Marketplace Management Portal, or directly through Catalog API using the [DescribeChangeSet](#) API operation.

## Synchronous Validations

The following schema validations are specific to UpdateTargeting actions in the AWS Marketplace Catalog API. These validations are performed when you call StartChangeSet. If the request doesn't meet the following requirements, it will fail with an HTTP response.

Input field	Validation rule	HTTP code
NegativeTargeting	Optional  Can have either one of the following : [CountryCodes ]	422
NegativeTargeting.CountryCodes	Optional  List size must be between 1 and 244  Country codes must be valid (ISO 3166-1 alpha-2)	422
PositiveTargeting	Optional  Can have either one of the following : [CountryCodes , BuyerAccounts ]	422
PositiveTargeting.BuyerAccounts	Optional  List size must be between 1 and 26  AWS account IDs must be in valid format (12-digit number)	422
PositiveTargeting.CountryCodes	Optional	422

Input field	Validation rule	HTTP code
	List size must be between 1 and 244	
	Country codes must be valid (ISO 3166-1 alpha-2)	

## Asynchronous Errors

The following errors are specific to UpdateTargeting actions in the AWS Marketplace Catalog API. These errors are returned when you call DescribeChangeSet after a change set is processing. For more information about using DescribeChangeSet to get the status of a change request, see [Working with change sets](#).

Error code	Error message
INVALID_BUYER_ACCOUNTS	Provide valid buyer accounts. Invalid accounts: [x].
INVALID_COUNTRY_CODES	Provide supported country codes.
INVALID_TARGETING	Use either negative or positive targeting on the same attribute.
INCOMPATIBLE_PRODUCT	Country-based targeting isn't supported for the product.
INCOMPATIBLE_RESALE_AUTHORIZATION	Provide BuyerAccounts that are compatible with the ResaleAuthorization.
INCOMPATIBLE_TARGETING	The requested change can't be performed after the offer is released.
INCOMPATIBLE_TARGETING	The requested change can't be performed after the offer is expired.
INCOMPATIBLE_TARGETING	Targeting can't be updated on a replacement offer. If the buyer isn't associated with the provided AgreementId, then create a new

Error code	Error message
	private offer by providing an AgreementId associated with the buyer.
TOO_MANY_BUYER_ACCOUNTS	Provide BuyerAccounts within the allowed limits.

## Update refund policy

You can use the Catalog API to update the refund policy of your offer in AWS Marketplace.

This change doesn't affect existing agreements. The support terms that aren't included in the latest request will be removed from the offer.

To update the refund policy, call the StartChangeSet API operation with the UpdateSupportTerms change type, as shown in the following example.

### Request Syntax

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

{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "UpdateSupportTerms",
      "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "offer-123456789"
      },
      "DetailsDocument": {
        "Terms": [
          {
            "Type": "SupportTerm",
            "RefundPolicy": "Updated refund policy description"
          }
        ]
      }
    }
  ]
}
```

```
]
}
```

Provide information for the fields to add the `UpdateSupportTerms` change type:

- **Entity** (object) (required) – Your offer.
  - **Type** (string) (required) – The Type is always `Offer@1.0`.
  - **Identifier** (string) (required) – Your offer ID. For more information, see [Identifier](#).
- **DetailsDocument** (object) (required) – The JSON value of specifics of the request.
- **Terms** (array of structures) (required) – List of support terms that you would like to update. Accepted support terms are:
  - **SupportTerm** (object) (required) – Defines the customer support available for the acceptors when they purchase the software.
    - **Type** (string) (required) – Type of the term being updated. This is the object value: `"SupportTerm"`.
    - **RefundPolicy** (string) (required) – Free-text field about the refund policy description that will be shown to customers as is on the website and console.

## Response Syntax

A change set is created for your request. The response to this request gives you the `ChangeSetId` and `ChangeSetArn` 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 includes validating information to ensure that it meets the AWS Marketplace guidelines. The validation process can take a few minutes.

You can check the status of the request through the AWS Marketplace Management Portal, or directly through Catalog API using the [DescribeChangeSet](#) API operation.

## Synchronous Validations

The following schema validations are specific to `UpdateSupportTerms` actions in the AWS Marketplace Catalog API. These validations are performed when you call `StartChangeSet`. If the request doesn't meet the following requirements, it will fail with an HTTP response.

Input field	Validation rule	HTTP code
Terms	Required	422
Terms[].RefundPolicy	Required Length must be between 1 and 500 Cannot lead or end with spaces	422
Terms[].Type	Required Can only be <code>SupportTerm</code>	422

## Asynchronous Errors

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

Error code	Error message
INCOMPATIBLE_PRODUCT	SupportTerm isn't supported in private offers for the product.
INCOMPATIBLE_TERMS	SupportTerm isn't supported for free trial offers.
INCOMPATIBLE_TERMS	The requested change can't be performed after the offer is released.
INCOMPATIBLE_TERMS	The requested change can't be performed after the offer is expired.

## Update legal resources

You can use the Catalog API to replace the existing legal documents, such as an end user license agreement (EULA). The legal terms that aren't included in the latest request will be removed from the offer.

To update legal resources of your offer, call the `StartChangeSet` API operation with the `UpdateLegalTerms` change type, as shown in the following example.

### Request Syntax

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

{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "UpdateLegalTerms",
      "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "offer-123456789"
      },
      "DetailsDocument": {
        "Terms": [
          {
            "Type": "LegalTerm",
            "Documents": [
              {
                "Type": "CustomEula",
                "Url": "https://s3.amazonaws.com/EULA/custom-eula-1234.txt"
              }
            ]
          }
        ]
      }
    }
  ]
}
```

Provide information for the fields to add the `UpdateLegalTerms` change type:

- **Entity** (object) (required) – Your offer.
  - **Type** (string) (required) – The Type is always Offer@1.0.
  - **Identifier** (string) (required) – Your offer ID. For more information, see [Identifier](#).
- **DetailsDocument** (object) (required) – The JSON value of specifics of the request.
- **LegalTerm** (object) (required) – Defines the list of text agreements to be proposed to the acceptors. One example of such an agreement is the end user license agreement (EULA).
  - **Type** (string) (required) – Type of the term being updated. This is the object value: "LegalTerm".
  - **Documents** (array of structures) (required) – List of references to legal resources to be proposed to the buyers. One example of such a resource is the end user license agreement (EULA). Each reference is made up of a Type and a URL:
    - **Type** (string) (required) – Type of document. Available document types are:
      - **CustomEula** – A custom EULA provided by you as seller. A URL for a EULA stored in an accessible S3 bucket is required for this document type.
      - **StandardEula** – Standard Contract For AWS Marketplace (SCMP). For more information about SCMP, see the AWS Marketplace Seller Guide. You don't provide a URL for this type because it is managed by AWS Marketplace.
    - **Url** (string) (conditionally required) – A URL to the legal document for buyers to read. Required when Type is one of the following [CustomEula].
    - **Version** (string) (conditionally required) – Version of standard contracts provided by AWS Marketplace. Required when Type is [StandardEula]. Available version:
      - **2022-07-14** – This version of the Standard Contract for AWS Marketplace is available from this Amazon S3 bucket: <https://s3.amazonaws.com/aws-mp-standard-contracts/Standard-Contract-for-AWS-Marketplace-2022-07-14.pdf>

## Response Syntax

A change set is created for your request. The response to this request gives you the ChangeSetId and ChangeSetArn 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 includes validating information to ensure that it meets the AWS Marketplace guidelines. The validation process can take a few minutes.

You can check the status of the request through the AWS Marketplace Management Portal, or directly through Catalog API using the [DescribeChangeSet](#) API operation.

## Synchronous Validations

The following schema validations are specific to UpdateLegalTerms actions in the AWS Marketplace Catalog API. These validations are performed when you call StartChangeSet. If the request doesn't meet the following requirements, it will fail with an HTTP response.

Input field	Validation rule	HTTP code
Terms	Required  Only LegalTerm is allowed in the list  List size must be 1	422
Terms[].Type	Required  Can only be LegalTerm	422
Terms[].LegalTerm.Documents	Required	422
Terms[].LegalTerm.Documents[].Type	Required  Allowed values: <ul style="list-style-type: none"> <li>• CustomEula</li> <li>• StandardEula</li> </ul>	422
Terms[].LegalTerm.Documents[].Url	Required and must be a valid URL when Type is CustomEula	422
Terms[].LegalTerm.Documents[].Version	Required and must be a valid Version when Type is StandardEula	422

Input field	Validation rule	HTTP code
	Valid StandardEula versions: ["2019-04-24", "2022-07-14"]	

## Asynchronous Errors

The following errors are specific to UpdateLegalTerms actions in the AWS Marketplace Catalog API. These errors are returned when you call DescribeChangeSet after a change set is processing. For more information about using DescribeChangeSet to get the status of a change request, see [Working with change sets](#).

Error code	Error message
INCOMPATIBLE_TERMS	The requested change can't be performed after the offer is released.
INCOMPATIBLE_TERMS	The requested change can't be performed after the offer is expired.
INVALID_LEGAL_DOCUMENTS	Provide URLs for legal documents stored in accessible S3 buckets.
INVALID_LEGAL_DOCUMENTS	Only the most recent version of StandardEula is supported for new offers.
INVALID_LEGAL_DOCUMENTS	Provide legal documents in the supported file formats.
INVALID_LEGAL_DOCUMENTS	Provide legal documents using the supported document types.
LIMIT_EXCEEDED_LEGAL_DOCUMENT_SIZE	Provide legal documents within the allowed size limits.

# Update pricing

You can use the Catalog API to replace the existing pricing terms completely. The pricing terms that aren't included in the latest request will be removed from the offer.

To update pricing terms for your offer, call the `StartChangeSet` API operation with the `UpdatePricingTerms` change type, as shown in the following example.

## Note

The following request syntax combines multiple examples. This combination doesn't work as a valid payload. For example, a `Terms` array can't include both the term type `FixedUpfrontPricingTerm` and the term type `ConfigurableUpfrontPricingTerm`. For examples of how different term types are combined for different pricing use cases, see [Manage offers with API](#) in the *AWS Marketplace seller workshop*.

## Request Syntax

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

{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "UpdatePricingTerms",
      "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "offer-123456789"
      },
      "DetailsDocument": {
        "PricingModel": "Usage",
        "Terms": [
          {
            "Type": "UsageBasedPricingTerm",
            "CurrencyCode": "USD",
            "RateCards": [
              {
                "RateCard": [
                  {
```

```

        "DimensionKey": "m3.large",
        "Price": "0.10"
    },
    {
        "DimensionKey": "m4.xlarge",
        "Price": "0.20"
    }
]
}
]
},
{
    "Type": "ConfigurableUpfrontPricingTerm",
    "CurrencyCode": "USD",
    "RateCards": [
        {
            "Selector": {
                "Type": "Duration",
                "Value": "P365D"
            },
            "RateCard": [
                {
                    "DimensionKey": "m3.large",
                    "Price": "300"
                },
                {
                    "DimensionKey": "m4.xlarge",
                    "Price": "400"
                }
            ],
            "Constraints": {
                "MultipleDimensionSelection": "Allowed",
                "QuantityConfiguration": "Allowed"
            }
        }
    ]
},
{
    "Type": "ByolPricingTerm"
},
{
    "Type": "RecurringPaymentTerm",
    "CurrencyCode": "USD",
    "BillingPeriod": "Monthly",

```



- **Usage** – Usage-based pricing model where buyers will be billed for their usage of your product.
- **Contract** – Contract-based pricing model where buyers are either billed in advance for the use of your product, or offered a flexible payment schedule. Buyers can also pay for an additional usage above their contract.
- **Free** – Free pricing model where buyers will not be charged for usage of product. When using this pricing model no pricing terms or payment schedule term can have non-zero rates.
- **Byol** – Byol pricing model where buyers will bring their own license for usage of the product.
- **Terms** (array of structures) (required) – List of pricing terms that you want to update. Supported pricing terms are:
  - **FreeTrialPricingTerm** (object) – Defines a short-term free pricing model where the buyers are not charged anything within a specified limit.
    - **Type** (string) – Type of the term being updated. This is the object value: "FreeTrialPricingTerm".
    - **Duration** (string) – Duration of the free trial period.
    - **Grants** (array of structures) – Entitlements that will be granted to the acceptor of a free trial as part of an agreement execution.
      - **DimensionKey** (string) – Unique dimension key defined in the product document. Dimensions represent categories of capacity in a product and are specified when the product is listed in AWS Marketplace.
      - **MaxQuantity** (integer) (optional) – Maximum amount of capacity that the buyer can be entitled to the given dimension of the product. If MaxQuantity is not provided, the buyer will be able to use an unlimited amount of the given dimension.
  - **UsageBasedPricingTerm** (object) – Defines a pay-as-you-go (PAYG) pricing model where the customers are charged based on product usage.
    - **Type** (string) (required) – Category of the term being updated. This is the object value: UsageBasedPricingTerm.
    - **CurrencyCode** (string) – Defines the currency for prices mentioned in this term. Currently, only USD is supported.
    - **RateCards** (array of structures) – List of rate cards.
      - **RateCard** (array of structures) – A rate card defines the per-unit rates for the product dimensions.

- **DimensionKey** (string) – Dimension that the given entitlement applies. Dimensions represent categories of capacity in a product and are specified when the product is listed in AWS Marketplace.
- **Price** (string) – Per-unit price for the product dimension that will be used for calculating the amount to be charged to the buyer.
- **ConfigurableUpfrontPricingTerm** (object) – Defines pre-paid payment model which allows buyers to configure the entitlements that they want to purchase and the duration of the entitlements. You can update the list of rates for each contract duration and entitlements for each dimension.
  - **Type** (string) (required) – Type of the term being updated. This is the object value: `ConfigurableUpfrontPricingTerm`.
  - **CurrencyCode** (string) (required) – Defines the currency for the prices mentioned in this term. Currently, only USD is supported.
  - **RateCards** (array of structures) (required) – List of rate cards.
    - **Selector** (object) (required) – Selector is used to differentiate between the mutually exclusive rate cards in the same pricing term, to be selected by the buyer.
      - **Type** (string) (required) – Category of Selector. At this time, only `Duration` is supported.
      - **Value** (string) (required) – Contract duration. This field supports the ISO 8601 format.
    - **RateCard** (array of structures) (required) – A rate card defines the per-unit rates for the product dimensions.
      - **DimensionKey** (string) (required) – Unique dimension key defined in the product document. Dimensions represent categories of capacity in a product and are specified when the product is listed in AWS Marketplace.
      - **Price** (string) (required) – Per-unit price for the product dimension which will be used for calculating the amount to be charged to the buyer.
  - **Constraints** (object) (required) – Defines constraints on how the term can be configured by acceptors.

**Note**

Currently, **MultipleDimensionSelection** and **QuantityConfiguration** values need to be same.

- **MultipleDimensionSelection** (string) (required) – Determines if buyers are allowed to select multiple dimensions in the rate card. Possible values are Allowed and Disallowed.
- **QuantityConfiguration** (string) (required) – Determines if acceptors are allowed to configure quantity for each dimension in rate card. Possible values are Allowed and Disallowed.
- **ByolPricingTerm** (object) – Enables you and your customers to move your existing agreements to AWS Marketplace. The customer won't be charged for product usage in AWS Marketplace because they already paid for the product outside of AWS Marketplace.
  - **Type** (string) (required) – Type of the term being updated. This is the object value: `ByolPricingTerm`.
- **RecurringPaymentTerm** (object) – Defines a pricing model where customers are charged a fixed recurring price at the end of each billing period.
  - **Type** (string) (required) – Type of the term being updated. This is the object value: `RecurringPaymentTerm`.
  - **BillingPeriod** (string) (required) – Defines the recurrence at which buyers are charged. Only `Monthly` is supported today.
  - **Price** (string) (required) – Amount charged to the buyer every billing period.
  - **CurrencyCode** (string) (required) – Defines the currency for the prices mentioned in this term. Currently, only USD is supported.
- **FixedUpfrontPricingTerm** (object) – Defines a pre-paid pricing model where the customers are charged a fixed upfront amount.
  - **Type** (string) (required) – Type of the term being updated. This is the object value: `FixedUpfrontPricingTerm`.
  - **CurrencyCode** (string) (required) – Defines the currency for the prices mentioned in this term. Currently, only USD is supported.
  - **Price** (string) (required) – Fixed amount to be charged to the customer when this term is accepted.
  - **Grants** (array of structures) (required) – Entitlements that will be granted to the acceptor of fixed upfront as part of agreement execution.

- **DimensionKey** (string) (required) – Unique dimension key defined in the product document. Dimensions represent categories of capacity in a product and are specified when the product is listed in AWS Marketplace.
- **MaxQuantity** (integer) (required) – Maximum amount of capacity that the buyer can be entitled to the given dimension of the product. If `MaxQuantity` is not provided, the buyer will be able to use an unlimited amount of the given dimension.
- **Duration** (string) (optional) – Defines the duration that the term remains active. This field supports the ISO 8601 format.

## Response Syntax

A change set is created for your request. The response to this request gives you the `ChangeSetId` and `ChangeSetArn` 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 includes validating information to ensure that it meets the AWS Marketplace guidelines. The validation process can take a few minutes.

You can check the status of the request through the AWS Marketplace Management Portal, or directly through Catalog API using the [DescribeChangeSet](#) API operation.

## Synchronous Validations

The following schema validations are specific to `UpdatePricingTerms` actions in the AWS Marketplace Catalog API. These validations are performed when you call `StartChangeSet`. If the request doesn't meet the following requirements, it will fail with an HTTP response.

Input field	Validation rule	HTTP
<code>PricingModel</code>	Required	422

Input field	Validation rule	HTTP
	Allowed pricing models: ["Byol", "Free", "Usage", "Contract"]	
Terms	Required  Allowed Terms: ["ConfigurableUpfrontPricingTerm", "ByolPricingTerm", "FreeTrialPricingTerm", "UsageBasedPricingTerm", "RecurringPaymentTerm", "FixedUpfrontPricingTerm"]	422
Terms[].ByolPricingTerm	Required	422
Terms[].ByolPricingTerm.Type	Required  Can only be "ByolPricingTerm"	422
Terms[].ConfigurableUpfrontPricingTerm	Required	422
Terms[].ConfigurableUpfrontPricingTerm.Type	Required  Can only be "ConfigurableUpfrontPricingTerm"	422
Terms[].ConfigurableUpfrontPricingTerm.CurrencyCode	Required  Supported currencies: ["USD"]	422
Terms[].ConfigurableUpfrontPricingTerm.RateCards	Required  List size must be between 1 and 5	422
Terms[].ConfigurableUpfrontPricingTerm.RateCards[].Constraints	Required	422

Input field	Validation rule	HTTP
Terms[].ConfigurableUpfrontPricingTerm.RateCards[].Constraints.MultipleDimensionSelection	Required Allowed values: ["Allowed", "Disallowed"]	422
Terms[].ConfigurableUpfrontPricingTerm.RateCards[].Constraints.QuantityConfiguration	Required Allowed values: ["Allowed", "Disallowed"]	422
Terms[].ConfigurableUpfrontPricingTerm.RateCards[].RateCard	Required List size must be between 1 and 800	422
Terms[].ConfigurableUpfrontPricingTerm.RateCards[].RateCard[].DimensionKey	Required Length must be between 1 and 100	422
Terms[].ConfigurableUpfrontPricingTerm.RateCards[].RateCard[].Price	Required Data type is "String" Non-negative decimals with up to 3 decimal places supported	422
Terms[].ConfigurableUpfrontPricingTerm.RateCards[].Selector	Required	422
Terms[].ConfigurableUpfrontPricingTerm.RateCards[].Selector.Type	Required Allowed values: ["Duration"]	422
Terms[].ConfigurableUpfrontPricingTerm.RateCards[].Selector.Value	Required Expected format per Selector type: ISO 8601 duration	422
Terms[].FixedUpfrontPricingTerm	Required	422

Input field	Validation rule	HTTP
Terms[].FixedUpfrontPricingTerm.Type	Required  Can only be "FixedUpfrontPricingTerm"	422
Terms[].FixedUpfrontPricingTerm.CurrencyCode	Required  Supported currencies: ["USD"]	422
Terms[].FixedUpfrontPricingTerm.Duration	Required  Expected format per Selector type: ISO 8601 duration	422
Terms[].FixedUpfrontPricingTerm.Grants	Required  List size must be between 1 and 200	422
Terms[].FixedUpfrontPricingTerm.Grants[].DimensionKey	Required  Length must be between 1 and 100	422
Terms[].FixedUpfrontPricingTerm.Grants[].MaxQuantity	RequiredValue must be greater than 0	422
Terms[].FixedUpfrontPricingTerm.Price	Required  Data type is "String"  Non-negative decimals with up to 3 decimal places supported	422
Terms[].FreeTrialPricingTerm	Required	422
Terms[].FreeTrialPricingTerm.Type	Required  Can only be "FreeTrialPricingTerm"	422

Input field	Validation rule	HTTP
Terms[].FreeTrialPricingTerm.Duration	Required Expected format: ISO 8601 duration	422
Terms[].FreeTrialPricingTerm.Grants	Required List size must be between 1 and 800	422
Terms[].FreeTrialPricingTerm.Grants[].DimensionKey	Required Length must be between 1 and 100	422
Terms[].FreeTrialPricingTerm.Grants[].MaxQuantity	Optional Value must be greater than 0	422
Terms[].RecurringPaymentTerm	Required	422
Terms[].RecurringPaymentTerm.Type	Required Can only be "RecurringPaymentTerm"	422
Terms[].RecurringPaymentTerm.BillingPeriod	Required Allowed values: ["Monthly"]	422
Terms[].RecurringPaymentTerm.CurrencyCode	Required Supported currencies: ["USD"]	422
Terms[].RecurringPaymentTerm.Price	Required Data type is "String" Non-negative decimals with up to 3 decimal places supported	422
Terms[].UsageBasedPricingTerm	Required	422

Input field	Validation rule	HTTP
Terms[].UsageBasedPricingTerm.Type	Required  Can only be "UsageBasedPricingTerm"	422
Terms[].UsageBasedPricingTerm.CurrencyCode	Required  Supported currencies: ["USD"]	422
Terms[].UsageBasedPricingTerm.RateCards	Required  Must be size of 1	422
Terms[].UsageBasedPricingTerm.RateCards[].RateCard	Required  List size must be between 1 and 800	422
Terms[].UsageBasedPricingTerm.RateCards[].RateCard[].DimensionKey	Required  Length must be between 1 and 100	422
Terms[].UsageBasedPricingTerm.RateCards[].RateCard[].Price	Required  Data type is "String"  Non-negative decimals with up to 8 decimal places supported	422

## Asynchronous Errors

The following errors are specific to UpdatePricingTerms actions in the AWS Marketplace Catalog API. These errors are returned when you call DescribeChangeSet after a change set is processing. For more information about using DescribeChangeSet to get the status of a change request, see [Working with change sets](#).

Error code	Error message
DUPLICATE_DIMENSION_KEYS	Provide Grants with a unique list of dimension keys in [x].
DUPLICATE_DIMENSION_KEYS	Provide RateCard with a unique list of dimension keys in [x].
DUPLICATE_SELECTORS	Provide a unique list of Selectors in ConfigurableUpfrontPricingTerm.
DUPLICATE_TERM_TYPES	Provide a unique list of term types.
INCOMPATIBLE_AGREEMENT	The following terms can't be removed from the replacement offer: [x, y, z].
INCOMPATIBLE_AGREEMENT	The following terms can't be added to the replacement offer: [x, y, z].
INCOMPATIBLE_CURRENCY_CODE	CurrencyCode can't be changed after the offer is released.
INCOMPATIBLE_PRODUCT	Usage pricing model isn't supported for the product.
INCOMPATIBLE_PRODUCT	Contract pricing model isn't supported for the product.
INCOMPATIBLE_PRODUCT	Byol pricing model isn't supported for the product.
INCOMPATIBLE_PRODUCT	Free pricing model isn't supported for the product.
INCOMPATIBLE_PRODUCT	[x] isn't supported in an offer for the product.
INCOMPATIBLE_PRODUCT	Provided payment and pricing terms are incompatible.

Error code	Error message
INCOMPATIBLE_PRODUCT	Use existing, available dimensions in the product in [x].
INCOMPATIBLE_PRODUCT	FreeTrialPricingTerm as the offer's only pricing term isn't supported for the product.
INCOMPATIBLE_PRODUCT	The following terms aren't supported for the product: [x,y,z].
INCOMPATIBLE_PRODUCT	Replacement offers are only supported for contract pricing model.
INCOMPATIBLE_PRODUCT	Provide pricing term(s) that are compatible with the product dimensions. Incompatible pricing terms: [x,y,z].
INCOMPATIBLE_RATE_CARD_CONSTRAINTS	Set MultipleDimensionSelection and QuantityConfiguration to Allowed in ConfigurableUpfrontPricingTerm for usage pricing model.
INCOMPATIBLE_RATE_CARD_CONSTRAINTS	Set MultipleDimensionSelection and QuantityConfiguration to Disallowed in ConfigurableUpfrontPricingTerm for usage pricing model.
INCOMPATIBLE_RATE_CARD_CONSTRAINTS	QuantityConfiguration in ConfigurableUpfrontPricingTerm can't be changed after the offer is released.
INCOMPATIBLE_RATE_CARD_CONSTRAINTS	MultipleDimensionSelection in ConfigurableUpfrontPricingTerm can't be changed after the offer is released.
INCOMPATIBLE_RATES	Set all charge amounts and prices to zero (0) when using Free pricing model.

Error code	Error message
INCOMPATIBLE_RATES	Only zero (0) prices are allowed in UsageBasedPricingTerm for a free trial offer for the product.
INCOMPATIBLE_RESALE_AUTHORIZATION	Provide the same CurrencyCode that is specified in the ResaleAuthorization.
INCOMPATIBLE_RESALE_AUTHORIZATION	Ensure Duration in FixedUpfrontPricingTerm matches duration specified in the ResaleAuthorization.
INCOMPATIBLE_RESALE_AUTHORIZATION	Provide term(s) that are compatible with the ResaleAuthorization. Incompatible terms: [x, y, z].
INCOMPATIBLE_SELECTOR_DURATION	Durations aren't allowed to be removed from rate cards in ConfigurableUpfrontPricingTerm after the offer released.
INCOMPATIBLE_TERMS	[x] isn't supported together with the following terms: [y,z].
INCOMPATIBLE_TERMS	The following terms can't be added after the offer is released: [x,y,z].
INCOMPATIBLE_TERMS	The following terms can't be removed after the offer is released: [x,y,z].
INCOMPATIBLE_TERMS	[x] isn't supported for private offers.
INCOMPATIBLE_TERMS	The following terms aren't supported with FreeTrialPricingTerm that grants unlimited usage: [x,y,z].
INCOMPATIBLE_TERMS	The following terms aren't supported with FreeTrialPricingTerm for the product: [x,y,z].

Error code	Error message
INCOMPATIBLE_TERMS	Provide zero (0) price for FixedUpfrontPricingTerm when the offer contains a PaymentScheduleTerm.
INCOMPATIBLE_TERMS	The following terms aren't compatible with the PricingModel: [x,y,z].
INCOMPATIBLE_TERMS	FixedUpfrontPricingTerm isn't supported when MarkupPercentage is greater than zero (0).
INCOMPATIBLE_TERMS	The requested change can't be performed after the offer is released.
INCOMPATIBLE_TERMS	The requested change can't be performed after the offer is expired.
INVALID_AGREEMENT_DURATION	Provide duration between [x] and [y] months.
INVALID_AGREEMENT_DURATION	Ensure duration granularity is at the day level for metered dimensions.
INVALID_CURRENCY_CODE	Provide a supported CurrencyCode.
INVALID_CURRENCY_CODE	Provide the same CurrencyCode across all pricing and payment terms.
INVALID_DURATION	Ensure Duration in FreeTrialPricingTerm is within the allowed range.
INVALID_DURATION	Provide Duration in FixedUpfrontPricingTerm that matches the duration between AgreementStartDate and AgreementEndDate.
INVALID_DURATION	Provide duration between [x] and [y] months.
INVALID_DURATION	Ensure duration granularity is at the day level for metered dimensions.

Error code	Error message
INVALID_GRANTS	Provide the same MaxQuantity for all Grants in FreeTrialPricingTerm.
INVALID_GRANTS	Provide Grants for all available metered dimensions in FreeTrialPricingTerm.
INVALID_PRICE_CHANGE	[x] can't be updated until [y] because you have requested a price increase in the past 120 days. To cancel your previous price increase request or for more information, contact the AWS Marketplace Managed Catalog Operations Team.
INVALID_PRICE_CHANGE	Price increase and dimension addition in [x] isn't supported in the same request. Add dimensions first.
INVALID_PRICE_CHANGE	Price increase and decrease in UsageBasedPricingTerm isn't supported in the same request. Decrease prices first.
INVALID_PRICE_CHANGE	Price increase in RecurringPaymentTerm and price decrease in UsageBasedPricingTerm isn't supported in the same request. Decrease prices first.
INVALID_PRICE_CHANGE	Price decrease in RecurringPaymentTerm and price increase in UsageBasedPricingTerm isn't supported in the same request. Decrease prices first.
INVALID_RATE_CARD	ConfigurableUpfrontPricingTerm is missing one or more-dimension keys for duration [x]. Provide prices for the same set of dimension keys for all durations.

Error code	Error message
INVALID_RATE_CARD	Provide a rate card for only metered dimensions in UsageBasedPricingTerm.
INVALID_RATE_CARD	Rates can't be removed from [x]. Provide prices for all dimensions in the existing rate card.
INVALID_RATE_CARD	Provide dimensions that have the same unit in [x].
INVALID_RATE_CARD	Provide either all metered or all entitled dimensions in [x].
INVALID_RATE_CARD	Provide only entitled dimensions in [x].
INVALID_RATE_CARD	Provide usage based rates for all available metered dimensions in UsageBasedPricingTerm.
INVALID_RATE_CARD	Provide usage based rates for all free trial dimensions.
INVALID_RATE_CARD	Provide prices with up to 3 decimal places in UsageBasedPricingTerm.
INVALID_SELECTOR_DURATION_VALUE	Provide duration between [x] and [y] months.
INVALID_SELECTOR_DURATION_VALUE	Ensure duration granularity is at the day level for metered dimensions.
INVALID_SELECTOR_DURATION_VALUE	Ensure Duration in ConfigurableUpfrontPricingTerm is within the allowed range.
INVALID_SELECTOR_DURATION_VALUE	Provide one or more supported contract durations.
INVALID_SELECTOR_DURATION_VALUE	Provide one or more supported contract durations or a single custom duration.

Error code	Error message
INVALID_SELECTOR_DURATION_VALUE	Provide Duration in ConfigurableUpfrontPricingTerm that matches the duration between AgreementStartDate and AgreementEndDate.
MISSING_DURATION	Provide Duration in FixedUpfrontPricingTerm.
MISSING_MANDATORY_TERMS	FixedUpfrontPricingTerm is only supported when paired with ByolPricingTerm or PaymentScheduleTerm.
MISSING_MANDATORY_TERMS	Provide at least one of [x,y,z].
MISSING_MANDATORY_TERMS	Provide a ByolPricingTerm when using Byol pricing model.
TOO_MANY_GRANTS	Provide up to [x] grants in [y].
TOO_MANY_RATE_CARDS	Only one rate card in ConfigurableUpfrontPricingTerm is allowed for the product.
TOO_MANY_RATE_CARDS	Up to [x] rate cards are allowed in ConfigurableUpfrontPricingTerm for the product.
TOO_MANY_RATES	Provide RateCards within the allowed limits in ConfigurableUpfrontPricingTerm.
TOO_MANY_RATES	Provide RateCards within the allowed limits in UsageBasedPricingTerm.

## Update the discoverability of the offer

You can use the Catalog API to control the discoverability of your offer in AWS Marketplace.

You can either choose to set a specific date in the future to restrict the discoverability of your offer or in the past to expire your offer. The UpdateAvailability change type doesn't affect existing agreements.

**Note**

- You can use the `UpdateAvailability` change type on a private offer that has already been [published](#) (also known as *released*). If buyers have already accepted the private offer, those existing agreements aren't affected.
- When modifying the `AvailabilityEndDate` of an existing private offer, the [constraints of the agreement duration](#) must be adhered to. If it's not, include an additional `UpdateValidityTerms` change type in this change set to modify the agreement duration to adhere to the new expiration. The `UpdateValidityTerms` change type can be used on a private offer that is either released or not yet released.
- When modifying the `AvailabilityEndDate` of an existing private offer, the [constraints of the payment schedule](#) must be adhered to. If it's not and the private offer is *not yet released*, include an additional `UpdatePaymentScheduleTerms` change type in this change set to modify the payment schedule to adhere to the new expiration. If the private offer is *already released*, you can only make changes to the `AvailabilityEndDate` as long as the new date adheres to the constraints of the payment schedule.

To control the discoverability of your offer, call the `StartChangeSet` API operation with the `UpdateAvailability` change type, as shown in the following example.

**Request Syntax**

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

{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "UpdateAvailability",
      "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "offer-123456789"
      },
      "DetailsDocument": {
        "AvailabilityEndDate": "2024-05-31"
      }
    }
  ]
}
```

```
    }  
  ]  
}
```

Provide information for the fields to add the `UpdateAvailability` change type:

- **Entity** (object) (required) – Your offer.
  - **Type** (string) (required) – The Type is always `Offer@1.0`.
  - **Identifier** (string) (required) – Your offer ID. For more information, see [Identifier](#).
- **DetailsDocument** (object) (required) – The JSON value of specifics of the request.
  - **AvailabilityEndDate** (string) (required) – This is the date until when the offer is discoverable and purchasable in AWS Marketplace. You can choose to set a specific date in the future to restrict the availability or in the past to expire the offer. Dates are represented in YYYY-MM-DD format.

