

# Release Procedure

## Abstract

### Warning

- This page only gives a brief step of Weex release, always reference the [release policy of ASF](#) if you have questions or confusions.
- Content of this page aimed at Weex release manager, PPMCs and committers of Weex. End-users may consider using the [artifacts of Weex release](#) directly.
- Release must be voted before publishing according to the ASF's policy.
- Release are always about source code, binary is only published for users' convenience.

This page describes the steps of Weex release under ASF policy.

Releases are, by definition, anything that is published beyond the group that owns it. A release is normally conducted by a release manager who is a committer or PPMC member of Weex.

## Before start

### Prepare environment

To generate and compile a weex release, you need to set up the following environment:

- [Weex build environment](#), which includes Android, iOS, JS, C++ toolchains.
- [OpenPGP environment](#), which is used to generate signature and checksum for release candidate. Please note that OpenPGP is not the same as your ssh key though both of them may use the same cryptography algorithm. You may simply use `brew install gpg` to install tools and setup your key pair by `gpg --full-gen-key`.

### Create release branch

The release branch should be named to `release/xxx`, like `release/0.24`. Then the release manager should push the release branch to the remote repository.

### Tip

*Normally, developers should create a release branch based on the latest commit of master. One can also break this rule if you have proper reasons, like there is an unstable commit in master.*

## Request buy-in

Remember to send an email to [dev@weex.apache.org](mailto:dev@weex.apache.org) to see that if there is any has any reason to postpone your release.

I'd like to start a release for Weex 0.xxxxxxx based on release/xxxxxxxxx branch.

Does anyone have any reason to delay a Weex release? Any vital patches needs to be merged?

If not, I will start the release process tomorrow.

## Release candidate

*Release Candidates are packages that have been proposed for approval as a release but have not yet been approved by the project.*

### Generate release candidate

Release manager could invoke `scripts/apache_release.sh $RELEASE_CANDIDATE $TAG_OF_LATEST_RELEASE` to generate a release candidate. The explanation of the variable is listed below:

- `$RELEASE_CANDIDATE` The full-name of the release candidate, like 0.24.0-RC3

- `$TAG_OF_LATEST_RELEASE` Weex tag of last release, like 0.19.0.2. `RELEASE_NOTE.md` came from the difference between `$RELEASE_CANDIDATE` and `$TAG_OF_LATEST_RELEASE`.

The above script will do following things for the release manager:

1. Generate `RELEASE_NOTE.md` based on the history of git commit and `CHANGELOG.md`.
2. Create a temp directory named `apache_release_temp` and copy files related to the release to this dir. Files like test file, local configure files, build directory, weex playground and etc. are exclude from a release by this script.
3. Run [Apache Rat](#), which is a release audit tool used for check license and output the result to `RELEASE_AUDIT.LOG`
  - You can execute Apache rat by `java -jar scripts/apache-rat-0.12.jar $RELEASE_CANDIDATE_DIRECTORY`
4. Create a tarball, signature and checksum for your release.

Release manager should always check the following things before you move to the next step:

1. Check `RELEASE_AUDIT.LOG` to see if there is any license problems in your release candidate.
2. Go to `apache_release_temp` and verify you can compile it from source by running `scripts/build_from_source.sh`

## Publish release candidate

### Warning

- *The procedure below may not work well if the release manager is not a PPMC member of Weex as it needs to publish artifact with the privilege of PPMC. If this is the case, please contact one of the PPMC member and ask him/her to execute the following script for you with the following information:*
  - The tarball of the source file, like `apache-weex-incubating-xxxxx-xxxxx-src.tar.gz`
  - The signature of the source file, like `apache-weex-incubating-xxxxx-xxxxx-src.tar.gz.asc`
  - The checksum of the source file, like `apache-weex-incubating-xxxxx-xxxxx-src.tar.gz.sha512`
  - The `KEYS` as a result of `gpg --list-sigs && gpg --armor --export >> KEYS`
- Remember to append your public-key info to `KEYS` file in repository <https://dist.apache.org/repos/dist/release/incubator/weex> and <https://dist.apache.org/repos/dist/dev/incubator/weex> within the help of PPMC member.

Release manager could invoke `scripts/publish_release_candidate.sh $RELEASE_CANDIDATE_PREFIX $RELEASE_CANDIDATE_SUFFIX $GIT_REMOTE` to publish a release candidate. The explanation of the variable is listed below:

- `$RELEASE_CANDIDATE_PREFIX`, Weex release candidate prefix, like 0.24.0
- `$RELEASE_CANDIDATE_SUFFIX`, The release candidate suffix, like RC3
- `$GIT_REMOTE` The name of your Github repository, like `github-Apache` whose URL should be `git@github.com:apache/incubator-weex.git`. You can view `.git/config` file to find the name of your repository.

