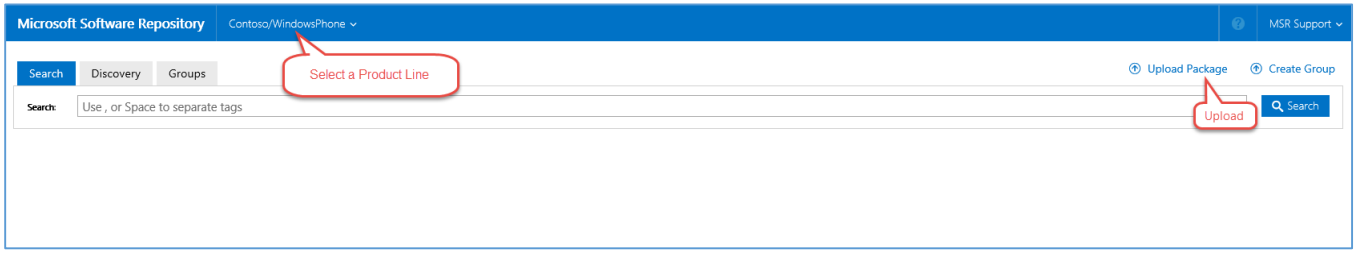
# Frequently Asked Questions

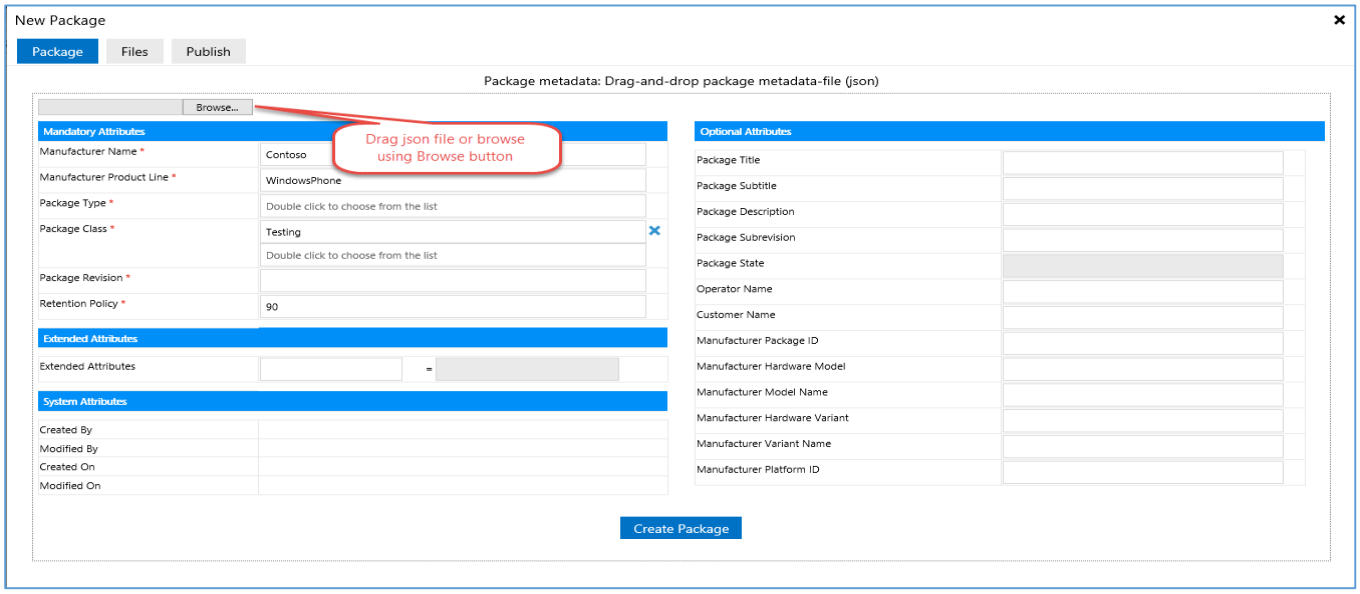
1. I want to upload a package
2. I want to create JSON file for my package
3. I want to get MD5 checksum for the files in my package
4. I want to publish/un-publish a package
5. I want to add members/admins to discovery group

