



eBook Gratuit

APPRENEZ

Codename One

eBook gratuit non affilié créé à partir des
contributeurs de Stack Overflow.

#codename

one

Table des matières

À propos	1
Chapitre 1: Démarrer avec Codename One	2
Remarques.....	2
Exemples.....	2
Installation et configuration.....	2
Installation	2
Installation du nom de code One In NetBeans.....	2
Installation du nom de code One In Eclipse.....	6
Installation de Codename One In IntelliJ IDEA.....	8
Qu'est-ce que Codename One et comment ça marche?.....	8
Comment fonctionne Codename One?	9
Pourquoi construire des serveurs?.....	10
Pourquoi ParparVM.....	11
Windows Phone / UWP.....	12
Port JavaScript.....	12
Desktop, Android, RIM & J2ME.....	12
Composants légers.....	12
Origine d'architecture légère.....	13
Dans le nom de code un.....	13
Versions dans le nom de code un.....	13
Crédits	14

À propos

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [codename-one](#)

It is an unofficial and free Codename One ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official Codename One.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

Chapitre 1: Démarrer avec Codename One

Remarques

Cette section fournit une vue d'ensemble de ce qu'est un nom de code et pourquoi un développeur peut vouloir l'utiliser.

Il devrait également mentionner tous les grands sujets dans le nom de code et les relier aux sujets connexes. Comme la documentation de nom de code est nouvelle, vous devrez peut-être créer des versions initiales de ces rubriques connexes.

