Guide d’installation – Mac OS X

Sommaire

Pré-requis 3

Installations 5

Fink 5

Préparer Sirius 8

Difficultés rencontrées 9

Résultats 10

# Pré-requis

Se procurer la dernière version de Sirius sur le site officiel.

**Lien :** [**http://sirius.clarity-lab.org/downloads/#sirius**](http://sirius.clarity-lab.org/downloads/#sirius)

La figure suivante montre le résultat obtenu en accédant au lien ci-dessus.

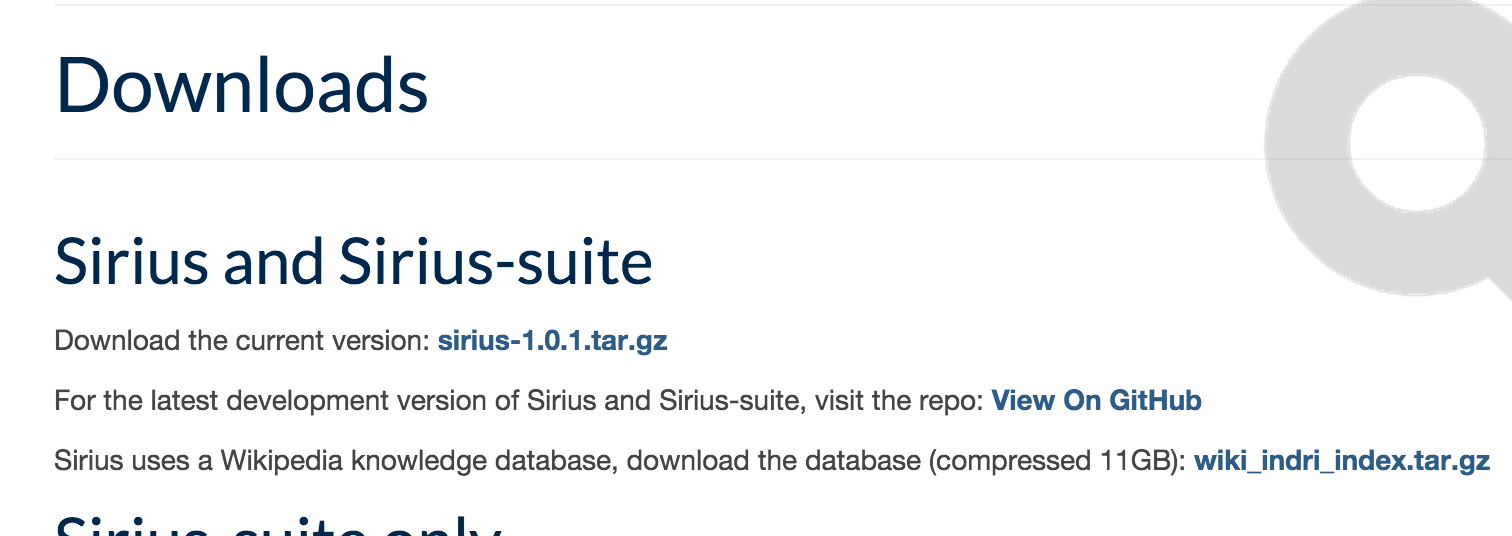


Figure Liens de téléchargement Sirius

Les éléments à télécharger sont :

* « Current version », la version 1.0.1 sur la **figure 1 ;**
* « Wikipedia knowledge database », la base de données de 11Go visible sur la **figure 1** également ;
* Le lien GitHub permet d’optimiser son niveau de mise à jour de Sirius, à chaque évolution il est possible de mettre à jour son projet ;

Sirius nécessite quelques prérequis pour fonctionner. Le script get-dependencies.sh permet de les mettre en place.

Voici la liste des modules utilisés par Sirius :

* [**Sphinx**](http://cmusphinx.sourceforge.net/) (sphinxbase and pocketsphinx)

**Lien : http://cmusphinx.sourceforge.net/wiki/download**

* [**Kaldi**](http://kaldi.sourceforge.net/)

**Lien : http://kaldi.sourceforge.net/install.html**

* [**Protobuf**](https://github.com/google/protobuf)

**Lien : https://github.com/google/protobuf**

* [**OpenCV**](http://opencv.org/downloads.html)

**Lien : http://opencv.org/downloads.html**

* **Java** (min. v1.7)
* **Fink**

**Lien : http://www.finkproject.org/download/srcdist.php?phpLang=en**

# Installations

## Fink

**Sirius** étant OpenSource, ses composants sont principalement des scripts et des commandes compatibles avec le système Linux GUI.