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

### Response Syntax

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

The change request is added to a queue and processed. It includes validating information to ensure that it meets the AWS Marketplace guidelines. The validation process can take a few minutes. You can check the status of the request through the AWS Marketplace Management Portal, or in the Catalog API with the `DescribeChangeSet` action.

### Synchronous Validations

The following schema validations are specific to `UpdateAvailability` actions in the AWS Marketplace Catalog API. These validations are performed when you call `StartChangeSet`. If the request doesn't meet the following requirements, it will fail with an HTTP response.

Input field	Validation rule	HTTP code
AvailabilityEndDate	Required Format: "YYYY-MM-DD"	422

## Asynchronous Errors

The following errors are specific to UpdateAvailability 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](#).

Error code	Error message
INVALID_AVAILABILITY_END_DATE	AvailabilityEndDate isn't supported for public offers.
INVALID_AVAILABILITY_END_DATE	Provide a future AvailabilityEndDate.
INVALID_AVAILABILITY_END_DATE	Provide an AvailabilityEndDate that is before AgreementEndDate.
MISSING_AVAILABILITY_END_DATE	Provide an AvailabilityEndDate that is before the agreement's end date.

## Define the expiration date of agreements created using the offer

You can use the Catalog API to define the expiration date details of agreements created using the offer in AWS Marketplace.

This change type doesn't affect existing agreements.

**Note**

You can use the `UpdateValidityTerms` change type on a private offer that has already been [published](#) (also known as *released*). If buyers have already accepted the private offer, those existing agreements aren't affected.

For **AMI-based** and **container-based** products, if your private offer [pricing terms](#) include a term type that has a `Duration` (for example, the term types `FixedUpfrontPricingTerm` or `ConfigurableUpfrontPricingTerm`), your `AgreementDuration` set in this change type must be greater than the following: the number of days from today to the [expiration of the private offer](#) plus the number of days set in the `Duration` of those term types. This is because after a buyer accepts the private offer and the agreement is created, they can optionally purchase additional entitlements specified in those term types until the private offer expires. Furthermore, all additional entitlements must end before the agreement does. For example, if the buyer accepts the private offer on the first available day and then purchases entitlements on the last available day, those entitlements must not end after the agreement end date.

To define the expiration date details of agreements created using the offer, call the `StartChangeSet` API operation with the `UpdateValidityTerms` change type, as shown in the following example.

**Request Syntax**

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

{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "UpdateValidityTerms",
      "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "offer-123456789"
      },
      "DetailsDocument": {
        "Terms": [
          {
            "Type": "ValidityTerm",
```

```

        "AgreementDuration": "P12M",
        "AgreementStartDate": "2021-08-01",
        "AgreementEndDate": "2022-08-01"
    }
  ]
}
]
}

```

Provide information for the fields to add the `UpdateValidityTerms` change type:

- **Entity** (object) (required) – Your offer.
  - **Type** (string) (required) – The Type is always `Offer@1.0`.
  - **Identifier** (string) (required) – Your offer ID. For more information, see [Identifier](#).
- **DetailsDocument** (object) (required) – The JSON value of specifics of the request.
- **Terms** (array of structures) – List of validity terms that you want to update. Supported validity terms are:
  - **ValidityTerm** (object) – Defines the conditions that will keep an agreement, created from this offer, valid.
    - **Type** (string) – Category of the term being updated. `ValidityTerm`
    - **AgreementDuration** (string) – Defines the duration that the agreement remains active. If `AgreementStartDate` isn't provided, agreement duration is relative to the agreement signature time. The duration is represented in the ISO\_8601 format.
    - **AgreementStartDate** (string) – Defines the date when agreement starts. `AgreementStartDate` is represented in YYYY-MM-DD format. The agreement starts at 00:00:00.000 UTC on the date provided. If `AgreementStartDate` isn't provided, agreement start date is determined based on agreement signature time.
    - **AgreementEndDate** (string) – Defines the date when the agreement ends. The `AgreementEndDate` is represented in YYYY-MM-DD format. The agreement ends at 23:59:59.999 UTC on the date provided. If `AgreementEndDate` isn't provided, the agreement end date is determined by the validity of individual terms.

## Response Syntax

A change set is created for your request. The response to this request gives you the `ChangeSetId` and `ChangeSetArn` 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 includes validating information to ensure that it meets the AWS Marketplace guidelines. The validation process can take a few minutes.

You can check the status of the request through the AWS Marketplace Management Portal, or directly through Catalog API using the [DescribeChangeSet](#) API operation.

### Synchronous Validations

The following schema validations are specific to UpdateValidityTerms actions in the AWS Marketplace Catalog API. These validations are performed when you call StartChangeSet. If the request doesn't meet the following requirements, it will fail with an HTTP response.

Input field	Validation rule	HTTP code
Terms	Required	422
Terms[].Type	Required Can only be "ValidityTerm"	422
Terms[].AgreementDuration	Optional Expected format per Selector type: ISO 8601 duration Can be stand alone or paired with AgreementStartDate	422
Terms[].AgreementEndDate	Optional Date must be formatted like "YYYY-MM-DD"	422
Terms[].AgreementStartDate	Optional	422

Input field	Validation rule	HTTP code
	Date must be formatted like "YYYY-MM-DD"	
	Can only be paired with Agreement EndDate and Agreement Duration	

## Asynchronous Errors

The following errors are specific to UpdateValidityTerms actions in the AWS Marketplace Catalog API. These errors are returned when you call DescribeChangeSet after a change set is processing. For more information about using DescribeChangeSet to get the status of a change request, see [Working with change sets](#).

Error code	Error message
INCOMPATIBLE_AGREEMENT	AgreementStartDate can't be in the future when the current agreement to be replaced isn't future dated.
INCOMPATIBLE_AGREEMENT_END_DATE	AgreementEndDate can't be updated after the offer is released.
INCOMPATIBLE_AGREEMENT_START_DATE	AgreementStartDate can't be updated after the offer is released.
INCOMPATIBLE_PRODUCT	AgreementStartDate in the future isn't supported.
INCOMPATIBLE_RESALE_AUTHORIZATION	Ensure the duration between Agreement StartDate and AgreementEndDate is compatible with the ResaleAuthorization.
INCOMPATIBLE_RESALE_AUTHORIZATION	Ensure AgreementStartDate is compatible with the ResaleAuthorization.

Error code	Error message
INCOMPATIBLE_RESALE_AUTHORIZATION	Ensure AgreementEndDate is compatible with the ResaleAuthorization.
INCOMPATIBLE_RESALE_AUTHORIZATION	Ensure the duration between AgreementStartDate and AgreementEndDate is compatible with the ResaleAuthorization.
INCOMPATIBLE_RESALE_AUTHORIZATION	Ensure AgreementDuration matches duration specified in the ResaleAuthorization.
INCOMPATIBLE_TERMS	ValidityTerm isn't supported for public offers.
INCOMPATIBLE_TERMS	The requested change can't be performed after the offer is expired.
INVALID_AGREEMENT_DURATION	Provide AgreementDuration that is greater than or equal to [x] days.
INVALID_AGREEMENT_END_DATE	Provide a future AgreementEndDate.
INVALID_AGREEMENT_END_DATE	Provide AgreementEndDate that is after or equal to [x].
INVALID_AGREEMENT_START_DATE	Provide an AgreementStartDate that is after AvailabilityEndDate.
INVALID_AGREEMENT_START_DATE	Provide an AgreementStartDate that is before the AgreementEndDate.
INVALID_AGREEMENT_START_DATE	Provide an AgreementStartDate that is within [x] years from today.
INVALID_AGREEMENT_TIME_INTERVAL	ValidityTerm with both AgreementDuration and AgreementEndDate isn't supported.
INVALID_AGREEMENT_TIME_INTERVAL	ValidityTerm with both AgreementStartDate and AgreementDuration isn't supported in an offer for the product.

Error code	Error message
INVALID_AGREEMENT_TIME_INTERVAL	ValidityTerm with AgreementStartDate isn't supported in an offer for the product.
INVALID_AGREEMENT_TIME_INTERVAL	ValidityTerm with only AgreementStartDate isn't supported.
INVALID_AGREEMENT_TIME_INTERVAL	AgreementEndDate isn't supported unless it's used in combination with a future AgreementStartDate or for replacement offers.
INVALID_AGREEMENT_TIME_INTERVAL	Provide AgreementStartDate and AgreementEndDate where the difference is less than or equal to [x] years.
MISSING_AGREEMENT_START_DATE	Ensure AgreementStartDate is present in ValidityTerm when used along with ConfigurableUpfrontPricingTerm.

## Update payment schedule details

You can use the Catalog API to update payment schedule details for your offer, such as flexible payment schedule, in AWS Marketplace.

### Note

You cannot use the `UpdatePaymentScheduleTerms` change type on an offer that has already been [published](#) (also known as *released*).

The private offer can be accepted any day between the creation of the private offer and its [expiration](#) (set in the `AvailabilityEndDate`). Only one `ChargeDate` value of the payment schedule can be a date on or before the last day the buyer can accept the private offer (the private offer expiration date). The remaining values of `ChargeDate` must be after the private offer expiration, but no later than the end of the agreement if the private offer was accepted immediately. The end of the agreement is based on when the private offer is accepted (creating the agreement) plus the [duration of the agreement](#).

To update payment schedule details for your offer, call the `StartChangeSet` API operation with the `UpdatePaymentScheduleTerms` change type, as shown in the following example.

## Request Syntax

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

{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "UpdatePaymentScheduleTerms",
      "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "offer-123456789"
      },
      "DetailsDocument": {
        "Terms": [
          {
            "Type": "PaymentScheduleTerm",
            "Schedule": [
              {
                "ChargeDate": "2021-12-01",
                "ChargeAmount": "200.00"
              },
              {
                "ChargeDate": "2022-03-01",
                "ChargeAmount": "250.00"
              }
            ]
          }
        ]
      }
    }
  ]
}
```

Provide information for the fields to add the `UpdatePaymentScheduleTerms` change type:

- **Entity** (object) (required) – Your offer.
- **Type** (string) (required) – The Type is always `Offer@1.0`.

- **Identifier** (string) (required) – Your offer ID. For more information, see [Identifier](#).
- **DetailsDocument** (object) (required) – The JSON value of specifics of the request.
- **Terms** (array of structures) – List of payment terms that you want to update. Supported payment terms are:
  - **PaymentScheduleTerm** (object) – Defines an installment-based pricing model where customers are charged a fixed price on different dates during the agreement validity period.
  - **Type** (string) – Type of the term being updated. This is the object value: "PaymentScheduleTerm".
  - **Schedule** (array of structures) – List of the payment schedule where each element defines one installment of payment. It contains the information necessary for calculating the price to be paid and the date on which the customer would be charged.
    - **ChargeDate** (string) – The date on which the customer would pay the price defined in this payment schedule term. ChargeDate is represented in YYYY-MM-DD format. Invoices are generated on the date provided.
    - **ChargeAmount** (string) – The price that the customer would pay on scheduled date (ChargeDate).

## Response Syntax

A change set is created for your request. The response to this request gives you the `ChangeSetId` and `ChangeSetArn` 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 includes validating information to ensure that it meets the AWS Marketplace guidelines. The validation process can take a few minutes.

You can check the status of the request through the AWS Marketplace Management Portal, or directly through Catalog API using the [DescribeChangeSet](#) API operation.

## Synchronous Validations

The following schema validations are specific to UpdatePaymentScheduleTerms actions in the AWS Marketplace Catalog API. These validations are performed when you call StartChangeSet. If the request doesn't meet the following requirements, it will fail with an HTTP response.

Input Field	Validation Rule	HTTP
Terms	Required  Only PaymentScheduleTerm is allowed  List size must be less than 2	422
Terms[].Type	Required  Can only be PaymentScheduleTerm	422
Terms[].PaymentScheduleTerm.CurrencyCode	Required  Supported currencies: ["USD"]	422
Terms[].PaymentScheduleTerm.Schedule[]	Required	422
Terms[].PaymentScheduleTerm.Schedule[].ChargeAmount	Required  Date type is "String"  Non-negative decimals with up to 2 decimal places supported	422
Terms[].PaymentScheduleTerm.Schedule[].ChargeDate	Required  Date must be formatted like "YYYY-MM-DD"	422

## Asynchronous Errors

The following errors are specific to UpdatePaymentScheduleTerms actions in the AWS Marketplace Catalog API. These errors are returned when you call DescribeChangeSet after a change set is processing. For more information about using DescribeChangeSet to get the status of a change request, see [Working with change sets](#).

Error code	Error message
DUPLICATE_CHARGE_DATES	Provide unique charge dates in PaymentScheduleTerm.
INCOMPATIBLE_MARKUP_PERCENTAGE	PaymentScheduleTerm isn't supported when MarkupPercentage is greater than zero (0).
INCOMPATIBLE_RESALE_AUTHORIZATION	Provide term(s) that are compatible with the ResaleAuthorization. Incompatible terms: [PaymentScheduleTerm].
INCOMPATIBLE_RESALE_AUTHORIZATION	Ensure the total ChargeAmounts in PaymentScheduleTerm is compatible with the ResaleAuthorization.
INCOMPATIBLE_TERMS	The requested change can't be performed after the offer is released.
INCOMPATIBLE_TERMS	The requested change can't be performed after the offer is expired.
INVALID_CHARGE_DATES	Provide charge dates before AgreementEndDate.
TOO_MANY_BACKDATED_CHARGES	Provide up to 1 scheduled payment before AvailabilityEndDate.

## Modify renewal options

You can use the Catalog API to control renewal options of the agreements that are created using this offer in AWS Marketplace.

For offers created through Catalog API, auto-renewal remains disabled by default until you call the `UpdateRenewalTerms` change type to allow auto-renewal. This change does not affect existing agreements.

To control renewal options of the agreements that are created using this offer, call the `StartChangeSet` API operation with the `UpdateRenewalTerms` change type, as shown in the following example.

## Request Syntax

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

{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "UpdateRenewalTerms",
      "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "offer-123456789"
      },
      "DetailsDocument": {
        "Terms": [
          {
            "Type": "RenewalTerm"
          }
        ]
      }
    }
  ]
}
```

Provide information for the fields to add the `UpdateRenewalTerms` change type:

- **Entity** (object) (required) – Your offer.
  - **Type** (string) (required) – The Type is always `Offer@1.0`.
  - **Identifier** (string) (required) – Your offer ID. For more information, see [Identifier](#).
- **DetailsDocument** (object) (required) – The JSON value of specifics of the request.
  - **Terms** (array of structures) – List of renewal terms that you want to update. Supported renewal terms are:

- **RenewalTerm** (object) – Defines that on graceful termination (expiration of the `ValidityTerm`, not buyer or AWS Marketplace cancellation) of the agreement, a new agreement will be created using the accepted terms on the existing agreement. In other words, the agreement will be renewed. Presence of `RenewalTerm` in the offer means that auto-renewal is allowed. Buyers will have the option to accept or decline auto-renewal at the offer acceptance/agreement creation.
- **Type** (string) – Type of the term being updated. `RenewalTerm`

## Response Syntax

A change set is created for your request. The response to this request gives you the `ChangeSetId` and `ChangeSetArn` 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 includes validating information to ensure that it meets the AWS Marketplace guidelines. The validation process can take a few minutes.

You can check the status of the request through the AWS Marketplace Management Portal, or directly through Catalog API using the [DescribeChangeSet](#) API operation.

## Synchronous Validations

The following schema validations are specific to `UpdateRenewalTerms` actions in the AWS Marketplace Catalog API. These validations are performed when you call `StartChangeSet`. If the request doesn't meet the following requirements, it will fail with an HTTP response.

Input field	Validation rule	HTTP code
Terms	Required	422
Terms[].Type	Required Can only be "RenewalTerm"	422

## Asynchronous Errors

The following errors are specific to `UpdateRenewalTerms` 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](#).

Error code	Error message
INCOMPATIBLE_PRODUCT	RenewalTerm isn't supported in private offers for the product.
INCOMPATIBLE_TERMS	RenewalTerm isn't supported together with PaymentScheduleTerm.
INCOMPATIBLE_TERMS	RenewalTerm isn't supported with the PricingModel.
INCOMPATIBLE_TERMS	The requested change can't be performed after the offer is released.
INCOMPATIBLE_TERMS	The requested change can't be performed after the offer is expired.

## Publish an offer

You can use the Catalog API to merge the information collected from all update change types, and then publish the offer.

Offers remain in a `Draft` state, until `ReleaseOffer` is called. After the offer is released, it's discoverable in AWS Marketplace.

To publish your offer, call the `StartChangeSet` API operation with the `ReleaseOffer` change type, as shown in the following example.

### Request Syntax

```
POST /StartChangeSet HTTP/1.1
```

```
Content-type: application/json

{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "ReleaseOffer",
      "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "offer-123456789"
      },
      "DetailsDocument": {}
    }
  ]
}
```

Provide information for the fields to add the `ReleaseOffer` change type:

- **Entity** (object) – The named type of entity being created. The `Identifier` is your offer ID, and the `Type` is always `Offer@1.0`. For more information, see [Identifier](#).
- **DetailsDocument** (object) – The JSON value of specifics of the request. It must be empty for `ReleaseOffer`.

## Response Syntax

A change set is created for your request. The response to this request gives you the `ChangeSetId` and `ChangeSetArn` 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 includes validating information to ensure that it meets the AWS Marketplace guidelines. The validation process can take a few minutes.

You can check the status of the request through the AWS Marketplace Management Portal, or directly through Catalog API using the [DescribeChangeSet](#) API operation.

## Synchronous Validations

The following schema validations are specific to `ReleaseOffer` actions in the AWS Marketplace Catalog API. These validations are performed when you call `StartChangeSet`. If the request doesn't meet the following requirements, it will fail with an HTTP response.

Input field	Validation rule
DetailsDocument	Must be empty ( {} )

## Asynchronous Errors

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

Error code	Error message
INCOMPATIBLE_PRODUCT	First create a public offer for the product.
INVALID_UPDATE_REQUEST	The requested change can't be performed after the offer is released.
MISSING_AGREEMENT_END_DATE	Provide an <code>AgreementEndDate</code> for replacement offers.
MISSING_AVAILABILITY_END_DATE	Provide an <code>AvailabilityEndDate</code> for private offer.
MISSING_BUYER_ACCOUNTS	Provide <code>PositiveTargeting</code> with <code>BuyersAccounts</code> for offers created using <code>ResaleAuthorization</code> .
MISSING_BUYER_ACCOUNTS	All offers for the product must be private. Provide <code>PositiveTargeting</code> with <code>BuyersAccounts</code> .

Error code	Error message
MISSING_DESCRIPTION	Set Description before releasing the offer.
MISSING_MANDATORY_TERMS	Add [x] to the offer.
MISSING_MANDATORY_TERMS	Provide a FixedUpfrontPricingTerm when the offer contains a PaymentScheduleTerm.
MISSING_NAME	Set Name before releasing the offer.
TOO_MANY_OFFERS	Only one public free trial offer can be created per product.
TOO_MANY_OFFERS	Only one public offer can be created per product.
INCOMPATIBLE_TARGETING	PreExistingAgreement is only supported for buyer targeted offers.

## Describe existing offer details

You can use the Catalog API to describe existing offer details in AWS Marketplace.

To describe existing offer details, call the `DescribeEntity` API operation with the `Offer@1.0` entity type, as shown in the following example.

### Request Syntax

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

Provide information for the fields to add the `DescribeEntity` change type:

- **catalog** (string) – The catalog related to the request. Fixed value: `AWSMarketplace`.
- **entityId** (string) – The unique ID of the offer to describe.

### Response Syntax

The response to this request gives you the offer details and looks like the following.

```
{
  "EntityType": "Offer@1.0",
  "EntityIdentifier": "offer-ad8EXAMPLE51@1",
  "EntityArn": "arn:aws:aws-marketplace:us-east-1:111122223333:AWSMarketplace/Offer/offer-ad8EXAMPLE51",
  "LastModifiedDate": "2021-03-10T21:57:16Z",
  "DetailsDocument": {
    "Id": "offer-3rEXAMPLErn",
    "State": "Released",
    "Name": "Test Offer",
    "Description": "Worldwide offer for Test Product",
    "PreExistingAgreement": {
      "AcquisitionChannel": "External",
      "PricingModel": "Contract"
    },
    "ProductId": "prod-ad8EXAMPLE51",
    "Terms": [
      {
        "Type": "SupportTerm",
        "RefundPolicy": "If you need to request a refund for software sold by Amazon Web Services, LLC, please contact AWS Customer Service."
      },
      {
        "Type": "LegalTerm",
        "Documents": [
          {
            "Type": "CustomEula",
            "Url": "https://s3.amazonaws.com/EULA/custom-eula-1234.txt"
          }
        ]
      }
    ],
    {
      "Type": "FreeTrialPricingTerm",
      "Duration": "P30D",
      "Grants": [
        {
          "DimensionKey": "m3.xlarge",
          "MaxQuantity": 10
        },
        {
          "DimensionKey": "m4.xlarge",
          "MaxQuantity": 10
        }
      ]
    }
  ]
}
```

```
    ]
  },
  {
    "Type": "ConfigurableUpfrontPricingTerm",
    "CurrencyCode": "USD",
    "RateCards": [
      {
        "Selector": {
          "Type": "Duration",
          "Value": "P365D"
        },
        "RateCard": [
          {
            "DimensionKey": "m3.large",
            "Price": "300.00"
          },
          {
            "DimensionKey": "m4.xlarge",
            "Price": "400.00"
          }
        ],
        "Constraints": {
          "MultipleDimensionSelection": "Allowed",
          "QuantityConfiguration": "Allowed"
        }
      }
    ]
  },
  {
    "Type": "UsageBasedPricingTerm",
    "CurrencyCode": "USD",
    "RateCards": [
      {
        "RateCard": [
          {
            "DimensionKey": "m3.large",
            "Price": "0.10"
          },
          {
            "DimensionKey": "m4.xlarge",
            "Price": "0.20"
          }
        ]
      }
    ]
  }
}
```

```
    ]
  },
  {
    "Type": "FixedUpfrontPricingTerm",
    "CurrencyCode": "USD",
    "Price": "200.00",
    "Grants": [
      {
        "DimensionKey": "Users",
        "MaxQuantity": 10
      }
    ]
  },
  {
    "Type": "RecurringPaymentTerm",
    "CurrencyCode": "USD",
    "BillingPeriod": "Monthly",
    "Price": "100.0"
  },
  {
    "Type": "PaymentScheduleTerm",
    "CurrencyCode": "USD",
    "Schedule": [
      {
        "ChargeDate": "2020-12-01T00:00:00.000Z",
        "ChargeAmount": "1000.00"
      },
      {
        "ChargeDate": "2021-06-15T00:00:00.000Z",
        "ChargeAmount": "1250.00"
      }
    ]
  },
  {
    "Type": "ByolPricingTerm"
  },
  {
    "Type": "RenewalTerm"
  }
],
"Rules": [
  {
    "Type": "TargetingRule",
    "PositiveTargeting": {
```

```

    "CountryCodes": [
      "US",
      "CA"
    ],
    "BuyerAccounts": [
      "444455556666"
    ]
  },
  "NegativeTargeting": {
    "CountryCodes": [
      "XX"
    ]
  }
},
{
  "Type": "AvailabilityRule",
  "AvailabilityEndDate": "2024-08-30T01:56:03.000Z"
}
]
}
}

```

The following is information about the fields you see in the DescribeEntity response.

- **EntityType** (string) – The named type of the entity, which is Offer@1.0.
- **EntityIdentifier** (string) – The identifier of the entity, in the format of EntityId@RevisionId.
- **EntityArn** (string) – The ARN associated to the unique identifier for the change set referenced in this request.
- **LastModifiedDate** (string) – The last modified date of the entity, in ISO 8601 format (for example: 2018-02-27T13:45:22Z).
- **Details** (string) – This stringified JSON object includes the following details of the entity:
  - **Id** (string) – Unique identifier for an offer entity in AWS Marketplace and is generated during the creation of an offer.
  - **State** (string) – The status of the offer.
  - **Name** (string) – The name associated with the offer for better readability to you and your customers. It will be displayed as part of Agreement information as well.
  - **Description** (string) – Description is a free-form text which is meant to be used only by you and will never be exposed to buyers.

- **PreExistingAgreement** (string) – Determines if this offer is a renewal for an existing agreement with an existing customer for the same underlying product. The existing agreement can be within or outside AWS Marketplace. AWS may audit and verify your offer is a renewal. If AWS is unable to verify your offer, then AWS may revoke the offer and entitlements from your customer.
- **AcquisitionChannel** (string) – Indicates if the existing agreement was signed outside AWS Marketplace or within AWS Marketplace. Possible values: `External`, `AwsMarketplace`.
- **PricingModel** (string) – Indicates which pricing model the existing agreement uses. Possible values: `Contract`, `Usage`, `Byo1`, `Free`.
- **ProductId** (string) – The unique identifier of the product being offered.
- **Terms** (array of structures) – List of terms.
- **Rules** (array of structures) – List of rules.

## Working with Resale Authorizations

You can use the AWS Marketplace Catalog API to automate tasks for working with Resale Authorizations.

While the *product* describes what is being sold in AWS Marketplace, the *Resale Authorization* (also known as an opportunity) describes the terms and rules regarding how this product is authorized to be resold in AWS Marketplace. The *CPPO* is the target of the Resale Authorization.

A Resale Authorization has a collection of terms and rules to be accepted for a reseller agreement between manufacturers and channel partners. Accepting the terms of the Resale Authorization allows the reseller to create offers for the product per the conditions expressed in the terms.

There are two types of rules in a Resale Authorization:

- **AvailabilityRule** – Controls the lifecycle of the Resale Authorization in AWS Marketplace.
- **PartnerTargetingRule** – Specifies whether the Resale Authorization should be accessible to a specific set of channel partners.

See the following resources:

- For end-to-end labs with working code examples, see [Lab: Authorize a reseller](#) in the *AWS Marketplace seller workshop*.

- For code examples of API requests, see [Python](#) and [Java](#) examples in *AWS Samples* on GitHub.
- For a video on creating resale authorizations, see [Create Resale Authorizations Using the AWS Marketplace Catalog API](#) on YouTube.

The following topics describe how to use the Catalog API to create and update Resale Authorizations:

## Topics

- [Resale Authorization prerequisites](#)
- [Create a new Resale Authorization](#)
- [Update buyer targeting](#)
- [Update availability](#)
- [Update the validity of a future dated agreement](#)
- [Update legal resources](#)
- [Update pricing](#)
- [Update payment schedule](#)
- [Update Resale Authorization details](#)
- [Restrict a Resale Authorization](#)
- [Release a Resale Authorization and make it visible to a Channel Partner](#)
- [Describe an existing Resale Authorization](#)

## Resale Authorization prerequisites

To use Resale Authorization, both independent software vendors (ISVs) and AWS Marketplace Channel Partners must create a service-linked role that provides resource-sharing permissions to AWS. If both groups don't perform this prerequisite, AWS can't share the authorization resource from the ISV to the AWS Marketplace Channel Partner. For more information, see [Using roles for Resale Authorization for AWS Marketplace](#) in the *AWS Marketplace Seller Guide*.

## Create a new Resale Authorization

You can use the Catalog API to create a new Resale Authorization in AWS Marketplace.

If your request is processed successfully, AWS Marketplace Catalog API generates a Resale Authorization in Draft state for you. It's an incomplete Resale Authorization and not visible to channel partners in AWS Marketplace.

Use the Update change types to complete the Resale Authorization. After the Resale Authorization is completed, use the ReleaseResaleAuthorization change type to complete the Resale Authorization creation process and release the Resale Authorization, which will validate the entire Resale Authorization and make your it visible to channel partners in AWS Marketplace.

To create a Resale Authorization in Draft state, call the StartChangeSet API operation with the CreateResaleAuthorization change type, as shown in the following example.

## Request Syntax

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

{
  "Catalog": "AWSMarketplace",
  "ChangeSet":
  [
    {
      "ChangeType": "CreateResaleAuthorization",
      "ChangeName": "xyz",
      "Entity":
      {
        "Type": "ResaleAuthorization@1.0"
      },
      "DetailsDocument":
      {
        "ProductId": "prod-ad8EXAMPLE51",
        "Name": "Test ResaleAuthorization",
        "Description": "Worldwide ResaleAuthorization for Test Product",
        "ResellerAccountId": "777788889999"
      }
    }
  ]
}
```

Provide information for the input fields to add the CreateResaleAuthorization change type:

- **Entity** (object) (required) – Your Resale Authorization.

- **Type** (string) (required) – The Type is always `ResaleAuthorization@1.0`.
- **DetailsDocument** (object) (required) – Specifics of the request.
  - **ProductId** (string) (required) – Product ID for which to create the resale authorization.
  - **Name** (string) (required) – Name associated with the `ResaleAuthorization` for better readability to you and your channel partners.
  - **Description** (string) (optional) – A free-form text field available to add details about the `ResaleAuthorization`.
  - **ResellerAccountId** (string) (required) – Add targeted channel partner's AWS account who can describe and use this `ResaleAuthorization` to create a private offer.

## Response Syntax

A change set is created for your request. The response to this request gives you the `ChangeSetId` and `ChangeSetArn` 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.

You can check the status of the request through the AWS Marketplace Management Portal, or directly through Catalog API using the [DescribeChangeSet](#) API operation.