Exemples

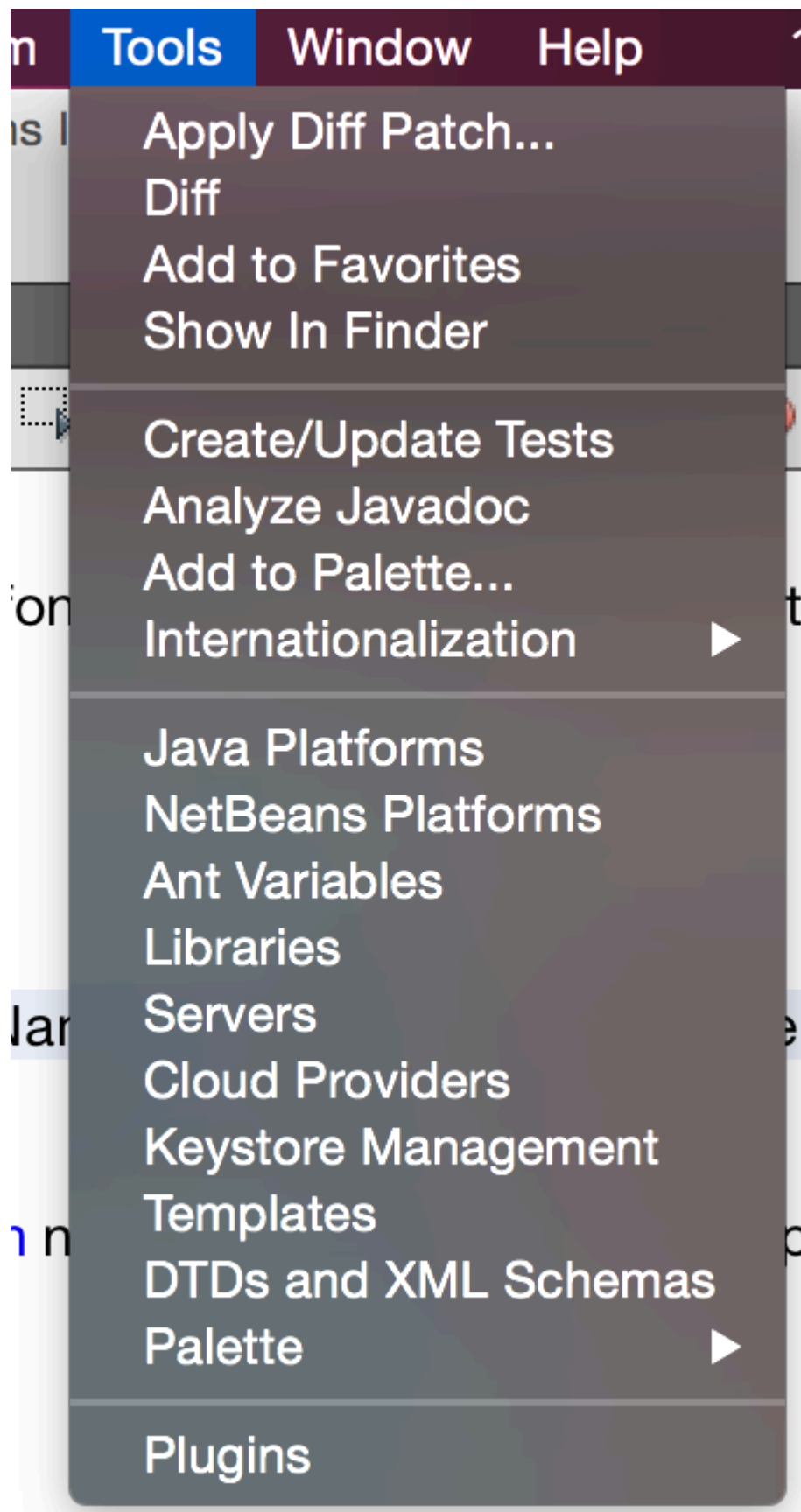
Installation et configuration

Installation

Installation du nom de code One In NetBeans

Ces instructions supposent que vous avez téléchargé une version récente de NetBeans (actuellement 8.x), que vous l'avez installée et lancée.

- Sélectionnez l'option de menu Outils-> Plugins



