

# Utilisation d'un client Git en mode graphique avec une Puppy Linux.

Pré-requis :

- distribution puppy precise 5.7.1 ou asri alpha 310.
- jre-1.7u67-i586\_310.sfs (dépôt asri)
- devx\_asriedu310.sfs (intègre le client git -dépôt asri)
- smartgithg-6\_0\_6\_310.sfs (dépôt asri)

Installation des packages dans la puppy linux :

- Monter les fichiers avec l'extension sfs avec sfs-load.

Configuration du client Git:

- Désactiver la vérification SSL (permet d'éviter le problème de chargement du certificat dans SmartGit/Hg), lancer les commandes :

```
git init
```

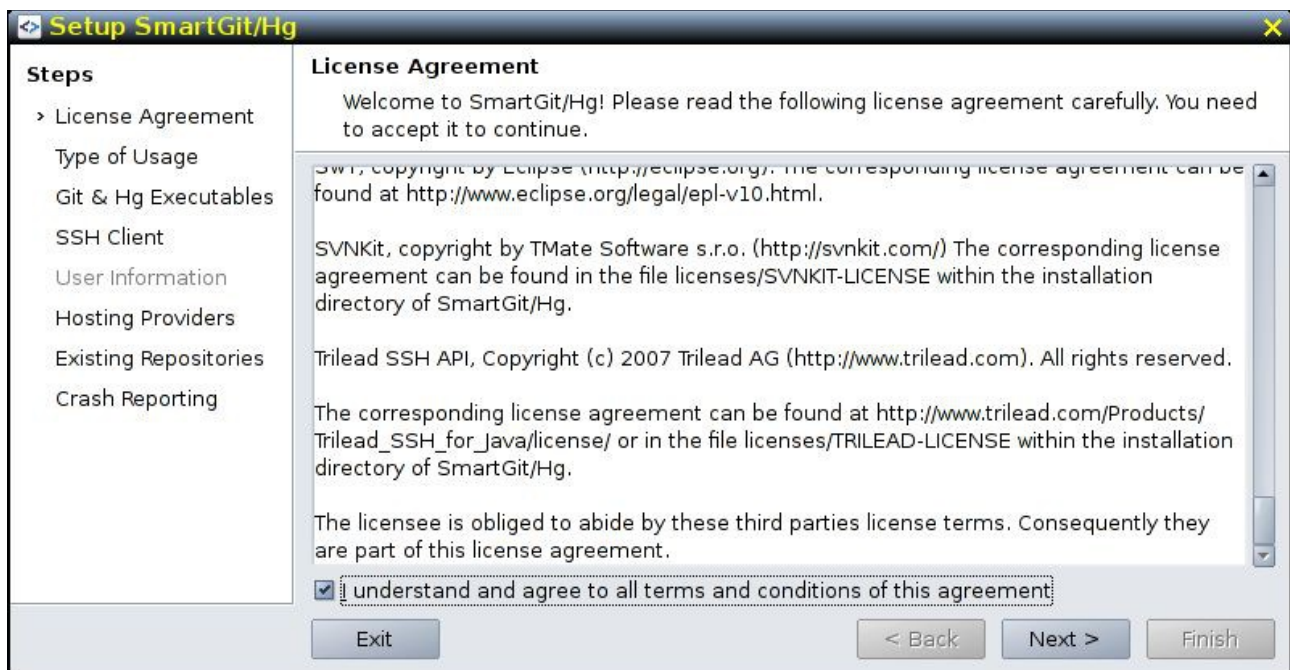
```
git config --global http.sslverify "false"
```

Assistant de configuration:

- Cet assistant n'apparaît qu'au premier démarrage du client.

- Validation de la licence:

Cocher la case après avoir lu et validé le contenu de la licence.

