Instructions for Journey Hotfix - Proposal

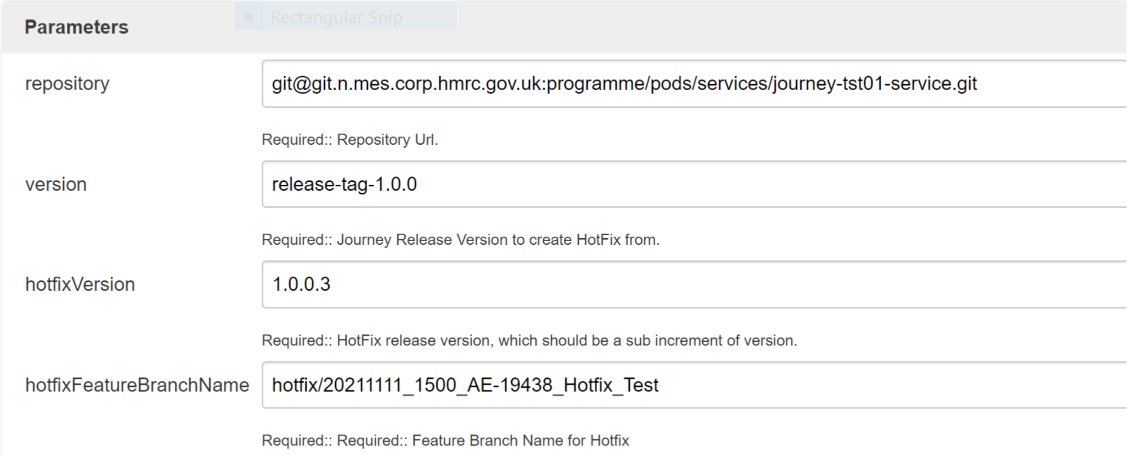
The instructions below assume that ***HMRC-GitFlow-Plugin** version **1.3*** is used. Any version of HMRC-GitFlow-Plugin lower than 1.3 will result in merge conflict errors in Jenkins Workflow as JGitFlow will attempt to merge into develop and master which will fail.

Performing a journey hotfix

In order to leverage on the current DevOps Jenkins Workflow pipeline, a dedicated hotfix workflow pipeline has been created to assist in building/testing/deploying hotfix's to Journey's. As this is a subtly different change to the normal feature workflow, the following steps **must** b e followed.

Hotfix-Start

1. Hotfix process requires two branches:
    a. **hotfix-feature-branch** where dev code changes are made.
    b. **hotfix-release-branch** where the changes will be released to and release-tag will be created from, hotfix-release-branch is used instead of develop branch for journey hotfixes.
    c. Initially these branches are created from the same commit Id which correspond to the release tag that requires the fix.
2. Create both hotfix branches by running **Jenkins-Job:workflow-v2-hotfix-start**
    a. http://jenkins-cma.n.mes.corp.hmrc.gov.uk/job/workflow-v2-hotfix-start/
    b. Following example shows parameters used to create hotfix branches for release-tag-1.0.0 of TST01 (sample journey)
    c. **Note:** *Hotfix release version* should be a sub-version of the release tag to be fixed so fixing release 1.0.0, a hotfix release version of 1.0.0.1 should be used to differentiate the release as being a child of 1.0.0
    d. **Note:** *Hotfix feature branch* must follow hotfix/<date>_<time>_<jira-ticket>_<fix-title> schematics used for feature branches as Workflow asserts this.



Parameters

repository: git@git.n.mes.corp.hmrc.gov.uk:programme/pods/services/journey-tst01-service.git

Required:: Repository Url.

version: release-tag-1.0.0

Required:: Journey Release Version to create HotFix from.