- Sélectionnez l'onglet Plugins disponibles
- Vérifiez le plugin CodenameOne



Welcome to the NetBeans IDE Plugin Installer

The installer will download, verify and then install the selected plugins.

The following plugins will be installed:

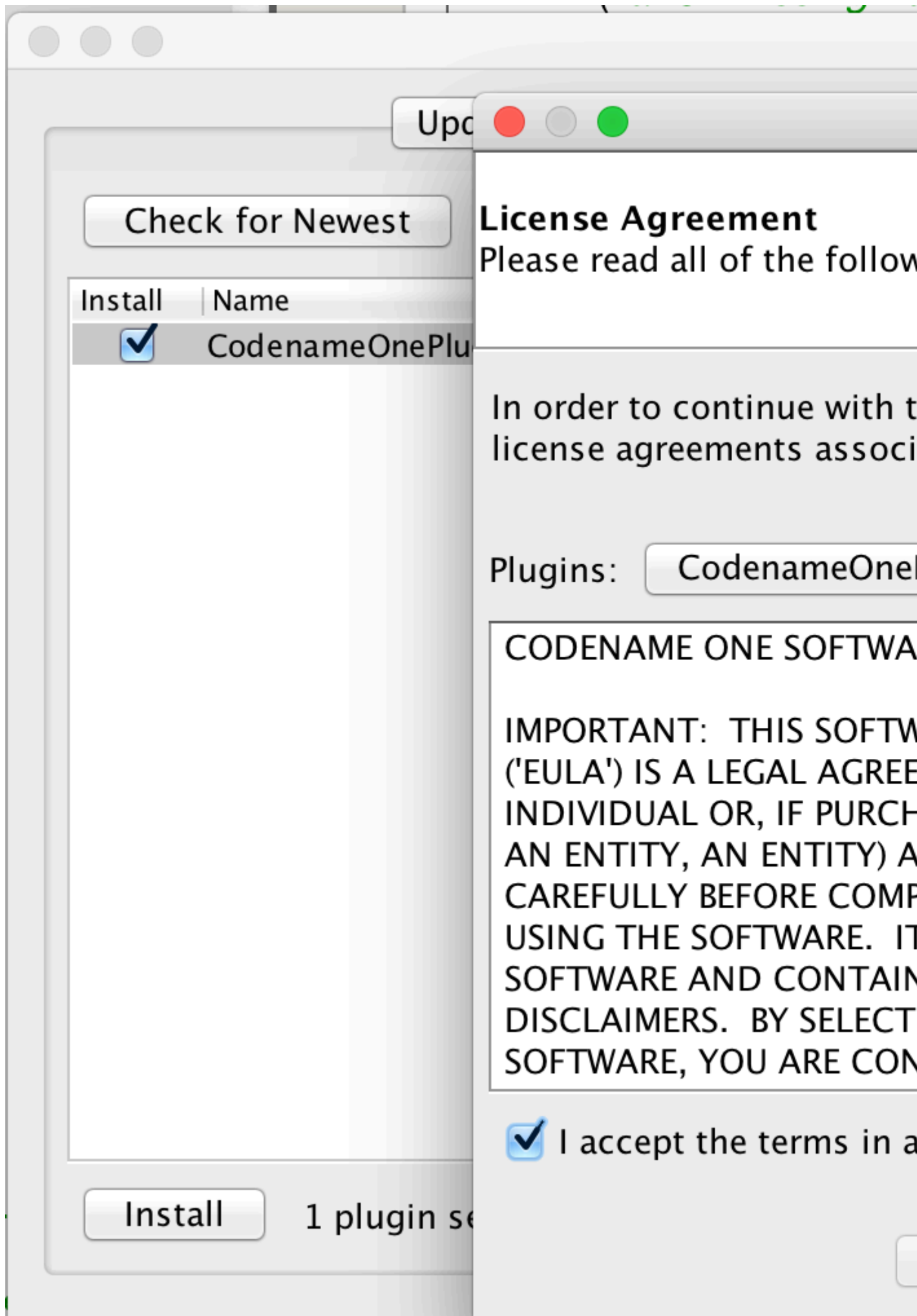
CodenameOnePlugin [3.2.6]

