**Content Release Steps**

**Steps if new publisher needs to be added**

1. Go to “C:\BuildSVN\Projects\Repurposing”
2. Create folder with publisher name

**Steps to download content for new title from SVN to local**

1. Create new folder with product short name.
2. Create “trunk” folder inside short name folder.
3. Now right click on folder “trunk” and select the “SVN Checkout…” option
4. Now check the URL of the repository to confirm “trunk” is the part of the url.

* URL of repository: Url of trunk directory stored on SVN
* Checkout directory: file path of the newly created trunk folder.
* Checkout Depth: Full recursive
* Revision: Head version

1. Now click on the “OK” button to checkout. This will download the content inside the newly created trunk folder.

**Steps to download updated content for new title from SVN to local**

1. Right click on folder “trunk” and select the “SVN Update” option

**Steps to create label**

1. Open CMD with administrator access (Right click on command prompt and select Run as administrator)
2. Run the following command

createlabel /u <user name> /p <password> /m “<message>” <label no.>

Example: createlabel /u dattatray /p datta1 /m “1st beta build” 1.0.0

Note: label number format

Beta: 1.0.0

Beta Update: 1.0.1

Release: 19.0.0

Release Update: 19.1.0

**Steps to run build**

1. Open CMD with administrator access (Right click on command prompt and select Run as administrator)
2. Run the following command

make <label no.> > log\_<lable no.>.txt

Example: make 1.0.0 > log\_1.0.0.txt

1. Verify following files after successful build
2. <shortname>\_bb.zip file under $/<shortname>/<label no.>/media/ota
3. <shortname>.ART file under $/<shortname>/content

**Steps to deploy the ART and Zip file**

1. Go to c:\ProductRelease
2. Open any existing perl script and save as with new title’s name
3. Make the required changes in the perl script from line # 10 to 20 and 23 by referring the data from control panel.
4. Save the file
5. Go to terminal run following commands

Beta build - beta.bat <publisher folder> <title name> <label no.>

Example: beta.bat LWW-M ClinSkill4 1.0.0

Release build - release.bat <publisher folder> <title name> <label no.>

Example: release.bat LWW-M ClinSkill4 19.0.0

Create entries in the control panel for the first beta build. i.e. 1.0.0

1. Go to Control Panel
2. Go to OTA Enabled table
3. In the 1st table create entries for Android and iPhone platforms

**To provide the content for testing we need to create the serial number of the product.**

1. Go to Control Panel
2. Go to Generate serial number
3. In the 1st table create entries for Android and iPhone platforms

**Why Jenkins?**

* Continuous Integration and Continuous Delivery
* Easy installation
* Easy configuration
* Plugins
* Extensible
* Distributed

How to configure Jenkins?

## Step 1 - Install Jenkins

1. Download the Jenkins **Mac OS X** native package from [http://jenkins-ci.org](http://jenkins-ci.org/).
2. Double click the .pkg file to install Jenkins.
3. Once done, your browser will open to [**http://localhost:8080**](http://localhost:8080/) where Jenkins lives.

Step 2 – Download required plugins

1. Go to Manage Jenkins -> Manage Plugins -> Available Plugins
2. Select and download the following plugins.

* GitHub Plugin
* Git parameter plugin
* Xcode integration
* Safe Restart
* Email Ext

1. Restart the Jenkins Server.

## Step 3 – Create Project and Clone the repository

1. Select New Item option from left panel.
2. Create Freestyle project
3. Select Git radio button available under **Source Code Management**

Setting Up with HTTP

1. Enter repository HTTP URL
2. Add and select the Git credentials (Kind: Username with Password)
3. Select the newly added Credentials
4. Enter the branch name to clone the specific branch say “master” / “development”
5. Apply the changes
6. Now clone the repository by clicking on build link

Setting UP with SSH

Open terminal and Enter the following Terminal

1. sudo su - jenkins
2. Enter your Jenkins password (If asked)
3. ssh-keygen -t rsa -b 4096 -C “[your\_email@example.com](mailto:your_email@example.com)”
4. When you're prompted to "Enter a file in which to save the key," press Enter. This accepts the default file location.
5. When you're prompted to " Enter passphrase (empty for no passphrase)" press Enter. Do not enter a passphrase while generating the key. I found that Jenkins was unable to authenticate with Git if a passphrase was needed.
6. pbcopy < ~/.ssh/id\_rsa.pub
7. Add this key to your Github by going in to github account

Open Jenkins and set the following

1. Enter repository SSH URL
2. Add and select the Git credentials.

* Kind: SSH username with private key
* Username
* Private Key: From the Jenkins master ~/.ssh

1. Select the newly added Credentials
2. Apply the changes
3. Now clone the repository by clicking on build link

## Step 4 – Fetch Git Tags and Populate list to build

1. Select This build is parameterized available below Project Name.
2. Select Git Parameter from the Add Parameter dropdown menu
3. Set Name say “TAG\_SELECTOR”
4. Set Parameter Type as “Tag”
5. Set Sort mode to “DESCENDING SMART”
6. Apply the changes.
7. Verify by clicking on the **Build with Parameters** link

(You can set the Branch, Branch filter and Tag filter as per your requirement)

## Step 6 – Build with Xcode

1. Select Xcode from the Add build step dropdown

**General build settings**

1. Target: Enter target to build.
2. Clean before build: Yes
3. Configuration: Release
4. Select checkbox of Pack application and build .ipa?
   1. Set IPA file name in “.ipa filename pattern”
   2. Output directory: The output directory for the .ipa file, relative to the build directory.
   3. **Advanced Xcode build options**
5. Xcode Schema File: Optional (Don’t set if target is already set)
6. Build output directory: You only need to supply this value if you want the product of the Xcode build to be in a location. I set it “${WORKSPACE}/build/”
7. Create the folder with the name mentioned in the output directory

Note: ipa file will be saved: “${WORKSPACE}/build/ Output directory /name.ipa”

## Step 7 – Shell script to keep logs of ipa files

1. Create new folder inside the workspace. Say “Logs”
2. Select Execute Shell from Add build step
3. Copy and paste following script



1. Apply the changes

## Step 8 – Setup email notification

## Step 3 – To login with the Jenkins user automatically (Optional)

1. Go to Mac’s **System Preferences\Users & Groups.**
2. Unlock the settings by clicking the **lock** in the bottom left corner of the page and entering the **login password** for the **current user**.
3. Select **Login Options** underneath the list of users.
4. Select **Automatic login** and choose **Jenkins**.
5. Enter the password of **Jenkins** user.
6. Restart your computer. Once finished, you should be automatically logged in as the OSX **Jenkins** user.

## Step 3 – Enable Security for Jenkins (Optional)

1. Go to Manage Jenkins
2. Select [Configure Global Security](http://192.168.2.2:8080/configureSecurity).
3. Click on Enable security checkbox
4. Select Jenkins’ own user database along with “Allow users to sign up”
5. Apply the changes
6. Go to login page by clicking on the login button present on top right corner
7. Click on [Create an account](http://192.168.2.2:8080/signup) and create new account.
8. Again Go to Manage Jenkins -> [Configure Global Security](http://192.168.2.2:8080/configureSecurity).
9. Select Authorization as Matrix-based security
10. Add username for the authorization by entering in to User/group to add- > Click on Add
11. Click on all checkboxes appearing against added user.
12. Apply the changes
13. Verify the credentials by login with the username