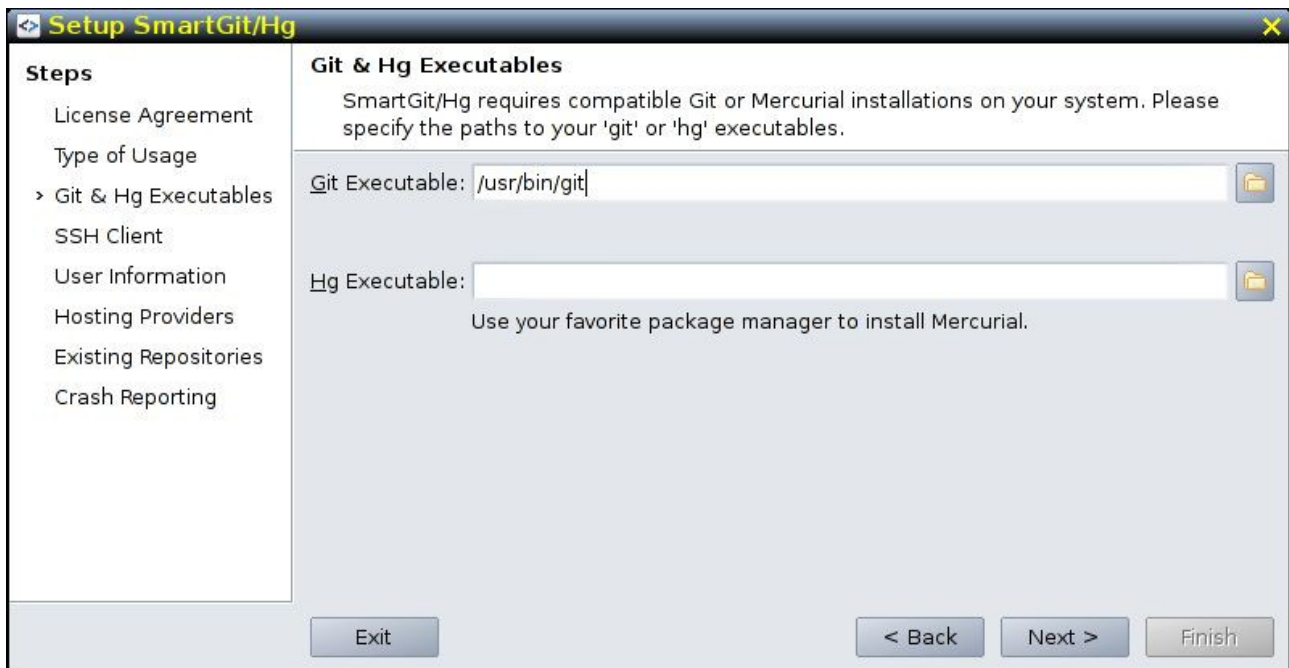


- Sélection du type de licence:



- Sélectionner le répertoire de l'exécutable Git:

Si aucune valeur n'est saisie dans le chemin de l'exécutable Git et que le bouton 'Suivant' est sélectionné, le programme tentera de détecter automatiquement l'emplacement de l'exécutable 'git'.