Help

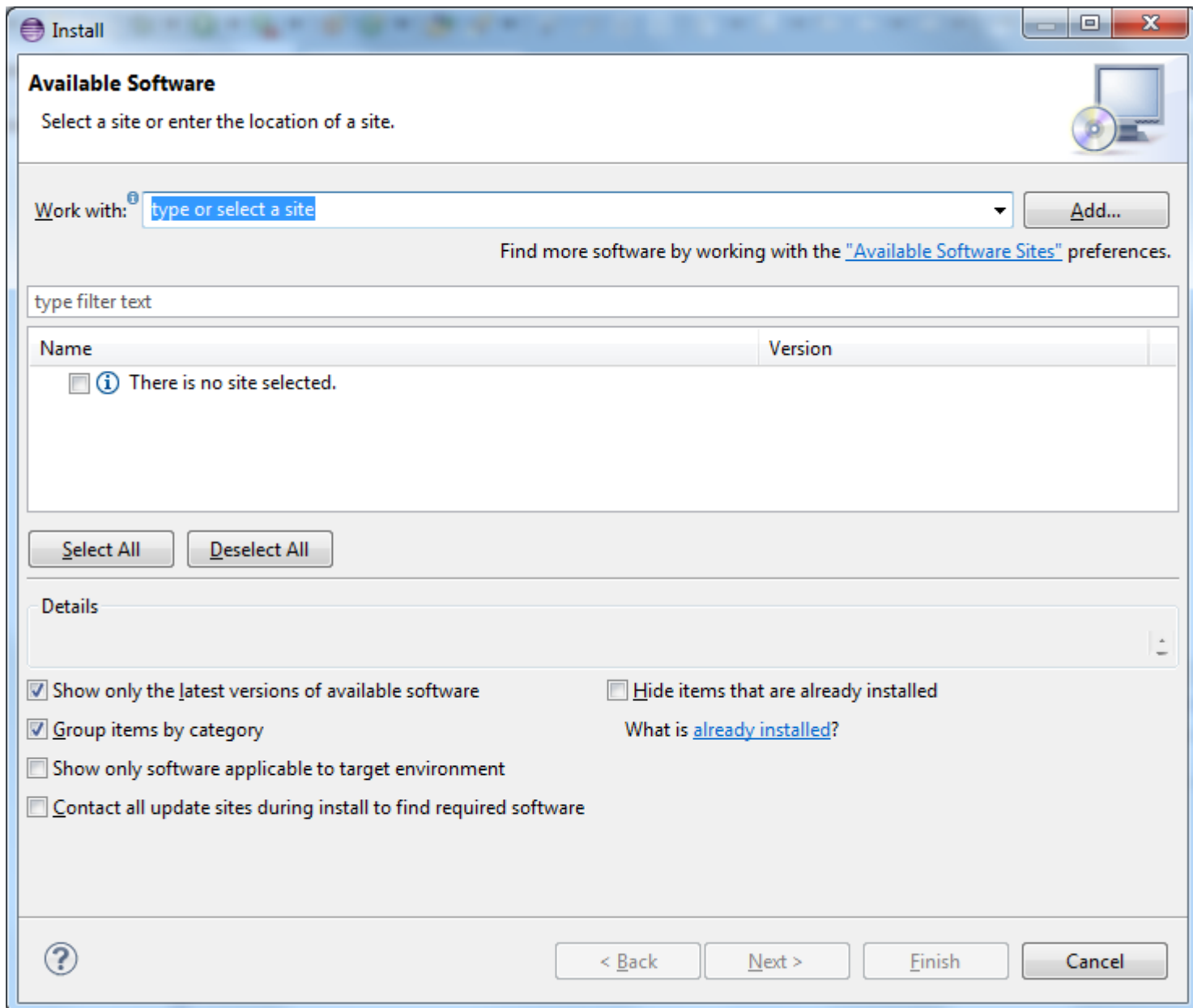
< Back

- cliquez sur le bouton d' `install` ci-dessous. Suivez les instructions de l'assistant pour