When the request is complete (if the Status is SUCCEEDED), a new `ResaleAuthorization` is generated. Although the SUCCEEDED status indicates that the `CreateResaleAuthorization` change type call is completed, the `ResaleAuthorization` status is still in Draft state.

The following shows the response from the [DescribeChangeSet](#) API operation.

```
{
  "ChangeSetId": "example123456789012abcdef",
  "ChangeSetArn": "arn:aws:aws-marketplace:us-east-
1:123456789012:AWSMarketplace/ChangeSet/example123456789012abcdef",
  "ChangeSetName": "Submitted by 123456789012",
  "StartTime": "2021-05-27T22:21:26Z",
  "EndTime": "2021-05-27T22:32:19Z",
}
```

```

"Status": "SUCCEEDED",
"ChangeSet": [
  {
    "ChangeType": "CreateResaleAuthorization",
    "Entity": {
      "Type": "ResaleAuthorization@1.0",
      "Identifier": "resaleauthz-123456789"
    },
  },
  "Details": "{
    \"ProductId\": \"prod-ad8EXAMPLE51\",
    \"Name\": \"Test ResaleAuthorization\",
    \"Description\": \"Worldwide ResaleAuthorization for Test Product\",
    \"ResellerAccountId\": \"777788889999\",
    \"BulkRequestId\": \"84977023-5093-4a66-8b24-ef2c5a2f8b1f\"
  }",
  "ErrorDetailList": []
}
]
}

```

## Synchronous Validations

The schema validations are specific to `CreateResaleAuthorization` actions in the AWS Marketplace Catalog API. The validations are performed when you call `StartChangeSet`. If the request doesn't meet the following requirements, it will fail with an HTTP response.

Input field	Validation rule	HTTP code
ProductId	Required  Must not be null or empty  Length must be between 1 and 50 characters	422
ProductId	User must be authorized to create ResaleAuthorization for the given product	403
ProductId	Must be an existing product in the catalog and not in Draft state	404

Input field	Validation rule	HTTP code
	Product should be supported to resell	
Name	Required  Must not be null or empty  Length must be between 1 and 100 characters  No special characters allowed	422
Description	Optional  Length must be between 1 and 255 characters  No special characters allowed	422
ResellerAccountId	Required  Must not be empty  AWS account IDs must be in valid format (12-digit number)	422
BulkRequestId	Optional  Length must be between 1 and 50 characters  Must be in UUID format	422
An unknown property	No additional properties are allowed	422

## Asynchronous Errors

The following errors are specific to `CreateResaleAuthorization` actions in the AWS Marketplace Catalog API. These errors are returned when you call `DescribeChangeSet` after a

change set is processing. For more information about using `DescribeChangeSet` to get the status of a change request, see [Working with change sets](#).

Error code	Error message
INVALID_RESELLER_ACCOUNT	Provide a valid reseller account.

## Update buyer targeting

You can use the Catalog API to update buyers targeting your Resale Authorization in AWS Marketplace.

Any existing targeting options that aren't included in the latest request are removed from the Resale Authorization. This change type is optional for release of the Resale Authorization.

To update buyers targeting your Resale Authorization, call the `StartChangeSet` API operation with the `UpdateBuyerTargetingTerms` change type, as shown in the following example.

### Request Syntax

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

{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "UpdateBuyerTargetingTerms",
      "Entity": {
        "Type": "ResaleAuthorization@1.0",
        "Identifier": "resaleauthz-123456789"
      },
      "Details": "{
        \"Terms\": [
          {
            \"Type\": \"BuyerTargetingTerm\",
            \"PositiveTargeting\": {
              \"BuyerAccounts\": [
                \"444455556666\"
              ]
            }
          }
        ]
      }"
```

```

    }
  ]
}

```

Provide information for the fields to add the `UpdateBuyerTargetingTerms` change type:

- **Entity** (object) (required) – Your Resale Authorization.
- **Type** (string) (required) – The Type is always `ResaleAuthorization@1.0`.
- **Identifier** (string) (required) – Your Resale Authorization ID. For more information, see [Identifier](#).
- **Details** (string) (required) – Specifics of the request.
- **Terms** (array of structures) (optional) – List of buyers targeting terms that you want to update. If the intentions aren't to target the `ResaleAuthorization` to any specific buyer, then terms field can be skipped. By default, `ResaleAuthorization` is targeted to all buyers. Supported terms are:
  - **BuyerTargetingTerms** (object) (optional) – Define buyer-specific targeting to your `ResaleAuthorization`.
    - **Type** (string) (required) – Category of the term being updated.
    - **PositiveTargeting** (object) (required) – Defines the criteria that any buyer's profile should fulfill to be allowed access to the `ResaleAuthorization`.
    - **BuyerAccounts** (array of strings) (optional) – List as optional. You can add the targeted buyer's AWS accounts. If the intention isn't to target `ResaleAuthorization` to specific buyers, then this field should be omitted. By default, all buyers are targeted. Targeted channel partners can choose to create a private offer and target a subset of buyers, if specified.

## Response Syntax

A change set is created for your request. The response to this request gives you the `ChangeSetId` and `ChangeSetArn` 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 includes validating information with the AWS Marketplace Seller Operations team to ensure it meets the AWS Marketplace guidelines. The validation process can take anywhere from a few minutes to a few hours.

You can check the status of the request through the AWS Marketplace Management Portal, or directly through Catalog API using the [DescribeChangeSet](#) API operation.

## Synchronous Validations

The schema validations are specific to UpdateBuyerTargetingTerms actions in the AWS Marketplace Catalog API. These validations are performed when you call StartChangeSet. If the request doesn't meet the following requirements, it will fail with an HTTP response.

Input field	Validation rule
Terms	<p>Optional</p> <p>Must not be null or empty</p> <p>Only "BuyerTargetingTerm" is allowed in the list</p> <p>List size must be 1 (there is no use case today that requires multiple buyer terms)</p>
BuyerTargetingTerm.PositiveTargeting	<p>Required</p> <p>Must not be empty</p>
BuyerTargetingTerm.PositiveTargeting.BuyerAccounts	<p>Optional</p> <p>AWS account IDs must be in valid format (12-digit number)</p>

Input field	Validation rule
	Must not contain more than 25 accounts
An unknown property	No additional properties are allowed

## Asynchronous Errors

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

Error code	Error message
INCOMPATIBLE_BUYER_TARGETING	At least one Buyer account must be present for ResaleAuthorization with PreExistingBuyerAgreement.

## Update availability

You can use the Catalog API to limit the availability of how many private offers are created or until what specific time a private offer can be created.

By default, the value is unlimited usage of this Resale Authorization, although you can check the availability under the rule list.

To control the availability and usability of your Resale Authorization, call the `StartChangeSet` API operation with the `UpdateAvailability` change type, as shown in the following example.

### Request Syntax

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

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

```
"ChangeSet":
[
  {
    "ChangeType": "UpdateAvailability",
    "Entity":
    {
      "Type": "ResaleAuthorization@1.0",
      "Identifier": "resaleauthz-123456789"
    },
    "DetailsDocument":
    {
      "AvailabilityEndDate": "2022-05-31",
      "OffersMaxQuantity": 1
    }
  }
]
```

Provide information for the fields to add the UpdateAvailability change type:

- **Entity** (object) (required) – Your Resale Authorization.
  - **Type** (string) (required) – The Type is always ResaleAuthorization@1.0.
  - **Identifier** (string) (required) – Your Resale Authorization ID. For more information, see [Identifier](#).
- **DetailsDocument** (object) (required) – Specifics of the request.
  - **AvailabilityEndDate** (string) (optional) – Define the end date until Channel Partners can leverage the ResaleAuthorization to create an offer. Channel Partners can use this ResaleAuthorization multiple times until the specified end date. Dates are represented in ISO\_8601 format.
  - **OffersMaxQuantity** (integer) (optional) – Define the maximum number of private offers that can be created using the ResaleAuthorization. This doesn't define the number of subscriptions.

## Response Syntax

A change set is created for your request. The response to this request gives you the ChangeSetId and ChangeSetArn 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 includes validating information with the AWS Marketplace Seller Operations team to ensure it meets the AWS Marketplace guidelines. The validation process can take anywhere from a few minutes to a few hours.

You can check the status of the request through the AWS Marketplace Management Portal, or directly through Catalog API using the [DescribeChangeSet](#) API operation.

## Synchronous Validations

The schema validations are specific to UpdateAvailability actions in the AWS Marketplace Catalog API. The validations are performed when you call StartChangeSet. If the request doesn't meet the following requirements, it will fail with an HTTP response

Input field	Validation rule
OffersMaxQuantity	Optional  Must be non-negative integer  Allowed value only "1" (Currently no use case to support multiple quantity)
AvailabilityEndDate	Optional  Must be ISO_8601 formatted  Must be date in the future
Availability	Provide either OffersMaxQuantity or AvailabilityEndDate.
An unknown property	No additional properties are allowed

## Asynchronous Errors

The following errors are specific to UpdateAvailability actions in the AWS Marketplace Catalog API. These errors are returned when you call DescribeChangeSet after a change set is

processing. For more information about using `DescribeChangeSet` to get the status of a change request, see [Working with change sets](#).

Error code	Error message
INVALID_AVAILABILITY_END_DATE	Provide an <code>AvailabilityEndDate</code> that is before all the <code>ChargeDate</code> in <code>ResalePaymentScheduleTerms</code> .
INVALID_AVAILABILITY_END_DATE	Provide a future <code>AvailabilityEndDate</code> .

## Update the validity of a future dated agreement

You can use the Catalog API to modify and control a future dated service start date in AWS Marketplace.

This change set is not mandatory to release a Resale Authorization.

To modify and control the product agreement duration of your Resale Authorization, call the `StartChangeSet` API operation with the `UpdateBuyerValidityTerms` change type, as shown in the following example.

### Request Syntax

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

{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "UpdateBuyerValidityTerms",
      "Entity": {
        "Type": "ResaleAuthorization@1.0",
        "Identifier": "resaleauthz-123456789"
      },
      "Details": "{
        \"Terms\": [
          {
            \"Type\": \"BuyerValidityTerm\",
```



The change request is added to a queue and processed. This includes validating information with the AWS Marketplace Seller Operations team to ensure it meets the AWS Marketplace guidelines. The validation process can take anywhere from a few minutes to a few hours.

You can check the status of the request through the AWS Marketplace Management Portal, or directly through Catalog API using the [DescribeChangeSet](#) API operation.

## Synchronous Validations

The schema validations are specific to UpdateBuyerValidityTerms actions in the AWS Marketplace Catalog API. The validations are performed when you call StartChangeSet. If the request doesn't meet the following requirements, it will fail with an HTTP response.

Input field	Validation rule
Terms	<p>Required</p> <p>Must not be null or empty</p> <p>Only "BuyerValidityTerm" is allowed in the list</p> <p>List size must be 1 (there's no use case today that requires multiple service availability terms)</p>
MaximumAgreementStartDate	<p>Required</p> <p>Must not be null or empty</p> <p>Must be future date and shouldn't exceed more than 3 years from now</p> <p>Must be ISO_8601 formatted</p>
An unknown property	No additional properties are allowed

## Asynchronous Errors

The following errors are specific to UpdateBuyerValidityTerms actions in the AWS Marketplace Catalog API. These errors are returned when you call DescribeChangeSet after a

change set is processing. For more information about using `DescribeChangeSet` to get the status of a change request, see [Working with change sets](#).

Error code	Error message
INCOMPATIBLE_PRODUCT	BuyerValidityTerm isn't supported for the product.
INVALID_MAXIMUM_AGREEMENT_START_DATE	Provide a future MaximumAgreementStartDate with in allowed limit.

## Update legal resources

You can use the Catalog API to replace the existing legal terms completely in AWS Marketplace.

The legal terms that aren't included in the latest request will be removed from the Resale Authorization. `BuyerLegalTerm` contains the EULA which will be included on the final buyer agreement and `LegalTerm` includes the Reseller Contract which will be included in the reseller agreement between the channel partner and the ISV.

To update legal terms of your `ResaleAuthorization`, call the `StartChangeSet` API operation with the `UpdateLegalTerms` change type, as shown in the following example.

### Request Syntax

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

{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "UpdateLegalTerms",
      "Entity": {
        "Type": "ResaleAuthorization@1.0",
        "Identifier": "resaleauthz-123456789"
      },
      "Details": "{
        \"Terms\": [
```

```

    {
      \"Type\": \"BuyerLegalTerm\",
      \"Documents\": [
        {
          \"Type\": \"CustomEula\",
          \"Url\": \"https://s3.amazonaws.com/EULA/custom-eula-1234.txt\"
        }
      ]
    },
    {
      \"Type\": \"ResaleLegalTerm\",
      \"Documents\": [
        {
          \"Type\":
            \"StandardResellerContract/CustomResellerContract\",
          \"Url\": \"https://s3.amazonaws.com/ResellerContract/custom-
reseller-contract-1234.txt\"
        }
      ]
    }
  ]
}

```

Provide information for the fields to add the `UpdateLegalTerms` change type:

- **Entity** (object) (required) – Your Resale Authorization.
  - **Type** (string) (required) – The Type is always `ResaleAuthorization@1.0`.
  - **Identifier** (string) (required) – Your Resale Authorization ID. For more information, see [Identifier](#).
- **Details** (string) (required) – Specifics of the request. This field is a JSON string field. It must be formatted properly for a single-line string field, including escaping characters (such as quotation marks) that can't directly be in a string.
- **Terms** (array of structures) (required) – List of legal terms. Supported legal terms are:
  - **BuyerLegalTerm** (object) (required) – Defines the list of text agreements to be proposed to acceptors. For example, the end user license agreement (EULA).
  - **Type** (string) (required) – Category of the term being updated.

- **Documents** (array of structures) (required) – List of references to legal resources to be proposed to the buyers. For example, the EULA. Each reference is made up of a Type and a URL:
- **Type** (string) (required) – Type of document. Available document types are:
  - **StandardEula** – Standard Contract for AWS Marketplace. For more information, see [SCMP](#) in the *AWS Marketplace Seller Guide*. You don't need to provide a URL for this type because it's managed by AWS Marketplace.
  - **EnterpriseEula** – Enterprise Contract for AWS Marketplace. For more information, see DSA in the *AWS Marketplace Seller Guide*. You don't need to provide a URL for this type because it's managed by AWS Marketplace.
  - **CustomEula** – Custom EULA provided by you as a manufacturer. A URL for the EULA stored in an accessible S3 bucket is required for this document type.
- **Url** (string) (conditionally required) – A URL to the legal document for buyers to read. This is required when category Type is CustomEula.
- **ResaleLegalTerm** (object) (optional) – Defines the list of text agreements to propose only to channel partners. This term won't be available to buyers.
  - **Type** (string) (required) – Category of term being updated.
  - **Documents** (array of structures) (required) – List of references to the reseller legal resources to be proposed to the channel partners.
    - **Type** (string) (required) – Category of the document. Available document types are:
      - **StandardResellerContract** – Standard Reseller Contract for AWS Marketplace.
      - **CustomResellerContract** – A custom reseller contract by you as a manufacturer. A URL for the reseller contract is stored in an accessible S3 bucket and is required for this document type.
    - **Url** (string) (conditionally required) – URL to the reseller contract document for channel partners to read. It's required when the Type is CustomResellerContract.

## Response Syntax

A change set is created for your request. The response to this request gives you the `ChangeSetId` and `ChangeSetArn` 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 includes validating information to ensure that it meets the AWS Marketplace guidelines. The validation process can take anywhere from a few minutes to a few hours.

You can check the status of the request through the AWS Marketplace Management Portal, or directly through Catalog API using the [DescribeChangeSet](#) API operation.

## Synchronous Validations

The schema validations are specific to UpdateLegalTerms actions in the AWS Marketplace Catalog API. The validations are performed when you call StartChangeSet. If the request doesn't meet the following requirements, it will fail with an HTTP response.

Input field	Validation rule	HTTP code
Terms	Required Must not be null or empty	422
Terms[].BuyerLegalTerm	Required Must not be null or empty	422
Terms[].ResaleLegalTerm	Optional Must not be null or empty if present	422
Terms[].BuyerLegalTerm.Documents	Required Must not be null or empty	422
Terms[].BuyerLegalTerm.Documents[].Type	Required Must not be null or empty Allowed values: • StandardEula	422

Input field	Validation rule	HTTP code
	<ul style="list-style-type: none"> <li>EnterpriseEula</li> <li>CustomEula</li> </ul>	
Terms[].BuyerLegalTerm.Documents[].Url	<p>Required and must be a valid URL when "Type" is "CustomEula"</p> <p>Must not be provided when "Type" is one of ["StandardEula", "EnterpriseEula"]</p>	422
Terms[].ResaleLegalTerm.Documents	<p>Required</p> <p>Must not be null or empty</p>	422
Terms[].ResaleLegalTerm.Documents[].Type	<p>Required</p> <p>Must not be null or empty Allowed values:</p> <ul style="list-style-type: none"> <li>StandardEula</li> <li>CustomResellerContract</li> </ul>	422
Terms[].ResaleLegalTerm.Documents[].Url	<p>Required and must be a valid URL when "Type" is "CustomResellerContract"</p> <p>Must not be provided when "Type" is one of ["StandardContract"]</p>	422
An unknown property	No additional properties are allowed	422

## Asynchronous Errors

The following errors are specific to UpdateLegalTerms actions in the AWS Marketplace Catalog API. These errors are returned when you call DescribeChangeSet, after a change set is processing. For more information about using DescribeChangeSet to get the status of a change request, see [Working with change sets](#).

Error code	Error message
INVALID_BUYER_LEGAL_DOCUMENTS	Provide URLs for buyer legal documents stored in accessible S3 buckets.
INVALID_RESALE_LEGAL_DOCUMENTS	Provide URLs for resale legal documents stored in accessible S3 buckets.
MISSING_MANDATORY_TERMS	Provide a BuyerLegalTerm.

## Update pricing

You can use the Catalog API to replace the existing pricing terms completely in AWS Marketplace.

Pricing terms that aren't included in the latest request will be removed from the Resale Authorization. You can update the discounted pricing for your product through this API.

To update pricing details for your Resale Authorizations, call the `StartChangeSet` API operation with the `UpdatePricingTerms` change type, as shown in the following example.

### Request Syntax

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

{
  "Catalog": "AWSMarketplace",
  "ChangeSet":
  [
    {
      "ChangeType": "UpdatePricingTerms",
      "Entity":
      {
        "Type": "ResaleAuthorization@1.0",
        "Identifier": "resaleauthz-123456789"
      },
      "DetailsDocument":
      {
        "PricingModel": "Contract",
        "Terms":
        [
```

```
{
  "Type": "ResaleUsageBasedPricingTerm",
  "CurrencyCode": "USD",
  "RateCards":
  [
    {
      "RateCard":
      [
        {
          "DimensionKey": "m3.large",
          "Price": "0.10"
        },
        {
          "DimensionKey": "m4.xlarge",
          "Price": "0.20"
        }
      ]
    }
  ],
},
{
  "Type": "ResaleConfigurableUpfrontPricingTerm",
  "CurrencyCode": "USD",
  "RateCards":
  [
    {
      "Selector":
      {
        "Type": "Duration",
        "Value": "P12M"
      },
      "RateCard":
      [
        {
          "DimensionKey": "m3.large",
          "Price": "300"
        },
        {
          "DimensionKey": "m4.xlarge",
          "Price": "400"
        }
      ]
    },
    "Constraints":
    {
```



- **Terms** (array of structures) (required) – List of pricing terms that you want to update. Supported pricing terms are:
  - **ResaleUsageBasedPricingTerm** (object) – Defines a pay-as-you-go (PAYG) pricing model where the customers are charged based on product usage.
    - **Type** (string) (required) – Category of the term.
    - **CurrencyCode** (string) – Defines the currency for prices mentioned in this term. Currently, only USD is supported.
    - **RateCards** (array of structures) – List of rate cards.
      - **RateCard** (array of structures) – A rate card defines the per-unit rates for the product dimensions.
        - **DimensionKey** (string) – Dimension for which the given entitlement applies. Dimensions represent categories of capacity in a product and are specified when the product is listed in AWS Marketplace.
        - **Price** (string) – Per unit price for the product dimension which is used for calculating the amount to be charged.
      - **Constraints** (object) (optional) – Defines limits on how the term can be configured by acceptors.
        - **MultipleDimensionSelection** (string) (optional) – Determines if buyers are allowed to select multiple dimensions in the rate card. Possible values are Allowed and Disallowed. Default value is Allowed.
        - **QuantityConfiguration** (string) (optional) – Determines if acceptors are allowed to configure quantity for each dimension in rate card. Possible values are Allowed and Disallowed. Default value is Allowed.
  - **ResaleFixedUpfrontPricingTerm** (object) – Defines a prepaid pricing model where the customers are charged a fixed upfront amount.
    - **Type** (string) (required) – Category of the term being updated.
    - **CurrencyCode** (string) – Defines the currency for prices mentioned in this term. Currently, only USD is supported.
    - **Price** (string) (required) – Fixed amount to be charged to the customer when this term is accepted.
    - **Duration** (string) (required) – Contract duration of the ResaleAuthorization. This field supports the ISO 8601 format.

- **Grants** (array of structures) (required) – Entitlements that will be granted to the acceptor of fixed upfront pricing as part of agreement execution.
- **DimensionKey** (string) (required) – Unique dimension key defined in the product document. Dimensions represent categories of capacity in a product and are specified when the product is listed in AWS Marketplace.
- **MaxQuantity** (integer) (required) – Maximum amount of capacity that the buyer can be entitled to the given dimension of the product. If MaxQuantity is not provided, the buyer will be able to use an unlimited amount of the given dimension.

## Response Syntax

A change set is created for your request. The response to this request gives you the `ChangeSetId` and `ChangeSetArn` 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 includes validating information to ensure that it meets the AWS Marketplace guidelines. The validation process can take anywhere from a few minutes to a few hours.

You can check the status of the request through the AWS Marketplace Management Portal, or directly through Catalog API using the [DescribeChangeSet](#) API operation.

## Synchronous Validations

The following schema validations are specific to `UpdatePricingTerms` actions in the AWS Marketplace Catalog API. The validations are performed when you call `StartChangeSet`. If the request doesn't meet the following requirements, it will fail with an HTTP response.

Input field	Validation rule
Terms	Required
	Must not be null or empty

Input field	Validation rule
	<p>Each term must present only single time</p> <p>Allowed terms:</p> <ul style="list-style-type: none"> <li>• * ResaleUsageBasedPricingTerm</li> <li>• * ResaleConfigurableUpfrontPricingTerm</li> <li>• * ResaleFixedupfrontPricingTerm</li> </ul>
Terms[].ResaleUsageBasedPricingTerm.CurrencyCode	<p>Required</p> <p>Allowed values: USD</p>
Terms[].ResaleUsageBasedPricingTerm.Validity	<p>Required</p> <p>Must not be null or empty</p> <p>Expected format: ISO 8601 duration</p>
Terms[].ResaleUsageBasedPricingTerm.RateCards	<p>Required</p> <p>Must not be null or empty</p>
Terms[].ResaleUsageBasedPricingTerm.RateCards[].DimensionKey	<p>Required</p> <p>Must not be null or empty</p> <p>Length must be between 1 and 60</p>
Terms[].ResaleUsageBasedPricingTerm.RateCards[].Price	<p>Required</p> <p>Must not be null or empty</p> <p>Data type is "String"</p> <p>Must be non-negative</p> <p>Support up to 6 Decimal</p> <p>No special characters supported</p>

Input field	Validation rule
Terms[].ResaleConfigurableUpfrontPricingTerm.CurrencyCode	Required  Allowed values: USD
Terms[].ResaleConfigurableUpfrontPricingTerm.RateCards[].Selector.Type	Required  Must not be null or empty  Allowed values: Duration
Terms[].ResaleConfigurableUpfrontPricingTerm.RateCards[].Selector.Value	Required  Must not be null or empty  Expected format: ISO 8601 duration
Terms[].ResaleConfigurableUpfrontPricingTerm.RateCards[].RateCard.DimensionKey	Required  Must not be null or empty  Length must be between 1 and 60
Terms[].ResaleConfigurableUpfrontPricingTerm.RateCards[].RateCard.Price	Required  Must not be null or empty  Data type is "String"  Must be non-negative  Support up to 6 Decimal  No special characters supported
Terms[].ResaleConfigurableUpfrontPricingTerm.RateCards[].Constraints	Optional
Terms[].ResaleFixedUpfrontPricingTerm.CurrencyCode	Required  Allowed values: USD

Input field	Validation rule
Terms[].ResaleFixedUpfrontPricingTerm.Price	Required Must not be null or empty Data type is "String" Must be non-negative Support up to 6 Decimal No special characters supported Allowed values: 0.0
Terms[].ResaleFixedUpfrontPricingTerm.Duration	Required Must not be null or empty Expected format: ISO 8601 duration
Terms[].ResaleFixedUpfrontPricingTerm.Grants[].DimensionKey	Required Must not be null or empty Length must be between 1 and 60
Terms[].ResaleFixedUpfrontPricingTerm.Grants[].MaxQuantity	Required Must not be null or empty
An unknown property	No additional properties are allowed

## Asynchronous Errors

The following errors are specific to UpdatePricingTerms actions in the AWS Marketplace Catalog API. These errors are returned when you call DescribeChangeSet after a change set is processing. For more information about using DescribeChangeSet to get the status of a change request, see [Working with change sets](#).

Error code	Error message
INVALID_CURRENCY_CODE	Provide the same CurrencyCode across all pricing and payment terms.
INCOMPATIBLE_PRODUCT	Use existing, available dimensions in the product in [x].
DUPLICATE_DIMENSION_KEYS	Provide rate card with a unique list of dimension keys in [x]
INVALID_RATE_CARD	Provide dimensions that have the same unit in [x]
INVALID_RATE_CARD	Provide a rate card for only metered dimensions in ResaleUsageBasedPricingTerm.
INVALID_RATE_CARD	Provide usage based rates for all available metered dimensions in ResaleUsageBasedPricingTerm.
TOO_MANY_RATES	Provide RateCards within the allowed limits in ResaleUsageBasedPricingTerm.
DUPLICATE_SELECTORS	Provide a unique list of Selectors in ResaleConfigurableUpfrontPricingTerm.
INVALID_RATE_CARD	ConfigurableUpfrontPricingTerm is missing one or more dimension keys for duration [x]. Provide prices for the same set of dimension keys for all durations.
INVALID_RATE_CARD	Provide either all metered or all entitled dimensions in [x].
INCOMPATIBLE_RATE_CARD_CONSTRAINTS	Set MultipleDimensionSelection and QuantityConfiguration to Disallowed in ResaleConfigurableUpfrontPricingTerm for the PricingModel.

Error code	Error message
TOO_MANY_RATE_CARDS	Only one rate card in ConfigurableUpfrontPricingTerm is allowed for the product.
INCOMPATIBLE_TERMS	The following terms aren't compatible with the PricingModel: [x,y,z].
TOO_MANY_RATES	Provide RateCards within the allowed limits in [x term].
TOO_MANY_GRANTS	Provide up to [N] grants in [x term].
INVALID_SELECTOR_DURATION_VALUE	Provide duration between [x] and [y] months in ResaleConfigurableUpfront
TOO_MANY_GRANTS	Provide duration between [x] and [y] months.
INVALID_SELECTOR_DURATION_VALUE	Ensure duration granularity is at the day level for metered dimensions in ResaleConfigurableUpfront
INVALID_DURATION	Ensure duration granularity is at the day level for metered dimensions in FixedUpfront.
INVALID_RATE_CARD	Provide only entitled dimensions in [x].
MISSING_DURATION	Provide a Duration in [x].
DUPLICATE_DIMENSION_KEYS	Provide Grants with a unique list of dimension keys in [x].

## Update payment schedule

You can use the Catalog API to change payment-associated details, such as a flexible payment schedule, in AWS Marketplace.

To update payment-associated details for your Resale Authorization, call the `StartChangeSet` API operation with the `UpdatePaymentScheduleTerms` change type, as shown in the following example.

## Request Syntax

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

{
  "Catalog": "AWSMarketplace",
  "ChangeSet":
  [
    {
      "ChangeType": "UpdatePaymentScheduleTerms",
      "Entity":
      {
        "Type": "ResaleAuthorization@1.0",
        "Identifier": "resaleauthz-123456789"
      },
      "DetailsDocument":
      {
        "Terms":
        [
          {
            "Type": "ResalePaymentScheduleTerm",
            "CurrencyCode": "USD",
            "Schedule":
            [
              {
                "ChargeDate": "2021-12-01T00:00:00.000Z",
                "ChargeAmount": "200.00"
              },
              {
                "ChargeDate": "2022-03-01T00:00:00.000Z",
                "ChargeAmount": "250.00"
              }
            ]
          }
        ]
      }
    }
  ]
}
```

```
}
```

Provide information for the fields to add the `UpdatePaymentScheduleTerms` change type:

- **Entity** (object) (required) – Your Resale Authorization.
  - **Type** (string) (required) – The Type is always `ResaleAuthorization@1.0`.
  - **Identifier** (string) (required) – Your Resale Authorization ID. For more information, see [Identifier](#).
- **DetailsDocument** (object) – Specifics of the request.
- **Terms** (array of structures) – List of payment terms that you want to update. Supported payment terms are:
  - **ResalePaymentScheduleTerm** (object) – Defines an installment-based pricing model where the customers are charged a fixed price on different dates during the agreement validity period.
    - **Type** (string) – Category of the term being updated.
    - **CurrencyCode** (string) (required) – Defines the currency for the payment mentioned in the schedule. Currently, only USD is supported.
    - **Schedule** (array of structures) – List of the payment schedule where each element defines one installment of payment. It contains the information necessary for calculating the price to be paid and the date on which the customer would be charged.
      - **ChargeDate** (string) (required) – The date the customer would pay the price defined in this payment schedule term. This field supports the ISO 8601 format.
      - **ChargeAmount** (string) (required) – The price the customer would pay on a scheduled date (`ChargeDate`).

## Response Syntax

A change set is created for your request. The response to this request gives you the `ChangeSetId` and `ChangeSetArn` 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 includes validating information to ensure that it meets the AWS Marketplace guidelines. The validation process can take anywhere from a few minutes to a few hours.

You can check the status of the request through the AWS Marketplace Management Portal, or directly through Catalog API using the [DescribeChangeSet](#) API operation.

## Synchronous Validations

The schema validations are specific to UpdatePaymentScheduleTerms actions in the AWS Marketplace Catalog API. The validations are performed when you call StartChangeSet. If the request doesn't meet the following requirements, it will fail with an HTTP response.

Input field	Validation rule	HTTP
Terms.Type	Required  Not supported for [x] product  Allowed terms: ResalePaymentScheduleTerm	422
Terms[].CurrencyCode	Required  Allowed values: USD	422
Terms[].ResalePaymentScheduleTerm.Schedule	Required  Length must be between 1 and 60	422
Terms[].ResalePaymentScheduleTerm.Schedule.ChargeDate	Required  Must be in ISO 8601 format  Date must be in the future	422
Terms[].ResalePaymentScheduleTerm.Schedule.ChargeAmount	Required  Must be non-negative	422
An unknown property	No additional properties are allowed	422

## Asynchronous Errors

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

Error code	Error message
INCOMPATIBLE_TERMS	OffersMaxQuantity and AvailabilityEndDate must be present with ResalePaymentScheduleTerm.
TOO_MANY_SCHEDULED_PAYMENTS	Provide up to 60 scheduled payments in ResalePaymentScheduleTerm.
DUPLICATE_CHARGE_DATES	Provide unique charge dates in ResalePaymentScheduleTerm.
INVALID_CHARGE_DATES	Provide a future ChargeDate.
INVALID_CHARGE_DATES	Provide a last charge date that is before [x].
MISSING_MANDATORY_TERMS	Provide a ResaleFixedUpfrontPricingTerm and ResalePaymentScheduleTerm together.
INVALID_CURRENCY_CODE	Provide the same CurrencyCode across all pricing and payment terms.

## Update Resale Authorization details

You can use the Catalog API to update Resale Authorization details in AWS Marketplace.

To update Resale Authorization details, call the `StartChangeSet` API operation with the `UpdateInformation` change type, as shown in the following example.

**Note**

The UpdateInformation change type only updates the sections provided in the request; all other information remains unchanged.

**Request Syntax**

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

{
  "Catalog": "AWSMarketplace",
  "ChangeSet":
  [
    {
      "ChangeType": "UpdateInformation",
      "Entity":
      {
        "Type": "ResaleAuthorization@1.0",
        "Identifier": "resaleauthz-123456789"
      },
      "DetailsDocument":
      {
        "Name": "TestResaleAuthorization",
        "Description": "Worldwide ResaleAuthorization for Test Product",
        "PreExistingBuyerAgreement":
        {
          "AcquisitionChannel": "AwsMarketplace",
          "PricingModel": "Contract"
        }
      }
    }
  ]
}
```

Provide information for the fields to add the UpdateInformation change type:

- **Entity** (object) (required) – Your Resale Authorization.
  - **Type** (string) (required) – The Type is always ResaleAuthorization@1.0.

