Required tools permissions to perform weekly release,

1. ClearQuest ( GETS Database ), Contact person – Jones, Matthew
2. GitHub (Write permissions to MCA Platform Software organization, in future Applications GitHub organization too ) Contact person – Tullai, Philip
3. MD ( MCA Projects ) – Kausal, Rana

Execution :

1. At the end of the week either Thursday or Friday before releasing the items to the IV&V we usually confirm with them about the release then based on their inputs we will make a release.

(Send the relase defects list to the Juli-Ann, IVV Team Lead and confirm with her about your current release.)



1. Once IV&V confirms the release, then we will run the CQ Query Developer\_Tested. (Check the following screenshot in CQ)

A screenshot of a computer

Description automatically generated

1. This query will show us the all the items from Platform, Systems (and applications but app team is not using CQ to track their defects at this time.) Now defects are ready to build which means development is completed by the developer and it is ready for testing.
2. Then we will export the developer\_tested query, we have developed a python script which reads the CQ Query and makes us a release. But to execute this query developers should follow some work instructions.
3. As of now we are not using the above-mentioned python script to make a release. We’re doing it manually.
4. To make this release manually first we need to understand the whether defect belongs to git item or MD item.
5. Check the Changelist under solution and implementation tab there you can identify that the defects belongs to MD or GIT. Developers mention the commit number under Changelist. (Check the following screenshot, in this under changelist you can see git commit # sometimes developers also mentions the PR number, then immediately check the 3rd Level Descriptor under main tab to know the repository name, if you did’t find the repository name under it then check the description or check with the developer submitter.)

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Before making a release make sure that the commit # or PR is merged with the develop branch in GIT.

1. If it is git item (make sure it is a service repository if not merge the develop with master which creates containers ) which means our Jenkins pipeline is developed in a way that it will creates the container with the release branch name is only for the service repos. So before making a release in git repo we have to make sure that the repo is a service repo. Apart from that merge the develop branch with master to get the container. ( Ask Jason Grosch from Platform team if you have any questions on this point )
2. Now make a release branch from the develop branch and the naming convention should be IVV-PLT/APP-YEAR-DATATIME (For E.g., IVV-PLT-2020.0218.1001).

A screenshot of a computer

Description automatically generated

Create a Pre-Release:

1. Now go to releases tab and click on the draft a new release then tag the particular release with same naming convention and enter the release details means physical week of the release and CQ ID’s for the release. And click the check mark “This is a pre-relase” then publish release. For your reference check the following image.

A screenshot of a computer

Description automatically generated

1. If it is MD item then it we have to tag the particular MD revision number with the same naming convention. (See the below steps to know how tag MD)
2. Once Containers are ready send the release information to the IVV and change the CQ status assign it to the IVV team lead.



1. Containers are available at <https://dtr.mca.local/harbor/projects>.

How to Tag MD Item:

1. Connect to the following server, 3.125.182.23 then run the magicdraw.py script.
2. It will ask you to enter some inputs, ( it supposed to be read these inputs from the ClearQuest excel sheet ) As I said if we follow the work instructions it will execute any manual inputs ) Let’s see how to run this with the manual inputs.
3. When you run the script, it will ask you to enter the Tag Name. It should be same as above which we already used it while creating the branch in GitHub IVV-PLT/APP-YEAR-DATATIME (For E.g., IVV-PLT-2020.0218.1001)
4. Then it will ask you to enter the MD project ID
5. Following are the MD Project ID’s

MCA Platform Prd - a9f1641e-5994-4065-ac83-5b65dad2e163

MCA Application Layer Prod - e3ac899c-040c-4267-848a-05a30b10564c

MCA System - d215df3d-f670-49dd-9931-f1875c245ce9

MCA IVV - 66e55a0c-6883-4a79-a16a-c51760aaf9ec

1. Then Script will ask you to enter the Tag description, here enter the associated CQ IDs (Note If there are multiple IDs enter them with the space don’t use comas)
2. Then It will ask you to enter the Version Number, chose the max number from the given CQ Excel spread sheet and enter that number, make sure all those given versions in excel sheet are associated with the same project.