

# Internal Release Management Scripts

## Introduction

These are internal scripts used by Release Managers for Tasks Automations

## Freeze users and update dev team

```
List<UserLogin> userLoginList = new List<UserLogin>(
    [Select Id, isFrozen from UserLogin where UserId IN(

        Select Id From User Where IsActive = true and Username IN (
            'andre.vieira-ext@syensqo.com.spp.bugfix', 'anne.gilles@syensqo.com.spp.bugfix', 'arturd.alves-ext@syensqo.com.
            spp.bugfix', 'barbara.lemos@syensqo.com.spp.bugfix', 'bartosz.borowiec-ext@syensqo.com.spp.bugfix', 'bartosz.
            jagiello-ext@syensqo.com.spp.bugfix', 'clemence.rovelon@syensqo.com.spp.bugfix', 'dina.mendes@syensqo.com.spp.
            bugfix', 'francisco.cardoso-ext@syensqo.com.spp.bugfix', 'fabio.martins-ext@solvay.com.spp.bugfix', 'goncalo.silva-
            ex@syensqo.com.spp.bugfix', 'gowtham.chalamcharla-ext@syensqo.com.spp.bugfix', 'hugo.rosario@syensqo.com.spp.
            bugfix', 'josiane.peytraud@syensqo.com.spp.bugfix', 'jose.fischer-ext@syensqo.com.spp.bugfix', 'julia.smutek-
            ext@syensqo.com.spp.bugfix', 'julien.gasqueton@syensqo.com.spp.bugfix', 'kristina.girniak-ext@syensqo.com.spp.
            bugfix', 'leila.radjah@syensqo.com.spp.bugfix', 'lou-ann.benett-ext@syensqo.com.spp.bugfix', 'luis.fonseca-
            ext@solvay.com.spp.bugfix', 'manuel-moya.ferrer-ext@syensqo.com.spp.bugfix', 'mariateresa.ferreira-ext@syensqo.
            com.spp.bugfix', 'marika.macchia@syensqo.com.spp.bugfix', 'marina.lyakh-ext@syensqo.com.spp.bugfix', 'marta.
            machado-ext@syensqo.com.spp.bugfix', 'miguel.duarte-ext@syensqo.com.spp.bugfix', 'mohan.devarapu@syensqo.com.spp.
            bugfix', 'nikola.milic-ext@syensqo.com.spp.bugfix', 'nina.song@syensqo.com.spp.bugfix', 'reddyrani.busannagari-
            ext@syensqo.com.spp.bugfix', 'ricardo.neves-ext@syensqo.com.spp.bugfix', 'romain.bablot-ext@syensqo.com.spp.
            bugfix', 'sandra.martins@syensqo.com.spp.bugfix', 'satish.chintapandu-ext@syensqo.com.spp.bugfix', 'simona.roman-
            croitoru-ext@syensqo.com.spp.bugfix', 'stan.turyn-ext@syensqo.com.spp.bugfix', 'susana.alves@syensqo.com.spp.
            bugfix', 'valeriya.khrebtova-ext@syensqo.com.spp.bugfix', 'vanessa.conceicao-ext@syensqo.com.spp.bugfix', 'yash.
            jahit-ext@syensqo.com.spp.bugfix', 'lukasz.kiwinski-ext@syensqo.com.spp.bugfix'))]);

system.debug('Size: ' + userLoginList.size());
system.debug('Checking if 1 at least 1 is Frozen: ' + userLoginList[0].isFrozen);
for (UserLogin u : userLoginList) {
    u.IsFrozen = false;
}
update userLoginList;
```

**Updating users after sandbox refresh, removing the .invalid suffix from the email - [CCCME-8177](#)**

```

public class UpdateUserAfterOrgRefresh {
    public void execute(List<String> emails) {
        List<User> usersToUpdate = [SELECT Id, Name, Email, ProfileId FROM User WHERE Email LIKE :emails];
        for (User user : usersToUpdate) {
            if (user.Email.endsWith('.invalid')) {
                user.Email = user.Email.substring(0, user.Email.length() - 8);
            }
            System.debug('Updated User - Name: ' + user.Name + ', New Email: ' + user.Email);
        }
        update usersToUpdate;
    }
}

List<String> emails = new List<String>{'%andre.vieira-ext@syensqo.com%', '%anne.gilles@syensqo.com%', '%arturd.
alves-ext@syensqo.com%', '%barbara.lemos@syensqo.com%', '%bartosz.borowiec-ext@syensqo.com%', '%bartosz.jagiello-
ext@syensqo.com%', '%clemence.rovelon@syensqo.com%', '%dina.mendes@syensqo.com%', '%francisco.cardoso-ext@syensqo.
com%', '%fabio.martins-ext@solvay.com%', '%goncalo.silva-ex@syensqo.com%', '%gowtham.chalamcharla-ext@syensqo.
com%', '%hugo.rosario@syensqo.com%', '%josiane.peytraud@syensqo.com%', '%jose.fischer-ext@syensqo.com%', '%julia.
smutek-ext@syensqo.com%', '%julien.gasqueton@syensqo.com%', '%kristina.girniak-ext@syensqo.com%', '%leila.
radjah@syensqo.com%', '%lou-ann.benett-ext@syensqo.com%', '%luis.fonseca-ext@solvay.com%', '%manuel-moya.ferrer-
ext@syensqo.com%', '%mariateresa.ferreira-ext@syensqo.com%', '%marika.macchia@syensqo.com%', '%marina.lyakh-
ext@syensqo.com%', '%marta.machado-ext@syensqo.com%', '%miguel.duarte-ext@syensqo.com%', '%mohan.devarapu@syensqo.
com%', '%nikola.milic-ext@syensqo.com%', '%nina.song@syensqo.com%', '%reddyrani.busannagari-ext@syensqo.com%', '%
ricardo.neves-ext@syensqo.com%', '%romain.bablot-ext@syensqo.com%', '%sandra.martins@syensqo.com%', '%satish.
chintapandu-ext@syensqo.com%', '%simona.roman-croitoru-ext@syensqo.com%', '%stan.turyn-ext@syensqo.com%', '%susana.
alves@syensqo.com%', '%valeriya.khrebtova-ext@syensqo.com%', '%vanessa.conceicao-ext@syensqo.com%', '%yash.jahit-
ext@syensqo.com%', '%lukasz.kiwinski-ext@syensqo.com%'};
new UpdateUserAfterOrgRefresh().execute(emails);