# I want to upload a package

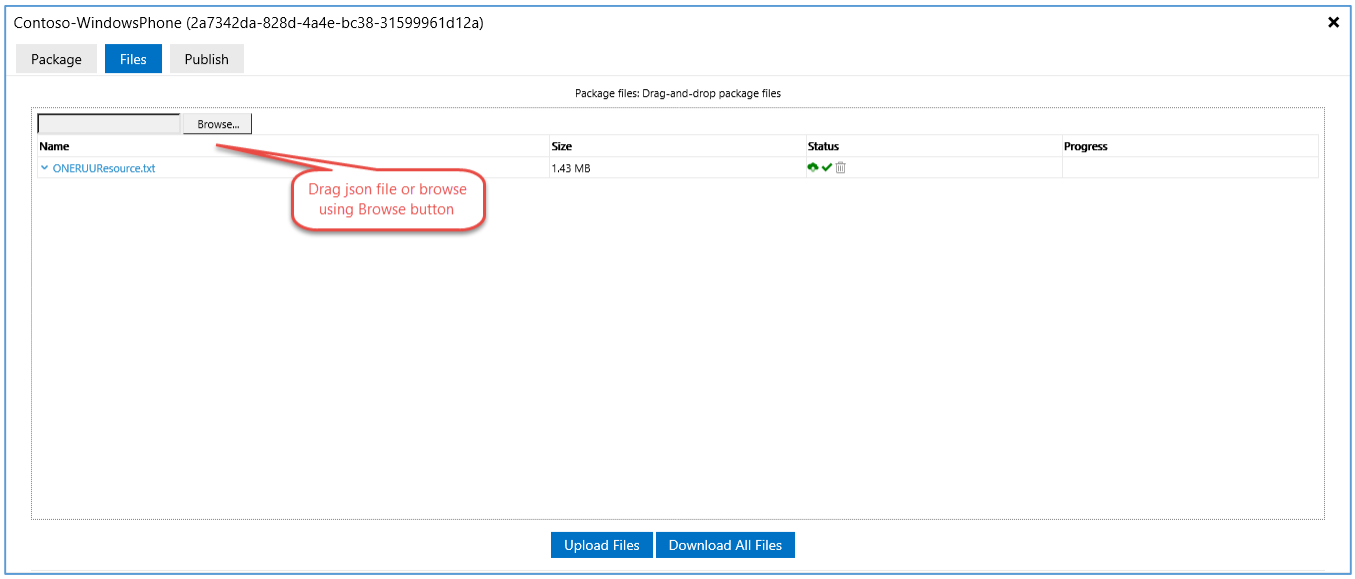
1. Select a Product Line from the drop down menu and then click the Upload Package link.




2. Package creation dialog shall pop up



3. Drag or Browse metadata JSON to Package metadata area
    a. Data can be edited on UI or JSON file can be edited and dragged again
4. Click ‘Create package’ button to create the package
    a. On successful package creation, a confirmation message shall be displayed.
5. Drag or Browse package files to Package files area
    a. Can be done only after creating the package.
    b. Once the package is created, Files tab will be visible through which files can be dragged or browsed.




6. Click ‘Upload files’ button to upload the files
7. Once package is created
    a. Editing and saving the metadata changes the created package
    b. In order to create another package dialog must be closed and ‘Create package’ in the main UI must be clicked again.

## I want to create json file of my package

1. Json file refers to the file containing metadata of the package..

2. The metadata file contains the details of the package and its associated files. Following are the attributes in the metadata.