installer le plugin

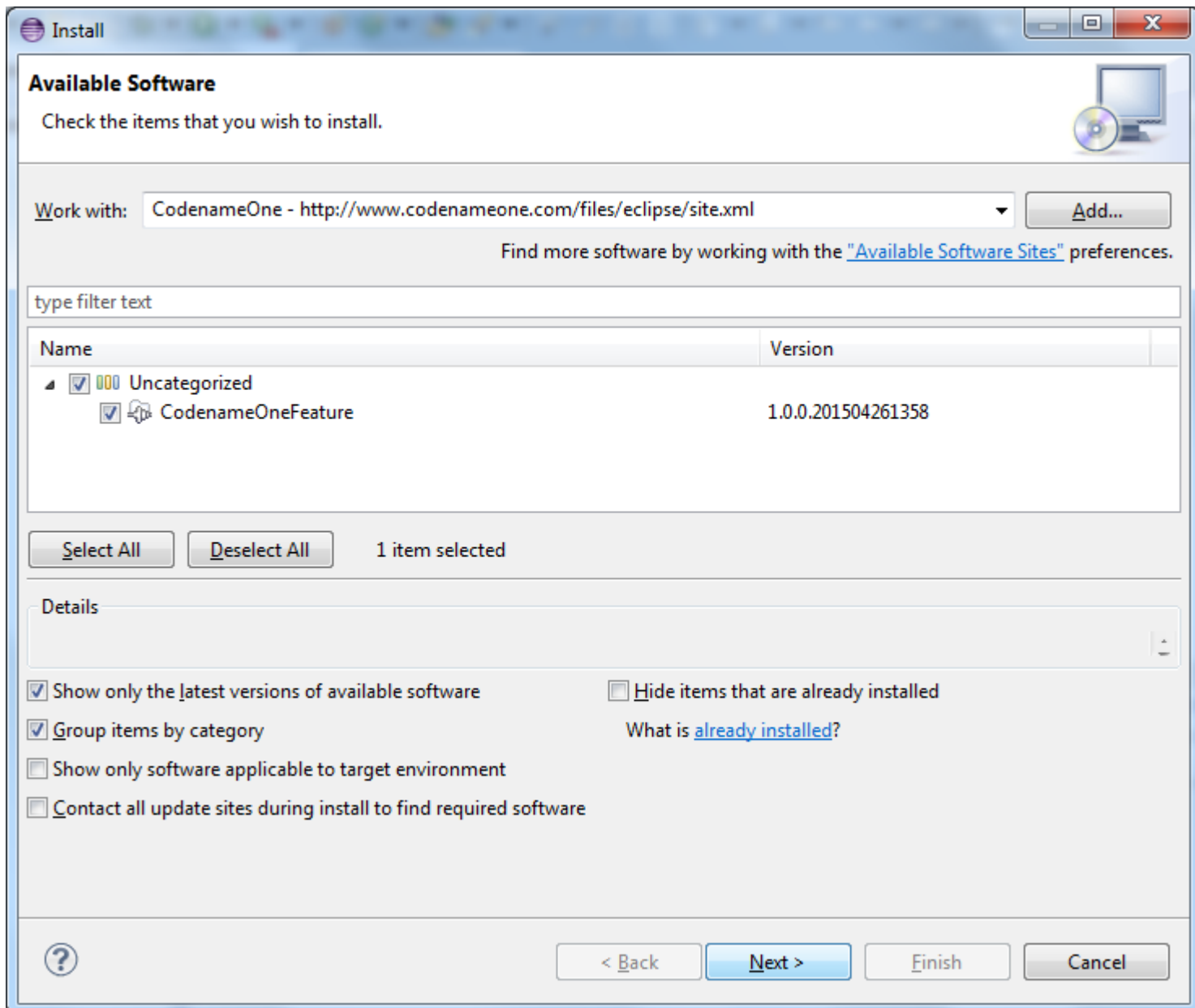


. Vous devriez avoir cette boîte de dialogue



Collez <https://www.codenameone.com/files/eclipse/site.xml> pour l'emplacement pour travailler avec et appuyez sur Entrée.

Sélectionnez les entrées et suivez l'assistant pour installer



Installation de Codename One In IntelliJ IDEA

Téléchargez et installez IntelliJ / IDEA. **Notez que Android Studio ne fonctionnera pas.**

Installez le plugin en utilisant le plugin center

Utilisez la fonctionnalité de recherche du centre de plug-ins pour rechercher et installer le plug-in Codename One.

Qu'est-ce que Codename One et comment ça marche?

Codename One est un ensemble d'outils pour le développement d'applications mobiles qui dérivent une grande partie de son architecture à partir de Java.

L'énoncé de mission de Codename One est le suivant:

Unifiez la tâche complexe et fragmentée de la programmation d'appareils mobiles en

