# Runtime Infrastructure - Support #6340

## move JNI and native dependencies into the jar

05/10/2022 12:21 PM - Greg Shah

| | | | |
|---|---|---|---|
| **Status:** | New | **Start date:** | |
| **Priority:** | Normal | **Due date:** | |
| **Assignee:** | | **% Done:** | 0% |
| **Category:** | | **Estimated time:** | 0.00 hour |
| **Target version:** | | | |
| **billable:** | No | **case_num:** | |
| **vendor_id:** | GCD | **version:** | |

| **Description** |
|---|
| |

| **Related issues:** | |
|---|---|
| Related to Runtime Infrastructure - Support #4549: reduce/eliminate installat... | **New** |

---

**History**

**#1 - 05/10/2022 12:33 PM - Greg Shah**

This article describes a technique to load our own native libraries (and other native dependencies) from our own jar file:

https://stackoverflow.com/questions/2937406/how-to-bundle-a-native-library-and-a-jni-library-inside-a-jar

The idea is to bundle multiple versions of the native binaries into the jar, extract them to a temp dir at runtime and then load them from there. This is a clever way to make it easier to deploy. There can also be a fall-back to use the current approach if there is any failure in the extraction or loading.

In addition to our own libp2j.so/p2j.dll we might want to also bundle libffi so that we can eliminate that dependency from the install requirements.

In combination with #4550, #5167 and #5568, we could eliminate all of the local install requirements. A critical requirement for the runtime system will be that a writable temp directory is available (with a certain amount of space available). That is OK.

**#2 - 05/10/2022 12:38 PM - Greg Shah**

*- Related to Support #4549: reduce/eliminate installation dependencies added*