| Attribute | Type | # | Description |
|---|---|---|---|
| id | String | 1 | Required. Unique system generated identifier for a software package. Not defined in new package creation. |
| packageType | String | 1 | Required. Defines type of software package allowing multiple types available for same device. |
| packageTitle | String | 0…1 | Recommended. Name of a software package. |
| packageSubtitle | String | 0…1 | Optional. More detailed naming of a software package. |
| packageDescription | String | 0…1 | Optional. Description of a software package or content of the software package. |
| packageClass | List of strings | 1…* | Required. Purpose of software package. |
| packageRevision | String | 1 | Required. Software version carried in software package. Syntax is limited to unlimited number numerical parts separated by dot (.) First part of revision is expected to be most significant part followed by less significant parts in descending significance and ending to least significant part. Comparing parts from most significant to least significant until there is difference in numerical value of a part will compare revisions to identify higher-lower revision. |
| packageSubRevision | String | 0…1 | Optional. Can be used to define revision order of multiple software packages with same package revision. Will be used in package revision comparison when value of packageRevision of two or more packages equal to get comparison order for packages. |
| packageState | String | 0…1 | Optional. Can be used to carry state information of package during its lifecycle. |
| manufacturerName | String | 1 | Required. A string to identify OEM manufacturer of a software package. |
| manufacturerProductLine | String | 1 | Required. A string to identify a product line of OEM manufacturer of a software package. |
| manufacturerModelName | List of strings | 0…* | Recommended [2]. Identifies a display name of OEM manufacturer product model. |
| manufacturerVariantName | List of strings | 0…* | Recommended [2]. Identifies OEM manufacturer product variant. |
| manufacturerPackageId | String | 0…1 | Recommended [2]. An identifier for software package within OEM manufacturer and product line scope. Value is intended to be unique unless omitted or NULL value defined. OEM shall take care of uniqueness of the attribute. |
| manufacturerPlatformId | List of strings | 0…* | Recommended [2]. Can be used to identify device platform for which software package is targeted. |
| manufacturerHardwareModel | List of strings | 0…* | Recommended [2]. Identifies OEM Manufacturer hardware model. |
| manufacturerHardwareVariant | List of strings | 0…* | Recommended [2]. Identifies OEM Manufacturer hardware variant. |

| | | | |
|---|---|---|---|
| operatorName | List of strings | 0…* | Recommended [2]. Identifies operator or operators for which product is targeted. |
| customerName | List of strings | 0…* | Recommended [2]. Identifies customer or customers for which product is targeted. |
| extendedAttributes | Map with flat list of strings | 0…* | Optional. Enables OEM specific attribute extensions as flat list of string type attributes |
| retentionPolicy | String | 1 | Required. Name of retention policy to be applied. Retention policies are listed and described in Software Repository Retention Policy Guide. |
| files | List | 0…* | Metadata of files associated with the package |

3. Example JSON file


```
{
    "manufacturerName": "Contoso",
    "manufacturerProductLine": "WindowsPhone",
    "packageType": "Firmware",
    "packageClass": ["Testing"],
    "packageTitle": "Phone firmware",
    "packageSubtitle": "",
    "packageDescription": "",
    "packageRevision": "1.0.20",
    "packageSubRevision": "",
    "packageState": "Completed",
    "manufacturerHardwareModel": [ "C-2014" ],
    "manufacturerHardwareVariant": [ "313", "314" ],
    "manufacturerPackageId": "Build_123.12",
    "manufacturerPlatformId": ["8.1"],
    "manufacturerModelName": ["The Headphone"],
    "manufacturerVariantName": ["Red"],
    "operatorName": [],
    "customerName": [],
    "extendedAttributes": {
        "myOwnAttribute1": "My Own Value1",
        "myOwnAttribute2": "My Own Value2"
    },
    "retentionPolicy": "90 days",
    "files": [
        {
            "fileName": "firmware.pkg",
            "fileSize": "10240543",
            "fileType": "image",
            "checksum": [
                {
                    "value": "8zpOSCYd3Jmkx96DXGtKsw==",
                    "type": "MD5"
                }
            ],
```

```
            "alternateUrl": [ "http://primary.local/path/firmware.pkg" ]
        },
        {
            "fileName": "firmware.signature.bin",
            "fileSize": 459,
            "fileType": "Signature",
            "checksum": [
                {
                    "value": "lHo9FcAD5r5KOAxKnIkX2g==",
                    "type": "MD5"
                }
            ],
            "alternateUrl": [ "http://primary.local/path/firmware.signature.bin"
]
        }
    ]
}
```

## I want to get MD5 checksum for the files in my package

1. For each file mentioned in the package metadata (JSON file), its MD5 hash value must be calculated by the source system/user before uploading to Software Repository.
2. It is used to ensure the file integrity during transit.
    a. Source systems/users could use this hash value to enforce file integrity upon upload to Software Repository.
    b. Clients could use this hash value for file integrity check after file download.
3. Software Repository requires MD5 hash value in Base64 encoded format.