- **Identifier** (string) (required) – Your Resale Authorization ID. For more information, see [Identifier](#).
- **DetailsDocument** (object) (required) – Details of the request, including the information you want to update information for the Resale Authorization.
  - **Name** (string) (optional) – The name associated with the ResaleAuthorization for better readability to you and your channel partners.
  - **Description** (string) (optional) – The description is free-form text where you can add details about the ResaleAuthorization.
  - **PreExistingBuyerAgreement** (object) (optional) – Determines if this offer is a renewal for an existing agreement with an existing customer for the same underlying product. The existing agreement can be within or outside AWS Marketplace. AWS may audit and verify your offer is a renewal. If AWS is unable to verify your offer, then AWS may revoke the offer and entitlements from your customer.
    - **AcquisitionChannel** (string) (required) – Indicates if the existing buyer agreement was signed outside AWS Marketplace or in AWS Marketplace.
- **PricingModel** (string) (required) – Indicates which pricing model the exiting agreement uses.

Possible values: External, AwsMarketplace

Possible values: Contract, Usage, BYOL, Free

## Response Syntax

A change set is created for your request. The response to this request gives you the `ChangeSetId` and `ChangeSetArn` 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 includes validating information to ensure that it meets the AWS Marketplace guidelines. The validation process can take anywhere from a few minutes to a few hours.

You can check the status of the request through the AWS Marketplace Management Portal, or directly through Catalog API using the [DescribeChangeSet](#) API operation.

## Synchronous Validations

The following schema validations are specific to UpdateInformation actions in the AWS Marketplace Catalog API. These validations are performed when you call StartChangeSet, and the request will fail with an HTTP error if the input does not meet the following requirements.

Input field	Validation rule	HTTP code
Name	Optional  Must not be null or empty  Length must be between 1 and 100 characters  Pattern <code>^[A-Za-z0-9]*\$</code>  No special character or white space allowed	422
Description	Optional  Length must be between 1 and 255 characters  Pattern <code>^[A-Za-z0-9\\s]*\$</code>  No special characters allowed	422
An unknown property	No additional properties are allowed	422

## Asynchronous Errors

The following errors are specific to UpdateInformation actions in the AWS Marketplace Catalog API. These errors are returned when you call DescribeChangeSet after a change set is processing, or more information about using DescribeChangeSet to get the status of a change request, see [Working with change sets](#).

Error code	Error message
INCOMPATIBLE_BUYER_TARGETING	At least one Buyer account must be present for ResaleAuthorization with PreExistingBuyerAgreement.

## Restrict a Resale Authorization

You can use the Catalog API to set restrict rules to a Resale Authorization in AWS Marketplace.

A restricted Resale Authorization can no longer be used by a channel partner to create a private offer. An existing private offer won't be impacted.

To restrict your Resale Authorization, call the `StartChangeSet` API operation with the `RestrictResaleAuthorization` change type, as shown in the following example.

### Important

This is a non-reversible operation. After the Resale Authorization is marked as Restricted, it can't be in an Active state again.

## Request Syntax

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

{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "RestrictResaleAuthorization",
      "Entity": {
        "Type": "ResaleAuthorization@1.0",
        "Identifier": "resaleauthz-123456789"
      },
      "Details": "{
    }"
    }
  ]
}
```

```
}
```

Provide information for the fields to add the `RestrictResaleAuthorization` change type:

- **Entity** (object) (required) – Your Resale Authorization.
  - **Type** (string) (required) – The Type is always `ResaleAuthorization@1.0`.
  - **Identifier** (string) (required) – Your Resale Authorization ID. For more information, see [Identifier](#).
- **Details** (string) (required) – Specifics of the request. It must be an empty object for `RestrictResaleAuthorization`.

## Response Syntax

A change set is created for your request. The response to this request gives you the `ChangeSetId` and `ChangeSetArn` 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 includes validating information to ensure that it meets the AWS Marketplace guidelines. The validation process can take anywhere from a few minutes to a few hours.

You can check the status of the request through the AWS Marketplace Management Portal, or directly through Catalog API using the [DescribeChangeSet](#) API operation.

## Synchronous Validations

The schema validations are specific to `RestrictResaleAuthorization` actions in the AWS Marketplace Catalog API. These validations are performed when you call `StartChangeSet`. If the request doesn't meet the following requirements, it will fail with an HTTP response.

Input field	Validation rule	HTTP code
Details	Must be empty	422

Input field	Validation rule	HTTP code
RestrictResaleAuthorization	Expired ResaleAuthorization can't be marked as Restricted	422
An unknown property	No additional properties are allowed	422

## Asynchronous Errors

The following errors are specific to RestrictResaleAuthorization actions in the AWS Marketplace Catalog API. These errors are returned when you call DescribeChangeSet after a change set is processing, or more information about using DescribeChangeSet to get the status of a change request, see [Working with change sets](#).

Error code	Error message
INCOMPATIBLE_STATUS	Expired ResaleAuthorization can't be marked as restricted.

## Release a Resale Authorization and make it visible to a Channel Partner

You can use the Catalog API to initiate your ResaleAuthorization to an Active state.

ReleaseResaleAuthorization makes your Resale Authorization active so that a Channel Partner can use your Resale Authorization to create private offers.

To release your Resale Authorization, call the StartChangeSet API operation with the ReleaseResaleAuthorization change type, as shown in the following example.

### Request Syntax

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

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

```
{
  "ChangeType": "ReleaseResaleAuthorization",
  "Entity": {
    "Type": "ResaleAuthorization@1.0",
    "Identifier": "resaleauthz-123456789"
  },
  "DetailsDocument": {}
}
```

Provide information for the fields to add the `ReleaseResaleAuthorization` change type:

- **Entity** (object) (required) – Your Resale Authorization.
- **Type** (string) (required) – The Type is always `ResaleAuthorization@1.0`.
- **Identifier** (string) (required) – Your Resale Authorization ID. For more information, see [Identifier](#).
- **DetailsDocument** (object) (required) – Specifics of the request. It must be empty for `ReleaseResaleAuthorization`.

## Response Syntax

A change set is created for your request. The response to this request gives you the `ChangeSetId` and `ChangeSetArn` 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 includes validating information to ensure that it meets the AWS Marketplace guidelines. The validation process can take anywhere from a few minutes to a few hours.

You can check the status of the request through the AWS Marketplace Management Portal, or directly through Catalog API using the [DescribeChangeSet](#) API operation.

## Synchronous Validations

The schema validations are specific to `ReleaseResaleAuthorization` actions in the AWS Marketplace Catalog API. The validations are performed when you call `StartChangeSet`. If the request doesn't meet the following requirements, it will fail with an HTTP response.

Input field	Validation rule	HTTP code
An unknown property	No additional properties are allowed	422

## Asynchronous Errors

The following errors are specific to `ReleaseResaleAuthorization` 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](#).

Error code	Error message
MISSING_MANDATORY_TERMS	Provide a <code>BuyerLegalTerm</code> .
MISSING_MANDATORY_TERMS	Provide a <code>PricingTerm</code> .
INCOMPATIBLE_PRODUCT	Use an active product in limited or public state.
INCOMPATIBLE_PRICING_TERM	<code>PaymentScheduleTerm</code> and <code>FixedUpfrontPricingTerm</code> must be present together.
INCOMPATIBLE_BUYER_TARGETING	At least one Buyer account must be present for <code>ResaleAuthorization</code> with <code>PreExistingBuyerAgreement</code> .
MISSING_MANDATORY_TERMS	Provide at least one of [x,y,z].
INCOMPATIBLE_STATUS	[x] request can't be performed after the resale authorization is released.

## Describe an existing Resale Authorization

To describe Resale Authorization details, call the DescribeEntity API operation with the ResaleAuthorization@1.0 entity type, as shown in the following example.

### Request Syntax

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

Provide information for the fields to add the DescribeEntity change type:

- **catalog** (string) – The catalog related to the request. Fixed value: AWSMarketplace.
- **entityId** (string) – The unique ID of the ResaleAuthorization to describe.

### Response Syntax

The response to this request gives you the offer details and looks like the following.

```
{
  "EntityType": "ResaleAuthorization@1.0",
  "EntityIdentifier": "resaleauthz-123456789",
  "EntityArn": "arn:aws:aws-marketplace:us-east-1:111122223333:AWSMarketplace/ResaleAuthorization/resaleauthz-123456789",
  "LastModifiedDate": "2021-03-10T21:57:16Z",
  "Details": {
    "Name": "TestResaleAuthorization",
    "Description": "ResaleAuthorization for Test Product",
    "ProductId": "prod-ad8EXAMPLE51",
    "ProductName": "TestProduct",
    "Status": "Active", /*Draft, Active, Restricted*/
    "PreExistingBuyerAgreement": {
      "AcquisitionChannel": "Unknown",
      "PricingModel": "Unknown"
    },
    "CreatedDate": "2023-07-18T16:39:31.335Z",
    "ManufacturerLegalName": "ChannelCAPI.Inc",
    "ManufacturerAccountId": "123456789012",
    "Dimensions": [
      {
        "Name": "Protected Resources",
        "Description": "Additional 100 protected resources",
        "Key": "hundredresources",
```

```

        "Unit": "Units",
        "Types": [
            "Entitled"
        ]
    },
],
"OfferDetails": {
    "OfferExtendedStatus": "Not Started", /* Not Started, Completed-Used, Completed-Usable*/
    "OfferCreatedCount": 0
},
"Terms": [
    {
        "Type": "ResaleUsageBasedPricingTerm",
        "Id": "term_id_placeholder",
        "CurrencyCode": "USD",
        "RateCards": [
            {
                "RateCard": [
                    {
                        "DimensionKey": "resource_number",
                        "Price": "0.05"
                    },
                    {
                        "DimensionKey": "scanned_data",
                        "Price": "0.05"
                    }
                ]
            }
        ]
    },
    {
        "Type": "ResaleConfigurableUpfrontPricingTerm",
        "Id": "term_id_placeholder",
        "CurrencyCode": "USD",
        "RateCards": [
            {
                "Selector": {
                    "Type": "Duration",
                    "Value": "P24M"
                },
                "RateCard": [
                    {
                        "DimensionKey": "hundredresources",

```

```

        "Price": "0.04"
      },
      {
        "DimensionKey": "tenTBData",
        "Price": "0.03"
      },
      {
        "DimensionKey": "channel_custom",
        "Price": "0.02"
      }
    ],
    "Constraints": {
      "MultipleDimensionSelection": "Allowed",
      "QuantityConfiguration": "Allowed"
    }
  }
]
},
{
  "Type": "ResaleFixedUpfrontPricingTerm",
  "Id": "term-sdh27fb2",
  "CurrencyCode": "USD",
  "Duration": "P180D",
  "Price": "0.0",
  "Grants": [
    {
      "DimensionKey": "sdf73rbns93n1120d10xm1",
      "MaxQuantity": 1
    }
  ]
},
{
  "Type": "ResalePaymentScheduleTerm",
  "Id": "term-sdh27fb2",
  "CurrencyCode": "USD",
  "Schedule": [
    {
      "ChargeDate": "2018-07-01T00:00:00.000Z",
      "ChargeAmount": "200.00"
    },
    {
      "ChargeDate": "2019-05-01T00:00:00.000Z",
      "ChargeAmount": "200.00"
    }
  ]
}

```

```

    ]
  },
  {
    "Type": "BuyerLegalTerm",
    "Id": "term_id_placeholder",
    "Documents": [
      {
        "Type": "StandardEula",
        "Url": "https://resale-auth-legal-terms-iad-beta.s3.us-east-1.amazonaws.com/09ae57d6-c75a-3a4c-aadf-9b866bae64ab/a85cace8-6d9d-40ca-a053-78fc265479bf?isSigned=yes"
      }
    ]
  },
  {
    "Type": "ResaleLegalTerm",
    "Id": "term_id_placeholder",
    "Documents": [
      {
        "Type": "StandardResellerContract",
        "Url": "https://resale-auth-legal-terms-iad-beta.s3.us-east-1.amazonaws.com/09ae57d6-c75a-3a4c-aadf-9b866bae64ab/bed55b56-7ab4-4c4c-b633-3bf4f6efcb98?isSigned=yes"
      }
    ]
  },
  {
    "Type": "BuyerValidityTerm",
    "Id": "term_id_placeholder",
    "MaximumAgreementStartDate": "2023-09-25T23:59:59.000Z"
  },
  {
    "Type": "BuyerTargetingTerm",
    "Id": "term_id_placeholder",
    "PositiveTargeting": {
      "BuyerAccounts": [
        {
          "AwsAccountId": "444455556666"
        }
      ]
    }
  }
],
"Rules": [

```

```

    {
      "Type": "AvailabilityRule",
      "Id": "availability_rule_id_placeholder",
      /* If the AvailabilityEndDate and OffersMaxQuantity not present Usage will be
Unlimited*/
      "Usage": "Limited",
      "AvailabilityEndDate": "2022-05-31T23:59:59Z",
      "OffersMaxQuantity": 1
    },
    {
      "Type": "PartnerTargetingRule",
      "Id": "partner_targeting_rule_id_placeholder",
      "ResellerAccountId": "777777777777",
      "ResellerLegalName": "ChannelCAPICP.Inc"
    }
  ]
}

```

The following is information about the fields you see in the DescribeEntity response.

- **EntityType** (string) – The named type of the entity, which is ResaleAuthorization@1.0.
- **EntityIdentifier** (string) – The identifier of the entity, in the format of EntityId@RevisionId.
- **EntityArn** (string) – The ARN associated to the unique identifier for the change set referenced in this request.
- **LastModifiedDate** (string) – The last modified date of the entity, in ISO 8601 format (2018-02-27T13:45:22Z).
- **Details** (string) – This JSON string includes the details of the entity.
  - **Name** (string) – Name associated with the ResaleAuthorization for better readability to you and your Channel Partners. It's displayed as part of the Agreement information.
  - **Description** (string) – Description is a free-form text which is meant to be used only by you and will never be exposed to buyers.
  - **ProductId** (string) – Description is a free-form text which is meant to be used only by you and will never be exposed to buyers.
  - **AgreementToken** (string) – Generated from content in ResaleAuthorization. It contains information about terms, rules, and proposer while creating an agreement. It's used for authorization checks and validations during procurement.
  - **Terms** (array of structures) – List of terms presented for acceptance.

- **Rules** (array of structures) – List of rules or set of instructions.

## Working with channel partner private offers

You can use the AWS Marketplace Catalog API to automate tasks for working with channel partner private offers (CPPOs).

When you create or update a CPPO, the draft offer will contain the terms and rules from a Resale Authorization and will be invisible to the buyer. It's possible, but not required, to involve multiple personas in your organization to create a private offer.

For example, one persona can be responsible for updating prices while a second persona can be responsible for updating the payment schedule. Then, a third persona can be responsible for updating legal terms. You can give a persona permission to update certain parts of an offer. However, you can give only read permissions to Resale Authorizations.

As a prerequisite for calling change types, you must have received one or more Resale Authorizations and be familiar working with AWS Marketplace Catalog API.

For more information, see [Channel partner private offers](#) in the *AWS Marketplace Seller Guide*.

The following topics describe how to use the Catalog API to create and update CPPOs:

### Topics

- [CPPO prerequisites](#)
- [Create a CPPO](#)
- [Create a channel partner private replacement offer](#)
- [Update markup](#)
- [Update targeting configuration](#)
- [Update legal resources](#)
- [Update the discoverability of the CPPO](#)
- [Define the expiration date of agreements](#)
- [Update pricing](#)
- [Update payment schedule details](#)
- [Publish the CPPO](#)
- [Define an existing CPPO](#)

## CPPO prerequisites

Resale Authorization manual service-linked role (SLR) setup is a mandatory pre-requisite to use resale authorization to create CPPO. Independent Software Vendors (ISVs) and Channel Partners (CPs) must create a custom AWS Identity and Access Management (IAM) role that provides permissions to AWS. This setup is required in order for the Resale Authorization Catalog API to work. For more information, see [the section called "Creating a custom IAM role"](#). ISVs and CPs must both perform the SLR setup in order for the Resale Authorization change types in the AWS Marketplace Catalog API to work. If both groups don't perform the setup, then we can't "resource share" the authorization from the ISV to the CP.

## Create a CPPO

You use a Resale Authorization targeted to you to create a channel partner private offer (CPPO) in Draft state in AWS Marketplace.

If your request is processed successfully, AWS Marketplace Catalog API generates an offer in Draft state for you with Resale Authorization terms. You can use `DescribeEntity` to see the terms applied to the draft offer from Resale Authorization. This is an incomplete offer and not visible to buyers in AWS Marketplace. You then use change types associated with the CPPO to complete the offer.

After the offer is completed, you use the `ReleaseOffer` change type to complete the offer creation process and release the offer. This will validate the entire offer and make your offer visible to buyers in AWS Marketplace.

To create a channel partner private offer, call the `StartChangeSet` API operation with the `CreateOfferUsingResaleAuthorization` change type, as shown in the following example.

### Request Syntax

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

{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "CreateOfferUsingResaleAuthorization",
      "Entity": {
        "Type": "Offer@1.0"
      }
    }
  ]
}
```

```
    },
    "DetailsDocument": {
      "ResaleAuthorizationId": "resaleauthz-123456789",
      "Name": "Test Offer"
    }
  }
]
```

Provide information for the fields to add the `CreateOfferUsingResaleAuthorization` change type:

- **Entity** (object) (required) – Your CPPO.
  - **Type** (string) (required) – The Type is always `Offer@1.0`.
- **DetailsDocument** (object) (required) – The JSON value of specifics of the request.
  - **ResaleAuthorizationId** (string) (required) – The unique identifier that includes product, terms, and rules that are being offered. Channel partners can add additional terms and rules using update change types. ResaleAuthorization must be available and targeted to you as a partner.
  - **Name** (string) (optional) – The name associated with the offer for better readability. It is displayed as a part of the agreement information.

## Response Syntax

A change set is created for your request. The response to this request gives you the `ChangeSetId` and `ChangeSetArn` 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.

You can check the status of the request through the AWS Marketplace Management Portal, or directly through Catalog API using the [DescribeChangeSet](#) API operation.

The response to this request gives you the status of the request. If the status is `SUCCEEDED`, then a new `OfferId` is generated.

```
{
  "ChangeSetId": "example123456789012abcdef",
  "ChangeSetArn": "arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/
ChangeSet/example123456789012abcdef",
  "ChangeSetName": "Submitted by 123456789012",
  "StartTime": "2021-05-27T22:21:26Z",
  "EndTime": "2021-05-27T22:32:19Z",
  "Status": "SUCCEEDED",
  "ChangeSet": [
    {
      "ChangeType": "CreateOfferUsingResaleAuthorization",
      "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "offer-123456789"
      },
      "DetailsDocument": {
        "ResaleAuthorizationId": "resaleauthz-123456789",
        "Name": "Test Offer"
      },
      "ErrorDetailList": []
    }
  ]
}
```

You can use the GET DescribeEntity request to describe the draft offer rules and terms created from ResaleAuthorization in the AWS Marketplace Catalog API Reference. For more information, see [DescribeChangeSet](#).

### Terms and rules from ResaleAuthorization

- **LegalTerms** – Provisions describing legal terms, such as the EULA in the ResaleAuthorization will be added to the draft offer. You can add legal terms using the UpdateLegalTerms change type. For more information, see [UpdateLegalTerms](#).
- **PricingTerms** – All the pricing terms (ConfigurableUpfrontPricingTerm, FixedUpfrontPricingTerm, UsageBasedPricingTerm, PaymentScheduleTerms) described by the Manufacturer in the ResaleAuthorization will be added to the draft offer. You can choose to increase the pricing (for each dimension) for your targeted buyers using the UpdateMarkup change type. For more information, see [UpdateMarkup](#) in this guide.
- **PaymentTerms** – If the manufacturer has defined the Future Payment Schedule in the ResaleAuthorization, then you will be able to see the payment terms in the draft offer.

You can choose to increase the payment schedule amount for your targeted buyers using the UpdateMarkup change type. If you want to set the payment schedule for your buyers, you can use UpdatePaymentScheduleTerms. For more information, see [UpdatePaymentScheduleTerms](#).

- **TargetingRule** – If the ResaleAuthorization is targeted to specific buyers, then channel partners can give private offers to a subset of buyers using PositiveTargeting. By default, the rule will include all the buyers from ResaleAuthorization. You can select specific buyers and update the draft offer using the UpdateTargeting change type. For more information, see [UpdateTargeting](#).

```
{
  "EntityType": "Offer@1.0",
  "EntityIdentifier": "offer-a5EXAMPLEEwzpu@1",
  "EntityArn": "arn:aws:aws-marketplace:us-east-1:444555666777:AWSMarketplace/Offer/offer-a5oEXAMPLEEzpu",
  "LastModifiedDate": "2021-03-10T21:57:16Z",
  "DetailsDocument": {
    "Id": "offer-3rb23tu92rn",
    "Name": "Test Offer",
    "Description": "Worldwide private offer for Test Product",
    "ProductId": "prod-0bc848d78b51",
    "ResaleAuthorizationId": "resaleauthz-123456789",
    "Terms": [
      {
        "Type": "LegalTerm",
        "Documents": [
          {
            "Type": "CustomEula",
            "Url": "https://s3.amazonaws.com/EULA/custom-eula-1234.txt"
          }
        ]
      }
    ],
  },
  {
    "Type": "ConfigurableUpfrontPricingTerm",
    "CurrencyCode": "USD",
    "RateCards": [
      {
        "Selector": {
          "Type": "Duration",
          "Value": "P12M"
        }
      }
    ],
  },
}
```

```
    "RateCard": [
      {
        "DimensionKey": "m3.large",
        "Price": "300.00"
      },
      {
        "DimensionKey": "m4.xlarge",
        "Price": "400.00"
      }
    ],
    "Constraints": {
      "MultipleDimensionSelection": "Allowed",
      "QuantityConfiguration": "Allowed"
    }
  }
],
{
  "Type": "UsageBasedPricingTerm",
  "CurrencyCode": "USD",
  "RateCards": [
    {
      "RateCard": [
        {
          "DimensionKey": "m3.large",
          "Price": "0.10"
        },
        {
          "DimensionKey": "m4.xlarge",
          "Price": "0.20"
        }
      ]
    }
  ]
},
{
  "Type": "PaymentScheduleTerm",
  "CurrencyCode": "USD",
  "Schedule": [
    {
      "ChargeDate": "2020-12-01T00:00:00.000Z",
      "ChargeAmount": "1000.00"
    },
    {

```

```

        "ChargeDate": "2021-06-15T00:00:00.000Z",
        "ChargeAmount": "1250.00"
    }
  ]
}
],
"Rules": [
  {
    "Type": "TargetingRule",
    "PositiveTargeting": {
      "BuyerAccounts": [
        "111222333444"
      ]
    }
  }
]
}
}
}

```

## Synchronous Validations

The following schema validations are specific to `CreateOfferUsingResaleAuthorization` actions in the AWS Marketplace Catalog API. These validations are performed when you call `StartChangeSet`. If the request doesn't meet the following requirements, it will fail with an HTTP response.

Input field	Validation rule	HTTP code
Name	Required  Length must be between 1 and 150 characters	422
Description	Required  Length must be between 1 and 255 characters	422
ResaleAuthorizationId	Required	422

Input field	Validation rule	HTTP code
	Length must be between 1 and 50 characters	
ResaleAuthorizationId	ResaleAuthorization must be targeted to the channel partner.	422
ResaleAuthorizationId	ResaleAuthorization must be active	422
Channel Partner	Channel Partner must be paid seller in AWS Marketplace	422

## Asynchronous Errors

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

Error code	Error message
INCOMPATIBLE_RESALE_AUTHORIZATION	Use a ResaleAuthorization in active state.

## Create a channel partner private replacement offer

You can use the Catalog API to create a channel partner private replacement offer in AWS Marketplace.

You use the `ResaleAuthorization` targeted to you and an Agreement of which you are the proposer to create a channel partner private replacement offer in Draft state in the Catalog API by calling `StartChangeSet` with the `CreateReplacementOfferUsingResaleAuthorization` change type, as shown in the following example. Replacement offers can be used to replace an agreement from a previous offer before it ends.

`CreateReplacementOfferUsingResaleAuthorization` will create a draft offer with the agreement acceptor in targeting. This targeting cannot be changed afterwards. The draft offer will also contain the source offer id of the agreement.

To create a channel partner private replacement offer, call the `StartChangeSet` API operation with the `CreateReplacementOfferUsingResaleAuthorization` change type, as shown in the following example.

## Request Syntax

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

{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "CreateReplacementOfferUsingResaleAuthorization",
      "Entity": {
        "Type": "Offer@1.0"
      },
      "DetailsDocument": {
        "ResaleAuthorizationId": "2bd2c761-3b7f-3771-a9a7-e8ad36517698",
        "Name": "CAPI-saas-abo-contract-fps",
        "AgreementId": "agmt-f2ooEXAMLEamt7mjj0j59gu"
      }
    }
  ]
}
```

Provide information for the fields to add the `CreateReplacementOfferUsingResaleAuthorization` change type:

- **Entity** (object) (required) – Your CPPO.
  - **Type** (string) (required) – The Type is always `Offer@1.0`.
- **DetailsDocument** (object) (required) – The JSON value of specifics of the request.
  - **ResaleAuthorizationId** (string) (required) – `ResaleAuthorizationId` is the unique identifier which includes product, terms and rules are being offered. Channel partners can add additional terms and rules using `Update` change types.

`ResaleAuthorization` must be available and targeted to you as a partner.

- **Name** (string) (optional) – Name associated with the offer for better readability to you and your customers. It will be displayed as part of Agreement information as well.

- **AgreementId** (string) (required) – AgreementId is the unique identifier of the agreement created when the targeted buyer accepted the previous offer you are trying to replace

## Response Syntax

A change set is created for your request. The response to this request gives you the ChangeSetId and ChangeSetArn 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.

You can check the status of the request through the AWS Marketplace Management Portal, or directly through Catalog API using the [DescribeChangeSet](#) API operation.

## Synchronous Validations

The following schema validations are specific to CreateReplacementOfferUsingResaleAuthorization actions in the AWS Marketplace Catalog API. These validations are performed when you call StartChangeSet. If the request doesn't meet the following requirements, it will fail with an HTTP response.

Input field	Validation rule
Name	Length must be between 1 and 150 characters
AgreementId	Provided agreement must be active Provided agreement must exist  Provided agreement must be owned by Channel Partner
ResaleAuthorizationId	Required  Length must be between 1 and 50 characters

Input field	Validation rule
ResaleAuthorizationId	ResaleAuthorization must be targeted to the channel partner.
ResaleAuthorizationId	ResaleAuthorization must be active
Channel Partner	Channel Partner must be paid seller in AWS Marketplace

## Asynchronous Errors

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

Error code	Error message
INCOMPATIBLE_RESALE_AUTHORIZATION	Use a ResaleAuthorization in active state.
INCOMPATIBLE_RESALE_AUTHORIZATION	The ResaleAuthorization must be for the same product that is associated with the agreement.
INCOMPATIBLE_RESALE_AUTHORIZATION	Use a ResaleAuthorization targeted to the acceptor of the agreement.

## Update markup

You can use the Catalog API to update pricing terms by a percentage value in your offer in AWS Marketplace.

This will apply the given percentage markup on all pricing terms and payment terms (for future payment schedules) that are defined by the manufacturer in the ResaleAuthorization. Any existing markup will be overwritten. You can view updated pricing and payment terms using `DescribeEntity`.

To update markup, call the `StartChangeSet` API operation with the `UpdateMarkup` change type, as shown in the following example.

## Request Syntax

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

{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "UpdateMarkup",
      "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "offer-123456789"
      },
      "DetailsDocument": {
        "Percentage": "5.0"
      }
    }
  ]
}
```

Provide information for the fields to add the `UpdateMarkup` change type:

- **Entity** (object) (required) – Your CPPO.
  - **Type** (string) (required) – The Type is always `Offer@1.0`.
  - **Identifier** (string) (required) – Your offer ID. For more information, see [Identifier](#).
- **DetailsDocument** (object) (required) – The JSON value of specifics of the request.
  - **Percentage** (string) (required) – Percentage value will be added to the manufacturer pricing or payment terms.

## Response Syntax

A change set is created for your request. The response to this request gives you the `ChangeSetId` and `ChangeSetArn` 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 includes validating information with the AWS Marketplace Seller Operations team to ensure it meets the AWS Marketplace guidelines. The validation process can take anywhere from a few minutes to a few hours.

You can check the status of the request through the AWS Marketplace Management Portal, or directly through Catalog API using the [DescribeChangeSet](#) API operation.

## Synchronous Validations

The following schema validations are specific to UpdateMarkup actions in the AWS Marketplace Catalog API. These validations are performed when you call StartChangeSet. If the request doesn't meet the following requirements, it will fail with an HTTP response.

Input field	Validation rule
Percentage	Required
	Data type is "String"
	Must be non-negative
	Allow up to 9 decimals

## Asynchronous Errors

The following errors are specific to UpdateMarkup actions in the AWS Marketplace Catalog API. These errors are returned when you call DescribeChangeSet after a change set is processing. For more information about using DescribeChangeSet to get the status of a change request, see [Working with change sets](#).

Error code	Error message
INCOMPATIBLE_MARKUP	The requested change can't be performed after the offer is released.

Error code	Error message
INCOMPATIBLE_RESALE_AUTHORIZATION	MarkupPercentage can't be updated when PaymentScheduleTerm or FixedUpfrontPricingTerm are present in offer and not present in ResaleAuthorization .
INVALID_MARKUP_PERCENTAGE	UpdateMarkup can only be invoked for offers created using ResaleAuthorization.

## Update targeting configuration

You can use the Catalog API to replace the existing targeting configuration completely in AWS Marketplace.

Any existing targeting options that are not included in the latest request will be removed from the offer. Manufacturers can mention specific targeted buyers in ResaleAuthorization. Channel partners can give private offers to a subset of buyers using PositiveTargeting in the UpdateTargeting change type.

To update targeting configurations of your offer, call the StartChangeSet API operation with the UpdateTargeting change type, as shown in the following example.

