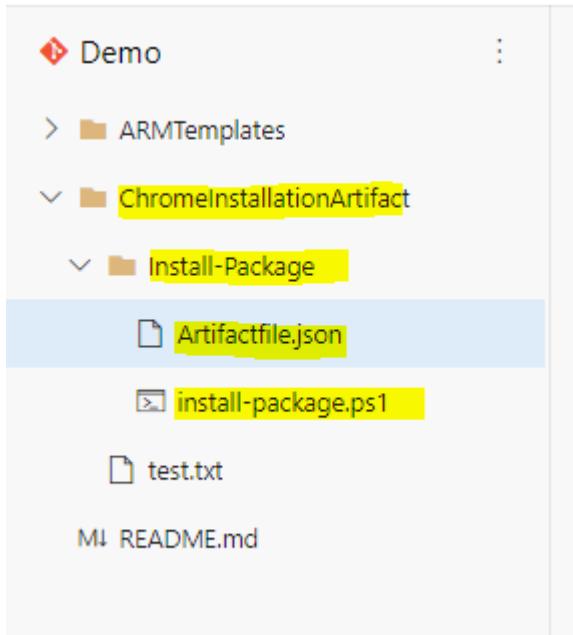


## How to add an custom artifact for Dev Test Lab and deploy it with ARM template from pipeline

### Step 1: Create an custom Artifact

- While creating an custom artifact create a folder with any name.
- Put the .ps1 script and an file “Artifactfile.json” as below. ([reference link](#))



- Configure Artifactfile.json to run the .ps1 file.

Sample Artifactfile.json will look like:

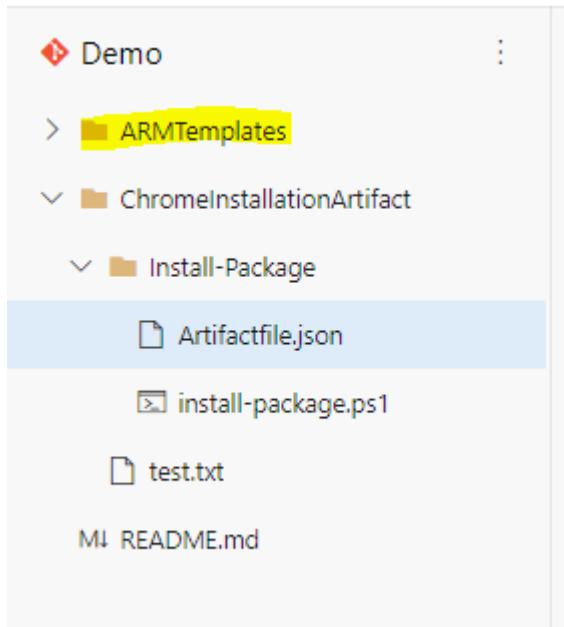
```
{  
  "$schema": "https://raw.githubusercontent.com/Azure/azure-devtestlab/master/schemas/2016-11-28/dtlArtifacts.json",  
  "title": "<>Expected title>>",  
  "publisher": "Saumya Mishra",  
  "description": "Description",  
  "tags": [  
    "Exchange"  
,  
  "targetOsType": "Windows",  
  "parameters": {  
    "pm1": {  
      "type": "string",  
      "displayName": "",  
      "description": ""  
    },  
  }]
```

```

"pm2": {
  "type": "string",
  "displayName": "",
  "description": ""
},
"pm3": {
  "type": "string",
  "displayName": "",
  "description": ""
},
"pm4": {
  "type": "securestring",
  "displayName": "PAT",
  "description": ""
}
},
"runCommand": {
  "commandToExecute": "[concat('powershell.exe -ExecutionPolicy bypass \\& ./install-package.ps1', ' -pm1 ', parameters(pm1), ' -pm2 ', parameters(pm2), ' -pm3 ', parameters(pm3), ' -PAT ', parameters(pm4))]"
}
}

```

- Create an ARM template for which we are going to access the artifact.



## Step 2: Add the Artifact to DevTestLab Repository

- 1) Create a new DevTest Lab
- 2) Go to configuration and policies:

The screenshot shows the 'Demo' DevTest Lab overview. On the left, there's a sidebar with links like 'Overview', 'Getting started', 'Internal support', 'My Lab' (with 'My virtual machines', 'Claimable virtual machines', 'All virtual machines', 'Security alerts', 'My data disks', 'Formulas (reusable bases)', 'My secrets', and 'Personal data'), and 'Settings'. Under 'Settings', the 'Configuration and policies' link is highlighted with a red circle. At the top, there are buttons for 'Refresh', 'Add', 'Claim any', 'Delete', 'MSDN forum', and 'Feedback'. Below the sidebar, it says 'Resource group (change) : [REDACTED]', 'Status : Ready', 'Location : Central US', 'Subscription (change) : [REDACTED]', and 'Subscription ID : [REDACTED]'. The main area is titled 'My virtual machines' and shows a table with one row: 'Nothing to display'.