```

The list of users is retrieved from <https://groups.google.com/a/syensqo.com/g/sco-sbs-is-appli-otc-crm.all/members>. Converting from this list in google to the list of emails in the above code snippet can be easily achieved through find & replace automations in Sublime Text or VSCode.

## Updating users profile after sandbox refresh, to have System Administrator profile - CCCME-8177

```

public class UpdateUserAfterOrgRefresh {
    public void execute(List<String> emails) {
        Profile sysAdminProfile = [SELECT Id, Name FROM Profile WHERE Name = 'System Administrator' LIMIT 1];
        List<User> usersToUpdate = [SELECT Id, Name, Email, ProfileId FROM User WHERE Email LIKE :emails];
        for (User user : usersToUpdate) {
            user.ProfileId = sysAdminProfile.Id;
            System.debug('Updated User - Name: ' + user.Name + ', New Profile: ' + user.ProfileId + ', New
Email: ' + user.Email);
        }
        update usersToUpdate;
    }
}

List<String> emails = new List<String>{'%andre.vieira-ext@syensqo.com%', '%anne.gilles@syensqo.com%', '%arturd.
alves-ext@syensqo.com%', '%barbara.lemos@syensqo.com%', '%bartosz.borowiec-ext@syensqo.com%', '%bartosz.jagiello-
ext@syensqo.com%', '%clemence.rovelon@syensqo.com%', '%dina.mendes@syensqo.com%', '%francisco.cardoso-ext@syensqo.
com%', '%fabio.martins-ext@solvay.com%', '%goncalo.silva-ex@syensqo.com%', '%gowtham.chalamcharla-ext@syensqo.
com%', '%hugo.rosario@syensqo.com%', '%josiane.peytraud@syensqo.com%', '%jose.fischer-ext@syensqo.com%', '%julia.
smutek-ext@syensqo.com%', '%julien.gasqueton@syensqo.com%', '%kristina.girniak-ext@syensqo.com%', '%leila.
radjah@syensqo.com%', '%lou-ann.benett-ext@syensqo.com%', '%luis.fonseca-ext@solvay.com%', '%manuel-moya.ferrer-
ext@syensqo.com%', '%mariateresa.ferreira-ext@syensqo.com%', '%marika.macchia@syensqo.com%', '%marina.lyakh-
ext@syensqo.com%', '%marta.machado-ext@syensqo.com%', '%miguel.duarte-ext@syensqo.com%', '%mohan.devarapu@syensqo.
com%', '%nikola.milic-ext@syensqo.com%', '%nina.song@syensqo.com%', '%reddyrani.busannagari-ext@syensqo.com%', '%
ricardo.neves-ext@syensqo.com%', '%romain.bablot-ext@syensqo.com%', '%sandra.martins@syensqo.com%', '%satish.
chintapandu-ext@syensqo.com%', '%simona.roman-croitoru-ext@syensqo.com%', '%stan.turyn-ext@syensqo.com%', '%susana.
alves@syensqo.com%', '%valeriya.khrebtova-ext@syensqo.com%', '%vanessa.conceicao-ext@syensqo.com%', '%yash.jahit-
ext@syensqo.com%', '%lukasz.kiwinski-ext@syensqo.com%'};
new UpdateUserAfterOrgRefresh().execute(emails);

```

Note that this is for dev sandboxes

- For UAT, INT - user needs to keep same access as in PROD
- For the other sandboxes assign Admin Lite

## Update Copado User stories for deployment, related to a specific release

```
List<copado__User_Story__c> updatedUserStories = new List<copado__User_Story__c>();

// Query user stories based on the provided externalIds
List<copado__User_Story__c> userStories = [SELECT Id, copado__Promote_Change__c
                                         FROM copado__User_Story__c
                                         WHERE copado__Release__r.Name = 'X'];

// Update the copado__Promote_Change__c field to true for the matching user stories
for (copado__User_Story__c story : userStories) {
    story.copado__Promote_Change__c = true;
    updatedUserStories.add(story);
}

// Perform the update
if (!updatedUserStories.isEmpty()) {
    try {
        update updatedUserStories;
    } catch (DmlException e) {
        // Handle any exceptions that might occur during the update
        System.debug('Error updating user stories: ' + e.getMessage());
    }
}

for (copado__User_Story__c story : updatedUserStories) {
    // Do something with the updated user story records
    System.debug('Updated User Story: ' + story.Id);
}
```