    Sample

    a. File (click to download) : `package_metadata.json`
    b. MD5 hash value : `li8HTGrUEMlLL7P1G0rj1A==`
4. Command prompt utility to get MD5 hash value of a file in Base64 string

    File Checksum Integrity Verifier (FCIV) is a command-prompt utility that computes and verifies cryptographic hash values of files.

    To get the base64 encoded hash value, utility usage is

    ```
    >fciv package_metadata.json -xml myMD5.xml
    ```
5. Code snippet to get MD5 hash value of a file in Base64 string

a. C#

```
using System;
using System.IO;
using System.Security.Cryptography;
…
private string GetMD5HashFromFile(string fileName)
{
    using (var md5 = MD5.Create())
    {
        using (var stream = File.OpenRead(fileName))
        {
            return Convert.ToBase64String(md5.ComputeHash(stream));
        }
    }
}
```

b. Java

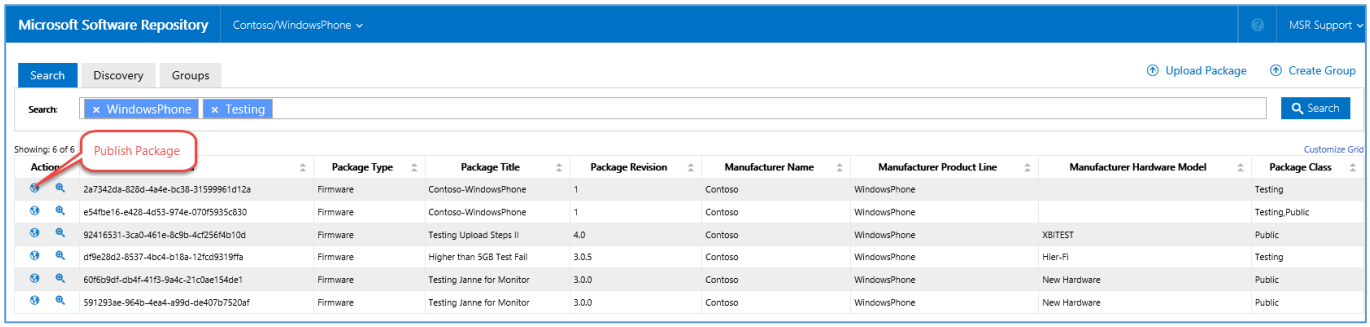
```
import org.apache.commons.codec.binary.Base64;
import org.apache.commons.codec.digest.DigestUtils;
…
private String GetMD5HashFromFile(String fileName)
{
       return Base64.encodeBase64String(
       DigestUtils.md5(Files.readAllBytes(Paths.get(fileName))));
}
```

c. Python

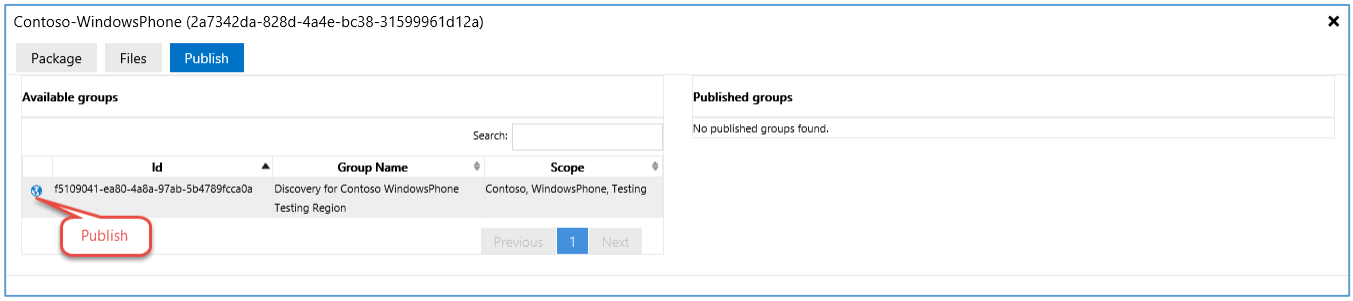
```
import hashlib
import base64
import sys

def md5base64(filePath):
    with open(filePath, 'rb') as fh:
        m = hashlib.md5()
        while True:
            data = fh.read(8192)
            if not data:
                break
            m.update(data)
        return base64.b64encode(m.digest()).decode("utf-8")

#provide the fully qualified filename as command line argument
print(md5base64(sys.argv[1]))
```