### Request Syntax

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

{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "UpdateTargeting",
      "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "offer-123456789"
      },
      "DetailsDocument": {
        "PositiveTargeting": {
          "CountryCodes": [
```

```

        "US",
        "CA"
    ],
    "BuyerAccounts": [
        "111222333444"
    ]
},
"NegativeTargeting": {
    "CountryCodes": [
        "XX"
    ]
}
}
}
]
}

```

Provide information for the fields to add the UpdateTargeting change type:

- **Entity** (object) (required) – Your CPPO.
  - **Type** (string) (required) – The Type is always Offer@1.0.
  - **Identifier** (string) (required) – Your offer ID. For more information, see [Identifier](#).
- **DetailsDocument** (object) (required) – The JSON value of specifics of the request.
  - **PositiveTargeting** (object) (optional) – Positive targeting defines the criteria which any buyer's profile should fulfill in order to be allowed to access the offer. This field is optional, but at least one targeting option should be provided when this field is present.
    - **CountryCodes** (array of strings) (optional) – List as option for allowing targeting based on country. If the intention isn't to target the offer to a country, this field should be omitted. If it's present, the list must contain at least one country code. Each element in this list should be a valid 2-letter country code, using this format: ISO 3166-1 alpha-2.
    - **BuyerAccounts** (array of strings) (optional) – List as an option to allow targeting based on AWS accounts (also known as, Private Offer). If the intention is to not target the offer to an AWS account, this field should be omitted.
  - **NegativeTargeting** (object) (optional) – Negative targeting defines the criteria which any customer's profile should fulfill to be restricted to access the offer. Although this field is optional, at least one targeting option should be provided when this field is present.
    - **CountryCodes** (array of strings) (required) – List as option for allowing targeting based on country. If the intention isn't to target the offer to a specific country, then this field should

be omitted. If it's present, the list must contain at least one country code. Each element in this list should be a valid 2-letter country code using this format: ISO 3166-1 alpha-2.

## Response Syntax

A change set is created for your request. The response to this request gives you the `ChangeSetId` and `ChangeSetArn` 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 includes validating information to ensure that it meets the AWS Marketplace guidelines. The validation process can take anywhere from a few minutes to a few hours.

You can check the status of the request through the AWS Marketplace Management Portal, or directly through Catalog API using the [DescribeChangeSet](#) API operation.

## Synchronous Validations

The following schema validations are specific to `UpdateTargeting` actions in the AWS Marketplace Catalog API. These validations are performed when you call `StartChangeSet`. If the request doesn't meet the following requirements, it will fail with an HTTP response.

Input field	Validation rule	HTTP code
Details	Required	422
PositiveTargeting	Optional	422
NegativeTargeting	Optional	422
PositiveTargeting.CountryCodes	Optional Country codes must be valid (ISO 3166-1 alpha-2)	422
PositiveTargeting.BuyerAccounts	Optional	422

Input field	Validation rule	HTTP code
	AWS account IDs must be in valid format (12-digit number)  Must not contain more than 25 accounts	
NegativeTargeting.CountryCodes	Optional  Country codes must be valid (ISO 3166-1 alpha-2)	422
NegativeTargeting.BuyerAccounts	Must not be provided (negative targeting on BuyerAccounts isn't supported)	422

## Asynchronous Errors

The following errors are specific to UpdateTargeting actions in the AWS Marketplace Catalog API. These errors are returned when you call DescribeChangeSet after a change set is processing. For more information about using DescribeChangeSet to get the status of a change request, see [Working with change sets](#).

Error code	Error message
INVALID_BUYER_ACCOUNTS	Provide valid buyer accounts. Invalid accounts: [x].
INVALID_COUNTRY_CODES	Provide supported country codes.
INVALID_TARGETING	Use either negative or positive targeting on the same attribute.
INCOMPATIBLE_PRODUCT	Country-based targeting isn't supported for the product.
INCOMPATIBLE_RESALE_AUTHORIZATION	Provide BuyerAccounts that are compatible with the ResaleAuthorization.

Error code	Error message
INCOMPATIBLE_TARGETING	The requested change can't be performed after the offer is released.
INCOMPATIBLE_TARGETING	The requested change can't be performed after the offer is expired.
INCOMPATIBLE_TARGETING	Targeting can't be updated on a replacement offer. If the buyer isn't associated with the provided AgreementId, then create a new private offer by providing an AgreementId associated with the buyer.
TOO_MANY_BUYER_ACCOUNTS	Provide BuyerAccounts within the allowed limits.

## Update legal resources

You can use the Catalog API to merge the Resale Authorization legal terms and replace the existing legal terms completely in AWS Marketplace.

This change doesn't affect existing agreements. The legal terms that aren't included in the latest request will be removed from the offer. You can view the merged legal terms by calling `DescribeEntity`.

To update legal terms of your offer, call the `StartChangeSet` API operation with the `UpdateLegalTerms` change type, as shown in the following example.

### Request Syntax

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

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



- **StandardEula** – Standard Contract For AWS Marketplace (SCMP). For more information about SCMP, see the AWS Marketplace Seller Guide. You don't provide a URL for this type because it is managed by AWS Marketplace.
- **Url** (string) (conditionally required) – A URL to the legal document for buyers to read. Required when Type is one of the following [CustomEula].
- **Version** (string) (conditionally required) – A version of standard contracts provided by AWS Marketplace. This is required when Type is StandardEula. Available versions are:
  - **2022-07-14** – This version of the Standard Contract for AWS Marketplace is available from this Amazon S3 bucket: <https://s3.amazonaws.com/aws-mp-standard-contracts/Standard-Contract-for-AWS-Marketplace-2022-07-14.pdf>

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

## Response Syntax

```
{
  "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 includes validating information with the AWS Marketplace Seller Operations team to ensure it meets the AWS Marketplace guidelines. The validation process can take anywhere from a few minutes to a few hours. You can check the status of the request through the AWS Marketplace Management Portal, or in the Catalog API with the DescribeChangeSet action.

## Synchronous Validations

The following schema validations are specific to UpdateLegalTerms actions in the AWS Marketplace Catalog API. These validations are performed when you call StartChangeSet. If the request doesn't meet the following requirements, it will fail with an HTTP response.

Input field	Validation rule	HTTP code
Terms	Required	422

Input field	Validation rule	HTTP code
	Only LegalTerm is allowed in the list	
	List size must be 1	
Terms[].LegalTerm.Documents	Required	422
Terms[].LegalTerm.Documents[].Type	Required  Allowed values: <ul style="list-style-type: none"> <li>• CustomEula</li> <li>• StandardEula</li> </ul>	422
Terms[].LegalTerm.Documents[].Url	Required and must be a valid URL when "Type" is one of <ul style="list-style-type: none"> <li>• CustomEula</li> </ul>	422

## Asynchronous Errors

The following errors are specific to UpdateLegalTerms actions in the AWS Marketplace Catalog API. These errors are returned when you call DescribeChangeSet after a change set is processing. For more information about using DescribeChangeSet to get the status of a change request, see [Working with change sets](#).

Error code	Error message
INCOMPATIBLE_TERMS	The requested change can't be performed after the offer is released.
INCOMPATIBLE_TERMS	The requested change can't be performed after the offer is expired.
INVALID_LEGAL_DOCUMENTS	Provide URLs for legal documents stored in accessible S3 buckets.

Error code	Error message
INVALID_LEGAL_DOCUMENTS	Only the most recent version of StandardEula is supported for new offers.
INVALID_LEGAL_DOCUMENTS	Provide legal documents in the supported file formats.
INVALID_LEGAL_DOCUMENTS	Provide legal documents using the supported document types.
LIMIT_EXCEEDED_LEGAL_DOCUMENT_SIZE	Provide legal documents within the allowed size limits.

## Update the discoverability of the CPPO

You can use the Catalog API to manage the discoverability of your offer in AWS Marketplace. This change type doesn't affect existing agreements.

You can either choose to set a specific date in the future to restrict the discoverability of your offer or in the past to expire your offer.

To manage the discoverability of your offer, call the StartChangeSet API operation with the UpdateAvailability change type, as shown in the following example.

### Request Syntax

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

{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "UpdateAvailability",
      "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "offer-123456789"
      },
      "DetailsDocument": {
        "AvailabilityEndDate": "2024-05-31"
      }
    }
  ]
}
```

```
    }
  }
]
}
```

Provide information for the fields to add the `UpdateAvailability` change type:

- **Entity** (object) (required) – Your CPPO.
  - **Type** (string) (required) – The Type is always `Offer@1.0`.
  - **Identifier** (string) (required) – Your offer ID. For more information, see [Identifier](#).
- **DetailsDocument** (object) (required) – The JSON value of specifics of the request.
  - **AvailabilityEndDate** (string) – Date until when the offer is discoverable and purchasable in AWS Marketplace. You can choose to set a specific date in the future to restrict the availability or in the past to expire the offer. Dates are represented in YYYY-MM-DD format. Offer expires at 23:59:59.999 UTC on the date provided.

## Response Syntax

A change set is created for your request. The response to this request gives you the `ChangeSetId` and `ChangeSetArn` 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 includes validating information to ensure that it meets the AWS Marketplace guidelines. The validation process can take anywhere from a few minutes to a few hours.

You can check the status of the request through the AWS Marketplace Management Portal, or directly through Catalog API using the [DescribeChangeSet](#) API operation.

## Synchronous Validations

The following schema validations are specific to `UpdateAvailability` actions in the AWS Marketplace Catalog API. These validations are performed when you call `StartChangeSet`. If the request doesn't meet the following requirements, it will fail with an HTTP response.

Input field	Validation rule	HTTP code
AvailabilityEndDate	Required  Format: "YYYY-MM-DD"	422

## Asynchronous Errors

The following errors are specific to UpdateAvailability actions in the AWS Marketplace Catalog API. These errors are returned when you call DescribeChangeSet after a change set is processing. For more information about using DescribeChangeSet to get the status of a change request, see [Working with change sets](#).

Error code	Error message
INVALID_AVAILABILITY_END_DATE	AvailabilityEndDate isn't supported for public offers.
INVALID_AVAILABILITY_END_DATE	Provide a future AvailabilityEndDate.
INVALID_AVAILABILITY_END_DATE	Provide an AvailabilityEndDate that is before AgreementEndDate.
MISSING_AVAILABILITY_END_DATE	Provide an AvailabilityEndDate that is before the agreement's end date.

## Define the expiration date of agreements

You can use the Catalog API to define the expiration date of the agreements that are created using this offer in AWS Marketplace.

This change does not affect existing agreements. The manufacturer could mention maximum agreement start date in a Resale Authorization. However, channel partners can't provide an agreement start date later than that date.

To define the expiration date of agreements, call the StartChangeSet API operation with the UpdateValidityTerms change type, as shown in the following example.

## Request Syntax

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

{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "UpdateValidityTerms",
      "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "offer-123456789"
      },
      "DetailsDocument": {
        "Terms": [
          {
            "Type": "ValidityTerm",
            "AgreementDuration": "P12M",
            "AgreementStartDate": "2021-08-01",
            "AgreementEndDate": "2022-08-01"
          }
        ]
      }
    }
  ]
}
```

Provide information for the fields to add the UpdateValidityTerms change type:

- **Entity** (object) (required) – Your CPPO.
  - **Type** (string) (required) – The Type is always Offer@1.0.
  - **Identifier** (string) (required) – Your offer ID. For more information, see [Identifier](#).
- **DetailsDocument** (object) (required) – The JSON value of specifics of the request.
  - **Terms** (array of structures) – List of validity terms that you want to update. Supported validity terms are:
    - **ValidityTerm** (object) – Defines the conditions that will keep an agreement, created from this offer, valid.
      - **Type** (string) – Category of the term being updated.

- **AgreementDuration** (string) – Defines the duration that the agreement remains active. If `AgreementStartDate` isn't provided, agreement duration is relative to the agreement signature time. The duration is represented in the ISO\_8601 format.
- **AgreementStartDate** (string) – Defines the date when agreement starts. `AgreementStartDate` is represented in YYYY-MM-DD format. The agreement starts at 00:00:00.000 UTC on the date provided. If `AgreementStartDate` isn't provided, agreement start date is determined based on agreement signature time.
- **AgreementEndDate** (string) – Defines the date when the agreement ends. The `AgreementEndDate` is represented in YYYY-MM-DD format. The agreement ends at 23:59:59.999 UTC on the date provided. If `AgreementEndDate` isn't provided, the agreement end date is determined by the validity of individual terms.

## Response Syntax

A change set is created for your request. The response to this request gives you the `ChangeSetId` and `ChangeSetArn` 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 includes validating information to ensure that it meets the AWS Marketplace guidelines. The validation process can take anywhere from a few minutes to a few hours.

You can check the status of the request through the AWS Marketplace Management Portal, or directly through Catalog API using the [DescribeChangeSet](#) API operation.

## Synchronous Validations

The following schema validations are specific to `UpdateValidityTerms` actions in the AWS Marketplace Catalog API. These validations are performed when you call `StartChangeSet`. If the request doesn't meet the following requirements, it will fail with an HTTP response.

Input field	Validation rule	HTTP code
Terms	<p>Required</p> <p>Only "ValidityTerm" is allowed in the list</p> <p>Must be empty or contain only 1 term</p>	422
Terms[].ValidityTerm	<p>Supported use cases:</p> <ol style="list-style-type: none"> <li>1. ValidityTerm with only Agreement Duration</li> <li>2. ValidityTerm with only Agreement StartDate</li> <li>3. ValidityTerm with only Agreement EndDate</li> <li>4. ValidityTerm with both AgreementStartDate and Agreement EndDate</li> </ol>	422
Terms[].ValidityTerm.AgreementDuration	<p>Optional</p> <p>Represented in ISO_8601 format.</p>	422
Terms[].ValidityTerm.AgreementStartDate	<p>Optional</p> <p>Format: "YYYY-MM-DD"</p>	422
Terms[].ValidityTerm.AgreementEndDate	<p>Optional</p> <p>Format: "YYYY-MM-DD"</p>	422

## Asynchronous Errors

The following errors are specific to UpdateValidityTerms actions in the AWS Marketplace Catalog API. These errors are returned when you call DescribeChangeSet after a change set is processing. For more information about using DescribeChangeSet to get the status of a change request, see [Working with change sets](#).

Error code	Error message
INCOMPATIBLE_AGREEMENT	AgreementStartDate can't be in the future when the current agreement to be replaced isn't future dated.
INCOMPATIBLE_AGREEMENT_END_DATE	AgreementEndDate can't be updated after the offer is released.
INCOMPATIBLE_AGREEMENT_START_DATE	AgreementStartDate can't be updated after the offer is released.
INCOMPATIBLE_PRODUCT	AgreementStartDate in the future isn't supported.
INCOMPATIBLE_RESALE_AUTHORIZATION	Ensure the duration between Agreement StartDate and AgreementEndDate is compatible with the ResaleAuthorization.
INCOMPATIBLE_RESALE_AUTHORIZATION	Ensure AgreementStartDate is compatible with the ResaleAuthorization.
INCOMPATIBLE_RESALE_AUTHORIZATION	Ensure AgreementEndDate is compatible with the ResaleAuthorization.
INCOMPATIBLE_RESALE_AUTHORIZATION	Ensure the duration between Agreement StartDate and AgreementEndDate is compatible with the ResaleAuthorization.
INCOMPATIBLE_RESALE_AUTHORIZATION	Ensure AgreementDuration matches duration specified in the ResaleAuthorization.
INCOMPATIBLE_TERMS	ValidityTerm isn't supported for public offers.

Error code	Error message
INCOMPATIBLE_TERMS	The requested change can't be performed after the offer is expired.
INVALID_AGREEMENT_DURATION	Provide AgreementDuration that is greater than or equal to [x] days.
INVALID_AGREEMENT_END_DATE	Provide a future AgreementEndDate.
INVALID_AGREEMENT_END_DATE	Provide AgreementEndDate that is after or equal to [x].
INVALID_AGREEMENT_START_DATE	Provide an AgreementStartDate that is after AvailabilityEndDate.
INVALID_AGREEMENT_START_DATE	Provide an AgreementStartDate that is before the AgreementEndDate.
INVALID_AGREEMENT_START_DATE	Provide an AgreementStartDate that is within [x] years from today.
INVALID_AGREEMENT_TIME_INTERVAL	ValidityTerm with both AgreementDuration and AgreementEndDate isn't supported.
INVALID_AGREEMENT_TIME_INTERVAL	ValidityTerm with both AgreementStartDate and AgreementDuration isn't supported in an offer for the product.
INVALID_AGREEMENT_TIME_INTERVAL	ValidityTerm with AgreementStartDate isn't supported in an offer for the product.
INVALID_AGREEMENT_TIME_INTERVAL	ValidityTerm with only AgreementStartDate isn't supported.
INVALID_AGREEMENT_TIME_INTERVAL	AgreementEndDate isn't supported unless it's used in combination with a future Agreement StartDate or for replacement offers.

Error code	Error message
INVALID_AGREEMENT_TIME_INTERVAL	Provide AgreementStartDate and AgreementEndDate where the difference is less than or equal to [x] years.
MISSING_AGREEMENT_START_DATE	Ensure AgreementStartDate is present in ValidityTerm when used along with ConfigurableUpfrontPricingTerm.

## Update pricing

You can use the Catalog API to replace the existing pricing terms completely.

The pricing terms that aren't included in the latest request will be removed from the offer. Channel partners can use this change type only to pass FixedUpfrontPricingTerm.

To update pricing terms for your offers, call the StartChangeSet API operation with the UpdatePricingTerms change type, as shown in the following example.

### Request Syntax

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

{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "UpdatePricingTerms",
      "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "offer-123456789"
      },
      "Details": {
        "PricingModel": "Contract",
        "Terms": [
          {
            "Type": "FixedUpfrontPricingTerm",
            "CurrencyCode": "USD",
```

```

    "Price": "200.00",
    "Duration": "P465D",
    "Grants": [
      {
        "DimensionKey": "Users",
        "MaxQuantity": 10
      }
    ]
  }
]
}

```

Provide information for the fields to add the UpdatePricingTerms change type:

- **Entity** (object) (required) – Your CPPO.
  - **Type** (string) (required) – The Type is always Offer@1.0.
  - **Identifier** (string) (required) – Your offer ID. For more information, see [Identifier](#).
- **Details** (string) (required) – Specifics of the request. This field is a JSON string field. It must be formatted properly for a single-line string field, including escaping characters (such as quotation marks) that can't be in a string.
- **PricingModel** (string) (required) – Pricing model for your offer. Possible values for pricing model are:
  - **Contract** – Contract-based pricing model where buyers are either billed in advance for the use of your product, or offered a flexible payment schedule. Buyers can also pay for an additional usage above their contract.
  - **Terms** (array of structures) (required) – List of pricing terms that you want to update. Supported pricing terms are:
    - **FixedUpfrontPricingTerm** (object) – Defines a pre-paid pricing model where the customers are charged a fixed upfront amount.
      - **Type** (string) (required) – Type of the term being updated.
      - **CurrencyCode** (string) (required) – Defines the currency for the prices mentioned in this term. Currently, only USD is supported.
      - **Price** (string) (required) – Fixed amount to be charged to the customer when this term is accepted.

- **Grants** (array of structures) (required) – Entitlements that will be granted to the acceptor of fixed upfront as part of agreement execution.
- **DimensionKey** (string) (required) – Unique dimension key defined in the product document. Dimensions represent categories of capacity in a product and are specified when the product is listed in AWS Marketplace.
- **MaxQuantity** (integer) (optional) – Maximum amount of capacity that the buyer can be entitled to the given dimension of the product. If `MaxQuantity` is not provided, the buyer will be able to use an unlimited amount of the given dimension.
- **Duration** (string) (optional) – Defines the duration that the term remains active. This field supports the ISO 8601 format.

## Response Syntax

A change set is created for your request. The response to this request gives you the `ChangeSetId` and `ChangeSetArn` 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 includes validating information to ensure that it meets the AWS Marketplace guidelines. The validation process can take anywhere from a few minutes to a few hours.

You can check the status of the request through the AWS Marketplace Management Portal, or directly through Catalog API using the [DescribeChangeSet](#) API operation.

## Synchronous Validations

The following schema validations are specific to `UpdatePricingTerms` actions in the AWS Marketplace Catalog API. These validations are performed when you call `StartChangeSet`. If the request doesn't meet the following requirements, it will fail with an HTTP response.

Input field	Validation rule
<code>PricingModel</code>	Required

Input field	Validation rule
	Allowed pricing models:  Contract
Terms	Required  Allowed terms:  FixedUpfrontPricingTerm
Terms[].FixedUpfrontPricingTerm.CurrencyCode	Required  Allowed values: [USD]  Allowed pricing models:  Contract
Terms[].FixedUpfrontPricingTerm.Price	Required  Data type is "String" Must be non-negative  Support up to 6 Decimals  No special character supported
Terms[].FixedUpfrontPricingTerm.Duration	Required  Expected format: ISO 8601 duration
Terms[].FixedUpfrontPricingTerm.Grants[].DimensionKey	Required  Length must be between 1 and 60
Terms[].FixedUpfrontPricingTerm.Grants[].MaxQuantity	Required

## Asynchronous Errors

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

Error code	Error message
DUPLICATE_DIMENSION_KEYS	Provide Grants with a unique list of dimension keys in [x].
DUPLICATE_DIMENSION_KEYS	Provide RateCard with a unique list of dimension keys in [x].
DUPLICATE_SELECTORS	Provide a unique list of Selectors in ConfigurableUpfrontPricingTerm.
DUPLICATE_TERM_TYPES	Provide a unique list of term types.
INCOMPATIBLE_AGREEMENT	The following terms can't be removed from the replacement offer: [x, y, z].
INCOMPATIBLE_AGREEMENT	The following terms can't be added to the replacement offer: [x, y, z].
INCOMPATIBLE_CURRENCY_CODE	CurrencyCode can't be changed after the offer is released.
INCOMPATIBLE_PRODUCT	Usage pricing model isn't supported for the product.
INCOMPATIBLE_PRODUCT	Contract pricing model isn't supported for the product.
INCOMPATIBLE_PRODUCT	Byol pricing model isn't supported for the product.
INCOMPATIBLE_PRODUCT	Free pricing model isn't supported for the product.
INCOMPATIBLE_PRODUCT	[x] isn't supported in an offer for the product.

Error code	Error message
INCOMPATIBLE_PRODUCT	Provided payment and pricing terms are incompatible.
INCOMPATIBLE_PRODUCT	Use existing, available dimensions in the product in [x].
INCOMPATIBLE_PRODUCT	FreeTrialPricingTerm as the offer's only pricing term isn't supported for the product.
INCOMPATIBLE_PRODUCT	The following terms aren't supported for the product: [x,y,z].
INCOMPATIBLE_PRODUCT	Replacement offers are only supported for contract pricing model.
INCOMPATIBLE_PRODUCT	Provide pricing term(s) that are compatible with the product dimensions. Incompatible pricing terms: [x,y,z].
INCOMPATIBLE_RATE_CARD_CONSTRAINTS	Set MultipleDimensionSelection and QuantityConfiguration to Allowed in ConfigurableUpfrontPricingTerm for usage pricing model.
INCOMPATIBLE_RATE_CARD_CONSTRAINTS	Set MultipleDimensionSelection and QuantityConfiguration to Disallowed in ConfigurableUpfrontPricingTerm for usage pricing model.
INCOMPATIBLE_RATE_CARD_CONSTRAINTS	QuantityConfiguration in ConfigurableUpfrontPricingTerm can't be changed after the offer is released.
INCOMPATIBLE_RATE_CARD_CONSTRAINTS	MultipleDimensionSelection in ConfigurableUpfrontPricingTerm can't be changed after the offer is released.

Error code	Error message
INCOMPATIBLE_RATES	Set all charge amounts and prices to zero (0) when using Free pricing model.
INCOMPATIBLE_RATES	Only zero (0) prices are allowed in UsageBasedPricingTerm for a free trial offer for the product.
INCOMPATIBLE_RESALE_AUTHORIZATION	Provide the same CurrencyCode that is specified in the ResaleAuthorization.
INCOMPATIBLE_RESALE_AUTHORIZATION	Ensure Duration in FixedUpfrontPricingTerm matches duration specified in the ResaleAuthorization.
INCOMPATIBLE_RESALE_AUTHORIZATION	Provide term(s) that are compatible with the ResaleAuthorization. Incompatible terms: [x, y, z].
INCOMPATIBLE_SELECTOR_DURATION	Durations aren't allowed to be removed from rate cards in ConfigurableUpfrontPricingTerm after the offer released.
INCOMPATIBLE_TERMS	[x] isn't supported together with the following terms: [y,z].
INCOMPATIBLE_TERMS	The following terms can't be added after the offer is released: [x,y,z].
INCOMPATIBLE_TERMS	The following terms can't be removed after the offer is released: [x,y,z].
INCOMPATIBLE_TERMS	[x] isn't supported for private offers.
INCOMPATIBLE_TERMS	The following terms aren't supported with FreeTrialPricingTerm that grants unlimited usage: [x,y,z].

Error code	Error message
INCOMPATIBLE_TERMS	The following terms aren't supported with FreeTrialPricingTerm for the product: [x,y,z].
INCOMPATIBLE_TERMS	Provide zero (0) price for FixedUpfrontPricingTerm when the offer contains a PaymentScheduleTerm.
INCOMPATIBLE_TERMS	The following terms aren't compatible with the PricingModel: [x,y,z].
INCOMPATIBLE_TERMS	FixedUpfrontPricingTerm isn't supported when MarkupPercentage is greater than zero (0).
INCOMPATIBLE_TERMS	The requested change can't be performed after the offer is released.
INCOMPATIBLE_TERMS	The requested change can't be performed after the offer is expired.
INVALID_AGREEMENT_DURATION	Provide duration between [x] and [y] months.
INVALID_AGREEMENT_DURATION	Ensure duration granularity is at the day level for metered dimensions.
INVALID_CURRENCY_CODE	Provide a supported CurrencyCode.
INVALID_CURRENCY_CODE	Provide the same CurrencyCode across all pricing and payment terms.
INVALID_DURATION	Ensure Duration in FreeTrialPricingTerm is within the allowed range.
INVALID_DURATION	Provide Duration in FixedUpfrontPricingTerm that matches the duration between AgreementStartDate and AgreementEndDate.
INVALID_DURATION	Provide duration between [x] and [y] months.

Error code	Error message
INVALID_DURATION	Ensure duration granularity is at the day level for metered dimensions.
INVALID_GRANTS	Provide the same MaxQuantity for all Grants in FreeTrialPricingTerm.
INVALID_GRANTS	Provide Grants for all available metered dimensions in FreeTrialPricingTerm.
INVALID_PRICE_CHANGE	[x] can't be updated until [y] because you have requested a price increase in the past 120 days. To cancel your previous price increase request or for more information, contact the AWS Marketplace Managed Catalog Operations Team.
INVALID_PRICE_CHANGE	Price increase and dimension addition in [x] isn't supported in the same request. Add dimensions first.
INVALID_PRICE_CHANGE	Price increase and decrease in UsageBasedPricingTerm isn't supported in the same request. Decrease prices first.
INVALID_PRICE_CHANGE	Price increase in RecurringPaymentTerm and price decrease in UsageBasedPricingTerm isn't supported in the same request. Decrease prices first.
INVALID_PRICE_CHANGE	Price decrease in RecurringPaymentTerm and price increase in UsageBasedPricingTerm isn't supported in the same request. Decrease prices first.

Error code	Error message
INVALID_RATE_CARD	ConfigurableUpfrontPricingTerm is missing one or more-dimension keys for duration [x]. Provide prices for the same set of dimension keys for all durations.
INVALID_RATE_CARD	Provide a rate card for only metered dimensions in UsageBasedPricingTerm.
INVALID_RATE_CARD	Rates can't be removed from [x]. Provide prices for all dimensions in the existing rate card.
INVALID_RATE_CARD	Provide dimensions that have the same unit in [x].
INVALID_RATE_CARD	Provide either all metered or all entitled dimensions in [x].
INVALID_RATE_CARD	Provide only entitled dimensions in [x].
INVALID_RATE_CARD	Provide usage based rates for all available metered dimensions in UsageBasedPricingTerm.
INVALID_RATE_CARD	Provide usage based rates for all free trial dimensions.
INVALID_RATE_CARD	Provide prices with up to 3 decimal places in UsageBasedPricingTerm.
INVALID_SELECTOR_DURATION_VALUE	Provide duration between [x] and [y] months.
INVALID_SELECTOR_DURATION_VALUE	Ensure duration granularity is at the day level for metered dimensions.
INVALID_SELECTOR_DURATION_VALUE	Ensure Duration in ConfigurableUpfrontPricingTerm is within the allowed range.

Error code	Error message
INVALID_SELECTOR_DURATION_VALUE	Provide one or more supported contract durations.
INVALID_SELECTOR_DURATION_VALUE	Provide one or more supported contract durations or a single custom duration.
INVALID_SELECTOR_DURATION_VALUE	Provide Duration in ConfigurableUpfrontPricingTerm that matches the duration between AgreementStartDate and AgreementEndDate.
MISSING_DURATION	Provide Duration in FixedUpfrontPricingTerm.
MISSING_MANDATORY_TERMS	FixedUpfrontPricingTerm is only supported when paired with ByolPricingTerm or PaymentScheduleTerm.
MISSING_MANDATORY_TERMS	Provide at least one of [x,y,z].
MISSING_MANDATORY_TERMS	Provide a ByolPricingTerm when using Byol pricing model.
TOO_MANY_GRANTS	Provide up to [x] grants in [y].
TOO_MANY_RATE_CARDS	Only one rate card in ConfigurableUpfrontPricingTerm is allowed for the product.
TOO_MANY_RATE_CARDS	Up to [x] rate cards are allowed in ConfigurableUpfrontPricingTerm for the product.
TOO_MANY_RATES	Provide RateCards within the allowed limits in ConfigurableUpfrontPricingTerm.
TOO_MANY_RATES	Provide RateCards within the allowed limits in UsageBasedPricingTerm.

## Update payment schedule details

You can use the Catalog API to change the payment schedule details, such as flexible payment schedule, in AWS Marketplace.

Channel partners can't override the payment schedule terms if the manufacturer provided terms in the Resale Authorization. You can increase the payment amount using the UpdateMarkup change type.

To update payment-associated details for your offer, call the StartChangeSet API operation with the UpdatePaymentScheduleTerms change type, as shown in the following example.

### Request Syntax

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

{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "UpdatePaymentScheduleTerms",
      "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "offer-123456789"
      },
      "DetailsDocument": {
        "Terms": [
          {
            "Type": "PaymentScheduleTerm",
            "Schedule": [
              {
                "ChargeDate": "2021-12-01",
                "ChargeAmount": "200.00"
              },
              {
                "ChargeDate": "2022-03-01",
                "ChargeAmount": "250.00"
              }
            ]
          }
        ]
      }
    }
  ]
}
```

```
    }  
  }  
]  
}
```

Provide information for the fields to add the `UpdatePaymentScheduleTerms` change type:

- **Entity** (object) (required) – Your CPPO.
  - **Type** (string) (required) – The Type is always `Offer@1.0`.
  - **Identifier** (string) (required) – Your offer ID. For more information, see [Identifier](#).
- **DetailsDocument** (object) (required) – The JSON value of specifics of the request.
- **Terms** (array of structures) – List of payment terms that you want to update. Supported payment terms are:
  - **PaymentScheduleTerm** (object) – Defines an installment-based pricing model where customers are charged a fixed price on different dates during the agreement validity period.
    - **Type** (string) – Type of the term being updated. This is the object value: `"PaymentScheduleTerm"`.
  - **Schedule** (array of structures) – List of the payment schedule where each element defines one installment of payment. It contains the information necessary for calculating the price to be paid and the date on which the customer would be charged.
    - **ChargeDate** (string) – The date on which the customer would pay the price defined in this payment schedule term. `ChargeDate` is represented in `YYYY-MM-DD` format. Invoices are generated on the date provided.
    - **ChargeAmount** (string) – The price that the customer would pay on scheduled date (`ChargeDate`).