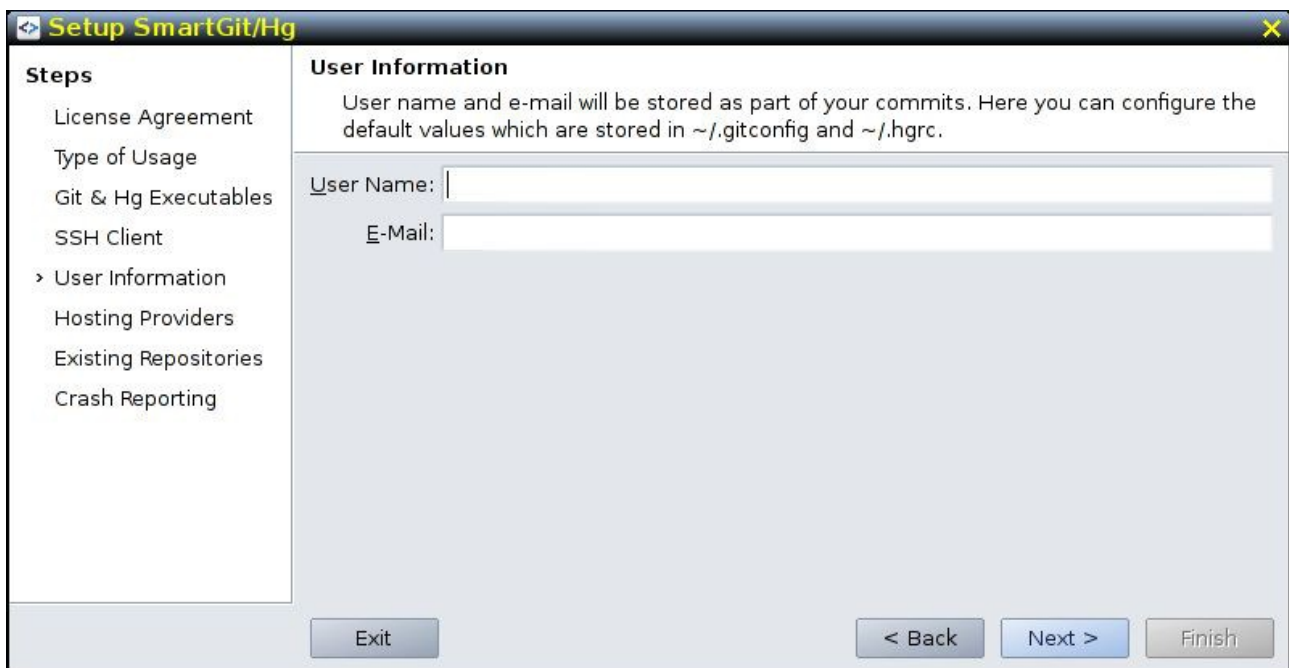


- Choix du client SSH:



- Saisir les informations utilisateur:

Si aucune information n'est saisie, l'utilisateur unix en cours au moment du commit sera utilisé.



- Sélection du fournisseur Git:

Sélectionner Github dans la liste déroulante.

Créer un compte Github si nécessaire sinon remplir le champ Account avec votre compte utilisateur Github,



Cliquer sur le bouton 'Generate API Token' pour autoriser la génération du token OAuth2.

Un mot de passe principal sera demandé, il sera utilisé dans l'application cliente pour protéger l'accès à votre compte Github à chaque démarrage de l'application. (il est conseillé d'utiliser un mot de passe différent de celui de votre compte github)



Le mot de passe de votre compte Github sera alors demandé afin de pouvoir générer le token.



Laisser désactiver l'option 'Use SSH instead of https to access repositories'.

- Spécifier l'emplacement d'un référentiel

Si un référentiel local Git est déjà disponible, vous pouvez le sélectionner, sinon vous pouvez passer cette étape.



Activation du rapport de crash:

Si l'option est active, un rapport sera renvoyé si un crash se produit.



Vous pouvez maintenant appuyer sur le bouton 'Finish'.

- Cloner/dupliquer un projet localement:

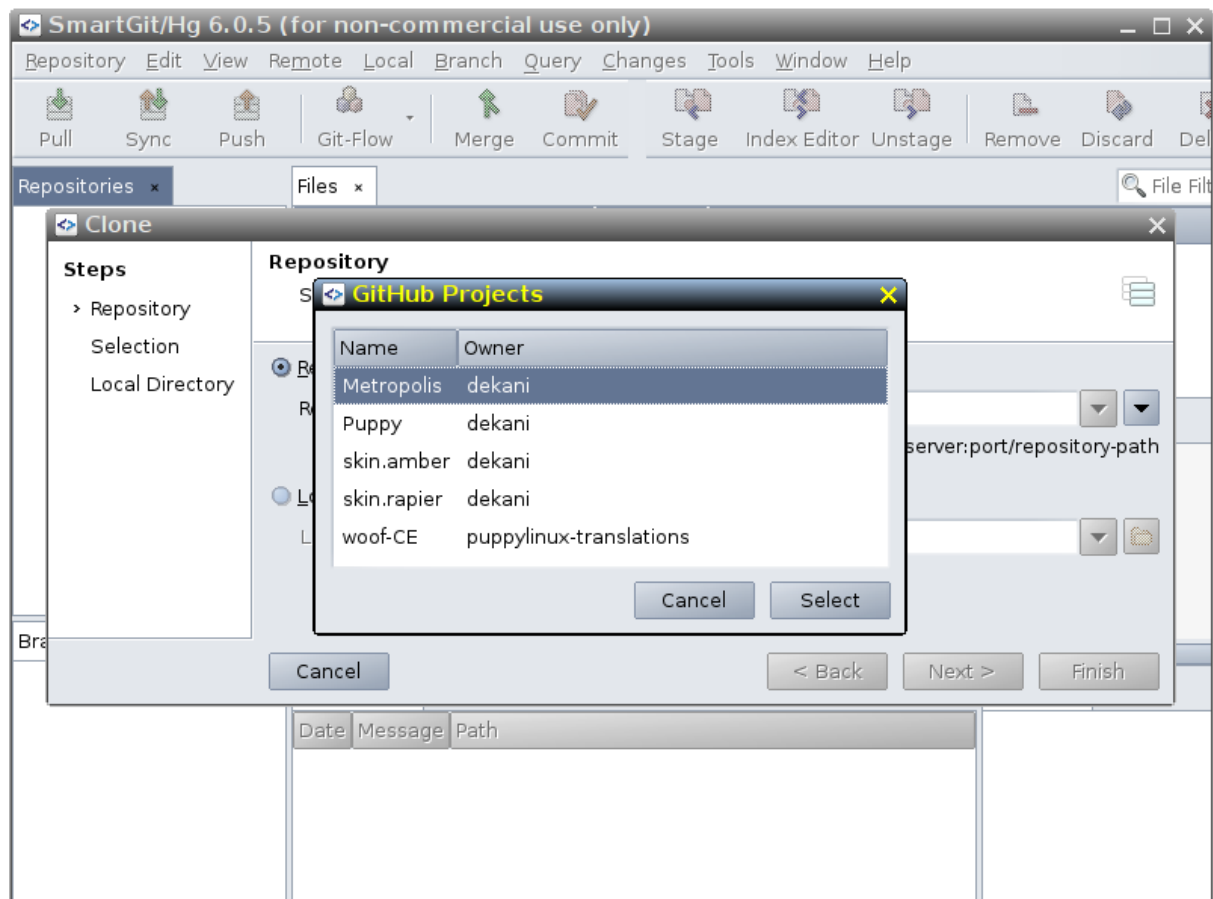
Sélectionner l'option 'Clone...' du menu 'Repository'

Cliquer sur bouton radio 'Remote Git' (référentiel distant)

Saisir l'URL de votre compte Github dans la zone de saisie.

Ex: <https://github.com/Compte Github/Puppy.git>

Cliquer sur le dernier bouton afin d'afficher le menu contextuel puis sélectionner le fournisseur 'github.com...'.  
Sélectionner le projet souhaité.

