Central Repositories

Overview
A central repository is a large repository managed by the JFrog Bintry team, that includes packages submitted by many contributing users.

Why Use a Central Repository
By including your package in a central repository:

- You expose your package to the largest relevant audience bringing the full force of the developer community to bear
- The package remains under your ownership giving you full rights to it and control of how it evolves
- Your package is in a curated repository with trusted content that the community can rely on

The two central repositories available on Bintry are JCenter and Conan-Center, and these are described in more detail below.

General Requirements
In order to include your packages in a central repository, you must meet the following requirements:

- Your package must be in one of your public Bintry repositories.
- You cannot be using Bintry on a trial account. You can only include your packages in a central repository if you are on an OSS or Premium account.
- In addition to these general requirements, each central repository (JCenter and Conan-Center) has its own specific requirements as described below.

JCenter
JCenter is a one-stop-shop for Maven artifacts in Bintry. It is a large, popular, Maven repository that is managed by Bintry and is publicly available to all Bintry users.

All connections to JCenter should be made using secure HTTPS transfer protocol. Until January 1, 2020 all requests using HTTP will be redirected to HTTPS. After that date, JCenter will accept only requests using HTTPS.

Including your Package in JCenter
You may include your packages in JCenter if they are Maven packages, which are hosted in a public repository. Even when your package is included in JCenter, it is still your package and is managed through your own repositories. This means that you can manage, modify and update it as you would any of your other packages on Bintry, and any modification will be updated automatically and will be available to the users accessing your package via JCenter.

As mentioned, to include a package in JCenter, you cannot be on a trial account (as described in the General Requirements above), and your package must be public. In addition, the following Maven-specific requirements must be met:

- The package must be in a public Maven repository and must contain Maven sources.
- The path of the files (entered in the Upload Files form of the Version page) must conform to Maven standards (the Group ID and Artifact ID combination must be unique, etc.; for more information about Maven standards, consult the appropriate Maven documentation)
- Every version within the package that includes files must also include a valid POM file.

To include your package in JCenter:

1. Access the Package page of the package you want to include in JCenter (you must be the owner of the package)
2. In the lower right corner, in the section **Included In**, click the **Add to JCenter** button.

The **Compose Message** page opens, with the addressee and subject already assigned for the request message.

3. In the Compose Message page, you can:
   - Check the **Host My Snapshot build artifacts** checkbox to include your Maven snapshots in Bintray (click [here](https://www.bintray.com) for more information).
   - You can also add some text to the message - a few lines about what your package is and why it is relevant for JCenter (optional).

4. Click **Send**.

   The request message is sent to the Bintray team, who will assign your package a path in the relevant part of the tree structure of the JCenter repository.

   If the request is approved, an indication that the package is included in JCenter is displayed in the Package page, in the **Included in** work area in the bottom right corner of the page.

   **Note:** This same **Included in** work area also indicated repos to which the package has been linked.

   Your package is now exposed to all the users who visit the large, central repository of JCenter. You are still the owner and can update it.
Conan-Center

Conan-Center is a one-stop-shop for C/C++ Conan packages in Bintray. It is a central Conan repository that is managed by Bintray and is publicly available to all Bintray users.

Bintray currently hosts two central repositories for C/C++ Conan packages:

- **conan-center**
  This repository hosts moderated, curated and well-maintained packages, and is the place where you can share your packages with the C/C++ community. To share your package, you upload it to your own (or your organization’s) repositories and submit a request to include it in Conan-Center.

- **conan-transit**
  This repository is an exact copy of the previous conan.io repository at the time it was migrated as described below. It will be a read-only repository meaning that you can download any packages that it hosts, but you will not be able to upload any new packages to it.

Including your Package in Conan-Center

You may include your packages in Conan-Center if they are C/C++ Conan packages, which are hosted in a public repository. Even when your package is included in Conan Center, it is still your package and is managed through your own repositories. This means that you can manage, modify and update it as you would any of your other packages on Bintray, and any modification will be updated automatically and will be available to the users accessing your package via Conan-Center.

As mentioned, to include a package in Conan-Center, you cannot be on a trial account (as described in the General Requirements above), and your package must be public. In addition, the following Conan-specific requirements must be met:

- If you are the author of an open source library, your package will be approved. Keep in mind that it is your responsibility to maintain acceptable standards of quality for all packages you submit for inclusion in Conan-Center.
- If you are packaging a third-party library, you need to follow the guidelines below:
  - The recipes must contain a `test_package`.
  - If the library supports it, the recipe has to be compatible with Windows, Linux and OSX.
  - Have CI enabled to test it.
  - Provide a general review of the recipe. The Bintray team will make suggestions for improvements or better/cleaner ways to implement the code.

To include your package in Conan-Center:

1. Access the Package page of the package you want to include in Conan-Center (you must be the owner of the package)
2. In the lower right corner, in the section **Included In**, click the **Add to Conan-Center** button.

![Compose Message page](image)

The **Compose Message** page opens, with the addressee and subject already assigned for the request message.

3. In the Compose Message page, you can add some text to the message - a few lines about what your package is and why it is relevant for Conan-Center (optional).

4. Click **Send**.

The request message is sent to the Bintray team, who will assign your package a path in the relevant part of the tree structure of the Conan-Center repository.

If the request is approved, an indication that the package is included in Conan-Center is displayed in the Package page, in the **Included in** work area in the bottom right corner of the page.

**Note:** This same **Included in** work area also indicates repositories to which the package has been linked.

Your package is now exposed to all the users who visit Conan-Center. You are still the owner and can update it.

**conan.io Package Consumers**

On **June 11, 2017**, all packages that were on conan.io were migrated to the Bintray conan-transit central repository.

Any requests to the old server conan.io domain are now redirected to the new conan-transit repository on Bintray. Since conan-transit contains a copy of all packages from the old conan.io, these requests will keep working transparently.
**Note:** Initially, following the migration from conan.io, Conan-Center will be empty. It will only be populated once users upload Conan C/C++ packages to their own Conan repositories and submit inclusion requests to the Bintray team.

If you are uploading packages for your own internal use, or for use by your team, then your personal or organization’s repositories are sufficient.

Here is how to configure your Conan client to start using the new Bintray repositories:

```
$ conan remote remove conan.io
$ conan remote add conan-center https://conan.bintray.com
$ conan remote add conan-transit https://conan-transit.bintray.com
```

If, for some reason, you still wanted to read from the old conan.io repository (which by now is read-only), the new remote should point to [https://legacy-server.conan.io/](https://legacy-server.conan.io/).

By specifying your remotes in this way, your Conan client will try to resolve packages and install them from repositories in the following order of priority:

1. Your own repository
2. conan-center
3. conan-transit

As described above, conan-transit contains a snapshot of conan.io at the time it was migrated to Bintray, including a copy of the packages you had uploaded to your own repositories, and these will all be read-only. If you now upload new versions to your repositories, conan-transit will become outdated, however, packages you had previously loaded before the migration will still be available to your consumers, so none of their builds will break.