- 3) Add a repository.

The screenshot shows the 'Configuration and policies | Repositories' page. On the left, there's a sidebar with 'Settings' (including 'Internal support', 'Allowed virtual machine sizes', 'Virtual machines per user', 'Virtual machines per lab', 'Lab settings', 'Lab announcement', 'Identity (Preview)'), 'Schedules' (including 'Auto-shutdown', 'Auto-start', 'Auto shutdown policy'), and 'External resources' (including 'Repositories' which is highlighted with a red box, and 'Mandatory artifacts'). At the top, there are buttons for 'Search (Ctrl+)', 'Add', 'Feedback', and 'Feedback'. The main area is titled 'Repository' and shows a single item: 'Public Artifact Repo'.

- 4) Fill the form to create an artifact

Home > Resource groups > ... > Demo > Configuration and policies | Repositories >

### Repository

Name of the repo

Name \*

Git clone URI \*

link to devops repo

Branch

Branch Name

Personal access token \*

PAT token

Folder paths

At least one folder path is required below

Artifact folder path

Artifact Folder

Azure Resource Manager template folder path

ARM Template folder name

```
graph LR; A[Name *] --> B[Git clone URI *]; B --> C[Branch]; C --> D[Personal access token *]; D --> E[Artifact folder path]; E --> F[Azure Resource Manager template folder path]
```

### Step 3 : Get the name (NOT display name) of the artifact.

- Goto [link](#) and run the command as below, the output of the API will give you the name of the Artifact.

List artifact sources in a given lab.

```
HTTP  
GET https://management.azure.com/subscriptions/{subscriptionId}/resourceGroups/{resourceGroupName}/providers/Microsoft.DevTestLab
```

Copy Try It

```
graph LR; A[HTTP] --> B[GET https://management.azure.com/subscriptions/{subscriptionId}/resourceGroups/{resourceGroupName}/providers/Microsoft.DevTestLab]; B --> C[Copy]; B --> D[Try It]
```

- Login with your account and provide the details as below and you should be able to see the name of the artifact

## REST API Try It

Try the REST API with the inputs below.

[Sign out](#)

**Request URL**

GET https://management.azure.com/subscriptions/{subscriptionId}/resourceGroups/{resourceGroupName}/providers/{providerNamespace}/{resourceType}/{resourceName}

**Parameters**

labName*	[REDACTED]
resourceGroupName*	[REDACTED]
subscriptionId*	[REDACTED] ▾
api-version*	2018-09-15

**Add parameter**   [+](#)

Sample output:

**Body**

[JSON](#) [Copy](#)

```
{  
  "value": [  
    {  
      "properties": {  
        "displayName": "[REDACTED]",  
        "uri": "[REDACTED]",  
        "sourceType": "VsoGit",  
        "folderPath": "[REDACTED]",  
        "armTemplateFolderPath": "[REDACTED]",  
        "branchRef": "[REDACTED]",  
        "status": "Enabled",  
        "createdDate": "[REDACTED]",  
        "provisioningState": "Succeeded",  
        "uniqueIdentifier": "[REDACTED]"  
      },  
      "id": "[REDACTED]",  
      "name": "[REDACTED]",  
      "type": "[REDACTED]"  
    }  
  ]  
}
```

**Name of the private artifact which we will be referring to in our ARM template**

#### Step 4: passing the parameter and referring to the artifact.

- Sample reference to the artifact will look like below:

The screenshot shows a GitHub commit for a file named `deployVms.json`. The commit message is "Added install-package.ps1". The code itself is a JSON object with the following structure:

```
1 [
2   "artifactId": "[resourceId('Microsoft.DevTestLab/labs/artifactSources/artifacts', 'name of custom artifact', 'Install-Package')]",
3   "parameters": [
4     {
5       "name": "pm1",
6       "value": "value for pm1"
7     },
8     {
9       "name": "pm2",
10      "value": "value for pm2"
11    },
12    {
13      "name": "pm3",
14      "value": "value for pm2"
15    },
16    {
17      "name": "PAT",
18      "value": "PAT token value"
19    }
20  ]
21 ]
```

Annotations with red boxes and arrows point to specific parts of the code:

- A box labeled "Lab Name" points to the string "Demo" in the `artifactId` field.
- A box labeled "Name of the custom artifact that we got from step 3" points to the string "'name of custom artifact'" in the `artifactId` field.
- A box labeled "folder name in which our .ps1 and artifact json file is present" points to the string "'Install-Package'".