Certaines de ces commandes ne sont pas connues par les systèmes MacOS sans leur apporter les modifications adéquates. C’est pourquoi il est nécessaire d’installer **Fink** qui permettra à notre système de comprendre certains éléments qu’il n’était pas capable de comprendre au préalable.

Pour cela, rendez vous sur le lien fourni dans les prérequis

Une fois sur le site de Fink, plusieurs versions sont proposées comme on peut le voir dans la **figure 2**. Choisir la version adaptée à celle de votre système.

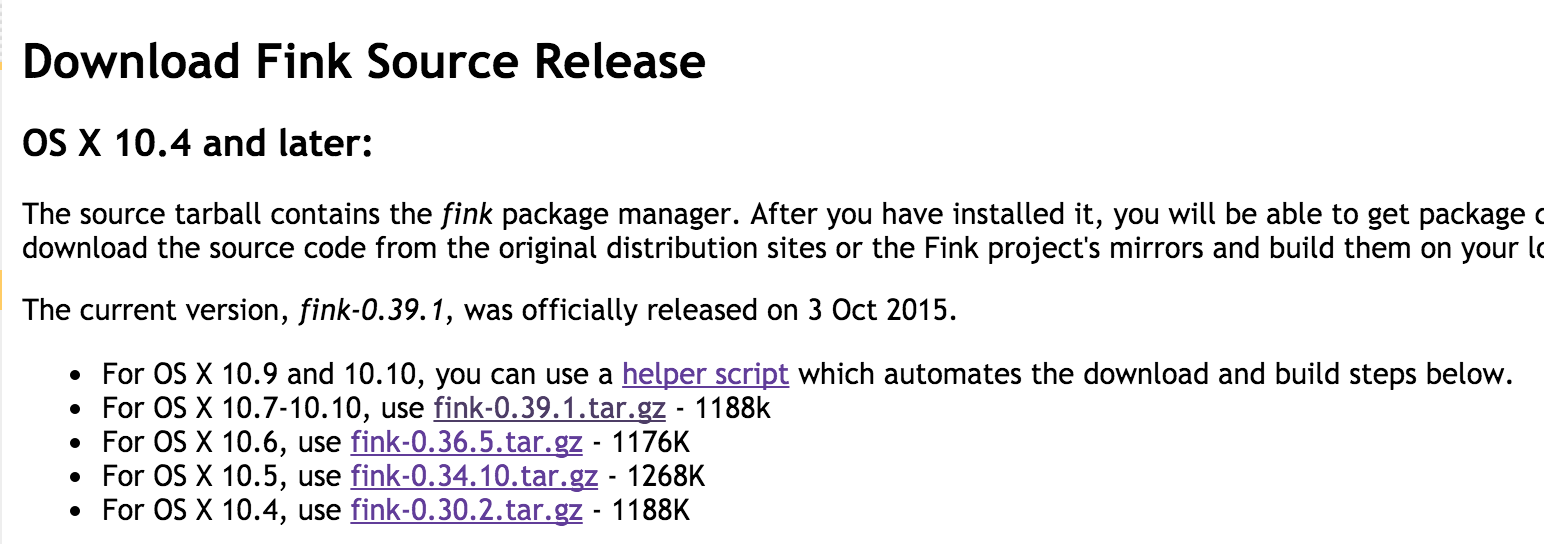


Figure Versions de téléchargement Fink

Dans notre cas, nous travaillons avec la dernière version de MacOS X El Capitan 10.11, prenons le fichier « fink-0.39.1.tar.gz » visible dans la capture ci-dessus.

