**Business Manager utility to update site-specific attributes**

(Reusable Component)



**PRESENTED BY: MEHUL KUMAR**

**Introduction**

This documentation aims to introduce a plugin cartridge **bm\_reusable\_utility** and how its functionalities can be a time saver for managing site-specific attributes.The documentation contains all the related information a merchant can get while using the cartridge. Components inside this cartridge aim to bulk update the process of attribute update with a user-friendly and attractive front-end design so that the merchants can save time on updating site-specific attributes and focus on other business needs. For more detailed information about the components and the setup process read along the whole document.

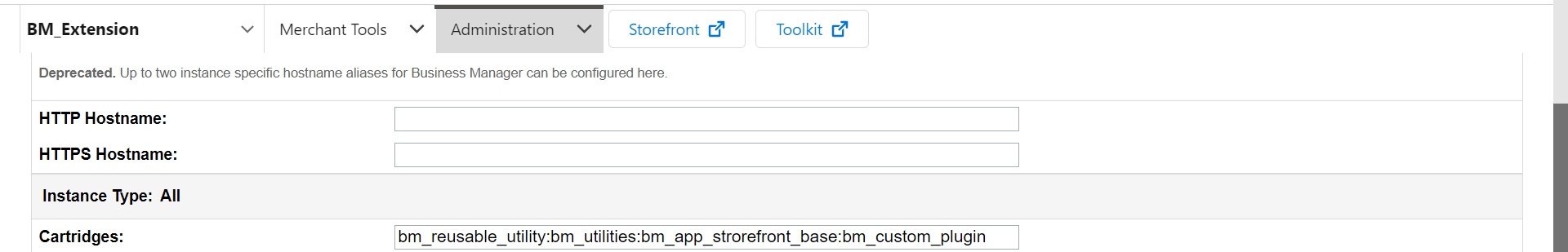
**Setup Process**

1. Setting up the cartridge

To use this cartridge upload it in your current code version and update the cartridge in the manage sites section as follows.

Go to Administration > Sites > Manage Sites > Business Manager – Settings

and enter the name of your cartridge as depicted in the figure below.



Now head over to the Administration > Organization > Roles > Administrator - Business Manager Modules section. Given that the cartridge was uploaded successfully to the active version, and you have done the above step then you will be able to see the Product Update Utility module as shown below in the picture.

Graphical user interface, text, application

Description automatically generated

Click on the update button at the bottom right to update the modules for the current site context.

Now if you refresh the business manager once you will be able to see the updated module in the Merchant Tool section.

On successful completion of the above steps, you will see a new module highlighted in the below picture.

Graphical user interface, application, email

Description automatically generated

**\*---------The cartridge part is done here---------\***

1. Setting up the job

Now head over to Administration > Operations > Jobs and create a job with the id

**Attributes\_update**. Inside this job create a job step with the same id (**Attributes\_update**) as the job id, now inside the job step click on configure the step and select **ImportCatalog** as the job step, select the id to be Attributes\_update.

1. Inside the WorkingFolder section go to Job Parameters > Add job parameter set the Parameter Name to **WorkingFolder**, and for WorkingFolder set the value to **attributesUpdate**.
2. Inside the FileNamePattern section go to Job Parameters > Add job parameter set the Parameter Name to **FileName**, and for FileNamePattern set the value to **attributes.xml**.
3. Set the import mode to Merge and let the rest of the attributes as it is.

**\*-------The job part is done here------\***

1. Setting up the services

We need to make three services. One is for getting an access token to access the resources. The second service will get all the site-specific attributes, and the third service is used to call the job which we have created in the previous step.

1. **Setting service for an access token**: - Go to Administration > Operations > Services > Service Credentials, and create a new credential set the Name field to **Create\_token**

set the URL field to [**https://account.demandware.com/dw/oauth2/access\_token**](https://account.demandware.com/dw/oauth2/access_token)**?**

set the User value to the client id of the ocapi data resource and set the Password field to the client secret key of the ocapi data resource.

Go to Administration > Operations > Services > Service Profiles and create a new profile with Name **Create\_token** set the Connection Timeout to 1000 ms.

Go to Administration > Operations > Services and create a service with Name **Create\_token,** set Type to HTTP, set Enable, and set Service Mode to Live. In the Profile and Credentials section select the Profile and Credentials we have just created with the name **Create\_token.**

1. **Setting service for site-specific attributes**: - Go to Administration > Operations > Services > Service Credentials, and create a new credential set the Name field to **getSitespecificattributes** set the URL field to **/system\_object\_definitions/Product/attribute\_definitions?**

set the User value to the client id of the ocapi data resource and set the Password field to the client secret key of the ocapi data resource.

Go to Administration > Operations > Services > Service Profiles and create a new profile with Name **getSitespecificattributes** set the Connection Timeout to 1000 ms.

Go to Administration > Operations > Services and create a service with Name **getSitespecificattributes,** set Type to HTTP, set Enable, and set Service Mode to Live. In the Profile and Credentials section select the Profile and Credentials we have just created with the name **getSitespecificattributes.**

1. **Setting service to call the job**: - Go to Administration > Operations > Services > Service Credentials, and create a new credential set the Name field to **ocapi\_job\_automation** leave the URL field to empty set the User value to the client id of the ocapi data resource and set the Password field to the client secret key of the ocapi data resource.

Go to Administration > Operations > Services > Service Profiles and create a new profile with Name **ocapi\_job\_automation** set the Connection Timeout to 1000 ms.

Go to Administration > Operations > Services and create a service with Name **ocapi\_job\_automation,** set Type to HTTP, set Enable, and set Service Mode to Live. In the Profile and Credentials section select the Profile and Credentials we have just created with the name **ocapi\_job\_automation.**

1. Setting up the resources

Go to Administration > Site Development > Open Commerce API Settings and select the **data** resource from the select type dropdown and select **Organization/your\_site** from the select context dropdown and paste the below JSON and validate.

{

"resource\_id": "/system\_object\_definitions/\*/attribute\_definitions",

"methods": [

"get"

],

"read\_attributes": "(\*\*)",

"write\_attributes": "(\*\*)"

},

{

"resource\_id": "/jobs/\*/executions",

"methods": [

"post"

],

"read\_attributes": "(\*\*)",

"write\_attributes": "(\*\*)"

}

**Advantages:**

1. Management of site-level product attributes can be managed by a single click and with a decent front-end design.
2. Good tool for merchants and whomsoever manages products on

the site level.