

# Xcraft CheatSheet



Mathieu Schroeter, v1.2

23 juin 2021

## Pré-requis

- System d'exploitation de type Linux (au moins de la génération des GNU/Linux Debian Jessie), Apple macOS 10.9.5 ou Microsoft Windows 7 SP1
- Espace disque d'au moins **60GB**, idéalement **100GB** (cela dépend de ce qui sera construit)
- Du bon hardware (CPU moyenne à haute gamme, 16GB de RAM pour être tranquille et un SSD)

**Linux** Installer git et un compilateur c/c++ de base (build-essential sous GNU/Linux Debian par exemple)

**macOS** L'exécution de git depuis un terminal va automatiquement demander et permettre alors d'installer les outils de développement en ligne de commande<sup>1</sup>.

**Windows** Installer Git pour Windows et contrairement à Linux et macOS, il ne faut surtout pas installer de buildtools dans le système. Les buildtools peuvent difficilement cohabiter en parallèle avec des versions différentes. Ici, Xcraft se chargera d'installer ce qui est nécessaire pour la compilation. Cela implique qu'Xcraft doit être exécuté en administrateur sous Windows contrairement à Linux et macOS où il s'exécute uniquement avec les droits utilisateur.

## SSH

Attention, les définitions de paquets Xcraft faisant référence à [git.epsitec.ch](https://git.epsitec.ch), utilisent toujours le protocole SSH pour accéder aux dépôts. La raison est relativement évidente car une authentification par SSH ne nécessite pas de devoir sauvegarder des credentials dans une valise. Il suffit d'autoriser sa propre clef SSH pour le serveur GitLab et le tour est joué. Voir <https://git.epsitec.ch/-/profile/keys> et suivre les procédures très bien documentées par GitLab pour tous les systèmes d'exploitations.

## Bootstrap

L'étape de bootstrap se charge de mettre en place les outils de base pour l'exploitation d'Xcraft. Cela implique la compilation (sous \*nix uniquement, pré-compilé sous Windows) de nodejs, l'installation des dépendances node pour Xcraft, lancement de la compilation de CMake puis compilation de wpkg. C'est uniquement une fois que wpkg est compilé et déployé, qu'il est possible de travailler avec Xcraft.

---

<sup>1</sup>. au besoin, installer Xcode

## Linux & macOS

Depuis les systèmes de type Linux et Unix, tout le processus de bootstrap doit se faire avec les droits d'un utilisateur courant. L'utilisateur root n'est jamais requis. Il faut commencer par cloner le dépôt Git de bootstrap pour \*nix et exécuter le script de bootstrap depuis un terminal.

```
mkdir ~/xcraft
cd ~/xcraft

# utiliser https://git.epsitec.ch/xcraft/nixdev.git pour HTTPS
git clone git@git.epsitec.ch:xcraft/nixdev.git

cd nixdev
. ./bootstrap.sh
```

L'emplacement du dossier de déploiement d'Xcraft n'a aucune importance. L'utilisation du chemin ~/xcraft n'est qu'une suggestion. Il est possible que pendant le bootstrap, le système demande le login ainsi que le mot de passe pour l'accès à GitLab.

## Windows

Avec un système de type Windows, il est nécessaire d'effectuer le bootstrap dans un emplacement pas trop éloigné de la lettre de lecteur ainsi qu'en mode élevé (administrateur). L'utilisation d'Xcraft devra se faire aussi en mode élevé afin que les buildtools puissent être installés proprement dans le système. Il faut commencer par ouvrir un command prompt `cmd.exe` en mode administrateur<sup>2</sup>.

```
cd /d c:\
mkdir xcraft
cd xcraft

REM utiliser git@git.epsitec.ch:xcraft/windev.git pour SSH
git clone https://git.epsitec.ch/xcraft/windev.git

cd windev
bootstrap
```

## Zog SHell

L'exploitation d'Xcraft se fait principalement par l'intermédiaire du Zog SHell qui n'est rien de plus qu'un client aux bus Xcraft du toolchain. Ce même client étant utilisable par les applications Westeros. La différence avec les applications Westeros provient du fait que Zog est utilisé ici, aussi pour démarrer le serveur Xcraft.

**Linux & macOS** Ouvrir un terminal (ne pas utiliser le compte root).

```
cd ~/xcraft/nixdev/sysroot/xcraft
./zog
```

**Windows** Ouvrir un command prompt `cmd.exe` en mode administrateur.

```
cd /d c:\xcraft\windev\sysroot\xcraft
zog
```

---

2. ce mode est indispensable pour pouvoir installer les buildtools Microsoft

La compilation d'un produit peut prendre énormément de temps la première fois. Pour se faire il est nécessaire de générer l'ensemble des paquets (outils) permettant de générer puis compiler le produit. Quelques commandes impliquées sont brièvement présentées ici.

**pacman.make** permet de construire des paquets Xcraft à partir de définitions

**pacman.build** tente de compiler un paquet source en binaire tout en résolvant les dépendances

**pacman.install** installe un paquet Xcraft dans un root (de développement ou de distribution)

**pacman.remove** désinstalle un paquet Xcraft

**pacman.upgrade** met à jour un root en fonction des versions disponibles dans les dépôts de paquets

## Exemple pour Linux & macOS

Ces commandes permettent (si tout se passe comme prévu) de construire Crésus Comptabilité pour Linux ou macOS en partant de zéro. Cela implique la compilation de la majorité des outils nécessaires à la création du produit. Selon les caractéristiques de la machine utilisée, il faut compter certainement plusieurs heures avant d'atteindre le résultat.

```
pacman.make epsitec+crossover-crcompta,@deps
pacman.build epsitec+crossover-crcompta
pacman.install epsitec+crossover-crcompta
```

## Exemple pour Windows

Ces commandes concernent la génération de la matrice Crésus pour Windows ainsi que de différents sous-paquets qui seront impliqués à la génération des templates CrossOver pour Linux et macOS. Dans le cadre de Windows, tous les outils ne sont pas compilés pour divers raisons comme l'absence de sources ou plus simplement à cause de procédures de build quasiment inexistantes voir même complètement obsolètes ou « magiques ». Des installeurs vont être exécutés pendant ces phases de constructions. Il est possible que des problèmes apparaissent car le système peut parfois nécessiter un redémarrage. Si tel est le cas, il suffit alors de recommencer avec la dernière commande qui a échoué après un redémarrage de la machine.

```
pacman.make epsitec+cresus,@deps
pacman.build epsitec+cresus
pacman.install epsitec+cresus
```

## Exemple de problèmes courants

- La création de certains paquets se terminent en erreur car les URLs pour le téléchargement des sources ou des binaires ne sont plus valables (momentanément ou non)
  - A tenter dans cet ordre : re-exécuter le **pacman.make**, effacer le fichier en cache, changer l'URL
- La compilation sous Windows avec MSVC provoque des erreurs comme quoi **new.h** n'existe pas
  - Dans ce cas il faut tenter un redémarrage de la machine pour finaliser une des installation de buildtools
- La commande **pacman.make** semble complètement bloquée
  - C'est peut être juste que votre clef SSH n'est pas valable pour le téléchargement du dépôt et que Git attend sur une action

Xcraft est un projet très complexe et nécessite une certaine expertise, veuillez me communiquer les problèmes directement car ce document ne fait que survoler le sujet.