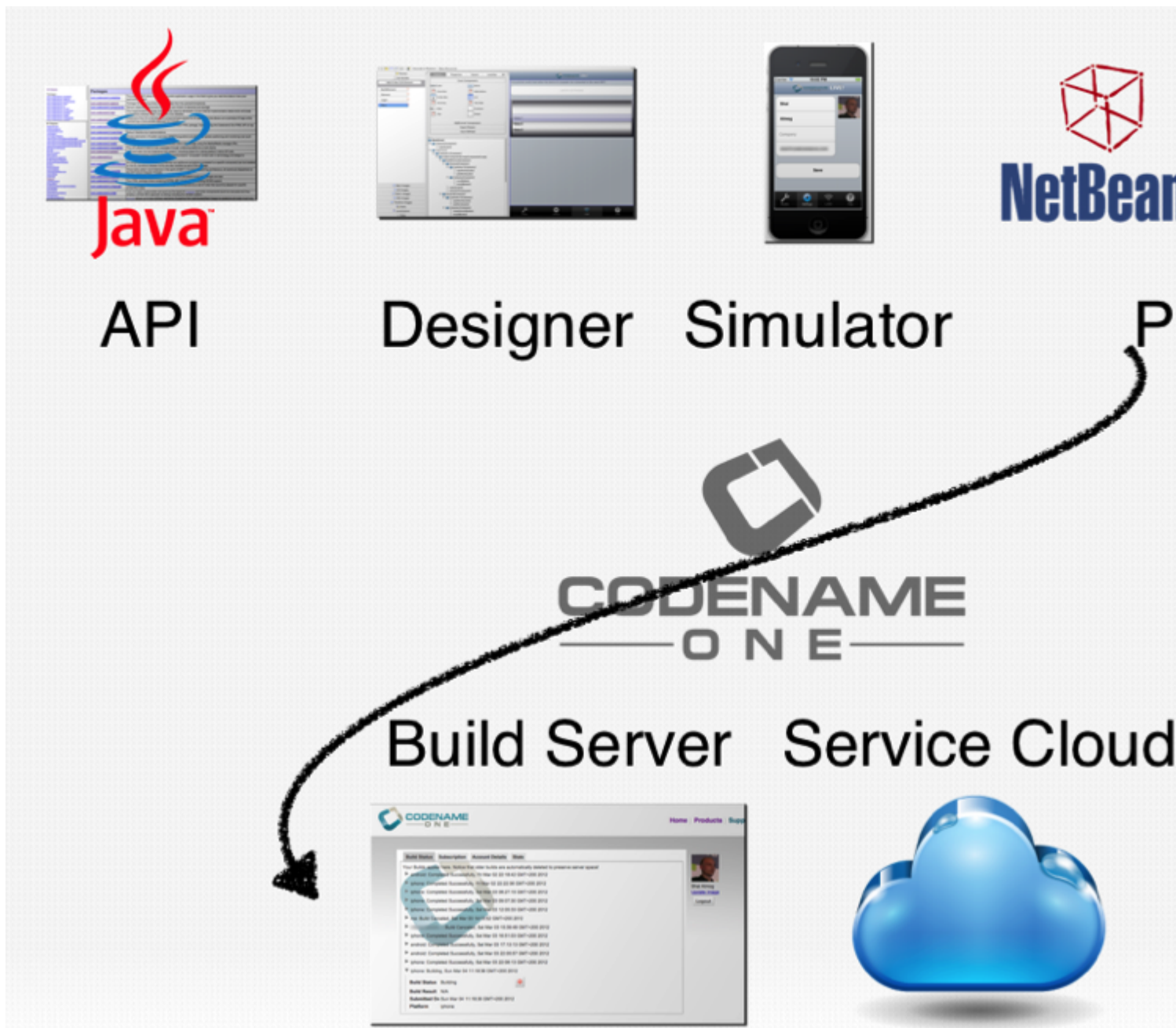
un seul ensemble d'outils, d'API et de services. En conséquence, créer une approche plus gérable du développement d'applications mobiles sans sacrifier la puissance / le contrôle donné aux développeurs.

Cela signifie que vous devez apporter ce vieux mantra Java «Write Once Run Anywhere» (WORA) aux appareils mobiles sans le «réduire» au plus petit dénominateur commun.

Comment fonctionne Codename One?

Codename One unifie plusieurs technologies et concepts en une seule façade:

- API - abstrait les différences entre les différents périphériques.
- Plugin - le seul logiciel installé sur les ordinateurs clients, il comprend les fonctionnalités suivantes:
 - Intégration IDE - Préférences, complétion, possibilité d'envoyer une version native
 - Simulateur - simulateur de périphérique natif qui s'exécute localement et permet de déboguer l'application
 - Designer / GUI Builder - outils de haut niveau
- Build Servers (Serveurs de génération) - Les serveurs de génération acceptent les générations de périphériques natives envoyées par le plug-in et convertissent les fichiers binaires (JAR, pas les sources) en applications natives, comme expliqué ci-dessous.
- Serveurs Cloud - Les serveurs Cloud fournissent des fonctionnalités telles que la notification push, la journalisation cloud, etc.



Pourquoi construire des serveurs?

Les serveurs de génération permettent la création d'applications iOS natives sans Mac et d'applications Windows natives sans ordinateur Windows. Ils éliminent le besoin d'installer / mettre à jour des chaînes d'outils complexes et de simplifier le processus de création d'une application native par un clic droit.