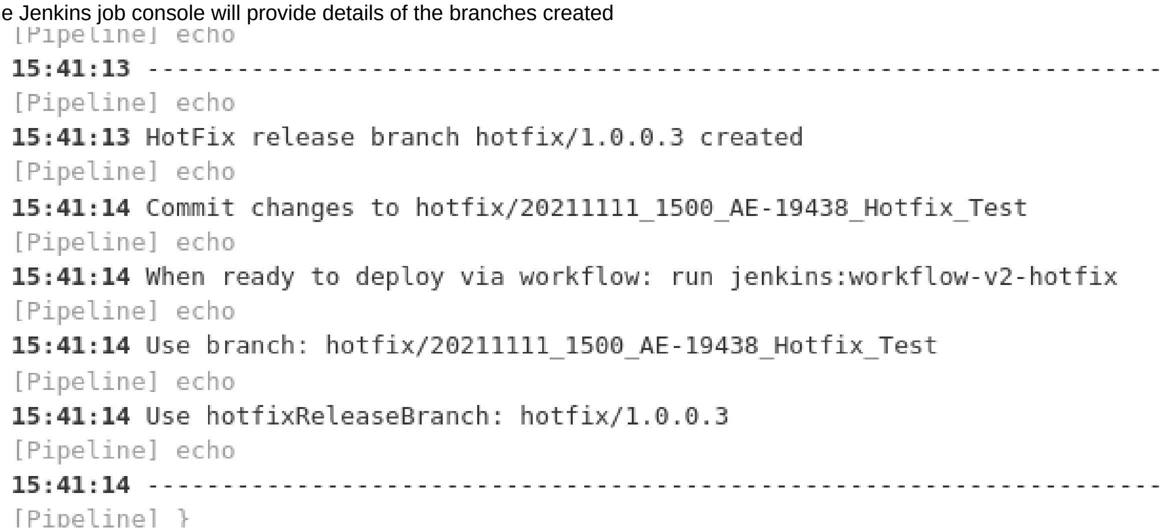
hotfixVersion: 1.0.0.3

Required:: HotFix release version, which should be a sub increment of version.

hotfixFeatureBranchName: hotfix/20211111_1500_AE-19438_Hotfix_Test

Required:: Required:: Feature Branch Name for Hotfix

3. The Jenkins job console will provide details of the branches created

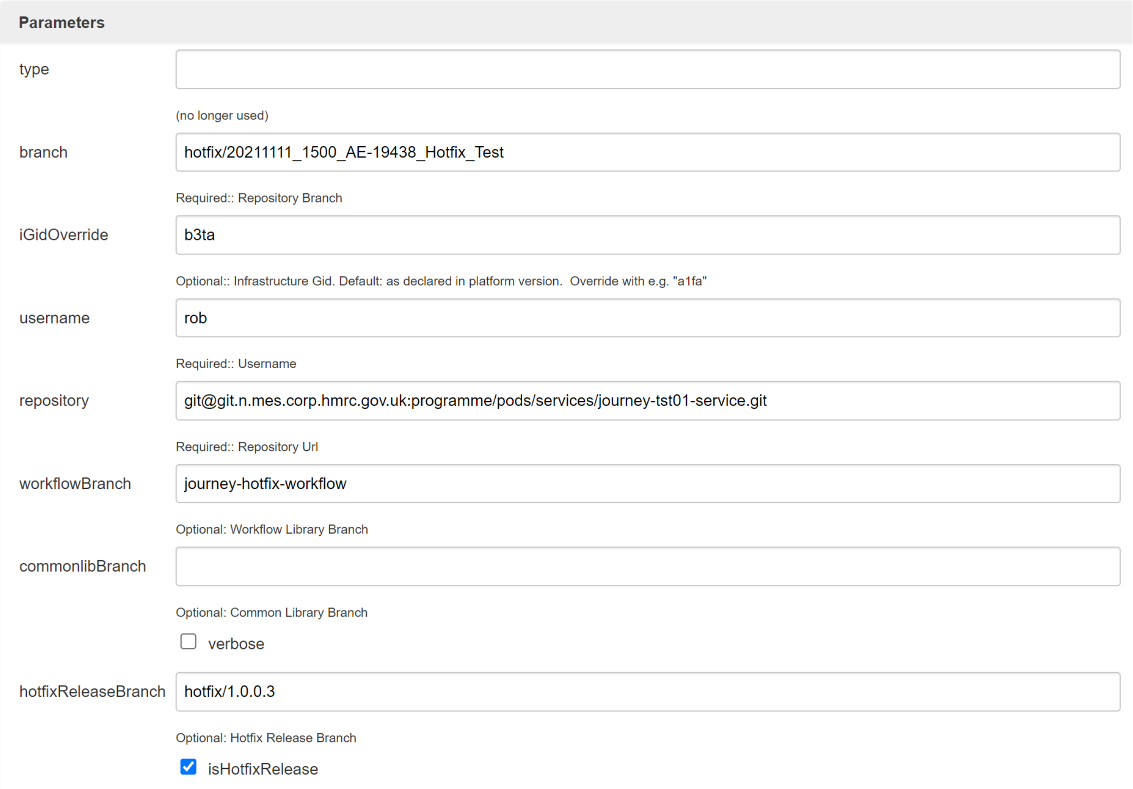


```
[Pipeline] echo
15:41:13 --------------------------------------------------------------------
[Pipeline] echo
15:41:13 HotFix release branch hotfix/1.0.0.3 created
[Pipeline] echo
15:41:14 Commit changes to hotfix/20211111_1500_AE-19438_Hotfix_Test
[Pipeline] echo
15:41:14 When ready to deploy via workflow: run jenkins:workflow-v2-hotfix
[Pipeline] echo
15:41:14 Use branch: hotfix/20211111_1500_AE-19438_Hotfix_Test
[Pipeline] echo
15:41:14 Use hotfixReleaseBranch: hotfix/1.0.0.3
[Pipeline] echo
15:41:14 --------------------------------------------------------------------
[Pipeline] }
```