## Response Syntax

A change set is created for your request. The response to this request gives you the `ChangeSetId` and `ChangeSetArn` 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 includes validating information to ensure that it meets the AWS Marketplace guidelines. The validation process can take anywhere from a few minutes to a few hours.

You can check the status of the request through the AWS Marketplace Management Portal, or directly through Catalog API using the [DescribeChangeSet](#) API operation.

## Synchronous Validations

The following schema validations are specific to UpdatePaymentScheduleTerms actions in the AWS Marketplace Catalog API. These validations are performed when you call StartChangeSet. If the request doesn't meet the following requirements, it will fail with an HTTP response.

Input field	Validation rule	HTTP
Terms	Required  Only "PaymentScheduleTerm" is allowed  List size must be less than 2	422
Terms[].PaymentScheduleTerm.CurrencyCode	Required  Supported currencies: USD	422
Terms[].PaymentScheduleTerm.Schedule[]	Required  List size must be between 1 and 60, inclusive	422
Terms[].PaymentScheduleTerm.Schedule[].ChargeDate	Required  Format: "YYYY-MM-DD"	422
Terms[].PaymentScheduleTerm.Schedule[].ChargeAmount	Required  Data type is "String"  Non-negative decimals with up to 2 decimal places supported	422

Input field	Validation rule	HTTP
	No additional properties are allowed	

## Asynchronous Errors

The following errors are specific to UpdatePaymentScheduleTerms actions in the AWS Marketplace Catalog API. These errors are returned when you call DescribeChangeSet after a change set is processing. For more information about using DescribeChangeSet to get the status of a change request, see [Working with change sets](#).

Error code	Error message
DUPLICATE_CHARGE_DATES	Provide unique charge dates in PaymentScheduleTerm.
INCOMPATIBLE_MARKUP_PERCENTAGE	PaymentScheduleTerm isn't supported when MarkupPercentage is greater than zero (0).
INCOMPATIBLE_RESALE_AUTHORIZATION	Provide term(s) that are compatible with the ResaleAuthorization. Incompatible terms: [PaymentScheduleTerm].
INCOMPATIBLE_RESALE_AUTHORIZATION	Ensure the total ChargeAmounts in PaymentScheduleTerm is compatible with the ResaleAuthorization.
INCOMPATIBLE_TERMS	The requested change can't be performed after the offer is released.
INCOMPATIBLE_TERMS	The requested change can't be performed after the offer is expired.
INVALID_CHARGE_DATES	Provide charge dates before AgreementEndDate.
TOO_MANY_BACKDATED_CHARGES	Provide up to 1 scheduled payment before AvailabilityEndDate.

## Publish the CPPO

You can use the Catalog API to merge the information collected from all update change types, and then publish the offer in AWS Marketplace.

Offers remain in a Draft state, until `ReleaseOffer` is called. After the offer is released, it's discoverable in AWS Marketplace.

To publish your offer, call the `StartChangeSet` API operation with the `ReleaseOffer` change type as shown in the following example.

### Request Syntax

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

{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "ReleaseOffer",
      "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "offer-123456789"
      },
      "DetailsDocument": {}
    }
  ]
}
```

Provide information for the fields to add the `ReleaseOffer` change type:

- **Entity** (object) (required) – Your CPPO.
  - **Type** (string) (required) – The Type is always `Offer@1.0`.
  - **Identifier** (string) (required) – Your offer ID. For more information, see [Identifier](#).
- **DetailsDocument** (object) (required) – The JSON value of specifics of the request. It must be empty for `ReleaseOffer`.

### Response Syntax

A change set is created for your request. The response to this request gives you the `ChangeSetId` and `ChangeSetArn` 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 includes validating information to ensure that it meets the AWS Marketplace guidelines. The validation process can take anywhere from a few minutes to a few hours.

You can check the status of the request through the AWS Marketplace Management Portal, or directly through Catalog API using the [DescribeChangeSet](#) API operation.

### Synchronous Validations

The following schema validations are specific to `ReleaseOffer` actions in the AWS Marketplace Catalog API. These validations are performed when you call `StartChangeSet`. If the request doesn't meet the following requirements, it will fail with an HTTP response.

Input field	Validation rule
Details	Must be empty ({})

### Asynchronous Errors

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

Error code	Error message
INCOMPATIBLE_PRODUCT	First create a public offer for the product.
INVALID_UPDATE_REQUEST	The requested change can't be performed after the offer is released.

Error code	Error message
MISSING_AGREEMENT_END_DATE	Provide an AgreementEndDate for replacement offers.
MISSING_AVAILABILITY_END_DATE	Provide an AvailabilityEndDate for private offer.
MISSING_MANDATORY_TERMS	Provide a FixedUpfrontPricingTerm when the offer contains a PaymentScheduleTerm.
MISSING_BUYER_ACCOUNTS	Provide PositiveTargeting with BuyersAccounts for offers created using ResaleAuthorization.
MISSING_BUYER_ACCOUNTS	All offers for the product must be private. Provide PositiveTargeting with BuyersAccounts.
MISSING_DESCRIPTION	Set Description before releasing the offer.
MISSING_MANDATORY_TERMS	Add [x] to the offer.
MISSING_MANDATORY_TERMS	Provide a FixedUpfrontPricingTerm when the offer contains a PaymentScheduleTerm.
MISSING_NAME	Set Name before releasing the offer.
TOO_MANY_OFFERS	Only one public offer can be created per product.
TOO_MANY_OFFERS	Only one public free trial offer can be created per product.

## Define an existing CPPO

You can use the Catalog API to define CPPO details in AWS Marketplace.

To define an existing CPPO, call the `DescribeEntity` API operation with the `Offer@1.0` entity type, as shown in the following example.

## Request Syntax

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

Provide information for the fields to add the `DescribeEntity` change type:

- **catalog** (string) – The catalog related to the request. Fixed value: `AWSMarketplace`.
- **entityId** (string) – The unique ID of the offer to describe.

## Response Syntax

The response to this request gives you the offer details and looks like the following.

```
{
  "EntityType": "Offer@1.0",
  "EntityIdentifier": "offer-a5oEXAMPLEzpu@1",
  "EntityArn": "arn:aws:aws-marketplace:us-east-1:446235747164:AWSMarketplace/Offer/offer-a5oEXAMPLEzpu",
  "LastModifiedDate": "2021-03-10T21:57:16Z",
  "Details": {
    "Id": "offer-3rEXAMPLErn",
    "State": "Released",
    "Name": "Test Offer",
    "Description": "Worldwide private offer for Test Product",
    "PreExistingAgreement": {
      "AcquisitionChannel": "External",
      "PricingModel": "Contract"
    },
    "ProductId": "prod-0bEXAMPLEb51",
    "ResaleAuthorizationId": "resaleauthz-123456789",
    "MarkupPercentage": "5.0",
    "Terms": [
      {
        "Type": "LegalTerm",
        "Documents": [
          {
            "Type": "CustomEula",
            "Url": "https://s3.amazonaws.com/EULA/custom-eula-1234.txt"
          }
        ]
      }
    ]
  }
}
```

```
]
},
{
  "Type": "ConfigurableUpfrontPricingTerm",
  "CurrencyCode": "USD",
  "RateCards": [
    {
      "Selector": {
        "Type": "Duration",
        "Value": "P12M"
      },
      "RateCard": [
        {
          "DimensionKey": "m3.large",
          "Price": "300.00"
        },
        {
          "DimensionKey": "m4.xlarge",
          "Price": "400.00"
        }
      ],
      "Constraints": {
        "MultipleDimensionSelection": "Allowed",
        "QuantityConfiguration": "Allowed"
      }
    }
  ]
},
{
  "Type": "UsageBasedPricingTerm",
  "CurrencyCode": "USD",
  "RateCards": [
    {
      "RateCard": [
        {
          "DimensionKey": "m3.large",
          "Price": "0.10"
        },
        {
          "DimensionKey": "m4.xlarge",
          "Price": "0.20"
        }
      ]
    }
  ]
}
```

```
    ]
  },
  {
    "Type": "PaymentScheduleTerm",
    "CurrencyCode": "USD",
    "Schedule": [
      {
        "ChargeDate": "2020-12-01T00:00:00Z",
        "ChargeAmount": "1000.00"
      },
      {
        "ChargeDate": "2021-06-15T00:00:00Z",
        "ChargeAmount": "1250.00"
      }
    ]
  }
],
"Rules": [
  {
    "Type": "TargetingRule",
    "PositiveTargeting": {
      "CountryCodes": [
        "US",
        "CA"
      ],
      "BuyerAccounts": [
        "118033953248"
      ]
    },
    "NegativeTargeting": {
      "CountryCodes": [
        "XX"
      ]
    }
  },
  {
    "Type": "AvailabilityRule",
    "AvailabilityEndDate": "2050-08-30T01:56:03Z"
  }
]
}
```

Provide information for the input fields for the DescribeEntity response:

- **EntityType** (string) – The named type of the entity, which is Offer@1.0.
- **EntityIdentifier** (string) – The identifier of the entity, in the format of EntityId@RevisionId.
- **EntityArn** (string) – The ARN associated to the unique identifier for the change set referenced in this request.
- **LastModifiedDate** (string) – The last modified date of the entity, in ISO 8601 format (2018-02-27T13:45:22Z).
- **DetailsDocument** (object) – The JSON object includes the details of the entity.
  - **Id** (string) – Unique identifier for an offer entity in AWS Marketplace and is generated during the creation of an offer.
  - **State** (string) – The status of the offer.
  - **Name** (string) – Name associated with the offer for better readability to you and your customers. It will be displayed as part of Agreement information as well.
  - **Description** (string) – Description is a free-form text which is meant to be used only by you and will never be exposed to buyers.
  - **PreExistingAgreement** (string) – Determines if this offer is a renewal for an existing agreement with an existing customer for the same underlying product. The existing agreement can be within or outside AWS Marketplace. AWS may audit and verify your offer is a renewal. If AWS is unable to verify your offer, then AWS may revoke the offer and entitlements from your customer.
    - **AcquisitionChannel** (string) – Indicates if the existing agreement was signed outside AWS Marketplace or within AWS Marketplace. Possible values: External, AwsMarketplace.
    - **PricingModel** (string) – Indicates which pricing model the existing agreement uses. Possible values: Contract, Usage, Byo1, Free.
  - **ProductId** (string) – Description is a free-form text which is meant to be used only by you and will never be exposed to buyers.
  - **ResaleAuthorizationId** (string) – ResaleAuthorization is used to create the private offer.
  - **MarkupPercentage** (string) – Percentage value that the channel partner passed in the UpdateMarkup change type. This markup is already applied to the terms.
  - **Terms** (array of structures) – List of terms.
  - **Rules** (array of structures) – List of rules.

## Working with a private marketplace

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

All change types can be called only from the organization's management account or by a member account that is a delegated administrator for private marketplace. If you're a current private marketplace customer without the AWS Organizations integration for private marketplace, you can create and manage a private marketplace from any account in your organization that has the `AWSPriateMarketplaceAdminFullAccess` IAM policy.

For more information about private marketplaces, see [Private marketplaces](#) in the *AWS Marketplace Buyer Guide*.

The following table details a set of tasks to manage private marketplaces and the change types that apply to each task.

Task	Action	Change types
<a href="#">the section called "Creating a private marketplace"</a>	StartChangeSet	CreateExperience CreateProcurementPolicy
<a href="#">the section called "Changing the branding of a private marketplace experience"</a>	StartChangeSet	CreateBrandingSettings UpdateBrandingSettings
<a href="#">the section called "Enabling or disabling a private marketplace experience"</a>	StartChangeSet	UpdateExperience
<a href="#">the section called "Enabling or disabling user requests"</a>	StartChangeSet	UpdateProcurementPolicy

Task	Action	Change types
<a href="#">the section called “Getting a list of products in a private marketplace experience”</a>	DescribeEntity	Not applicable
<a href="#">the section called “Adding or removing products from a private marketplace”</a>	StartChangeSet	AllowProductProcurement DenyProductProcurement
<a href="#">the section called “Finding products”</a>	Not applicable	Not applicable
<a href="#">the section called “Working with private marketplaces for AWS Organizations”</a>	Not applicable	Not applicable
<a href="#">the section called “Associating principals to experiences”</a>	StartChangeSet	AssociateAudience DisassociateAudience
<a href="#">Archiving and reactivating a private marketplace experience</a>	StartChangeSet	RestrictExperience ReviveExperience

## 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, you can use the grouping from AWS Organizations called [organizational unit \(OU\)](#) to associate to an experience. 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 OU in the organization. You can also have a different list of products for individual accounts 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:

- Experience entity – This entity is at the highest level of the experience and contains two child entities.
- ProcurementPolicy entity – This entity represents the products that have been allowed and denied in your private marketplace.
- BrandingSettings entity – You can also create a BrandingSettings entity to define how your private marketplace looks to your users.
- Audience entity – You must also associate one or more Audience entities, which define the set of AWS accounts, OUs, or organization 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 principals with your experience. A principal can be an AWS account, OU, or the organization.
5. Enable the experience.

 **Note**

If your account is part of an organization in AWS Organizations, see [Working with private marketplaces for AWS Organizations](#).

## 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",
      "DetailsDocument":
      {
        "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](#).

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](#).

## 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",
      "DetailsDocument":
      {
        "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](#).

You can again use `DescribeChangeSet` on the `CreateProcurementPolicy` change type 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 principals 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",
      "DetailsDocument":
      {
        "Name": "ExampleBrandingSettingsName"
      },
      "Entity":
      {
        "Type": "Experience@1.0",
        "Identifier": "exp-1234example@2"
      }
    }
  ]
}
```

This example 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](#).

**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 DetailsDocument object.

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

{
  "Catalog": "AWSMarketplace",
  "ChangeSet":
  [
    {
      "ChangeType": "UpdateBrandingSettings",
      "DetailsDocument":
      {
        "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**

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",
      "DetailsDocument":
      {
        "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 `DetailsDocument` 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 default experience for the organization). 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 ([Private Marketplace](#)), 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",
      "DetailsDocument":
      {
        "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](#).

## 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.

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

```

### Note

The `DetailsDocument` attribute contains the entity details as a JSON object. The legacy `Details` attribute contains the same JSON object as a string.

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.

```

{
  "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\\"}}",
  "DetailsDocument":
  {
    "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 example, 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",
      "DetailsDocument":
      {
        "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 `DetailsDocument` 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](#).

## 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:

- Before you can use private marketplace feature in an organization, you must [enable trusted access](#) to provide private marketplace service (`private-marketplace.marketplace.amazonaws.com`) access to your AWS Organizations data. You must also [create the private marketplace service-linked role](#) in the management account. This role includes all the permissions that private marketplace requires to describe AWS Organizations and update private marketplace resources on your behalf. These actions can only be performed by the management account. It is recommended to perform this enablement using private marketplace administrator page. If you are a new customer, see [Private marketplaces](#) in the *AWS Marketplace Buyer Guide*. If you are an existing customer, see [Creating and managing a private marketplace](#) in the *AWS Marketplace Buyer Guide*.
- Private marketplace resources in an organization are created in the management account and shared with the member account that is a delegated administrator for private marketplace.
- When listing objects in a private marketplace from a member account that is a delegated administrator for private marketplace, you must specifically request them with the `SharedWithMe` filter. This applies to both `ListEntities` and `ListChangeSets` actions.

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 principals to experiences

A private marketplace experience must have one or more principals 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 principals.

### Note

- The experience that is associated with the organization is the default for all other accounts in the organization. Associating a member account or OU with a different experience directly sets a different experience for the member account or child accounts of the OU.
- If you are a current private marketplace customer without the AWS Organizations integration for private marketplace, the experience that is associated with the management account is the default for all other accounts in the organization.

To associate a principal 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",
      "DetailsDocument":
      {
        "Name": "AudienceName",
        "Description": "Audience example.",
        "Principals":
        [
          "012345678901",
          "ou-abcd-01234567",
          "o-0123456789"
        ]
      }
    }
  ],
  "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, organizational unit, or organization defined by its ID. `Principals` is a list, so you can include multiple principals 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",
      "DetailsDocument":
      {
        "Principals":
        [
          "012345678901",
          "ou-abcd-01234567",
          "o-0123456789"
        ]
      }
    }
  ],
  "ChangeSetName": "Disassociate audience example"
}
```

### Note

A principal can only be directly associated with one experience. To move a principal from being directly associated with one experience to another experience, you must disassociate it from the initial experience, then associate it with the second.

## Archiving and reactivating a private marketplace experience

You can remove a private marketplace experience by archiving it. Archived experiences can't be updated or used to govern accounts in your organization. If you have audiences associated with an archived experience, you can associate them with a different experience. If you decide to use the experience at a later time, you can always reactivate it. Administrators from the management

account or a member account that is a delegated administrator for private marketplace have permissions to archive and reactivate experiences. If you're a current private marketplace customer without the AWS Organizations integration for private marketplace, administrators from the account that created the experience have permissions to archive and reactivate experiences.

**Note**

Before archiving an experience, you must disable it. For information about disabling an experience, see [Configuring your private marketplace](#) in the *AWS Marketplace Buyer Guide*.

To archive an experience, use the `RestrictExperience` 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":
  [
    {
      "ChangeType": "RestrictExperience",
      "DetailsDocument":
      {},
      "Entity":
      {
        "Type": "Experience@1.0",
        "Identifier": "exp-1234example"
      }
    }
  ]
}
```

To reactivate an experience, use the `ReviveExperience` 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":
  [
    {
      "ChangeType": "ReviveExperience",
      "DetailsDocument":
      {},
      "Entity":
      {
        "Type": "Experience@1.0",
        "Identifier": "exp-1234example"
      }
    }
  ]
}

```

## 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
<b>Errors returned directly by the StartChangeSet action</b>			
All	422	Document not valid JSON format	Invalid JSON input used, check your syntax.
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.

Change type	Error code	Error message	Description
<b>Errors found by calling the DescribeChangeSet action</b>			
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.

<b>Change type</b>	<b>Error code</b>	<b>Error message</b>	<b>Description</b>
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.
CreateExperience	CALLER_NOT_AUTHORIZED	Caller not authorized to create experience.	You must have permissions to create an experience.
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.

Change type	Error code	Error message	Description
AssociateAudience, DisassociateAudience	ENTITY_IN_USE	There is already a conflicting change in progress for the selected account. Try again later	You can't change the association with an account while another change request to change the association is already in progress.

## Entity types defined by private marketplace

The following table lists the private marketplace entity types, purpose, and actions on which each can be specified. Each entity type can be used to specify a resource Amazon Resource Name (ARN) that can be used in the AWS Identity and Access Management (IAM) policy. For more details on ARN formats, see [the section called "Catalog API entities"](#).

Entity	Purpose	Actions
Experience	Stores the top-level settings for a private marketplace	StartChangeSet DescribeEntity
BrandingSettings	Stores the branding settings for a private marketplace	DescribeEntity
ProcurementPolicy	Stores the procurement settings and lists of products in a private marketplace	DescribeEntity
Audience	Stores the details of principals associated with a private marketplace	DescribeEntity

# Working with AWS RAM to share resources

AWS Marketplace Catalog API integrates with AWS Resource Access Manager (AWS RAM) to enable resource sharing. A *resource* is an entity that users can work with in AWS Marketplace, such as a product, an offer, or a resale authorization. With AWS RAM, you can share some AWS Marketplace Catalog API resources with other AWS accounts. You share resources that you own by creating a *resource share*. A resource share specifies the resources that you want to share and the consumers with whom to share them.

## Contents

- [Prerequisites for sharing AWS Marketplace entities](#)
- [Sharing an AWS Marketplace entity](#)

## Prerequisites for sharing AWS Marketplace entities

Before sharing entities in AWS Marketplace Catalog API, you must meet the following prerequisites:

- You can only have one resource policy attached to your AWS Marketplace entity.
- To share an AWS Marketplace entity, you must own it in your AWS account. This requirement means that the entity must be allocated or provisioned in your account. You can't share an AWS Marketplace entity that has been shared with you.

## Sharing an AWS Marketplace entity

With AWS Marketplace resource sharing, entity owners can share their entities with other AWS accounts in AWS Marketplace. Entity-owners can be ISVs and channel partners. Entities that can be shared are products, offers, and resale authorizations.

### Note

At this time, you can only share entities. Entities in AWS Marketplace include `AmiProduct`, `Audience`, `BrandingSettings`, `ContainerProduct`, `Experience`, and `ProcurementPolicy`.

For more information about AWS RAM, see the [AWS RAM User Guide](#). For more information about managing your shared resources, see [Using shared AWS resources](#) in the *AWS RAM User Guide*.

As a *sharing account*, you can set read-only or both read/write on the resources that you want to share. These permissions determine what operations a *consuming account* can perform on the resources that are shared with them.

- **Sharing account** – The resource that is shared and in which the AWS RAM administrator creates the AWS resource share by using AWS RAM.
- **Consuming account** – The AWS account to which a resource is shared. The resource share can specify an entire account as the principal, or for some resource types, individual roles or users in the account.

To share an AWS Marketplace entity, you must add it to a resource share. A resource share is an AWS RAM resource that lets you share your resources across AWS accounts. A resource share specifies the resources to share, and the consumers with whom they are shared. When you share an entity using the AWS Marketplace console, you add it to an existing resource share. To add the AWS Marketplace entity to a new resource share, you must first create the resource share using the [AWS RAM console](#).

You can share an AWS Marketplace entity that you own using the AWS Marketplace console, AWS RAM console, or the AWS Command Line Interface (AWS CLI).

### To share an AWS Marketplace entity that you own using the AWS RAM console

See [Creating a Resource Share](#) in the *AWS RAM User Guide*.

### To share an AWS Marketplace entity that you own using the AWS CLI

Use the [create-resource-share](#) command.

#### Note

For resource types such as entities that support resource-based policies, you can use AWS RAM to share resources to use additional AWS RAM features. For more information, see [Resource-based policy](#) in the *AWS RAM User Guide*. AWS RAM uses the AWS Marketplace Catalog API to automatically construct the resource policy from permissions in a resource share and manages that resource policy for you.

For information about how to set, view, or delete AWS resource-based policies on your AWS Marketplace entity through AWS RAM, see [Allowing actions on all resources](#) in the *AWS RAM User Guide*.

## Differences between sharing an entity through AWS RAM and the AWS Marketplace Catalog API

In addition to sharing your entity through AWS RAM, you can also set, view, or delete AWS resource-based policies on your entities through the AWS Marketplace Catalog API. However, there are a few differences between sharing your entity through AWS RAM and through the AWS Marketplace Catalog API.

When you share an entity through AWS RAM:

- If you share your entity with accounts that are outside of AWS Organizations, the consuming account must first accept your sharing request before the entity is shared.
- The consuming account can discover the shared entity through `ListEntities` with `OwnershipType` set to `SHARED`.
- You must adhere to several resource quotas. For more information, see [Service quotas for AWS RAM](#) in the *AWS RAM User Guide*.

When you share an entity through the AWS Marketplace Catalog API:

- Your entity will be shared as soon as the `PutResourcePolicy` request succeeds with no input from the consuming account.
- The consuming account can't discover the shared entity through `ListEntities` with `OwnershipType` set to `SHARED`. Instead, the owner of the sharing account must inform the consuming account of the shared entity ID.

### Note

If your use case requires sharing resources that might exceed AWS RAM service quotas, or if you want to share resources without direct input from the consuming account, consider sharing through the AWS Marketplace Catalog API. For all other use cases, consider using AWS RAM to share AWS Marketplace resources.

The following sections detail how you can set, view, or delete AWS resource-based policies on your entities through the AWS Marketplace Catalog API.

## Topics

- [Attach read-only policy to your resource](#)
- [Attach read and write resource policy to your resource](#)
- [View resource policy set on your resource](#)
- [Delete resource policy on your resource](#)
- [View all resources owned by you and shared with you](#)

## Attach read-only policy to your resource

You can create a read-only resource-based policy on your shared resource using a sharing account. With this policy, the principal can only view the details of the resource that is shared with them.

### Request

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

{
  "ResourceArn": "arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/
AmiProduct/example2-abcd-1234-5ef6",
  "Policy": {
    "Version": "2012-10-17",
    "Statement": {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::222233334444:root"
      },
      "Action": [
        "aws-marketplace:DescribeEntity"
      ],
      "Resource": [
        "arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/
AmiProduct/example2-abcd-1234-5ef6"
      ]
    }
  }
}
```

```
}
```

## Response

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

```
{}
```

## Attach read and write resource policy to your resource

As a sharing account, you can create a read and write resource-based policy on your shared resource. With this policy, the principal can view the details and perform write operations on the resource that is shared with them.

## Request

```
POST /PutResourcePolicy HTTP/1.1  
Content-type: application/json  
  
{  
  "ResourceArn": "arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/  
AmiProduct/example2-abcd-1234-5ef6",  
  "Policy": {  
    "Version": "2012-10-17",  
    "Statement": {  
      "Effect": "Allow",  
      "Principal": {  
        "AWS": "arn:aws:iam::222233334444:root"  
      },  
      "Action": [  
        "aws-marketplace:DescribeEntity",  
        "aws-marketplace:StartChangeSet"  
      ],  
      "Resource": [  
        "arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/  
AmiProduct/example2-abcd-1234-5ef6"  
      ]  
    }  
  }  
}
```

## Response

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

{}
```

## View resource policy set on your resource

As a sharing account, you can view the resource policy that is set on your shared resource.

## Request

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

{
  "ResourceArn": "arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/
  AmiProduct/example2-abcd-1234-5ef6"
}
```

## Response

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

{
  "Policy": {
    "Version": "2012-10-17",
    "Statement": {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::222233334444:root"
      },
      "Action": [
        "aws-marketplace:DescribeEntity",
        "aws-marketplace:StartChangeSet"
      ],
      "Resource": [
        "arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/
        AmiProduct/example2-abcd-1234-5ef6"
      ]
    }
  }
}
```

```
    }  
  }  
}
```

## Delete resource policy on your resource

As a sharing account, you can delete the resource policy that is set on your shared resource.

### Request

```
POST /DeleteResourcePolicy HTTP/1.1  
Content-type: application/json  
  
{  
  "ResourceArn": "arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/  
AmiProduct/example2-abcd-1234-5ef6"  
}
```

### Response

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

## View all resources owned by you and shared with you

As a consuming account, you can view the resources that are shared with you.

### Note

You can view the resources shared with you only if the resources were shared through AWS RAM.

### Request

```
POST /ListEntities HTTP/1.1
```

```
Content-type: application/json
{
  "Catalog": "AWSMarketplace",
  "EntityType": "AmiProduct",
  "FilterList": [
    {
      "Name": "EntityId",
      "ValueList": [ "example2-abcd-1234-5ef6" ]
    }
  ],
  "OwnershipType": "SHARED"
}
```

## Response

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

{
  "EntitySummaryList": [
    {
      "EntityArn": "arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/
AmiProduct/example2-abcd-1234-5ef6",
      "EntityId": "example2-abcd-1234-5ef6",
      "EntityType": "AmiProduct",
      "LastModifiedDate": "2018-02-27T13:45:22Z",
      "Name": "TestProduct",
      "Visibility": "public"
    }
  ],
  "NextToken": ""
}
```

# Logging and notifications

The AWS Marketplace Catalog API supports logging with CloudTrail and notifications with Amazon EventBridge.

## Topics

- [Logging AWS Marketplace Catalog API calls with CloudTrail](#)
- [AWS Marketplace Catalog API Amazon EventBridge events](#)

## 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 Simple Storage Service (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 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": "ABCDEF GHIJKLMNOP12345",
    "arn": "arn:aws:iam::123456789010:user/CloudTrailTestUser",
    "accountId": "123456789010",
    "accessKeyId": "ABCDEF GHIJKLMNOP1234",
    "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"
}
]

```

## AWS Marketplace Catalog API Amazon EventBridge events

AWS Marketplace is integrated with Amazon EventBridge, formerly called Amazon CloudWatch Events. EventBridge is an event bus service that you can use to connect your applications with data from a variety of sources.

For information on how sellers, channel partners, and private marketplace administrators can receive ChangeSet status events using EventBridge, see [Events for change sets](#) in the *AWS Marketplace Seller Guide*.

# AWS Marketplace Discovery API

The AWS Marketplace Catalog API service provides an API interface to manage the products that you create as a seller in AWS Marketplace, and to manage your private marketplace. However, for discovery use cases, you must use the AWS Marketplace Discovery API (Discovery API).

The Discovery API enables your buyers to have a frictionless discovery experience of AWS Marketplace listings on your web properties. You can use the Discovery API to:

- Get listing information from AWS Marketplace, such as long and short product descriptions, marketplace categories, badges, media, pricing model/unit, information, publisher, reviews, and more.
- Populate your website, platform, and private marketplace with AWS Marketplace listings.
- Create a custom view of AWS Marketplace listings for your customers where you show offerings from other sellers and offer value-added functionality.

With the Discovery API, you can create a browse and search functionality of the entire AWS Marketplace catalog or a curated view that best serves your customers. This dynamic experience helps direct your customers to the most up-to-date and relevant AWS Marketplace pages to purchase the products they need. Additionally, you can access the full list of AWS Marketplace products to help populate your private marketplace.

## Note

The Discovery API is currently available to select AWS customers and is available upon request to, and approval by, the AWS Marketplace Team. For more information, see the next section, [Getting access to the Discovery API](#).

## Getting access to the Discovery API

To request access, or to get answers to questions about the Discovery API, reach out to your current AWS Marketplace contact. If you don't have an AWS Marketplace contact, you don't know who your contact is, or you don't have an AWS Marketplace Seller account, submit a general inquiry to the [AWS Marketplace Seller Operations team](#).

After you receive access to the Discovery API, you will receive a unique Integration ID, API documentation, and a SDK to help you integrate with and call the Discovery API.

## Logging Discovery API calls using AWS CloudTrail

The Discovery API is integrated with AWS CloudTrail, a service that provides a record of actions taken by a user, role, or an AWS service in AWS Marketplace. CloudTrail captures all API calls for the Discovery API as events. The calls captured include calls from the AWS Marketplace console and code calls to the Discovery API operations.

If you create a trail, you can enable continuous delivery of CloudTrail events to an Amazon S3 bucket, including events for the Discovery API. A *trail* enables CloudTrail to deliver log files 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 that was made to the Discovery API, the IP address from which the request was made, who made the request, when it was made, and additional details.

For more information about CloudTrail, see the [AWS CloudTrail User Guide](#).

## Discovery API information in CloudTrail

CloudTrail is enabled on your AWS account when you create the account. When activity occurs in the Discovery 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, including events for the Discovery API, 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 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 the following:

- [Overview for creating a trail](#)
- [CloudTrail supported services and integrations](#)
- [Configuring Amazon SNS notifications for CloudTrail](#)

- [Receiving CloudTrail log files from multiple regions](#) and [Receiving CloudTrail log files from multiple accounts](#)

All Discovery API actions are logged by CloudTrail and are documented in this API Reference. For example, calls to the `SearchListings`, `GetSearchFacets`, and `GetListingView` 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 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 Discovery API 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 aren't an ordered stack trace of the public API calls, so they don't appear in any specific order.

The following example shows a CloudTrail log entry that demonstrates the `SearchListings` action.