The above script will do following things for the release manager:

1. Push git tag with the name `${RELEASE_CANDIDATE_PREFIX}-${RELEASE_CANDIDATE_SUFFIX}` to `${GIT_REMOTE}` repository.
2. Push the release candidate to <https://dist.apache.org/repos/dist/dev/incubator/weex/>

## Call a vote in PPMC

### Warning

- All the links in your emails must be `https` instead of `http`.
- It's recommendation that release manager should send email during release with him/her Apache email account instead of personal account.

## Start Vote

Release manager should call a vote for your release in [dev@weex.apache.org](mailto:dev@weex.apache.org). To get the vote passed, you need three positive vote (namely +1 binding vote) and more positive vote than negative vote (namely -1).

### Tip

- In general community members are encouraged to vote, but their votes are only advisory. Only PMC members have formally binding vote.
- Though there are no vetoed in release vote, but negative vote should be considered seriously by the release manager.
- It is really normal that your release candidate failed in the vote. Don't be frustrated. Read the message that makes you fail and try to fix them. Then start another release vote like RC2, RC3 and so on.

One may reference the following email template:

Subject: [VOTE] Release Apache Weex (Incubating) XXXXXXXXXX-RCX

I am going to call a vote for release Apache Weex (Incubating) XXXXX-RCXXXXXX

- \* Git tag for this Release:
- \* The source tarball could be found at :
- \* The signature of the source tarball could be found at :
- \* The SHA-512 checksum of the source tarball could be found at :
- \* The source tarball is signed with Key: XXXXXXXXXXXXXXXXXXXXXXXXXXXX, which could be found in the key file:
- \* ChangeLog about this version:

One can build the binary from source according to XXXXXXXXXXXXXXXX

This vote will remain open for at least 72 hours, until we get enough votes. Please vote on releasing this RC.

[ ] +1 approve  
[ ] +0 no opinion  
[ ] -1 disapprove (and reason why)

## Close the vote

If your vote passed and the vote itself last over 72 hours. You can close the vote by sending another email to [dev@weex.apache.org](mailto:dev@weex.apache.org).

Subject: [RESULT][VOTE] Release Apache Weex (Incubating) XXXXXXXXXX-RCX

The vote for release Apache Weex (Incubating) XXXXXXXXXX has passed with XXXXXXXXXX +1 binding votes and XXXXXXXXXX -1 vote.

- \* +1 vote:
  - \* XXXXXXXXXX (binding)
  - \* XXXXXXXXXX (binding)
  - \* XXXXXXXXXX (not binding)
- \* +0 vote:
  - \* XXXXXXXXXX (binding)
  - \* XXXXXXXXXX (binding)
  - \* XXXXXXXXXX (not binding)
- \* -1 vote:
  - \* XXXXXXXXXX (binding)
  - \* XXXXXXXXXX (binding)
  - \* XXXXXXXXXX (not binding)

Vote Thread: XXXXXXXXXXXXXXXXXXXX

### Tip

- You can use [Apache mailing list archives](#) to find the link of your vote thread.
- If you vote don't get passed, you can start another vote directly without sending the email of closing vote. Of course, you can still send an email to close the vote even if it don't get passed. It's all your choice. Either way is acceptable.

## Call a vote in IPMC

After your vote passed in [dev@weex.apache.org](mailto:dev@weex.apache.org), release manager should bring this to [general@incubator.apache.org](mailto:general@incubator.apache.org). The vote rules are similar to vote in PPMC, where only difference are that only Incubator PMC votes are binding.

### Warning

*A Release vote must be passed by PPMC and IPMC in order.*

One may reference the following email template:

Subject: [VOTE] Release Apache Weex (Incubating) XXXXXXXXXX-RCX

The Apache Weex community has voted and approved the proposal to release Apache Weex (Incubating) version XXXXXXXXXXXXXXXX, we now kindly request the Incubator PMC members to review and vote on this incubator release.

- \* Vote thread:
- \* Vote result thread:
- \* Git tag for this Release:
- \* The source tarball could be found at :
- \* The signature of the source tarball could be found at :
- \* The SHA-512 checksum of the source tarball could be found at :
- \* The source tarball is signed with Key: XXXXXXXXXXXXXXXXXXXXXXXXXXXX, which could be found in the key file:
- \* ChangeLog about this version:

One can build the binary from source according to XXXXXXXXXXXXXXXXXXXX

This vote will remain open for at least 72 hours, until we get enough votes. Please vote on releasing this RC.

```
[ ] +1 approve
[ ] +0 no opinion
[ ] -1 disapprove (and reason why)
```

## Close Vote

Similarly, release manager need to close the vote if it get passed the vote itself last over 72 hours. Please remember send the email of closing vote to [general@incubator.apache.org](mailto:general@incubator.apache.org) and [dev@weex.apache.org](mailto:dev@weex.apache.org) together.