## I want to publish/un-publish a package

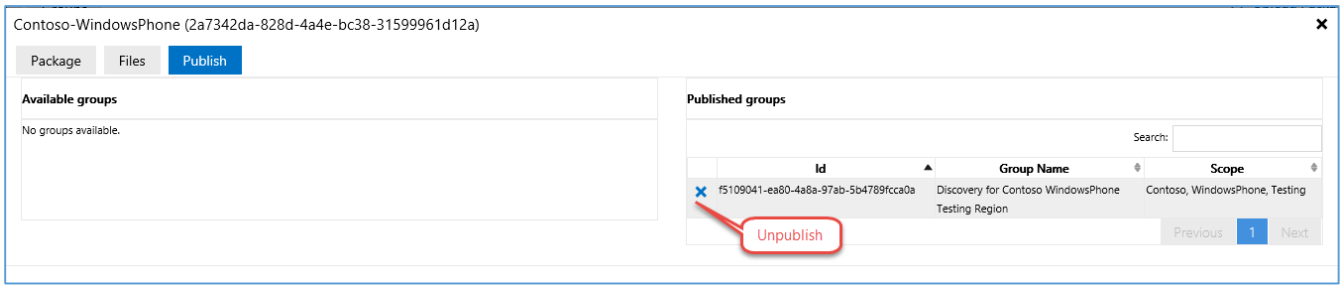
1. Search for the package.




2. Click the globe icon against the package to be published from the Search result.
3. Package edit dialog shall pop up.
4. "Available groups" section lists all the group that you have access to.
    a. Click the globe icon against a group to publish the package to it.




5. "Published groups" section lists all the group to which the package is already published to.
    a. Click the cross icon against a group to unpublish the package from it.

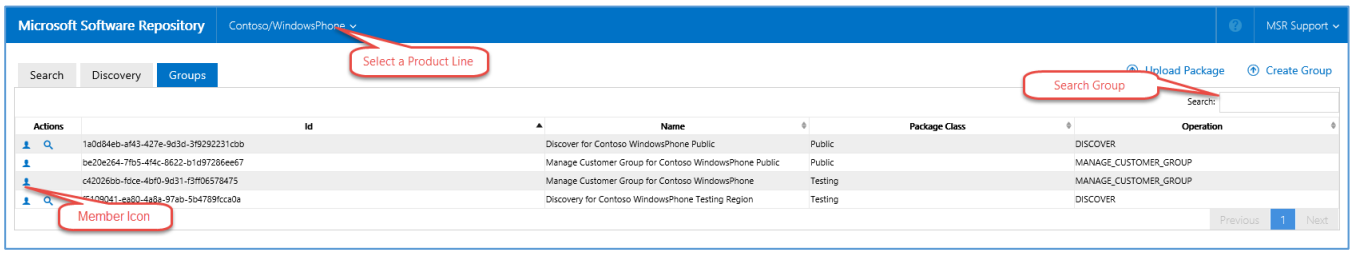



## I want to add members/admins to discovery group

1. Select a Product Line from the drop down menu.
    a. Click the Groups tab.

b. On right corner, you can search the discovery group to which members needs to be added by using search box

c. Click Member icon at the left corner of the particular group.