```
{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "ABCDEFGHJKLMNOP12345",
    "arn": "arn:aws:iam::123456789010:user/CloudTrailTestUser",
    "accountId": "123456789010",
    "accessKeyId": "ABCDEFGHJKLMNOP1234",
    "userName": "CloudTrailTestUser",
    "sessionContext": {
      "sessionIssuer": {},
      "webIdFederationData": {},
```

```
        "attributes": {
            "creationDate": "2022-09-01T20:33:21Z",
            "mfaAuthenticated": "false"
        }
    },
    "eventTime": "2022-09-01T20:33:58Z",
    "eventSource": "discovery-marketplace.amazonaws.com",
    "eventName": "SearchListings",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "12.34.567.890",
    "userAgent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15; rv:91.0) Gecko/20100101
Firefox/91.0",
    "requestParameters": {
        "maxResults": 20,
        "sort": {
            "sortBy": "RELEVANT",
            "sortOrder": "DESCENDING"
        },
        "requestContext": {
            "integrationId": "HIDDEN_DUE_TO_SECURITY_REASONS"
        }
    },
    "responseElements": null,
    "requestID": "fEXAMPLE-cb3e-4e21-86fd-6b3EXAMPLEd1",
    "eventID": "7EXAMPLE-97d6-4139-91e3-01aEXAMPLE48",
    "readOnly": true,
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "123456789010",
    "eventCategory": "Management",
    "tlsDetails": {
        "tlsVersion": "TLSv1.2",
        "cipherSuite": "ECDHE-RSA-AES128-GCM-SHA256",
        "clientProvidedHostHeader": "discovery.marketplace.us-east-1.amazonaws.com"
    },
    "sessionCredentialFromConsole": "true"
}
```

## Release notes for AWS Marketplace Discovery API

The release notes for AWS Marketplace Discovery API (Discovery API) provides details about the service's features, improvements, fixes, and announcements by release date.

# Discovery API release notes for 2022

## May 20, 2022

Published on May 20, 2022

Existing Discovery API customers can access the updated Discovery API documentation and SDK on the Amazon Simple Storage Service (Amazon S3) bucket that the Discovery API team shared with them previously. Customers can refer to the Change Log in the private documentation for more details.

Discovery API announces the following launch, and improvements:

### Launch announcements

- Discovery API launched in two additional AWS Regions:
  - US West (Oregon) – us-west-2
  - Europe (Ireland) – eu-west-1
- Discovery API Private SDK is now available in Java 2.x:
  - For more information about how to use the AWS SDK for Java 2.x, see the [AWS SDK for Java 2.x Developer Guide](#).
  - For more information about migration, see [migrating from version 1.x to 2.x of the AWS SDK for Java](#) in the *AWS SDK for Java 2.x Developer Guide*.
  - For more information about changes between versions 1.11.x and 2.x of the AWS SDK for Java 2.x, see [1.11 to 2.x Changelog](#) on the GitHub website.

### Improvements

- Enhanced sorting functionality for the SearchListings API operation by introducing new options for:
  - `SortBy` – AVERAGE\_CUSTOMER\_RATING, CREATION\_TIME, LAST\_MODIFIED\_TIME
  - `SortOrder` – ASCENDING
- SDK updated for all the existing languages with the latest AWS SDK artifacts.
- Documentation updated to include SDK usage section for all languages.

# Actions

The following actions are supported:

- [CancelChangeSet](#)
- [DeleteResourcePolicy](#)
- [DescribeChangeSet](#)
- [DescribeEntity](#)
- [GetResourcePolicy](#)
- [ListChangeSets](#)
- [ListEntities](#)
- [ListTagsForResource](#)
- [PutResourcePolicy](#)
- [StartChangeSet](#)
- [TagResource](#)
- [UntagResource](#)

# 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

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

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

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

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](#).

### **AccessDeniedException**

Access is denied.

HTTP status code: 403

HTTP Status Code: 403

### **InternalServerErrorException**

There was an internal service exception.

HTTP status code: 500

HTTP Status Code: 500

### **ResourceNotFoundException**

The specified resource wasn't found.

HTTP status code: 404

HTTP Status Code: 404

### **ThrottlingException**

Too many requests.

HTTP status code: 429

HTTP Status Code: 429

### **ValidationException**

An error occurred during validation.

HTTP status code: 422

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 V3](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

# DeleteResourcePolicy

Deletes a resource-based policy on an entity that is identified by its resource ARN.

## Request Syntax

```
DELETE /DeleteResourcePolicy?resourceArn=ResourceArn HTTP/1.1
```

## URI Request Parameters

The request uses the following URI parameters.

### ResourceArn

The Amazon Resource Name (ARN) of the entity resource that is associated with the resource policy.

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

Pattern: `^arn:[\w+=/, .@-]+:aws-marketplace:[\w+=/, .@-]*:[0-9]+:[\w+=, .@-]+(\/[\w+=, .@-]+)*$`

Required: Yes

## Request Body

The request does not have a request body.

## Response Syntax

```
HTTP/1.1 200
```

## Response Elements

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

## Errors

For information about the errors that are common to all actions, see [Common Errors](#).

## **AccessDeniedException**

Access is denied.

HTTP status code: 403

HTTP Status Code: 403

## **InternalServiceException**

There was an internal service exception.

HTTP status code: 500

HTTP Status Code: 500

## **ResourceNotFoundException**

The specified resource wasn't found.

HTTP status code: 404

HTTP Status Code: 404

## **ThrottlingException**

Too many requests.

HTTP status code: 429

HTTP Status Code: 429

## **ValidationException**

An error occurred during validation.

HTTP status code: 422

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 V3](#)
- [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

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

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",
      "DetailsDocument": JSON value,
      "Entity": {
        "Identifier": "string",
        "Type": "string"
      },
      "ErrorDetailList": [
        {
          "ErrorCode": "string",
          "ErrorMessage": "string"
        }
      ]
    }
  ],
  "ChangeSetArn": "string",
  "ChangeSetId": "string",
  "ChangeSetName": "string",
  "EndTime": "string",
  "FailureCode": "string",
  "FailureDescription": "string",
  "Intent": "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

An array of `ChangeSummary` objects.

Type: Array of [ChangeSummary](#) objects

## 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:*/-]+$`

## ChangeSetId

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

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

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

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

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

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

Pattern: ^(. )+\$

## Intent

The optional intent provided in the StartChangeSet request. If you do not provide an intent, APPLY is set by default.

Type: String

Valid Values: VALIDATE | APPLY

## StartTime

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

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](#).

### AccessDeniedException

Access is denied.

HTTP status code: 403

HTTP Status Code: 403

### InternalServiceException

There was an internal service exception.

HTTP status code: 500

HTTP Status Code: 500

### ResourceNotFoundException

The specified resource wasn't found.

HTTP status code: 404

HTTP Status Code: 404

### ThrottlingException

Too many requests.

HTTP status code: 429

HTTP Status Code: 429

### ValidationException

An error occurred during validation.

HTTP status code: 422

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 V3](#)
- [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

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

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",
  "DetailsDocument": JSON value,
  "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

This stringified JSON object includes the details of the entity.

Type: String

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

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

### DetailsDocument

The JSON value of the details specific to the entity.

Type: JSON value

### EntityArn

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

Type: String

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

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

### EntityIdentifier

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

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

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

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\`

## Errors

For information about the errors that are common to all actions, see [Common Errors](#).

### **AccessDeniedException**

Access is denied.

HTTP status code: 403

HTTP Status Code: 403

### **InternalServiceException**

There was an internal service exception.

HTTP status code: 500

HTTP Status Code: 500

### **ResourceNotFoundException**

The specified resource wasn't found.

HTTP status code: 404

HTTP Status Code: 404

### **ResourceNotSupportedException**

Currently, the specified resource is not supported.

HTTP Status Code: 415

### **ThrottlingException**

Too many requests.

HTTP status code: 429

HTTP Status Code: 429

### **ValidationException**

An error occurred during validation.

HTTP status code: 422

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 V3](#)

- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

# GetResourcePolicy

Gets a resource-based policy of an entity that is identified by its resource ARN.

## Request Syntax

```
GET /GetResourcePolicy?resourceArn=ResourceArn HTTP/1.1
```

## URI Request Parameters

The request uses the following URI parameters.

### ResourceArn

The Amazon Resource Name (ARN) of the entity resource that is associated with the resource policy.

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

Pattern: `^arn:[\w+=/, .@-]+:aws-marketplace:[\w+=/, .@-]*:[0-9]+:[\w+=, .@-]+(/[\w+=, .@-]+)*$`

Required: Yes

## Request Body

The request does not have a request body.

## Response Syntax

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

{
  "Policy": "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.

## Policy

The policy document to set; formatted in JSON.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 10240.

Pattern: `^[\u0009\u000A\u000D\u0020-\u00FF]+$`

## Errors

For information about the errors that are common to all actions, see [Common Errors](#).

### **AccessDeniedException**

Access is denied.

HTTP status code: 403

HTTP Status Code: 403

### **InternalServiceException**

There was an internal service exception.

HTTP status code: 500

HTTP Status Code: 500

### **ResourceNotFoundException**

The specified resource wasn't found.

HTTP status code: 404

HTTP Status Code: 404

### **ThrottlingException**

Too many requests.

HTTP status code: 429

HTTP Status Code: 429

## ValidationException

An error occurred during validation.

HTTP status code: 422

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 V3](#)
- [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

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

An array of filter objects.

Type: Array of [Filter](#) objects

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

Required: No

### MaxResults

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

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

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

Type: [Sort](#) 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

Array of [ChangeSetSummaryListItem](#) objects.

Type: Array of [ChangeSetSummaryListItem](#) objects

### NextToken

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](#).

### AccessDeniedException

Access is denied.

HTTP status code: 403

HTTP Status Code: 403

### InternalServiceException

There was an internal service exception.

HTTP status code: 500

HTTP Status Code: 500

### ThrottlingException

Too many requests.

HTTP status code: 429

HTTP Status Code: 429

### ValidationException

An error occurred during validation.

HTTP status code: 422

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 V3](#)
- [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",
  "EntityTypeFilters": { ... },
  "EntityTypeSort": { ... },
  "FilterList": [
    {
      "Name": "string",
      "ValueList": [ "string" ]
    }
  ],
  "MaxResults": number,
  "NextToken": "string",
  "OwnershipType": "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

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

The type of entities to retrieve. Valid values are: `AmiProduct`, `ContainerProduct`, `DataProduct`, `SaaSProduct`, `ProcurementPolicy`, `Experience`, `Audience`, `BrandingSettings`, `Offer`, `Seller`, `ResaleAuthorization`.

Type: String

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

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

Required: Yes

## EntityTypeFilters

A Union object containing filter shapes for all `EntityTypes`. Each `EntityTypeFilter` shape will have filters applicable for that `EntityType` that can be used to search or filter entities.

Type: [EntityTypeFilters](#) object

**Note:** This object is a Union. Only one member of this object can be specified or returned.

Required: No

## EntityTypeSort

A Union object containing `Sort` shapes for all `EntityTypes`. Each `EntityTypeSort` shape will have `SortBy` and `SortOrder` applicable for fields on that `EntityType`. This can be used to sort the results of the filter query.

Type: [EntityTypeSort](#) object

**Note:** This object is a Union. Only one member of this object can be specified or returned.

Required: No

## FilterList

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

Type: Array of [Filter](#) objects

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

Required: No

## MaxResults

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 50.

Required: No

## NextToken

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

## OwnershipType

Filters the returned set of entities based on their owner. The default is SELF. To list entities shared with you through AWS Resource Access Manager (AWS RAM), set to SHARED. Entities shared through the AWS Marketplace Catalog API `PutResourcePolicy` operation can't be discovered through the SHARED parameter.

Type: String

Valid Values: SELF | SHARED

Required: No

## Sort

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

Type: [Sort](#) object

Required: No

## Response Syntax

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

{
  "EntitySummaryList": [
    {
      "AmiProductSummary": {
        "ProductTitle": "string",
        "Visibility": "string"
      },
      "ContainerProductSummary": {
        "ProductTitle": "string",
        "Visibility": "string"
      },
      "DataProductSummary": {
        "ProductTitle": "string",
        "Visibility": "string"
      },
      "EntityArn": "string",
      "EntityId": "string",
      "EntityType": "string",
      "LastModifiedDate": "string",
      "Name": "string",
      "OfferSummary": {
        "AvailabilityEndDate": "string",
        "BuyerAccounts": [ "string" ],
        "Name": "string",
        "ProductId": "string",
        "ReleaseDate": "string",
        "ResaleAuthorizationId": "string",
        "State": "string",

```

```

    "Targeting": [ "string" ]
  },
  "ResaleAuthorizationSummary": {
    "AvailabilityEndDate": "string",
    "CreateDate": "string",
    "ManufacturerAccountId": "string",
    "ManufacturerLegalName": "string",
    "Name": "string",
    "OfferExtendedStatus": "string",
    "ProductId": "string",
    "ProductName": "string",
    "ResellerAccountID": "string",
    "ResellerLegalName": "string",
    "Status": "string"
  },
  "SaaSProductSummary": {
    "ProductTitle": "string",
    "Visibility": "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

Array of EntitySummary objects.

Type: Array of [EntitySummary](#) objects

### NextToken

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+=.:@-\v]*$`

## Errors

For information about the errors that are common to all actions, see [Common Errors](#).

### AccessDeniedException

Access is denied.

HTTP status code: 403

HTTP Status Code: 403

### InternalServiceException

There was an internal service exception.

HTTP status code: 500

HTTP Status Code: 500

### ResourceNotFoundException

The specified resource wasn't found.

HTTP status code: 404

HTTP Status Code: 404

### ThrottlingException

Too many requests.

HTTP status code: 429

HTTP Status Code: 429

### ValidationException

An error occurred during validation.

HTTP status code: 422

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 V3](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

# ListTagsForResource

Lists all tags that have been added to a resource (either an [entity](#) or [change set](#)).

## Request Syntax

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

```
{
  "ResourceArn": "string"
}
```

## URI Request Parameters

The request does not use any URI parameters.

## Request Body

The request accepts the following data in JSON format.

### [ResourceArn](#)

Required. The Amazon Resource Name (ARN) associated with the resource you want to list tags on.

Type: String

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

Pattern: `^arn:[\w+=/, .@-]+:aws-marketplace:[\w+=/, .@-]*:[0-9]+:[\w+=, .@-]+(\/[\w+=, .@-]+)*$`

Required: Yes

## Response Syntax

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

```
{
  "ResourceArn": "string",
  "Tags": [
    {
      "Key": "string",
      "Value": "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.

### ResourceArn

Required. The ARN associated with the resource you want to list tags on.

Type: String

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

Pattern: `^arn:[\w+=/, .@-]+:aws-marketplace:[\w+=/, .@-]*:[0-9]+:[\w+=, .@-]+(/[\w+=, .@-]+)*$`

### Tags

Required. A list of objects specifying each key name and value. Number of objects allowed: 1-50.

Type: Array of [Tag](#) objects

Array Members: Minimum number of 1 item. Maximum number of 200 items.

## Errors

For information about the errors that are common to all actions, see [Common Errors](#).

### **AccessDeniedException**

Access is denied.

HTTP status code: 403

HTTP Status Code: 403

### **InternalServerErrorException**

There was an internal service exception.

HTTP status code: 500

HTTP Status Code: 500

### **ResourceNotFoundException**

The specified resource wasn't found.

HTTP status code: 404

HTTP Status Code: 404

### **ThrottlingException**

Too many requests.

HTTP status code: 429

HTTP Status Code: 429

### **ValidationException**

An error occurred during validation.

HTTP status code: 422

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 V3](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

# PutResourcePolicy

Attaches a resource-based policy to an entity. Examples of an entity include: AmiProduct and ContainerProduct.

## Request Syntax

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

```
{
  "Policy": "string",
  "ResourceArn": "string"
}
```

## URI Request Parameters

The request does not use any URI parameters.

## Request Body

The request accepts the following data in JSON format.

### Policy

The policy document to set; formatted in JSON.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 10240.

Pattern: `^[^\u0009\u000A\u000D\u0020-\u00FF]+$`

Required: Yes

### ResourceArn

The Amazon Resource Name (ARN) of the entity resource you want to associate with a resource policy.

Type: String

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

Pattern: `^arn:[\w+=/, .@-]+:aws-marketplace:[\w+=/, .@-]*:[0-9]+:[\w+=, .@-]+(#[\w+=, .@-]+)*$`

Required: Yes

## Response Syntax

```
HTTP/1.1 200
```

## Response Elements

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

## Errors

For information about the errors that are common to all actions, see [Common Errors](#).

### AccessDeniedException

Access is denied.

HTTP status code: 403

HTTP Status Code: 403

### InternalServerError

There was an internal service exception.

HTTP status code: 500

HTTP Status Code: 500

### ResourceNotFoundException

The specified resource wasn't found.

HTTP status code: 404

HTTP Status Code: 404

## ThrottlingException

Too many requests.

HTTP status code: 429

HTTP Status Code: 429

## ValidationException

An error occurred during validation.

HTTP status code: 422

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 V3](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

# StartChangeSet

Allows you to request changes for your entities. Within a single ChangeSet, you can't 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 change set has completed (either succeeded, cancelled, or failed). If you try to start a change set containing a change against an entity that is already locked, you will receive a `ResourceInUseException` error.

For example, you can't start the ChangeSet described in the [example](#) later in this topic because it contains two changes to run 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](#). For information about change types for single-AMI products, see [Working with single-AMI products](#). Also, for more information about change types available for container-based products, see [Working with container products](#).

## Request Syntax

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

{
  "Catalog": "string",
  "ChangeSet": [
    {
      "ChangeName": "string",
      "ChangeType": "string",
      "Details": "string",
      "DetailsDocument": JSON value,
      "Entity": {
        "Identifier": "string",
        "Type": "string"
      },
      "EntityTags": [
        {
          "Key": "string",
          "Value": "string"
        }
      ]
    }
  ]
}
```

```
    }
  ],
  "ChangeSetName": "string",
  "ChangeSetTags": [
    {
      "Key": "string",
      "Value": "string"
    }
  ],
  "ClientRequestToken": "string",
  "Intent": "string"
}
```

## URI Request Parameters

The request does not use any URI parameters.

## Request Body

The request accepts the following data in JSON format.

### Catalog

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

Array of change object.

Type: Array of [Change](#) objects

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

Required: Yes

## ChangeSetName

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

## ChangeSetTags

A list of objects specifying each key name and value for the ChangeSetTags property.

Type: Array of [Tag](#) objects

Array Members: Minimum number of 1 item. Maximum number of 200 items.

Required: No

## ClientRequestToken

A unique token to identify the request to ensure idempotency.

Type: String

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

Pattern: `^[!-~]+$`

Required: No

## Intent

The intent related to the request. The default is APPLY. To test your request before applying changes to your entities, use VALIDATE. This feature is currently available for adding versions to single-AMI products. For more information, see [Add a new version](#).

Type: String

Valid Values: VALIDATE | APPLY

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](#)

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](#)

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](#).

## **AccessDeniedException**

Access is denied.

HTTP status code: 403

HTTP Status Code: 403

## **InternalServiceException**

There was an internal service exception.

HTTP status code: 500

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

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

HTTP Status Code: 429

## **ValidationException**

An error occurred during validation.

HTTP status code: 422

HTTP Status Code: 422

## Examples

### Example

You can't start this ChangeSet because it contains two changes to run 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 V3](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

# TagResource

Tags a resource (either an [entity](#) or [change set](#)).

## Request Syntax

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

{
  "ResourceArn": "string",
  "Tags": [
    {
      "Key": "string",
      "Value": "string"
    }
  ]
}
```

## URI Request Parameters

The request does not use any URI parameters.

## Request Body

The request accepts the following data in JSON format.

### [ResourceArn](#)

Required. The Amazon Resource Name (ARN) associated with the resource you want to tag.

Type: String

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

Pattern: `^arn:[\w+=/, .@-]+:aws-marketplace:[\w+=/, .@-]*:[0-9]+:[\w+=, .@-]+(\/[\w+=, .@-]+)*$`

Required: Yes

## Tags

Required. A list of objects specifying each key name and value. Number of objects allowed: 1-50.

Type: Array of [Tag](#) objects

Array Members: Minimum number of 1 item. Maximum number of 200 items.

Required: Yes

## Response Syntax

```
HTTP/1.1 200
```

## Response Elements

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

## Errors

For information about the errors that are common to all actions, see [Common Errors](#).

### AccessDeniedException

Access is denied.

HTTP status code: 403

HTTP Status Code: 403

### InternalServiceException

There was an internal service exception.

HTTP status code: 500

HTTP Status Code: 500

### ResourceNotFoundException

The specified resource wasn't found.

HTTP status code: 404

HTTP Status Code: 404

### **ThrottlingException**

Too many requests.

HTTP status code: 429

HTTP Status Code: 429

### **ValidationException**

An error occurred during validation.

HTTP status code: 422

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 V3](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

# UntagResource

Removes a tag or list of tags from a resource (either an [entity](#) or [change set](#)).

## Request Syntax

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

{
  "ResourceArn": "string",
  "TagKeys": [ "string" ]
}
```

## URI Request Parameters

The request does not use any URI parameters.

## Request Body

The request accepts the following data in JSON format.

### [ResourceArn](#)

Required. The Amazon Resource Name (ARN) associated with the resource you want to remove the tag from.

Type: String

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

Pattern: `^arn:[\w+=/, .@-]+:aws-marketplace:[\w+=/, .@-]*:[0-9]+:[\w+=, .@-]+(\/[\w+=, .@-]+)*$`

Required: Yes

### [TagKeys](#)

Required. A list of key names of tags to be removed. Number of strings allowed: 0-256.

Type: Array of strings

Array Members: Minimum number of 1 item. Maximum number of 200 items.

Length Constraints: Minimum length of 1. Maximum length of 128.

Pattern:  $^([\p{L}\p{Z}\p{N}_\cdot :/=+\-@]^\ast)\$$

Required: Yes

## Response Syntax