Par exemple: la création d'applications iOS natives nécessite une machine Mac OS X avec une version récente de xcode. Codename One gère ces machines dans le cloud. Lorsque les développeurs envoient une version iOS, un tel Mac sera utilisé pour générer le code source C à l'aide de [ParparVM](#), puis il compilera le code source C à l'aide de xcode et signera le binaire résultant à l'aide de xcode. Vous pouvez installer le binaire sur votre appareil ou créer un fichier binaire de distribution pour le magasin d'applications. Étant donné que le code C est généré, cela signifie également que votre application sera «à l'épreuve du futur» en cas de modifications

apportées par Apple. Vous pouvez également injecter du code natif Objective-C dans l'application tout en le maintenant 100% portable grâce à la fonctionnalité "interfaces natives" de Codename One.

Les abonnés peuvent recevoir le code source C à l'aide de la fonctionnalité include sources de Codename One et utiliser ces sources pour l'étalonnage, le débogage sur les appareils, etc.

La même chose est vraie pour la plupart des autres plates-formes. Pour Android, J2ME et Blackberry, le code Java standard est exécuté tel quel.

La syntaxe Java 8 est prise en charge par [retrolambda](#) installée sur les serveurs Codename One. Ceci est utilisé pour convertir le bytecode de manière transparente aux niveaux de syntaxe Java 5. La syntaxe Java 5 est traduite dans le sous-ensemble JDK 1.3 clc sur J2ME / Blackberry pour fournir les fonctionnalités linguistiques et les API de tous les périphériques. Ceci est fait en utilisant un processeur de code bytec basé sur le serveur basé sur retroweaver et beaucoup de code personnalisé. Notez que cette architecture est transparente pour les développeurs, car les serveurs de génération suppriment la plupart des différences douloureuses entre les périphériques.

Pourquoi ParparVM

Sur iOS, Codename One utilise [ParparVM](#) qui [convertit le](#) code octet Java en code C et dispose d'un GC non bloquant ainsi que d'un support 64 bits / bitcode. Cette VM est entièrement open source dans le [dépôt git Codename One](#) . Dans le passé, Codename One utilisait [XMLVM](#) pour générer du code natif de manière très similaire, mais la solution XMLVM était trop générique pour les besoins de Codename One. [ParparVM](#) possède une architecture unique de traduction du code en C (similaire à XMLVM), car Codename One est la seule solution de ce type pouvant **garantir la** compatibilité future avec iOS puisque la chaîne d'outils iOS officiellement prise en charge est toujours utilisée.

REMARQUE: XMLVM pourrait garantir cela en théorie, mais il n'est plus maintenu.

Les principaux avantages de ParparVM par rapport aux autres approches sont les suivants:

- Vraiment natif - puisque le code est traduit en C plutôt que directement vers le code ARM ou LLVM, l'application est "plus native". Il utilise les outils et les approches officiels d'Apple et peut tirer parti de leurs avancées, par exemple les dernières fonctionnalités en matière de bitcode ou de profilage.
- Bibliothèque de classes plus petite - ParparVM inclut un très petit segment des JavaAPI complets, ce qui se traduit par des binaires finaux plus petits que les alternatives par ordre de grandeur. Cela correspond directement aux performances et à la mémoire.
- Simple et extensible - pour travailler avec ParparVM, vous avez besoin d'une compréhension de base de C. Ceci est crucial pour le monde en évolution rapide du développement mobile, car Apple change les choses à gauche et à droite, nous avons besoin d'une VM plus agile.

Windows Phone / UWP

Le nom de code One comporte 2 principaux ports VM Windows et 3 ou 4 pipelines de rendu dans ces ports.

L'ancien port Windows Phone utilisait XMLVM pour traduire le bytecode Java en C#. Notez que le backend XMLVM qui se traduit en C# est très différent de celui utilisé auparavant pour traduire le code pour iOS.

Le nom de code One cible désormais UWP en exploitant une version modifiée de iKVM pour créer des applications universelles Windows natives.

Port JavaScript

Le port JavaScript de Codename One est basé sur le travail incroyable du [projet TeaVM](#). L'équipe derrière TeaVM a effectivement construit une JVM qui traduit le bytecode Java en code source JavaScript tout en maintenant la sémantique des threads en utilisant une approche très imaginative.