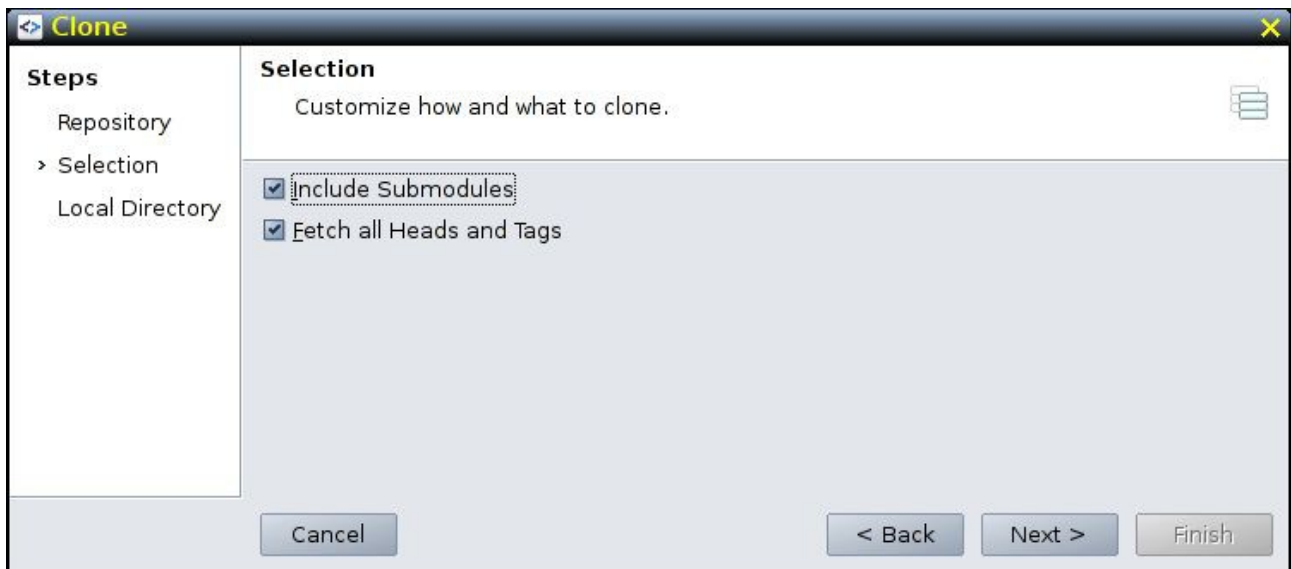


Votre mot de passe principal est demandé. (si vous en avez saisi un)

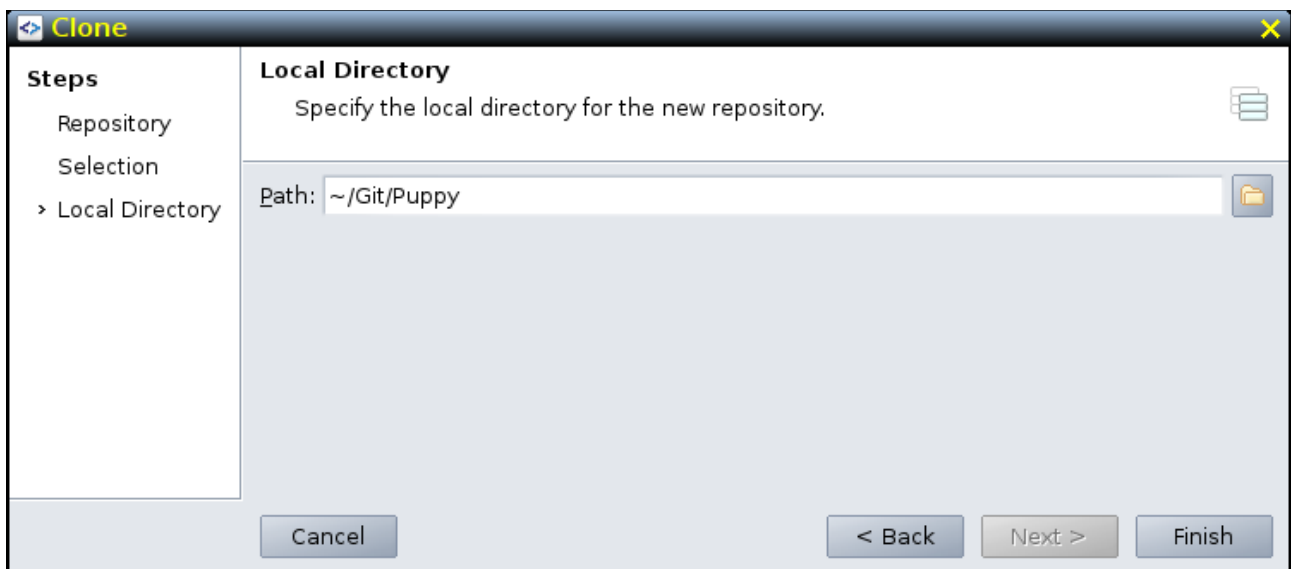
Puis la liste des projets associés à votre compte est affichée.