Avant de procéder à son installation il est nécessaire d’installer des outils de ligne de commande Xcode. Pour cela veuillez taper les lignes suivantes dans un Terminal :

* Rendez vous dans le répertoire où se situe votre application Xcode.app (si vous ne l’avez pas déjà téléchargez la dans l’AppleStore). Elle se situe généralement dans votre répertoire « Applications »

Commande : cd /Applications

* Tapez ensuite

Commande : sudo xcode-select -switch ./Xcode.app/Contents/Developer

Commande : sudo xcodebuild -license

* Désarchivez le fichier que nous avons téléchargé. Pour cela rendez-vous dans le répertoire dans lequel il a été téléchargé

Commande : cd $HOME/Downloads

Commande : tar -xvf fink-0.39.1.tar.gz

Adaptez bien sûr vos actions à votre contexte (emplacement, nom, version… de vos fichiers ou répertoires)

* Les commandes qui suivent permettent d’installer et paramétrer Fink, il faut vous rendre dans le répertoire où il a été désarchivé

Commande : cd fink-0.39.1

Commande : ./bootstrap

Par défaut un répertoire est créé pour accueillir les sources d’installation : /sw

* Une fois l’installation complétée, il reste à configurer votre environnement et récupérer les éventuels patchs ayant pu être publié ultérieurement à la dernière version de Fink

Commande : /sw/bin/pathsetup.sh

Ouvrez un nouveau Terminal pour appliquer les changements apportés par la commande précédente, puis continuez avec la mise à niveau de Fink

Commande : fink selfupdate-rsync

Commande : fink index -f

A ce stade il est dorénavant possible d’utiliser des commandes Unix qui ne sont pas interprétables par un système MacOS.

## Préparer Sirius

Veuillez suivre les étapes suivantes pour préparer Sirius :

* Lancer un Terminal
* Vous placer dans votre espace de travail où vous avez téléchargé Sirius
* Décompresser le fichier « sirius-X.X.X.tar.gz »

Commande : tar xzf sirius-1.0.1.tar.gz

* Vous placer dans le répertoire « sirius-application »

Commande : cd sirius/sirius-application

* Décompresser le fichier « question-answer »

Commande : tar xzf question-answer.tar.gz

* Exécuter le script « get-dependencies.sh »

Commande : sudo ./get-dependencies.sh

* Exécuter le script « get-opencv.sh »

Commande : sudo ./get-opencv.sh

* Exécuter le script « get-kaldi.sh »

Commande : sudo ./get-kaldi.sh

* Exécuter le script « compile-sirius-servers.sh »

Commande : sudo ./compile-sirius-servers.sh

Les scripts get-xxx.sh servent à préparer l’installation complète établie par le script compile-sirius-servers.sh. Ce dernier compile tous les serveurs Sirius.

# Difficultés rencontrées

## Prise en main

Le fait de devoir installer plusieurs outils tiers permettant le bon déroulement de l’installation Sirius, ainsi que son bon fonctionnement, rend le processus de préparation difficile à prendre en main pour une personne novice.

En effet il y a beaucoup de commandes influant le comportement du système.

## Conflits entre les sources d’installation et le système cible

Effectivement, les scripts comportent des commandes Linux qui ne sont pas toute interprétables par un environnement MacOs et cela même avec la solution Fink.

Le résultat suivant expose le type de conflit cité :

$ sudo ./get-dependencies.sh

./get-dependencies.sh: line 7: add-apt-repository: command not found

./get-dependencies.sh: line 13: apt-add-repository: command not found

Hit <http://bindist.finkmirrors.net> stable/main Packages

Hit <http://bindist.finkmirrors.net> stable/main Release

Reading Package Lists... Done

Building Dependency Tree... Done

Reading Package Lists... Done

Building Dependency Tree... Done

Package git has no available version, but exists in the database.

This typically means that the package was mentioned in a dependency and

never uploaded, has been obsoleted or is not available with the contents

of sources.list

E: Package git has no installation candidate

Reading Package Lists... Done

Building Dependency Tree... Done

E: Couldn't find package build-essential

Reading Package Lists... Done

Building Dependency Tree... Done

E: Couldn't find package tesseract-ocr

Reading Package Lists... Done

Building Dependency Tree... Done

E: Couldn't find package libatlas-dev

Reading Package Lists... Done

Building Dependency Tree... Done

E: Couldn't find package libprotobuf-dev

./get-dependencies.sh: line 42: pip: command not found

Je rappelle que le script get-dependencies.sh met en place les dépendances nécessaires pour faire fonctionner Sirius.

# Résultats