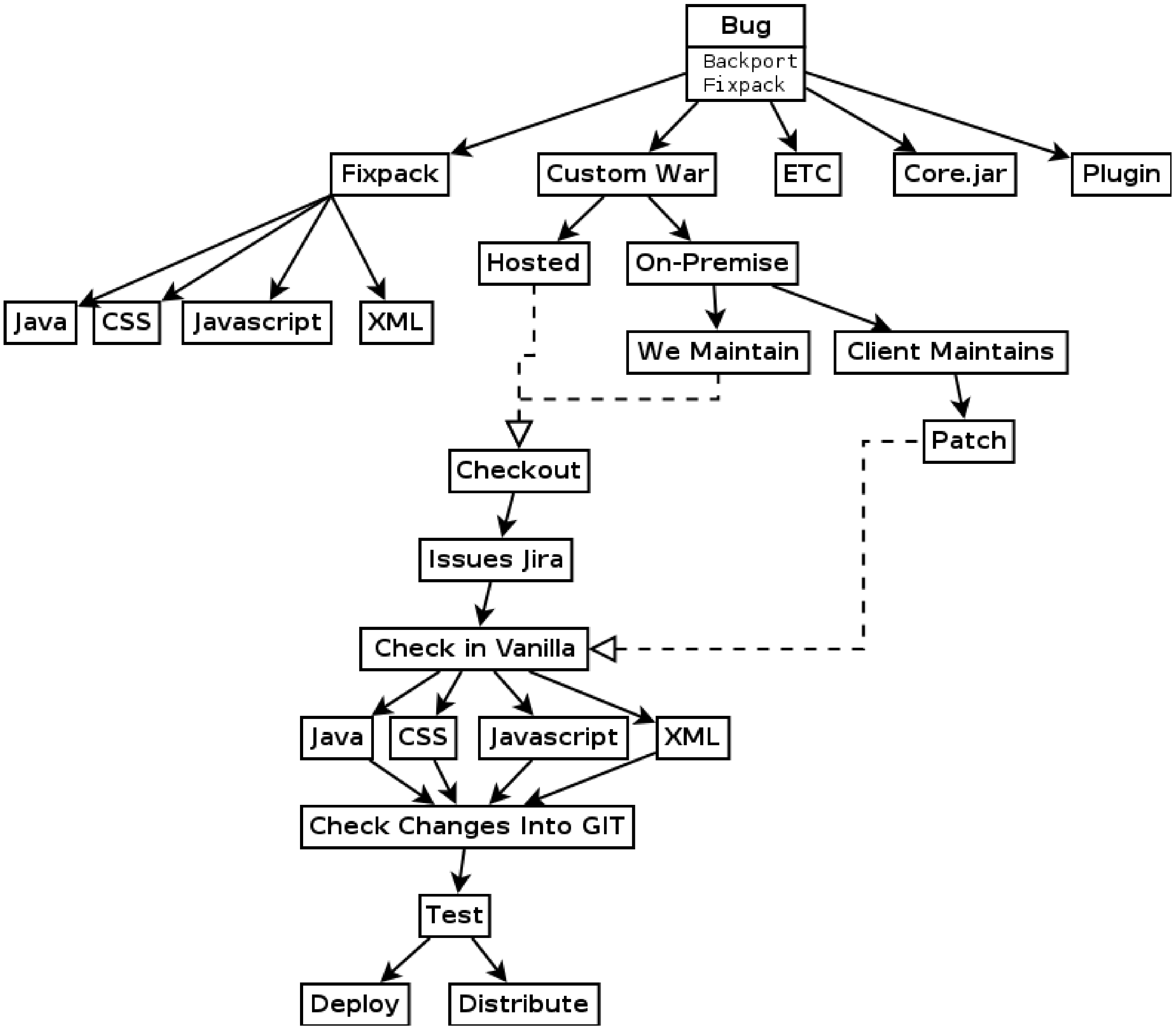
Big Book of Backporting



Overview

Before you begin, read the following summary of what a backport is and the risks involved. What is a Backport? | Jive Community

Backports start out as a "Backport Request". It is up to the backline engineer to determine if it is worth doing. They can reject the "backport request" for any number of reasons (i.e. too complex, to risky).
When I pick up a "backport request" I usually cut/paste the following into the case:

> Hi XXXXX,
> I'm the backline engineer that will review this request. In the mean time, please take a look at this document explaining what a backport is and the risks involved: What is a Backport? | Jive Community

More technically, a backport is the process of taking specific code from a later version of jive and applying it to the customer's code, a more recent version of jive. For example the customer is on version 9.0.1.0 but wants some code from version 9.0.3.0 because that is the version that fixed the bug they are encountering. Note, however, this later version of jive (i.e. version 9.0.3.0) has has many code changes besides the one bug fix (multiple bugs may have been fixed, possibly new features added, security enhancements, etc.). Therefore a backport is the process of filtering out only the code changes that fixed a single bug, and applying those specific code changes to the customer's instance (i.e. version 9.0.1.0).

Your last option should be a backport, because it opens the door for a lot of risk:
- backports do not go through significant testing
- require a full restart (many customers don't like downtime of their site)
- cannot guarantee it will fix the problem
- can introduce side effects (which could be discovered weeks/months later and difficult to trace back to the backport as the cause).

- Therefore, if you identify a customer's issue as being tied to a bug in Jira AND that bug has been fixed in a later version of jive, then ideally you'll want to recommend an upgrade to the customer first.

Keep in mind there is an associated JIRA ticket with the case

When a customer requests a backport, then a case **AND** a Jira ticket are created.
- Read Backport Request Process
- The Jira ticket is created along with the case because the Jira system is a more efficient way to search for and gather reporting data on backports than the JiveWorks case system. (using two

"issue trackers" is a bit redundant and we may want to get rid of one or the other. But for now, there should be a Jira ticket for each backport case).

- When working the backport, assign both the case and the Jira ticket to yourself
- Approve the jira ticket (assuming you are approving the backport request, otherwise reject it and you are done)
  When you close out the case, be sure to close out the JIRA ticket also.
  - If you create a fixpack JAR (discussed below) for the backport please attach that JAR with a diff of the changes to the JIRA ticket. You can do this by going to the backport request ticket in jira and clicking More -> Attach File(s).

Determine if there is a custom WAR

If there is a custom WAR, the process for backporting is very different and much more involved.

Is there a custom WAR?

Determining the Fix Method

- Determining the Fix Method - Important for filling out the 'Custom WAR?' field in the backport request as well as where to proceed next

Getting ready

- How to create a patch file
- Maintaining the PS Internal JIRA

Doing work

- It's a fixpack
- It's a Plugin
- It's a Soy/FTL file
- Patching the EAE service
- Patching the EAE service-client

Important!!- As of Jan 2016 (and likely ongoing for a few months) PS is transitioning from SVN to Stash.

They do not want any work done in SVN. Therefore, you must first check Stash to find your customer.

- If the customer is in Stash, then proceed to follow the steps outlined in Backporting to a Custom WAR using git and stash
- If the customer is not in Stash, then proceed to use SVN and follow the steps outlines inBackporting to a Custom WAR using SVN
- If the customer is not in Stash or SVN and needs a Custom WAR, put in a request to the PS Response Team to have them generate a blank WAR within Jenkins for you.

- How to test your backport
- Removing a backport - If you are removing a fixpack or JAR from an instance you should only be doing so if you will be restarting directly afterward! You cannot stage removal of a fixpack or JAR as this can cause exceptions to be thrown and application errors to ensue.

Delivery

- Maintaining PS Jenkins
- Delivering the fix

Miscellaneous help and troubleshooting

- Big Book of Backport Troubleshooting
- How to overlay what ...
- Maven tips
- What's an etc.zip?
- How-To: Finding Customer Backport Requests in JIRA
- How To - Tell what fixpacks a customer has
- Non-Dairy Soy Plugin- It's delicious! - helpful when doing theme related backports
- So you have a complicated backport... Scoping out the issue

Is your backport question not answered within this documentation?

- See if the work has already been done before:
  - How to find if a fixpack has already been created
- Ask in Chime in BL:ON chat.
- Create a discussion within the The Kraken (Backline Support)

Once you've had your question answered, please update the documentation above so that others have access to it.