Sélectionner le projet souhaité.

Sélectionner les options du clonage:

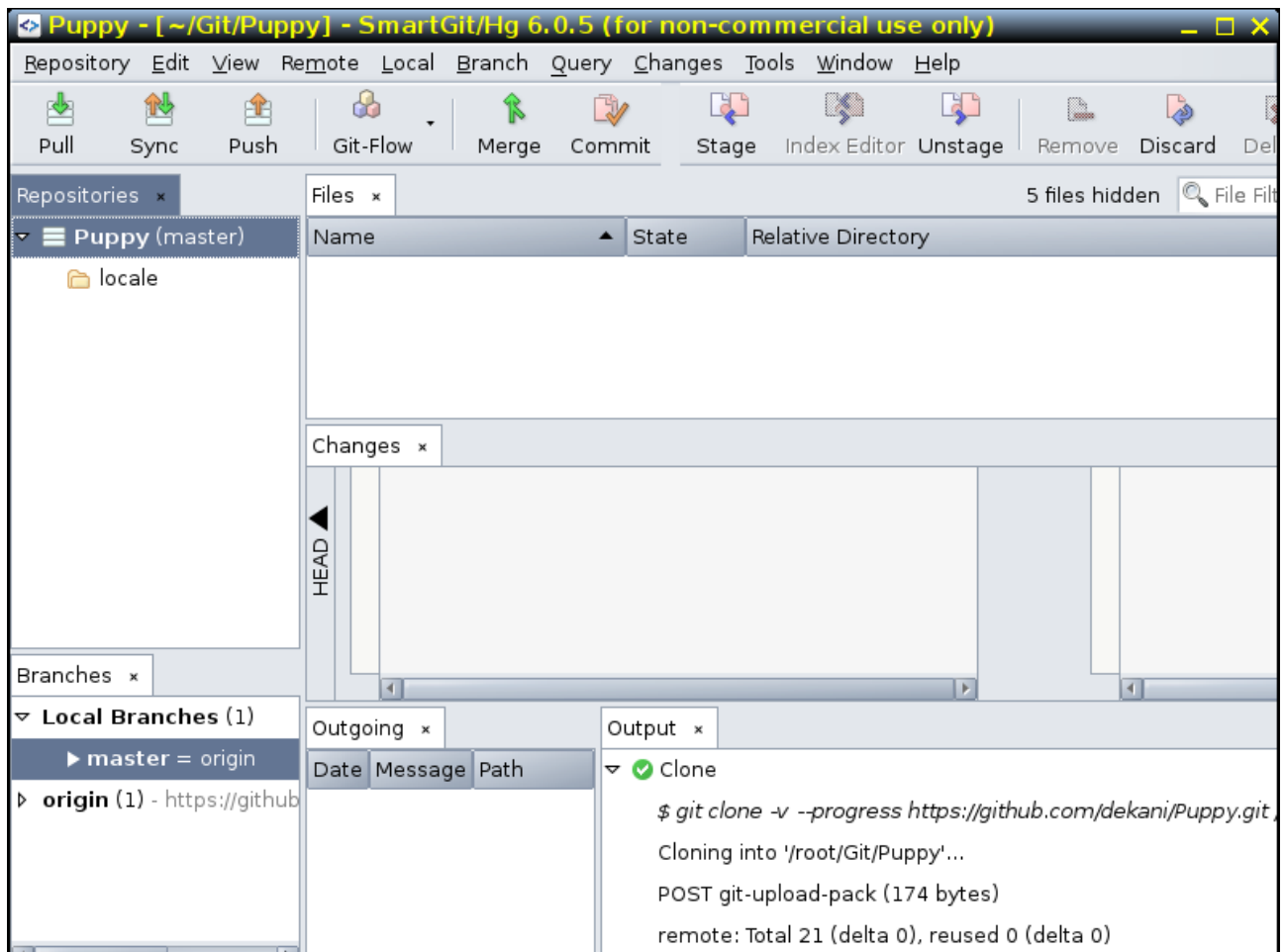


Sélectionner ensuite le répertoire local où sera cloné le projet:



Cliquer sur le bouton 'Finish' pour valider le clonage sur le disque local.