Le port JavaScript permet aux applications Codename One non modifiées de s'exécuter dans un navigateur de bureau ou mobile. Le port lui-même est basé sur l'API HTML5 Canvas pour fournir une implémentation parfaite de l'API Codename One.

REMARQUE: Le port JavaScript est uniquement disponible pour les abonnés de qualité Entreprise de Codename One.

Desktop, Android, RIM & J2ME

Les autres ports de Codename One utilisent la machine virtuelle disponible sur les machines / environnements hôtes pour exécuter le moteur d'exécution.

<https://github.com/orfjackal/retrolambda> [Retrolambda] est utilisé pour fournir des fonctionnalités de langage Java 8 de manière portable, pour les anciens appareils, retroweaver est utilisé pour apporter des fonctionnalités Java 5.

Le port Android utilise les outils Android natifs, y compris l'environnement de construction graduel dans les dernières versions.

Le port de bureau crée une application JavaSE standard qui est fournie avec JRE et un installateur.

REMARQUE: le port Desktop est uniquement disponible pour les abonnés de qualité professionnelle de Codename One.

Composants légers

Ce qui distingue Codename One, c'est l'approche de l'interface utilisateur qui utilise une

«architecture légère», ce qui permet à l'interface utilisateur de fonctionner de manière transparente sur toutes les plates-formes. En conséquence, la majeure partie de l'interface utilisateur est développée en Java et est donc remarquablement portable et débogable. L'architecture légère inclut toujours la possibilité d'incorporer des widgets "lourds" parmi les "poids légers".

Origine d'architecture légère

Les composants légers remontent aux frameworks Smalltalk, cette notion a été popularisée dans le monde Java par Swing. Swing était la principale source d'inspiration du prédécesseur de Codename One, LWUIT. De nombreux frameworks ont adopté cette approche au fil des ans, y compris JavaFX et, plus récemment, Ionic dans le monde JavaScript.

Dans le nom de code un

Un composant léger est un composant entièrement écrit en Java, il dessine sa propre interface et gère ses propres événements / états. Cela présente de grands avantages en matière de portabilité, car le même code s'exécute sur toutes les plates-formes, mais il comporte de nombreux avantages supplémentaires.

Les composants légers sont personnalisables à l'infini en utilisant l'héritage standard et la manipulation de peinture / événement. Un composant léger étant entièrement écrit en Java, les développeurs peuvent prévisualiser l'application avec précision dans le générateur de simulateurs et d'interface graphique. Cela évite de nombreux pièges courants des autres solutions WORA où le comportement spécifique à la plate-forme a déjoué tout effort enregistré. Par conséquent, tous les efforts enregistrés dans le codage ont été perdus lors du débogage des bizarreries de périphériques ésotériques uniquement.

Codename One réalise des performances rapides en dessinant en utilisant les API de jeu natives de la plupart des plates-formes, par exemple OpenGL ES sur iOS.

Versions dans le nom de code un

L'une des choses déroutantes à propos de Codename One sont les versions. Comme Codename One est un produit SaaS, la gestion des versions n'est pas aussi simple que celle d'un moniker 2.x ou 3.x. Cependant, pour se conformer à cette convention, le Codename One crée des versions versionnées qui contribuent à la confusion générale.

Lorsqu'une version de Codename One est publiée, le numéro de version fait référence aux bibliothèques au moment de la publication. Ces bibliothèques sont ensuite congelées et sont mis à disposition pour les développeurs qui utilisent la [Versioned Builds](#) fonction. Le plug-in, qui inclut le concepteur ainsi que tout développement sans rapport avec les versions versionnées, continue avec ses mises à jour régulières immédiatement après la publication. La même chose est vraie pour les serveurs de génération qui passent directement au cycle de mise à jour standard.

Lire [Démarrer avec Codename One en ligne](#):

<https://riptutorial.com/fr/codenameone/topic/1077/demarrer-avec-codename-one>

Crédits

S. No	Chapitres	Contributeurs
1	Démarrer avec Codename One	Community , kaya , Shai Almog