```
HTTP/1.1 200
```

## Response Elements

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

## Errors

For information about the errors that are common to all actions, see [Common Errors](#).

### AccessDeniedException

Access is denied.

HTTP status code: 403

HTTP Status Code: 403

### InternalServerError

There was an internal service exception.

HTTP status code: 500

HTTP Status Code: 500

### ResourceNotFoundException

The specified resource wasn't found.

HTTP status code: 404

HTTP Status Code: 404

## ThrottlingException

Too many requests.

HTTP status code: 429

HTTP Status Code: 429

## ValidationException

An error occurred during validation.

HTTP status code: 422

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 V3](#)
- [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:

- [AmiProductEntityIdFilter](#)
- [AmiProductFilters](#)
- [AmiProductLastModifiedDateFilter](#)
- [AmiProductLastModifiedDateFilterDateRange](#)
- [AmiProductSort](#)
- [AmiProductSummary](#)
- [AmiProductTitleFilter](#)
- [AmiProductVisibilityFilter](#)
- [Change](#)
- [ChangeSetSummaryListItem](#)
- [ChangeSummary](#)
- [ContainerProductEntityIdFilter](#)
- [ContainerProductFilters](#)
- [ContainerProductLastModifiedDateFilter](#)
- [ContainerProductLastModifiedDateFilterDateRange](#)
- [ContainerProductSort](#)
- [ContainerProductSummary](#)
- [ContainerProductTitleFilter](#)
- [ContainerProductVisibilityFilter](#)
- [DataProductEntityIdFilter](#)

- [DataProductFilters](#)
- [DataProductLastModifiedDateFilter](#)
- [DataProductLastModifiedDateFilterDateRange](#)
- [DataProductSort](#)
- [DataProductSummary](#)
- [DataProductTitleFilter](#)
- [DataProductVisibilityFilter](#)
- [Entity](#)
- [EntitySummary](#)
- [EntityTypeFilters](#)
- [EntityTypeSort](#)
- [ErrorDetail](#)
- [Filter](#)
- [OfferAvailabilityEndDateFilter](#)
- [OfferAvailabilityEndDateFilterDateRange](#)
- [OfferBuyerAccountsFilter](#)
- [OfferEntityIdFilter](#)
- [OfferFilters](#)
- [OfferLastModifiedDateFilter](#)
- [OfferLastModifiedDateFilterDateRange](#)
- [OfferNameFilter](#)
- [OfferProductIdFilter](#)
- [OfferReleaseDateFilter](#)
- [OfferReleaseDateFilterDateRange](#)
- [OfferResaleAuthorizationIdFilter](#)
- [OfferSort](#)
- [OfferStateFilter](#)
- [OfferSummary](#)
- [OfferTargetingFilter](#)
- [ResaleAuthorizationAvailabilityEndDateFilter](#)

- [ResaleAuthorizationAvailabilityEndDateFilterDateRange](#)
- [ResaleAuthorizationCreatedDateFilter](#)
- [ResaleAuthorizationCreatedDateFilterDateRange](#)
- [ResaleAuthorizationEntityIdFilter](#)
- [ResaleAuthorizationFilters](#)
- [ResaleAuthorizationLastModifiedDateFilter](#)
- [ResaleAuthorizationLastModifiedDateFilterDateRange](#)
- [ResaleAuthorizationManufacturerAccountIdFilter](#)
- [ResaleAuthorizationManufacturerLegalNameFilter](#)
- [ResaleAuthorizationNameFilter](#)
- [ResaleAuthorizationOfferExtendedStatusFilter](#)
- [ResaleAuthorizationProductIdFilter](#)
- [ResaleAuthorizationProductNameFilter](#)
- [ResaleAuthorizationResellerAccountIDFilter](#)
- [ResaleAuthorizationResellerLegalNameFilter](#)
- [ResaleAuthorizationSort](#)
- [ResaleAuthorizationStatusFilter](#)
- [ResaleAuthorizationSummary](#)
- [SaaSProductEntityIdFilter](#)
- [SaaSProductFilters](#)
- [SaaSProductLastModifiedDateFilter](#)
- [SaaSProductLastModifiedDateFilterDateRange](#)
- [SaaSProductSort](#)
- [SaaSProductSummary](#)
- [SaaSProductTitleFilter](#)
- [SaaSProductVisibilityFilter](#)
- [Sort](#)
- [Tag](#)

# AmiProductEntityIdFilter

Object that allows filtering on entity id of an AMI product.

## Contents

### ValueList

A string array of unique entity id values to be filtered on.

Type: Array of strings

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

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

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

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](#)

# AmiProductFilters

Object containing all the filter fields for AMI products. Client can add only one wildcard filter and a maximum of 8 filters in a single `ListEntities` request.

## Contents

### EntityId

Unique identifier for the AMI product.

Type: [AmiProductEntityIdFilter](#) object

Required: No

### LastModifiedDate

The last date on which the AMI product was modified.

Type: [AmiProductLastModifiedDateFilter](#) object

Required: No

### ProductTitle

The title of the AMI product.

Type: [AmiProductTitleFilter](#) object

Required: No

### Visibility

The visibility of the AMI product.

Type: [AmiProductVisibilityFilter](#) object

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](#)

# AmiProductLastModifiedDateFilter

Object that allows filtering based on the last modified date of AMI products.

## Contents

### DateRange

Dates between which the AMI product was last modified.

Type: [AmiProductLastModifiedDateFilterDateRange](#) object

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](#)

# AmiProductLastModifiedDateFilterDateRange

Object that contains date range of the last modified date to be filtered on. You can optionally provide a `BeforeValue` and/or `AfterValue`. Both are inclusive.

## Contents

### AfterValue

Date after which the AMI product was last modified.

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

### BeforeValue

Date before which the AMI product was last modified.

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

## 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](#)

# AmiProductSort

Objects that allows sorting on AMI products based on certain fields and sorting order.

## Contents

### SortBy

Field to sort the AMI products by.

Type: String

Valid Values: EntityId | LastModifiedDate | ProductTitle | Visibility

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](#)

# AmiProductSummary

Object that contains summarized information about an AMI product.

## Contents

### ProductTitle

The title of the AMI product.

Type: String

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

Pattern: `^(.)+$`

Required: No

### Visibility

The lifecycle of the AMI product.

Type: String

Valid Values: `Limited` | `Public` | `Restricted` | `Draft`

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](#)

# AmiProductTitleFilter

Object that allows filtering on product title.

## Contents

### ValueList

A string array of unique product title values to be filtered on.

Type: Array of strings

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

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

Pattern: `^(.)+$`

Required: No

### WildcardValue

A string that will be the `wildCard` input for product tile filter. It matches the provided value as a substring in the actual value.

Type: String

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

Pattern: `^(.)+$`

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](#)

# AmiProductVisibilityFilter

Object that allows filtering on the visibility of the product in the AWS Marketplace.

## Contents

### ValueList

A string array of unique visibility values to be filtered on.

Type: Array of strings

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

Valid Values: Limited | Public | Restricted | Draft

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](#)

# Change

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

## Contents

### 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. For more information about change types available for single-AMI products, see [Working with single-AMI products](#). Also, for more information about change types available for container-based products, see [Working with container products](#).

Type: String

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

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

Required: Yes

### Entity

The entity to be changed.

Type: [Entity](#) object

Required: Yes

### ChangeName

Optional name for the change.

Type: String

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

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

Required: No

### Details

This object contains details specific to the change type of the requested change. For more information about change types available for single-AMI products, see [Working with single-](#)

[AMI products](#). Also, for more information about change types available for container-based products, see [Working with container products](#).

Type: String

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

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

Required: No

## DetailsDocument

Alternative field that accepts a JSON value instead of a string for ChangeType details. You can use either `Details` or `DetailsDocument`, but not both.

Type: JSON value

Required: No

## EntityTags

The tags associated with the change.

Type: Array of [Tag](#) objects

Array Members: Minimum number of 1 item. Maximum number of 200 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](#)

# 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

### DetailsDocument

The JSON value of the details specific to the change type of the requested change.

Type: JSON value

Required: No

### Entity

The entity to be changed.

Type: [Entity](#) object

Required: No

### ErrorDetailList

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

Type: Array of [ErrorDetail](#) 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](#)

# ContainerProductEntityIdFilter

Object that allows filtering on entity id of a container product.

## Contents

### ValueList

A string array of unique entity id values to be filtered on.

Type: Array of strings

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

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

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

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](#)

# ContainerProductFilters

Object containing all the filter fields for container products. Client can add only one wildcard filter and a maximum of 8 filters in a single `ListEntities` request.

## Contents

### EntityId

Unique identifier for the container product.

Type: [ContainerProductEntityIdFilter](#) object

Required: No

### LastModifiedDate

The last date on which the container product was modified.

Type: [ContainerProductLastModifiedDateFilter](#) object

Required: No

### ProductTitle

The title of the container product.

Type: [ContainerProductTitleFilter](#) object

Required: No

### Visibility

The visibility of the container product.

Type: [ContainerProductVisibilityFilter](#) object

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](#)

# ContainerProductLastModifiedDateFilter

Object that allows filtering based on the last modified date of container products.

## Contents

### DateRange

Dates between which the container product was last modified.

Type: [ContainerProductLastModifiedDateFilterDateRange](#) object

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](#)

# ContainerProductLastModifiedDateFilterDateRange

Object that contains date range of the last modified date to be filtered on. You can optionally provide a `BeforeValue` and/or `AfterValue`. Both are inclusive.

## Contents

### AfterValue

Date after which the container product was last modified.

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

### BeforeValue

Date before which the container product was last modified.

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

## 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](#)

# ContainerProductSort

Objects that allows sorting on container products based on certain fields and sorting order.

## Contents

### SortBy

Field to sort the container products by.

Type: String

Valid Values: EntityId | LastModifiedDate | ProductTitle | Visibility

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](#)

# ContainerProductSummary

Object that contains summarized information about a container product.

## Contents

### ProductTitle

The title of the container product.

Type: String

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

Pattern: `^(.)+$`

Required: No

### Visibility

The lifecycle of the product.

Type: String

Valid Values: `Limited` | `Public` | `Restricted` | `Draft`

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](#)

# ContainerProductTitleFilter

Object that allows filtering on product title.

## Contents

### ValueList

A string array of unique product title values to be filtered on.

Type: Array of strings

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

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

Pattern: `^(.)+$`

Required: No

### WildcardValue

A string that will be the `wildCard` input for product tile filter. It matches the provided value as a substring in the actual value.

Type: String

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

Pattern: `^(.)+$`

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](#)

# ContainerProductVisibilityFilter

Object that allows filtering on the visibility of the product in the AWS Marketplace.

## Contents

### ValueList

A string array of unique visibility values to be filtered on.

Type: Array of strings

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

Valid Values: Limited | Public | Restricted | Draft

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](#)

# DataProductEntityIdFilter

Object that allows filtering on entity id of a data product.

## Contents

### ValueList

A string array of unique entity id values to be filtered on.

Type: Array of strings

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

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

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

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](#)

# DataProductFilters

Object containing all the filter fields for data products. Client can add only one wildcard filter and a maximum of 8 filters in a single `ListEntities` request.

## Contents

### EntityId

Unique identifier for the data product.

Type: [DataProductEntityIdFilter](#) object

Required: No

### LastModifiedDate

The last date on which the data product was modified.

Type: [DataProductLastModifiedDateFilter](#) object

Required: No

### ProductTitle

The title of the data product.

Type: [DataProductTitleFilter](#) object

Required: No

### Visibility

The visibility of the data product.

Type: [DataProductVisibilityFilter](#) object

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](#)

# DataProductLastModifiedDateFilter

Object that allows filtering based on the last modified date of data products.

## Contents

### DateRange

Dates between which the data product was last modified.

Type: [DataProductLastModifiedDateFilterDateRange](#) object

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](#)

# DataProductLastModifiedDateFilterDateRange

Object that contains date range of the last modified date to be filtered on. You can optionally provide a `BeforeValue` and/or `AfterValue`. Both are inclusive.

## Contents

### AfterValue

Date after which the data product was last modified.

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

### BeforeValue

Date before which the data product was last modified.

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

## 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](#)

# DataProductSort

Objects that allows sorting on data products based on certain fields and sorting order.

## Contents

### SortBy

Field to sort the data products by.

Type: String

Valid Values: EntityId | ProductTitle | Visibility | LastModifiedDate

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](#)

# DataProductSummary

Object that contains summarized information about a data product.

## Contents

### ProductTitle

The title of the data product.

Type: String

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

Pattern: `^(.)+$`

Required: No

### Visibility

The lifecycle of the data product.

Type: String

Valid Values: `Limited` | `Public` | `Restricted` | `Unavailable` | `Draft`

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](#)

# DataProductTitleFilter

Object that allows filtering on product title.

## Contents

### ValueList

A string array of unique product title values to be filtered on.

Type: Array of strings

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

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

Pattern:  $^(.)+\$$

Required: No

### WildcardValue

A string that will be the wildcard input for product tile filter. It matches the provided value as a substring in the actual value.

Type: String

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

Pattern:  $^(.)+\$$

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](#)

# DataProductVisibilityFilter

Object that allows filtering on the visibility of the product in the AWS Marketplace.

## Contents

### ValueList

A string array of unique visibility values to be filtered on.

Type: Array of strings

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

Valid Values: Limited | Public | Restricted | Unavailable | Draft

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

### Type

The type of entity.

Type: String

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

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

Required: Yes

### Identifier

The identifier for the entity.

Type: String

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

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

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](#)



# 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

### AmiProductSummary

An object that contains summary information about the AMI product.

Type: [AmiProductSummary](#) object

Required: No

### ContainerProductSummary

An object that contains summary information about the container product.

Type: [ContainerProductSummary](#) object

Required: No

### DataProductSummary

An object that contains summary information about the data product.

Type: [DataProductSummary](#) object

Required: No

### 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

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

## Name

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

Type: String

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

Pattern: `^\\S+[\\S\\s]*`

Required: No

## OfferSummary

An object that contains summary information about the offer.

Type: [OfferSummary](#) object

Required: No

## ResaleAuthorizationSummary

An object that contains summary information about the Resale Authorization.

Type: [ResaleAuthorizationSummary](#) object

Required: No

## SaaSProductSummary

An object that contains summary information about the SaaS product.

Type: [SaaSProductSummary](#) object

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

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

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

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](#)

# EntityTypeFilters

Object containing all the filter fields per entity type.

## Contents

### Important

This data type is a UNION, so only one of the following members can be specified when used or returned.

### AmiProductFilters

A filter for AMI products.

Type: [AmiProductFilters](#) object

Required: No

### ContainerProductFilters

A filter for container products.

Type: [ContainerProductFilters](#) object

Required: No

### DataProductFilters

A filter for data products.

Type: [DataProductFilters](#) object

Required: No

### OfferFilters

A filter for offers.

Type: [OfferFilters](#) object

Required: No

## ResaleAuthorizationFilters

A filter for Resale Authorizations.

Type: [ResaleAuthorizationFilters](#) object

Required: No

## SaaSProductFilters

A filter for SaaS products.

Type: [SaaSProductFilters](#) object

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](#)

# EntityTypeSort

Object containing all the sort fields per entity type.

## Contents

### Important

This data type is a UNION, so only one of the following members can be specified when used or returned.

### AmiProductSort

A sort for AMI products.

Type: [AmiProductSort](#) object

Required: No

### ContainerProductSort

A sort for container products.

Type: [ContainerProductSort](#) object

Required: No

### DataProductSort

A sort for data products.

Type: [DataProductSort](#) object

Required: No

### OfferSort

A sort for offers.

Type: [OfferSort](#) object

Required: No

## ResaleAuthorizationSort

A sort for Resale Authorizations.

Type: [ResaleAuthorizationSort](#) object

Required: No

## SaaSProductSort

A sort for SaaS products.

Type: [SaaSProductSort](#) object

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

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

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

Required: No

### ErrorMessage

The message for the error.

Type: String

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

Pattern: `^(.)+$`

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.

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

Pattern:  $^(.)+\$$

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](#)

# OfferAvailabilityEndDateFilter

Allows filtering on the AvailabilityEndDate of an offer.

## Contents

### DateRange

Allows filtering on the AvailabilityEndDate of an offer with date range as input.

Type: [OfferAvailabilityEndDateFilterDateRange](#) object

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](#)

# OfferAvailabilityEndDateFilterDateRange

Allows filtering on the AvailabilityEndDate of an offer with date range as input.

## Contents

### AfterValue

Allows filtering on the AvailabilityEndDate of an offer after a date.

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

### BeforeValue

Allows filtering on the AvailabilityEndDate of an offer before a date.

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

## 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](#)



# OfferBuyerAccountsFilter

Allows filtering on the BuyerAccounts of an offer.

## Contents

### WildcardValue

Allows filtering on the BuyerAccounts of an offer with wild card input.

Type: String

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

Pattern: `^(.)+$`

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](#)

# OfferEntityIdFilter

Allows filtering on the entity id of an offer.

## Contents

### ValueList

Allows filtering on entity id of an offer with list input.

Type: Array of strings

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

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

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

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](#)

# OfferFilters

Object containing all the filter fields for offers entity. Client can add only one wildcard filter and a maximum of 8 filters in a single `ListEntities` request.

## Contents

### AvailabilityEndDate

Allows filtering on the `AvailabilityEndDate` of an offer.

Type: [OfferAvailabilityEndDateFilter](#) object

Required: No

### BuyerAccounts

Allows filtering on the `BuyerAccounts` of an offer.

Type: [OfferBuyerAccountsFilter](#) object

Required: No

### EntityId

Allows filtering on `EntityId` of an offer.

Type: [OfferEntityIdFilter](#) object

Required: No

### LastModifiedDate

Allows filtering on the `LastModifiedDate` of an offer.

Type: [OfferLastModifiedDateFilter](#) object

Required: No

### Name

Allows filtering on the `Name` of an offer.

Type: [OfferNameFilter](#) object

Required: No

## ProductId

Allows filtering on the ProductId of an offer.

Type: [OfferProductIdFilter](#) object

Required: No

## ReleaseDate

Allows filtering on the ReleaseDate of an offer.

Type: [OfferReleaseDateFilter](#) object

Required: No

## ResaleAuthorizationId

Allows filtering on the ResaleAuthorizationId of an offer.

### Note

Not all offers have a ResaleAuthorizationId. The response will only include offers for which you have permissions.

Type: [OfferResaleAuthorizationIdFilter](#) object

Required: No

## State

Allows filtering on the State of an offer.

Type: [OfferStateFilter](#) object

Required: No

## Targeting

Allows filtering on the Targeting of an offer.

Type: [OfferTargetingFilter](#) object

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](#)

# OfferLastModifiedDateFilter

Allows filtering on the LastModifiedDate of an offer.

## Contents

### DateRange

Allows filtering on the LastModifiedDate of an offer with date range as input.

Type: [OfferLastModifiedDateFilterDateRange](#) object

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](#)

# OfferLastModifiedDateFilterDateRange

Allows filtering on the LastModifiedDate of an offer with date range as input.

## Contents

### AfterValue

Allows filtering on the LastModifiedDate of an offer after a date.

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

### BeforeValue

Allows filtering on the LastModifiedDate of an offer before a date.

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

## 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](#)



# OfferNameFilter

Allows filtering on the Name of an offer.

## Contents

### ValueList

Allows filtering on the Name of an offer with list input.

Type: Array of strings

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

Length Constraints: Minimum length of 1. Maximum length of 150.

Pattern: `^(.)+$`

Required: No

### WildcardValue

Allows filtering on the Name of an offer with wild card input.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 150.

Pattern: `^(.)+$`

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](#)



# OfferProductIdFilter

Allows filtering on the ProductId of an offer.

## Contents

### ValueList

Allows filtering on the ProductId of an offer with list input.

Type: Array of strings

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

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

Pattern:  $^(.)+\$$

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](#)

# OfferReleaseDateFilter

Allows filtering on the ReleaseDate of an offer.

## Contents

### DateRange

Allows filtering on the ReleaseDate of an offer with date range as input.

Type: [OfferReleaseDateFilterDateRange](#) object

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](#)

# OfferReleaseDateFilterDateRange

Allows filtering on the ReleaseDate of an offer with date range as input.

## Contents

### AfterValue

Allows filtering on the ReleaseDate of offers after a date.

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

### BeforeValue

Allows filtering on the ReleaseDate of offers before a date.

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

## 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](#)



# OfferResaleAuthorizationIdFilter

Allows filtering on the ResaleAuthorizationId of an offer.

## Note

Not all offers have a ResaleAuthorizationId. The response will only include offers for which you have permissions.

## Contents

### ValueList

Allows filtering on the ResaleAuthorizationId of an offer with list input.

Type: Array of strings

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

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

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

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](#)

# OfferSort

Allows to sort offers.

## Contents

### SortBy

Allows to sort offers.

Type: String

Valid Values: EntityId | Name | ProductId | ResaleAuthorizationId | ReleaseDate | AvailabilityEndDate | BuyerAccounts | State | Targeting | LastModifiedDate

Required: No

### SortOrder

Allows to sort offers.

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](#)

# OfferStateFilter

Allows filtering on the State of an offer.

## Contents

### ValueList

Allows filtering on the State of an offer with list input.

Type: Array of strings

Array Members: Minimum number of 1 item. Maximum number of 2 items.

Valid Values: Draft | Released

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](#)

# OfferSummary

Summarized information about an offer.

## Contents

### AvailabilityEndDate

The availability end date of the offer.

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

### BuyerAccounts

The buyer accounts in the offer.

Type: Array of strings

Array Members: Minimum number of 0 items. Maximum number of 26 items.

Length Constraints: Fixed length of 12.

Pattern: `^\d{12}$`

Required: No

### Name

The name of the offer.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 150.

Pattern: `^(.)+$`

Required: No

## ProductId

The product ID of the offer.

Type: String

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

Pattern: `^(.)+$`

Required: No

## ReleaseDate

The release date of the offer.

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

## ResaleAuthorizationId

The ResaleAuthorizationId of the offer.

Type: String

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

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

Required: No

## State

The status of the offer.

Type: String

Valid Values: Draft | Released

Required: No

## Targeting

The targeting in the offer.

Type: Array of strings

Array Members: Minimum number of 0 items. Maximum number of 4 items.

Valid Values: BuyerAccounts | ParticipatingPrograms | CountryCodes | None

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](#)

# OfferTargetingFilter

Allows filtering on the Targeting of an offer.

## Contents

### ValueList

Allows filtering on the Targeting of an offer with list input.

Type: Array of strings

Array Members: Minimum number of 1 item. Maximum number of 4 items.

Valid Values: BuyerAccounts | ParticipatingPrograms | CountryCodes | None

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](#)

# ResaleAuthorizationAvailabilityEndDateFilter

Allows filtering on AvailabilityEndDate of a ResaleAuthorization.

## Contents

### DateRange

Allows filtering on AvailabilityEndDate of a ResaleAuthorization with date range as input

Type: [ResaleAuthorizationAvailabilityEndDateFilterDateRange](#) object

Required: No

### ValueList

Allows filtering on AvailabilityEndDate of a ResaleAuthorization with date value as input.

Type: Array of strings

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

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

## 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](#)

# ResaleAuthorizationAvailabilityEndDateFilterDateRange

Allows filtering on AvailabilityEndDate of a ResaleAuthorization with date range as input.

## Contents

### AfterValue

Allows filtering on AvailabilityEndDate of a ResaleAuthorization after a date.

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

### BeforeValue

Allows filtering on AvailabilityEndDate of a ResaleAuthorization before a date.

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

## 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](#)



# ResaleAuthorizationCreatedDateFilter

Allows filtering on CreatedDate of a ResaleAuthorization.

## Contents

### DateRange

Allows filtering on CreatedDate of a ResaleAuthorization with date range as input.

Type: [ResaleAuthorizationCreatedDateFilterDateRange](#) object

Required: No

### ValueList

Allows filtering on CreatedDate of a ResaleAuthorization with date value as input.

Type: Array of strings

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

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

## 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](#)

# ResaleAuthorizationCreatedDateFilterDateRange

Allows filtering on CreatedDate of a ResaleAuthorization with date range as input.

## Contents

### AfterValue

Allows filtering on CreatedDate of a ResaleAuthorization after a date.

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

### BeforeValue

Allows filtering on CreatedDate of a ResaleAuthorization before a date.

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

## 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](#)



# ResaleAuthorizationEntityIdFilter

Allows filtering on EntityId of a ResaleAuthorization.

## Contents

### ValueList

Allows filtering on EntityId of a ResaleAuthorization with list input.

Type: Array of strings

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

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

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

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](#)

# ResaleAuthorizationFilters

Object containing all the filter fields for resale authorization entity. Client can add only one wildcard filter and a maximum of 8 filters in a single `ListEntities` request.

## Contents

### AvailabilityEndDate

Allows filtering on the `AvailabilityEndDate` of a `ResaleAuthorization`.

Type: [ResaleAuthorizationAvailabilityEndDateFilter](#) object

Required: No

### CreatedDate

Allows filtering on the `CreatedDate` of a `ResaleAuthorization`.

Type: [ResaleAuthorizationCreatedDateFilter](#) object

Required: No

### EntityId

Allows filtering on the `EntityId` of a `ResaleAuthorization`.

Type: [ResaleAuthorizationEntityIdFilter](#) object

Required: No

### LastModifiedDate

Allows filtering on the `LastModifiedDate` of a `ResaleAuthorization`.

Type: [ResaleAuthorizationLastModifiedDateFilter](#) object

Required: No

### ManufacturerAccountId

Allows filtering on the `ManufacturerAccountId` of a `ResaleAuthorization`.

Type: [ResaleAuthorizationManufacturerAccountIdFilter](#) object

Required: No

## ManufacturerLegalName

Allows filtering on the `ManufacturerLegalName` of a `ResaleAuthorization`.

Type: [ResaleAuthorizationManufacturerLegalNameFilter](#) object

Required: No

## Name

Allows filtering on the `Name` of a `ResaleAuthorization`.

Type: [ResaleAuthorizationNameFilter](#) object

Required: No

## OfferExtendedStatus

Allows filtering on the `OfferExtendedStatus` of a `ResaleAuthorization`.

Type: [ResaleAuthorizationOfferExtendedStatusFilter](#) object

Required: No

## ProductId

Allows filtering on the `ProductId` of a `ResaleAuthorization`.

Type: [ResaleAuthorizationProductIdFilter](#) object

Required: No

## ProductName

Allows filtering on the `ProductName` of a `ResaleAuthorization`.

Type: [ResaleAuthorizationProductNameFilter](#) object

Required: No

## ResellerAccountID

Allows filtering on the `ResellerAccountID` of a `ResaleAuthorization`.

Type: [ResaleAuthorizationResellerAccountIDFilter](#) object

Required: No

## ResellerLegalName

Allows filtering on the ResellerLegalName of a ResaleAuthorization.

Type: [ResaleAuthorizationResellerLegalNameFilter](#) object

Required: No

## Status

Allows filtering on the Status of a ResaleAuthorization.

Type: [ResaleAuthorizationStatusFilter](#) object

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](#)

# ResaleAuthorizationLastModifiedDateFilter

Allows filtering on the LastModifiedDate of a ResaleAuthorization.

## Contents

### DateRange

Allows filtering on the LastModifiedDate of a ResaleAuthorization with date range as input.

Type: [ResaleAuthorizationLastModifiedDateFilterDateRange](#) object

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](#)

# ResaleAuthorizationLastModifiedDateFilterDateRange

Allows filtering on the LastModifiedDate of a ResaleAuthorization with date range as input.

## Contents

### AfterValue

Allows filtering on the LastModifiedDate of a ResaleAuthorization after a date.

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

### BeforeValue

Allows filtering on the LastModifiedDate of a ResaleAuthorization before a date.

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

## 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](#)



# ResaleAuthorizationManufacturerAccountIdFilter

Allows filtering on the `ManufacturerAccountId` of a `ResaleAuthorization`.

## Contents

### ValueList

Allows filtering on the `ManufacturerAccountId` of a `ResaleAuthorization` with list input.

Type: Array of strings

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

Length Constraints: Fixed length of 12.

Pattern: `^\d{12}$`

Required: No

### WildcardValue

Allows filtering on the `ManufacturerAccountId` of a `ResaleAuthorization` with wild card input.

Type: String

Length Constraints: Fixed length of 12.

Pattern: `^\d{12}$`

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](#)

# ResaleAuthorizationManufacturerLegalNameFilter

Allows filtering on the `ManufacturerLegalName` of a `ResaleAuthorization`.

## Contents

### ValueList

Allows filtering on the `ManufacturerLegalName` of a `ResaleAuthorization` with list input.

Type: Array of strings

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

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

Pattern: `^(.)+$`

Required: No

### WildcardValue

Allows filtering on the `ManufacturerLegalName` of a `ResaleAuthorization` with wild card input.

Type: String

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

Pattern: `^(.)+$`

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](#)

# ResaleAuthorizationNameFilter

Allows filtering on the Name of a ResaleAuthorization.

## Contents

### ValueList

Allows filtering on the Name of a ResaleAuthorization with list input.

Type: Array of strings

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

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

Pattern: `^(.)+$`

Required: No

### WildcardValue

Allows filtering on the Name of a ResaleAuthorization with wild card input.

Type: String

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

Pattern: `^(.)+$`

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](#)



# ResaleAuthorizationOfferExtendedStatusFilter

Allows filtering on the OfferExtendedStatus of a ResaleAuthorization.

## Contents

### ValueList

Allows filtering on the OfferExtendedStatus of a ResaleAuthorization with list input.

Type: Array of strings

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

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

Pattern:  $^(.)+\$$

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](#)

# ResaleAuthorizationProductIdFilter

Allows filtering on the ProductId of a ResaleAuthorization.

## Contents

### ValueList

Allows filtering on the ProductId of a ResaleAuthorization with list input.

Type: Array of strings

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

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

Pattern:  $^(.)+\$$

Required: No

### WildcardValue

Allows filtering on the ProductId of a ResaleAuthorization with wild card input.

Type: String

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

Pattern:  $^(.)+\$$

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](#)



# ResaleAuthorizationProductNameFilter

Allows filtering on the ProductName of a ResaleAuthorization.

## Contents

### ValueList

Allows filtering on the ProductName of a ResaleAuthorization with list input.

Type: Array of strings

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

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

Pattern:  $^(.)+\$$

Required: No

### WildcardValue

Allows filtering on the ProductName of a ResaleAuthorization with wild card input.

Type: String

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

Pattern:  $^(.)+\$$

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](#)



# ResaleAuthorizationResellerAccountIDFilter

Allows filtering on the ResellerAccountID of a ResaleAuthorization.

## Contents

### ValueList

Allows filtering on the ResellerAccountID of a ResaleAuthorization with list input.

Type: Array of strings

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

Length Constraints: Fixed length of 12.

Pattern: `^\d{12}$`

Required: No

### WildcardValue

Allows filtering on the ResellerAccountID of a ResaleAuthorization with wild card input.

Type: String

Length Constraints: Fixed length of 12.

Pattern: `^\d{12}$`

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](#)



# ResaleAuthorizationResellerLegalNameFilter

Allows filtering on the ResellerLegalName of a ResaleAuthorization.

## Contents

### ValueList

Allows filtering on the ResellerLegalNameProductName of a ResaleAuthorization with list input.

Type: Array of strings

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

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

Pattern: `^(.)+$`

Required: No

### WildcardValue

Allows filtering on the ResellerLegalName of a ResaleAuthorization with wild card input.

Type: String

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

Pattern: `^(.)+$`

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](#)



# ResaleAuthorizationSort

Allows to sort ResaleAuthorization.

## Contents

### SortBy

Allows to sort ResaleAuthorization.

Type: String

Valid Values: EntityId | Name | ProductId | ProductName |  
ManufacturerAccountID | ManufacturerLegalName | ResellerAccountID  
| ResellerLegalName | Status | OfferExtendedStatus | CreatedDate |  
AvailabilityEndDate | LastModifiedDate

Required: No

### SortOrder

Allows to sort ResaleAuthorization.

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](#)

# ResaleAuthorizationStatusFilter

Allows filtering on the Status of a ResaleAuthorization.

## Contents

### ValueList

Allows filtering on the Status of a ResaleAuthorization with list input.

Type: Array of strings

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

Valid Values: Draft | Active | Restricted

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](#)

# ResaleAuthorizationSummary

Summarized information about a Resale Authorization.

## Contents

### AvailabilityEndDate

The availability end date of the ResaleAuthorization.

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

### CreatedDate

The created date of the ResaleAuthorization.

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

### ManufacturerAccountId

The manufacturer account ID of the ResaleAuthorization.

Type: String

Length Constraints: Fixed length of 12.

Pattern: `^\d{12}$`

Required: No

## **ManufacturerLegalName**

The manufacturer legal name of the ResaleAuthorization.

Type: String

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

Pattern:  $^(.)+\$$

Required: No

## **Name**

The name of the ResaleAuthorization.

Type: String

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

Pattern:  $^(.)+\$$

Required: No

## **OfferExtendedStatus**

The offer extended status of the ResaleAuthorization

Type: String

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

Pattern:  $^(.)+\$$

Required: No

## **ProductId**

The product ID of the ResaleAuthorization.

Type: String

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

Pattern:  $^(.)+\$$

Required: No

### **ProductName**

The product name of the ResaleAuthorization.

Type: String

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

Pattern: `^(.)+$`

Required: No

### **ResellerAccountID**

The reseller account ID of the ResaleAuthorization.

Type: String

Length Constraints: Fixed length of 12.

Pattern: `^\d{12}$`

Required: No

### **ResellerLegalName**

The reseller legal name of the ResaleAuthorization

Type: String

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

Pattern: `^(.)+$`

Required: No

### **Status**

The status of the ResaleAuthorization.

Type: String

Valid Values: Draft | Active | Restricted

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](#)

# SaaSProductEntityIdFilter

Object that allows filtering on entity id of a SaaS product.

## Contents

### ValueList

A string array of unique entity id values to be filtered on.

Type: Array of strings

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

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

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

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](#)

# SaaSProductFilters

Object containing all the filter fields for SaaS products. Client can add only one wildcard filter and a maximum of 8 filters in a single `ListEntities` request.

## Contents

### EntityId

Unique identifier for the SaaS product.

Type: [SaaSProductEntityIdFilter](#) object

Required: No

### LastModifiedDate

The last date on which the SaaS product was modified.

Type: [SaaSProductLastModifiedDateFilter](#) object

Required: No

### ProductTitle

The title of the SaaS product.

Type: [SaaSProductTitleFilter](#) object

Required: No

### Visibility

The visibility of the SaaS product.

Type: [SaaSProductVisibilityFilter](#) object

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](#)

# SaaSProductLastModifiedDateFilter

Object that allows filtering based on the last modified date of SaaS products

## Contents

### DateRange

Dates between which the SaaS product was last modified.

Type: [SaaSProductLastModifiedDateFilterDateRange](#) object

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](#)

# SaaSProductLastModifiedDateFilterDateRange

Object that contains date range of the last modified date to be filtered on. You can optionally provide a `BeforeValue` and/or `AfterValue`. Both are inclusive.

## Contents

### AfterValue

Date after which the SaaS product was last modified.

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

### BeforeValue

Date before which the SaaS product was last modified.

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

## 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](#)

# SaaSProductSort

Objects that allows sorting on SaaS products based on certain fields and sorting order.

## Contents

### SortBy

Field to sort the SaaS products by.

Type: String

Valid Values: EntityId | ProductTitle | Visibility | LastModifiedDate

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](#)

# SaaSProductSummary

Object that contains summarized information about a SaaS product.

## Contents

### ProductTitle

The title of the SaaS product.

Type: String

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

Pattern: `^(.)+$`

Required: No

### Visibility

The lifecycle of the SaaS product.

Type: String

Valid Values: `Limited` | `Public` | `Restricted` | `Draft`

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](#)

# SaaSProductTitleFilter

Object that allows filtering on product title.

## Contents

### ValueList

A string array of unique product title values to be filtered on.

Type: Array of strings

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

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

Pattern:  $^(.)+\$$

Required: No

### WildcardValue

A string that will be the wildcard input for product tile filter. It matches the provided value as a substring in the actual value.

Type: String

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

Pattern:  $^(.)+\$$

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](#)

# SaaSProductVisibilityFilter

Object that allows filtering on the visibility of the product in the AWS Marketplace.

## Contents

### ValueList

A string array of unique visibility values to be filtered on.

Type: Array of strings

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

Valid Values: Limited | Public | Restricted | Draft

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) and `EntityId`. In addition to `LastModifiedDate` and `EntityId`, each `EntityType` might support additional fields.

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](#)



# Tag

A list of objects specifying each key name and value.

## Contents

### Key

The key associated with the tag.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 128.

Pattern:  $^([\p{L}\p{Z}\p{N}_\p{-}:/=+\p{-}@]^*)\$$

Required: Yes

### Value

The value associated with the tag.

Type: String

Length Constraints: Minimum length of 0. Maximum length of 256.

Pattern:  $^([\p{L}\p{Z}\p{N}_\p{-}:/=+\p{-}@]^*)\$$

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](#)

# 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 [Signing AWS API requests](#) in the *IAM User Guide*.

## 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 [Create a signed AWS API request](#) in the *IAM User Guide*.

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 [Elements of an AWS API request signature](#) in the *IAM User Guide*.

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 STS, see [AWS services that work with IAM](#) in the *IAM User Guide*.

Condition: If you're using temporary security credentials from AWS STS, 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 [Create a signed AWS API request](#) in the *IAM User Guide*.

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

## **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*.

Change	Description	Date
<a href="#">AWS Marketplace Catalog API topic updates</a>	Added notes to <a href="#">Working with offers</a> to clarify constraints.	April 17, 2024
<a href="#">AWS Marketplace Catalog API supports service-linked for resale authorization</a>	Updated resale authorization prerequisites for service-linked role.	March 20, 2024
<a href="#">AWS Marketplace Catalog API supports organization units</a>	Added content to enable private marketplace support at Organization Unit (OU) level.	February 16, 2024
<a href="#">AWS Marketplace Catalog API supports setting intent on requests</a>	Sellers now can request changes for entities with specific intent using the AWS Marketplace Catalog API.	February 9, 2024
<a href="#">AWS Marketplace Catalog API supports wildcard filter validation</a>	Added wildcard filter validation in ListEntities API.	February 5, 2024
<a href="#">AWS Marketplace Catalog API supports Amazon EKS add-ons</a>	Added content and error messages related to publishing to Amazon EKS add-ons from AWS Marketplace container-based product.	January 29, 2024
<a href="#">AWS Marketplace Catalog API supports listing details about entities</a>	Sellers can now list details about entities using the AWS Marketplace Catalog API.	December 19, 2023

<a href="#">AWS Marketplace Catalog API supports the ability to create products, offers, Resale Authorizations, and CPPOs</a>	Sellers can now use the AWS Marketplace Catalog API to create and update <a href="#">products</a> , <a href="#">offers</a> , <a href="#">Resale Authorizations</a> , and <a href="#">channel partner private offers (CPPOs)</a> .	November 29, 2023
<a href="#">AWS Marketplace Catalog API supports enhanced filtering and sorting capabilities</a>	Sellers can now sort and filter products using the AWS Marketplace Catalog API.	November 29, 2023
<a href="#">AWS Marketplace Catalog API supports resource sharing</a>	The AWS Marketplace Catalog API integrates with AWS Resource Access Manager (AWS RAM) to enable resource sharing. See <a href="#">Working with AWS RAM to share resources</a> .	April 12, 2023
<a href="#">AWS Marketplace Discovery API topic update</a>	The AWS Marketplace Discovery API now supports CloudTrail. See <a href="#">Logging AWS Marketplace Discovery API calls using AWS CloudTrail</a> .	December 15, 2022
<a href="#">AWS Marketplace supports archiving private marketplace experiences</a>	Buyers can now archive and reactivate private marketplace experiences in AWS Marketplace. See <a href="#">Working with a private marketplace</a> .	December 12, 2022
<a href="#">AWS Marketplace Private marketplace granular permissions</a>	Buyers now have more granular permissions to manage private marketplace experiences. See <a href="#">Working with a private marketplace</a> .	September 8, 2022

---

<a href="#">AWS Marketplace Discovery API release notes</a>	Added <a href="#">Release notes</a> for the AWS Marketplace Discovery API.	May 20, 2022
<a href="#">AWS Marketplace Discovery API topic update</a>	Documentation-only update to the <a href="#">AWS Marketplace Discovery API topic</a> .	January 14, 2022
<a href="#">Support for Helm chart delivery options and QuickLaunch for container-based products</a>	Added documentation for adding or updating Helm chart delivery options in container-based product versions, including enabling QuickLaunch for buyers. See <a href="#">Working with container-based products</a> .	November 29, 2021
<a href="#">Support for managing seller products</a>	Added the ability to manage AMI and container products programmatically. See <a href="#">Working with seller products</a> .	March 26, 2021
<a href="#">Support for managing private marketplaces</a>	Added the ability to create and maintain private marketplaces for AWS Organizations programmatically. See <a href="#">Working with a private marketplace</a> .	December 3, 2020
<a href="#">The AWS Marketplace Discovery API is now available</a>	The Discovery API provides programmatic access to find products in the AWS Marketplace. For details, see <a href="#">Discovery API</a> .	September 30, 2020

[The AWS Marketplace Catalog API is now generally available](#)

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