4. You can now apply your changes to the feature branch hotfix/<date>_<time>_<jira-ticket>_<fix-title>

Jenkins Workflow-v2-Hotfix

1. **IMPORTANT:** Update the jgitflow plugin version to 1.3 **<jgitflow-maven-plugin.version>1.3</jgitflow-maven-plugin.version>.**
    a. Without this plugin update, JGitFlow will attempt to merge changes into develop and master branches and fail due to merge conflicts.
    b. Do not update parent pom to default to this value, jgitflow-maven-plugin version 1.3 is a Beta version
2. You may also need to update journey platform version to the same one used in production, and check that the stack also matches. This will only be applied in personal/dev. To set CI platform to production version you will need to talk to devops.
3. Once changes are ready to deploy to rancher, Jenkins Job workflow-v2-hotfix should be used. This basically swaps the hotfix-release-branch such as hotfix/1.0.0.3 for develop in the process, and does not merge result to develop or master on completion to avoid merge conflicts.
4. http://jenkins-cma.n.mes.corp.hmrc.gov.uk/job/workflow-v2-hotfix/
5. Set mandatory **params**: **repository**, **branch**, **hotfixReleaseBranch & user** using information given in output of **Jenkins-Job:workflow-v2-hotfix-start**
6. **IMPORTANT:** For now accept default value for workflowBranch which should be '**journey-hotfix-workflow**', once we are happy this hotfix strategy works, we will ask DevOps to merge the given workflowBranch into master.
7. Ensure **isHotfix** is ticked



Parameters

type

(no longer used)

branch: hotfix/20211111_1500_AE-19438_Hotfix_Test

Required:: Repository Branch

iGidOverride: b3ta

Optional:: Infrastructure Gid. Default: as declared in platform version. Override with e.g. "a1fa"

username: rob

Required:: Username

repository: git@git.n.mes.corp.hmrc.gov.uk:programme/pods/services/journey-tst01-service.git

Required:: Repository Url

workflowBranch: journey-hotfix-workflow

Optional: Workflow Library Branch

commonlibBranch

Optional: Common Library Branch

☐ verbose

hotfixReleaseBranch: hotfix/1.0.0.3

Optional: Hotfix Release Branch

☑ isHotfixRelease

8. Run workflow as normal through to end of dev deployment pipeline.
9. Hotfix-Workflow will prompt for **branch** to be merged into **hotfixReleaseBranch,** please do so via GitHub Merge-Request, once this has been done you can proceed into CI pipeline.
10. Hotfix-Workflow will run jgitflow:hotfix-finish and push the result to GIT. A release tag will be created and built modules pushed to artifactory.
11. Hotfix-Workflow will build the release docker image and push to artifactory before preceeding to deploy to Rancher CI Environment.
12. The workflow will conclude the CI pipeline once GenericTests have successfully completed.
13. The **hotfixReleaseBranch** will be left in GIT and will need to be manually merged and released to develop/master.

CI pipeline: merge conficts Ensure jgitflow maven plugin version is set to 1.3 or higher in journey pom. jGirFlow should not be trying to merge changes to either develop or master

Merge Hotfix-Release-Branch to Develop/Master

1. Create a normal development feature branch from **hotfixReleaseBranch** which can be found in the GIT repository
2. Release the feature branch to develop/master via normal Jenkins: Workflow-v2 Job, this ensures generic tests suite is passed
3. You can delete the **hotfixReleaseBranch** once merged to develop

Rancher

Their are several potential issues with the deployment of older journey releases to Rancher. Some DevOps support or a dedicated Rancher instance may be required to facilitate deployment.

Potential Issues

- Deployment of versions less than currently deployed (i.e. replacing GER01:1.0.4 with GER01:1.02.1)
- Deployment of versions where Live Platform version is less than that deployed to CI/NFR/SIT envs.
- Live Stack or infrastructure grid versions being lower then CI/NFR/SIT

You may require help from your friendly DevOps expert to facilitate deployment in Rancher, Good Luck!.

Backend Mock changes

If the hotfix requires changes to the backend mock, but the backend version is kept as is, then you will need to restart Tomcat pods to pick up the latest version of the mock war since the name has not changed. e.g. mock_XRS_getexchangeratexrs_v1.0.war takes its version from the backend spec it is mocking, unless that backend spec changes its version, an updated mock war will simply be uploaded to artifactory with the same name/path. The only way to install this into tomcat replacing the previous version is to restart the Tomcat pod.

Externalised Deployment Config

Many journeys now use externalised deployment config, which may need changes which are specific for the hotfix. This will need to be managed to minimise deployment conflicts. Generally you should suspend working / deployment of other active feature development on the journey while the hotfix is being deployed and tested.