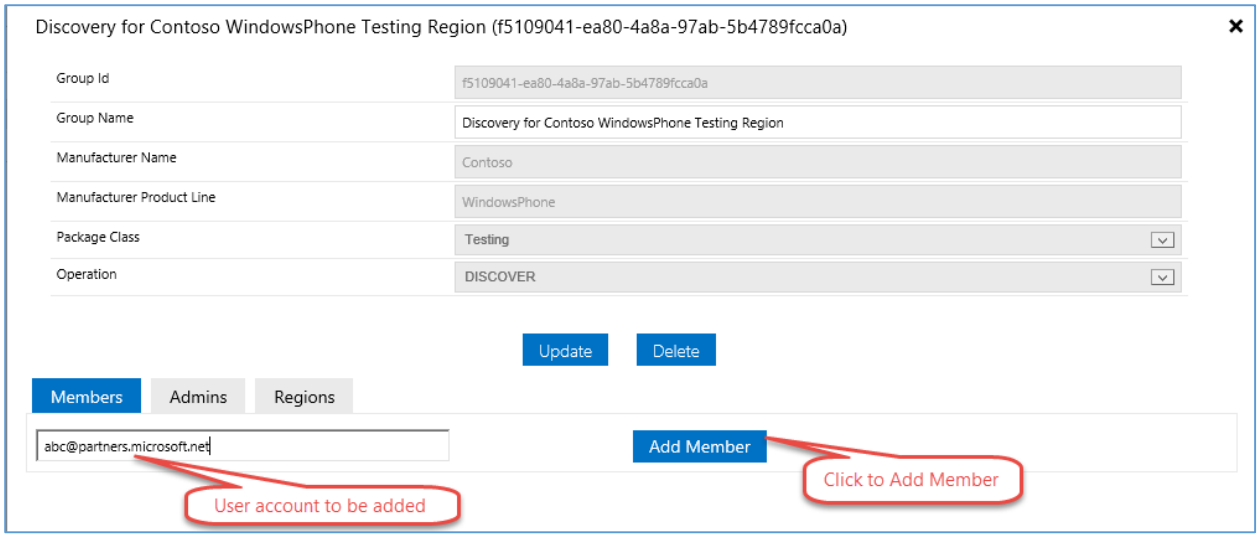



2. Group details dialog shall pop up.

3. Choose Members tab if you need to add members.

   a. Enter the user account details in the text box.

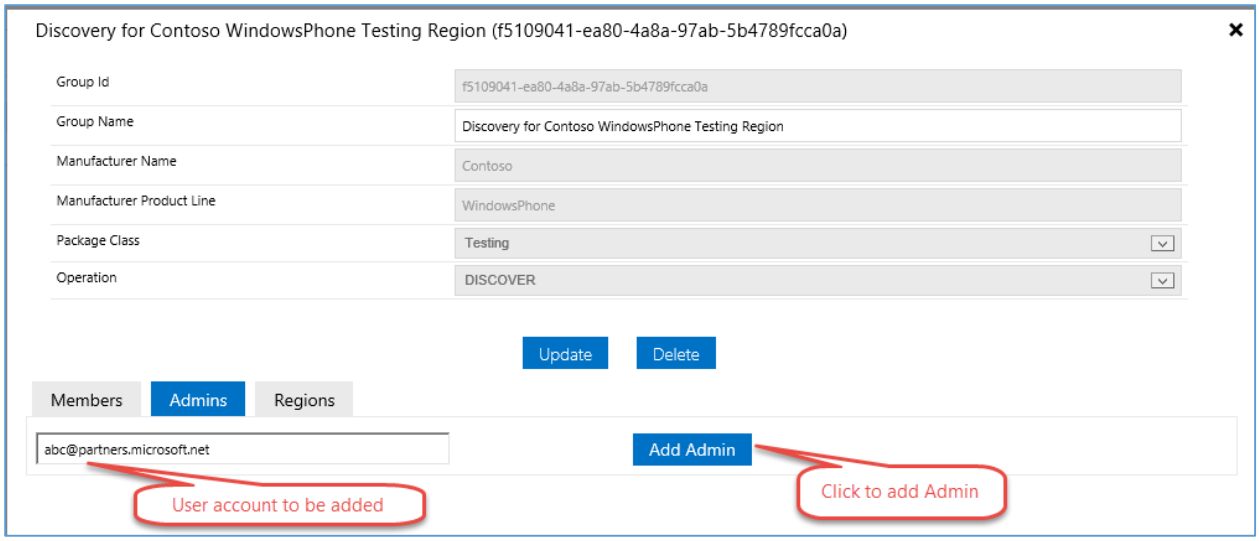
   b. Click Add Member button.




4. Choose Admins tab if you need to add admins.

   a. Enter the user account details in the text box.

   b. Click Add Admin button.


Discovery for Contoso WindowsPhone Testing Region (f5109041-ea80-4a8a-97ab-5b4789fcca0a)
Group Id
f5109041-ea80-4a8a-97ab-5b4789fcca0a
Group Name
Discovery for Contoso WindowsPhone Testing Region
Manufacturer Name
Contoso
Manufacturer Product Line
WindowsPhone
Package Class
Testing
Operation
DISCOVER
Update
Delete
Members
Admins
Regions
abc@partners.microsoft.net
Add Admin
User account to be added
Click to add Admin