If you don't know what git hooks are, this tutorial is probably not relevant to you. Text below explains how you can synchronize various business-central git repositories into the central public git repo. This can be helpful when there are multiple developers on the projects.

Git hooks support has been introduced since B*MS 6.2.0.GA. Please note that currently only "post-commit" hook is supported.
You can simply put your hook script under
$JBOSS_HOME/bin/.niogit/<repo-name>.git/hooks directory.
The following procedure is an example of "push when post-commit".
1) [Authoring]->[Administration]
2) [Repositories]->[Clone repository]
3) Provide information then click OK

Raw
```
Repository Name: <as you like>
Organization Unit: <as you like>
Git URL: <your-https-url> for example,
https://github.com/<username>/<repo-name>.git
User Name: leave empty
Password: leave empty
```

use https or git protocol here.

4) Go to the created repository
Raw
```
$ cd $JBOSS_HOME/bin/.niogit/<repo-name>.git
```

5) Change the remote URL
Raw
```
$ git remote set-url origin git@github.com:<username>/<repo-name>.git
```

Make sure that you can access the remote repository via ssh in command line. (e.g. the private ssh key for this remote repo exists under ~/.ssh)
If you are using github, make sure the public key is uploaded to github as explained in:
https://help.github.com/articles/adding-a-new-ssh-key-to-your-github-account/

6) Confirm if the remote URL was set correctly

Raw

```
$ git remote -v
origin  git@github.com:<username>/<repo-name>.git (fetch)
origin  git@github.com:<username>/<repo-name>.git (push)
```

7) Create a file named "post-commit" with executable permission (rwxr--r--) under $JBOSS_HOME/bin/.niogit/<repo-name>.git/hooks with the following content.

Raw

```
#!/bin/sh
git push origin master
```

On Windows you probably need to create post-commit.bat executable script file, with "git push origin master" content. Make sure git utility is in your PATH. Note this is not tested as I don't have access to Windows OS.

8) Confirm that 'push' is issued when you create a new "Guided Rule" via business-central.

Once you configure the above, following will be achieved:

-> Every change in your jBPM/BPM Suite repository will be immediately pushed to both:

Local git repository

Remote (i.e. github) repository

Example workflow:

- Imagine there are two developers in the project team, both are using business-central for process and rule authoring.
- First developer can create a git repository in business-central and create an empty project.
- He can also create an empty public git repository and initialize it with the empty project created earlier:

```
- $ git clone <mygithub.repo>
- $ git clone .niogit/pv207.git
- $ cp -r  myInitialRepo/myEmptyProject myGithubRepo/
- $ gd myGithubRepo
- $ git add myEmptyProject/
- $ git commit -m "initializing github repo"
- $ git push
```

- Now he can configure the post-hook script as explained earlier in this document

- Now second developer can start authoring - he can clone the public github repo and he needs to configure it with post-commit hook too (this is important, all developers needs to configure git hooks!)

- Now imagine first developer is doing some changes in his business-central pv207.git repo. All saves are propagated to the "master" github repository too, because of the git hooks - but they are not propagated automatically into the local git repository of second developer!

- Second developer come few hours later, because he needs to author some process

- Remember, when he cloned the github repository earlier, business-central created an internal git repo, which does not include the latest changes done by the first developer. The latest changes are present only in the master github repo.

- He can fetch the latest commits from the centralized github repo by this command:
  ```
  - $ cd .niogit/myPV207.git
  - $ git fetch
  ```

- Now he can log in to the business-central and do some authoring…

- Afterwards, first developer will come, and again..he will do `git fetch` to stay up to date with the master github repo, and he will continue with work.

- So remember, if you decide to use git hooks, you need to do `git fetch` as a first thing before you log into the business-central! Only this will ensure consistent state of your data.