Le projet est ensuite affiché:

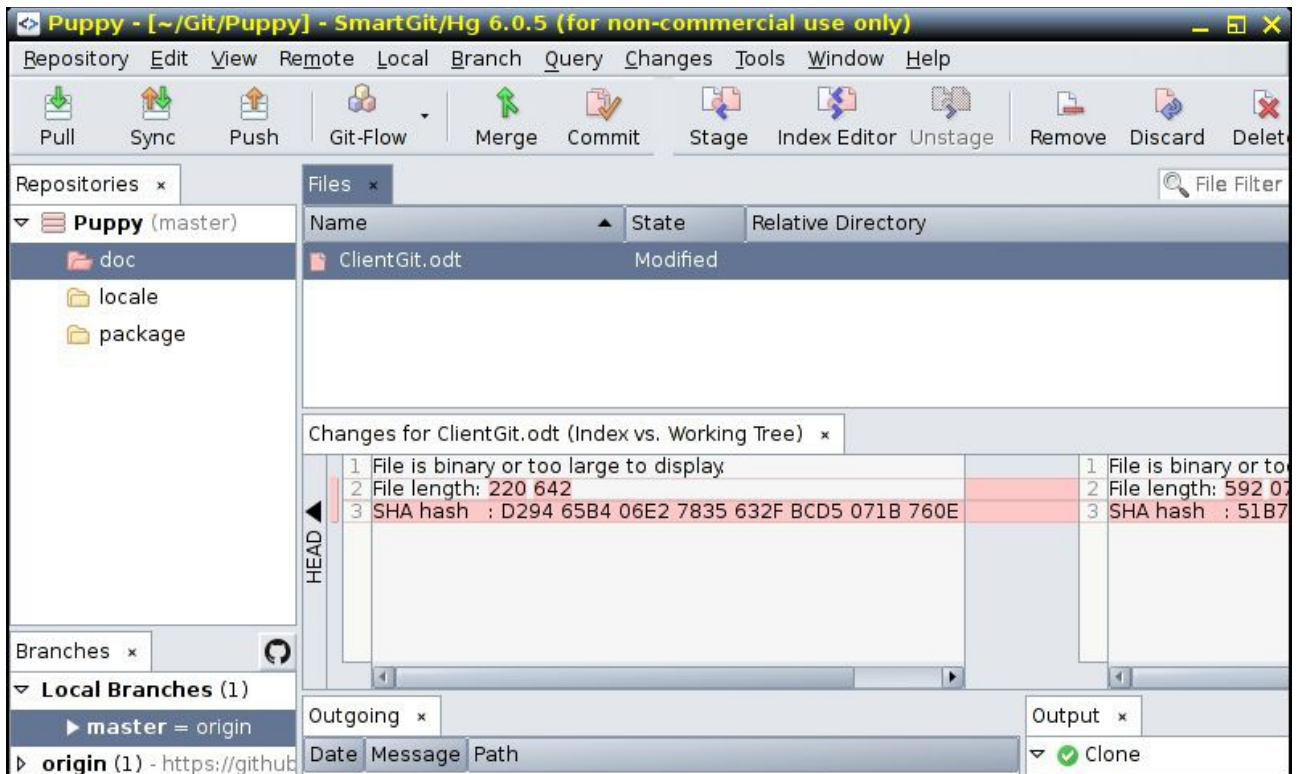


La fenêtre 'Output' affiche la liste des commandes Git lancées.