## Official release

*Releases are packages that have been approved for general public release, with varying degrees of caveat regarding their perceived quality or potential for change.*

In order to make your release official, here are the next steps.

## Publish release

### Warning

*The procedure below may not work well if the release manager is not a PPMC member of Weex as it needs to publish artifact with the privilege of PPMC. If this is the case, please ask help from one of the PPMC member.*

Release manager could invoke `scripts/publish_release_official.sh $RELEASE_CANDIDATE_PREFIX $RELEASE_CANDIDATE_SUFFIX $TAG_OF_LATEST_RELEASE $GIT_REMOTE $GITHUB_PERSONAL_TOKEN $JCENTER_TOKEN` to publish the release officially. The explanation of the variable is listed below:

- `$RELEASE_CANDIDATE_PREFIX`, Weex release candidate prefix, like 0.24.0
- `$RELEASE_CANDIDATE_SUFFIX`, The release candidate suffix, like RC3
- `$TAG_OF_LATEST_RELEASE` Weex tag of last release, like 0.19.0.2. `RELEASE_NOTE.md` came from the difference between `${RELEASE_CANDIDATE_PREFIX}-${RELEASE_CANDIDATE_SUFFIX}` and `$TAG_OF_LATEST_RELEASE`.
- `$GIT_REMOTE` The name of your Github repository, like `github-Apache` whose URL should be `git@github.com:apache/incubator-weex.git`
- `$GITHUB_PERSONAL_TOKEN` The personal access token of your [github Account](#), which should have write privilege to `git@github.com:apache/incubator-weex.git`. The personal access token is used to publish Github Release
- `$JCENTER_TOKEN` The private key for [JCenter](#), which is the distribution channel for Android

The above script will do following things for the release manager:

1. Generate `RELEASE_NOTE.md` based on the history of git commit and `CHANGELOG.md`.
2. Download release candidate from `https://dist.apache.org/repos/dist/dev/incubator/weex/${RELEASE_CANDIDATE_PREFIX}/${RELEASE_CANDIDATE_SUFFIX}`, rename it to `apache-weex-incubating-${RELEASE_CANDIDATE_PREFIX}-src.tar.gz`, then upload it to <https://dist.apache.org/repos/dist/release/incubator/weex/>
3. Push git tag named `${RELEASE_CANDIDATE_PREFIX}` to `${GIT_REMOTE}` repository and generate a corresponding [Github Release](#) with `${GITHUB_PERSONAL_TOKEN}`. As the script will install [release-it](#) with `npm install -g release-it` to publish github release, make sure your npm environment is ready.
4. Publish the convenience binary of Android to JCenter with your `${JCENTER_TOKEN}`. You can see the [JCenter token](#) if you are a PPMC member of Weex.

### Warning

*The convenience binary of iOS can only be published manually now.*

You should also archive your old release by deleting them from <https://dist.apache.org/repos/dist/release/incubator/weex/> once your latest release is published according to [ASF' release policy](#).

## Update website

Please remember update the [website](#) with your latest release information and correct link.

### Warning

- *The download link of the latest release source must be provided in the format of mirror, like `https://www.apache.org/dyn/closer.cgi?filename=incubator/weex/${RELEASE_CANDIDATE_PREFIX}/apache-weex-incubating-${RELEASE_CANDIDATE_PREFIX}-src.tar.gz&action=download` according to [ASF' policy](#).*
- *As the old release is archived, you should probably update the link of your previous release to new address like, `https://archive.apache.org/dist/incubator/weex/${PREVIOUS_RELEASE}/apache-weex-incubating-${PREVIOUS_RELEASE}-src.tar.gz`*

#### Tip

- *Remember to move old latest version to Archived Release section.*
- *The version information like `pod version` in [Github Page](#) will get updated within 24 hours automatically. Please be patient.*

## Announce it

You are almost there. Just send an email to [dev@weex.apache.org](mailto:dev@weex.apache.org) to announce it.

Subject: [ANNOUNCEMENT] Weex XXXXXXXXXXXXXXXX Released

Weex XXXXXXXXXXXXXXXX is released now, one can download source or convenience binary through the link in our website [1].

[1] <https://weex.apache.org/download/download.html#latest-release>