Branch Preparation Guide

1. Open Visual Studio Code & Press F1 -> Git Clone

The clone URL should be of the below format.

https://<<sso>>:<<personal access token>>@github.inbcu.com/SalesforceCentreOfExcellence/AdSales.git

1. Discard any pending changes
2. Create a new branch

Press F1 -> Create New Branch from -> enter branch name -> select origin/master as the source branch

1. Authorize an ORG (source org)
2. Update the package.xml in Manifest folder with components & save.
3. Right click on the package.xml file & ‘Retrieve source in Manifest from Org’
4. Stage all the changes -> enter commit message -> commit
5. Publish changes

Key Points:

1. The xml which is used to retrieve components should not have below tags
   1. <customlabels> 🡪 If present, then remove.
   2. <members>Insights\_Measurement\_Request\_\_c</members>

<name>Workflow</name>

This will fetch all the workflows on above object.

It should be like below

<members>Project\_Handling\_\_c.Billing finance approved alert</members>

<name>WorkflowRule</name>

1. Keep the xml used to fetch components (for our use) & the xml used for deployment (deploy.xml -> provided to COE team) separate.
2. Once components are fetched, copy paste content of deploy.xml into package.xml (in manifest folder) & stage, commit and publish.

Branch Validation Using Migration Assistant

1. Commit all the components to the branch.
2. Generate deploy.xml

With the correct branch selected, run the below command & select target org username.

./migration\_assistant.exe –c

Copy the deployment command & keep.

Verify the deploy.xml generated & ensure all the components are included.

1. Authorize the target org

If target org is a sandbox, use below command

sfdx force:auth:web:login -r <https://test.salesforce.com>

If the target org is production, use below command

sfdx force:auth:web:login -r <https://nbcusales.my.salesforce.com>

1. Validate the branch components

The deployment command will be similar to below (don’t run this yet)

cd C:\Users\lg135218\ProdDep\AdSales\deploy\2019-11-19 & sfdx force:source:deploy -u lalitasuresh.gawas@nbcuni.com -x C:\Users\lg135218\ProdDep\AdSales\deploy\2019-11-19\deploy.xml --checkonly

Append ‘-l’ to the command (indicates we need to run test classes)

Append ‘RunSpecifiedTests –r’ followed by all the test classes names separated by comma

The new command will look like below

cd C:\Users\lg135218\ProdDep\AdSales\deploy\2019-11-19 & sfdx force:source:deploy -u lalitasuresh.gawas@nbcuni.com -x C:\Users\lg135218\ProdDep\AdSales\deploy\2019-11-19\deploy.xml --checkonly -l RunSpecifiedTests -r TestMethods\_ShowMultiCreate,IMCreateClassTest,MultiSelectLookupCtrl\_Test,PaT\_TicketComponent\_Controller\_Test,PaT\_All\_Trigger\_Test,Test\_LEX\_Misc\_Billing

Navigate to deploy\2019-11-19 --- date will change

Cd deploy\2019-11-19

Run the command after the ‘&’ sign similar to below

sfdx force:source:deploy -u lalitasuresh.gawas@nbcuni.com -x C:\Users\lg135218\ProdDep\AdSales\deploy\2019-11-19\deploy.xml --checkonly -l RunSpecifiedTests -r TestMethods\_ShowMultiCreate,IMCreateClassTest,MultiSelectLookupCtrl\_Test,PaT\_TicketComponent\_Controller\_Test,PaT\_All\_Trigger\_Test,Test\_LEX\_Misc\_Billing

Key Points:

1. For reports, in deploy.xml, we have to add the report folder before the report name.

It will be as below

<members>New\_Users\_Report\_pcI</members>

<name>Report</name>

Add the report folder & it should be as below

<members>Digital\_Sales\_Management\_Reports/New\_Users\_Report\_pcI</members>

<name>Report</name>