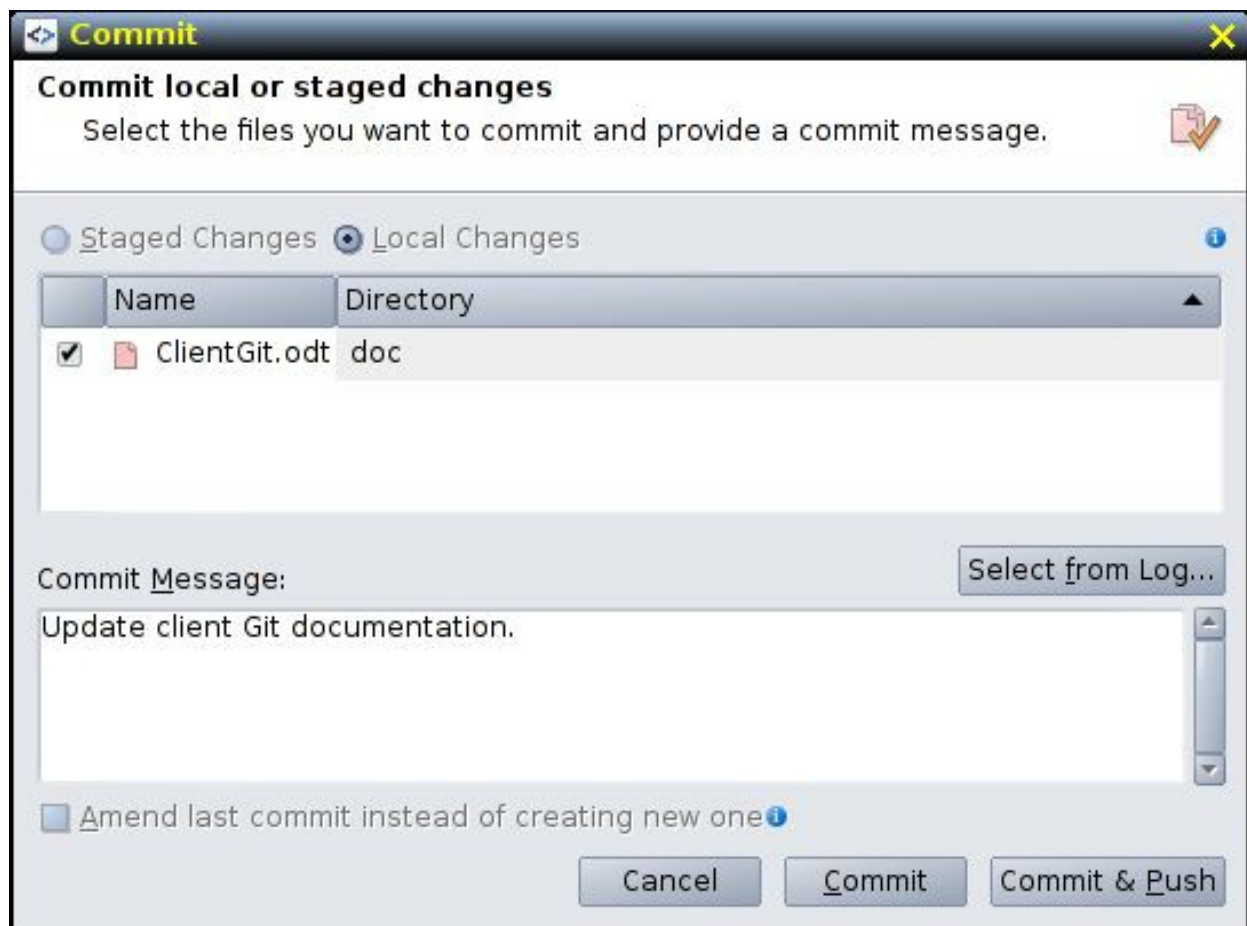


Validation des modifications.

Il est possible de rafraîchir l'écran principal afin de voir les modifications locales effectuées. Utiliser l'option 'Refresh' du menu 'View' ou le raccourci-clavier F5.



Vous pouvez valider les modifications (commit) en sélectionnant l'option 'Commit...' du menu 'Local' ou le raccourci-clavier Ctrl+K.



Pour pouvoir accéder aux boutons 'Commit', il faut saisir, dans la zone de texte 'Commit Message', un commentaire décrivant les modifications effectuées.

Il est possible d'annuler l'action (Cancel), valider les modifications (Commit) ou les valider et les envoyer directement vers le serveur Git. (Commit & Push)