FREE eBook

LEARNING Codename One

Free unaffiliated eBook created from
Stack Overflow contributors.

#codename one

Table of Contents

About

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.

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

Chapter 1: Getting started with Codename One

Remarks

This section provides an overview of what codenameone is, and why a developer might want to use it.

It should also mention any large subjects within codenameone, and link out to the related topics. Since the Documentation for codenameone is new, you may need to create initial versions of those related topics.

Examples

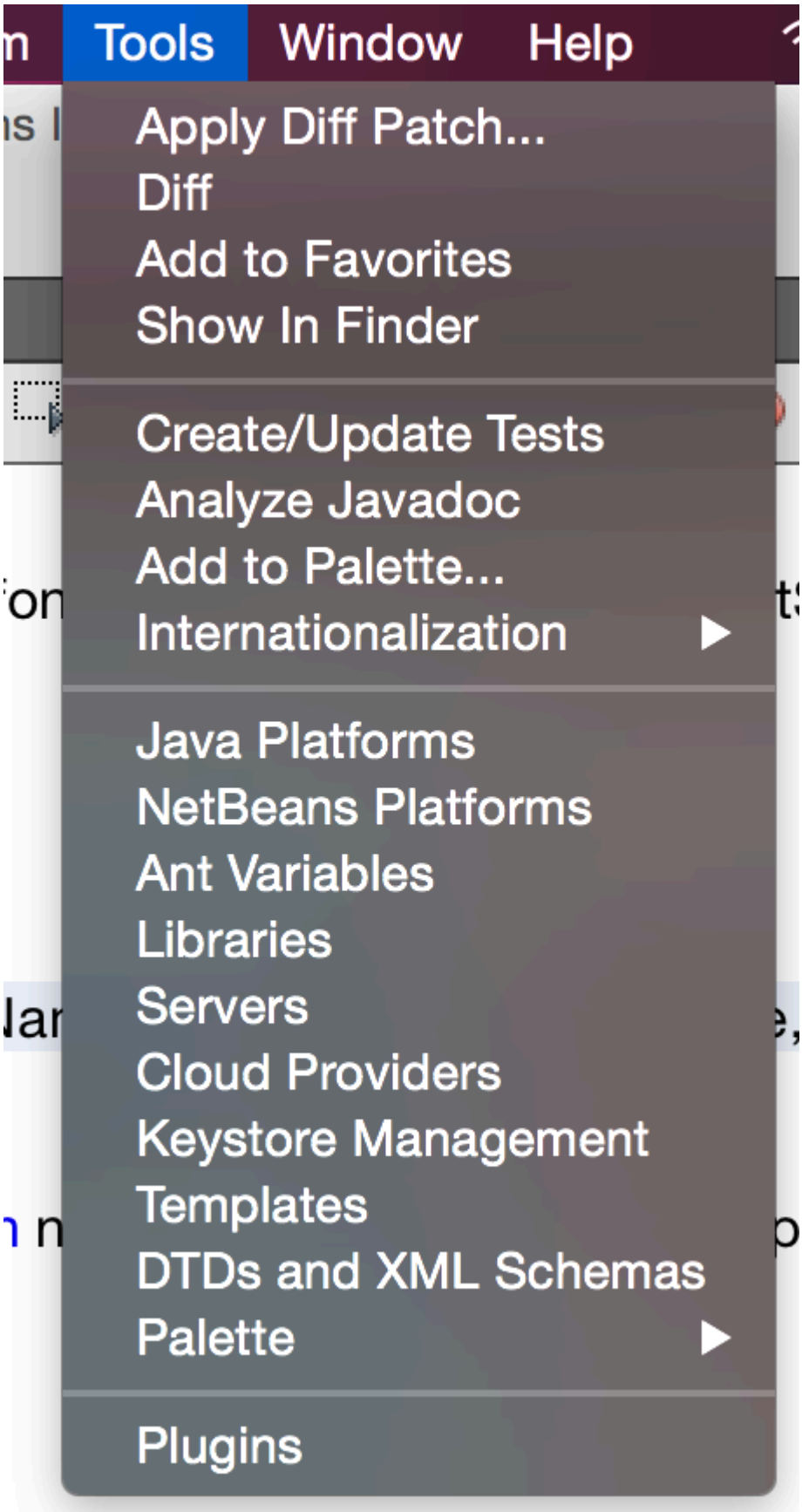
Installation & Setup

Installation

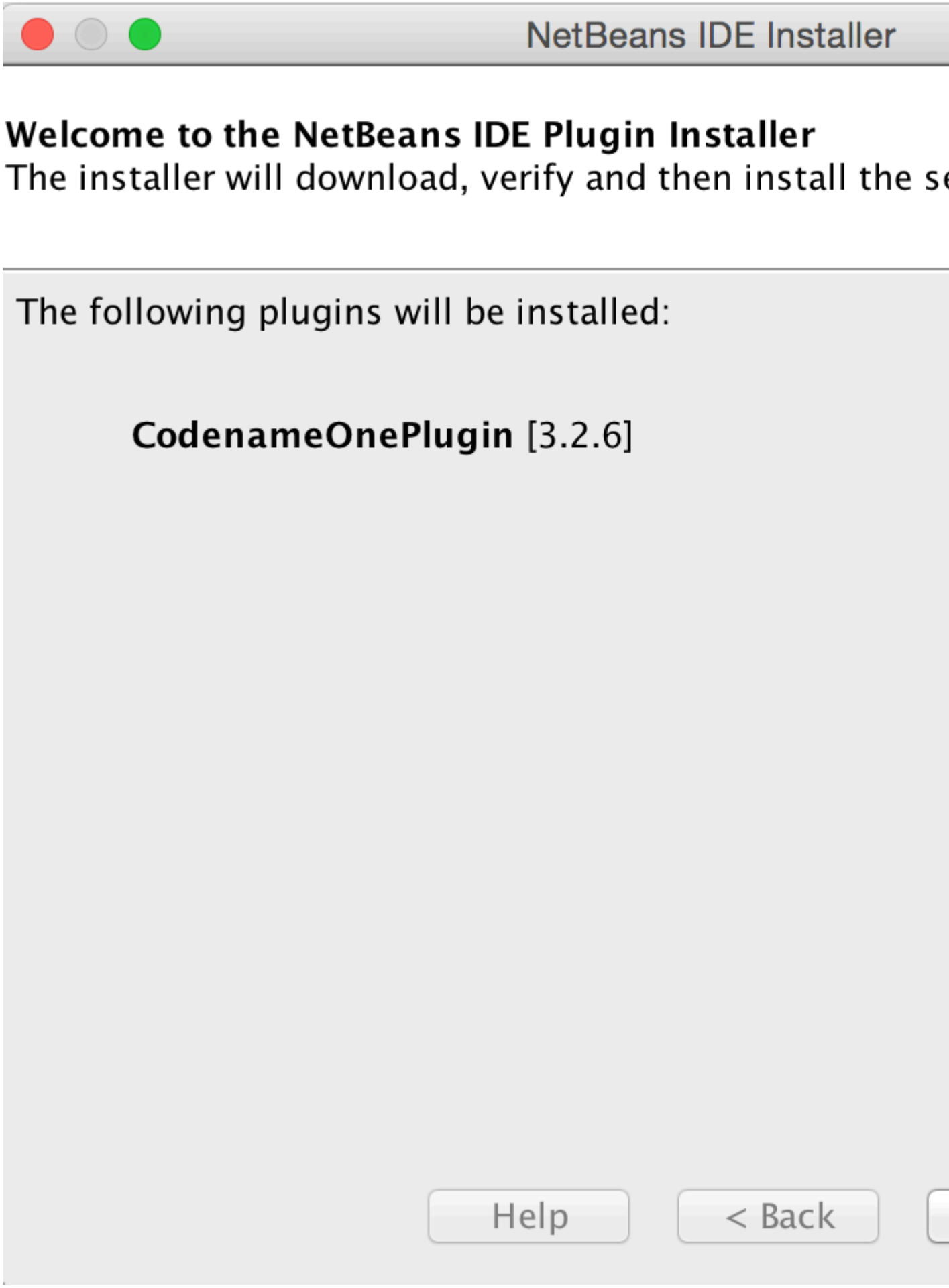
Installing Codename One In NetBeans

These instructions assume you have downloaded a recent version of NetBeans (at this time 8.x), installed and launched it.

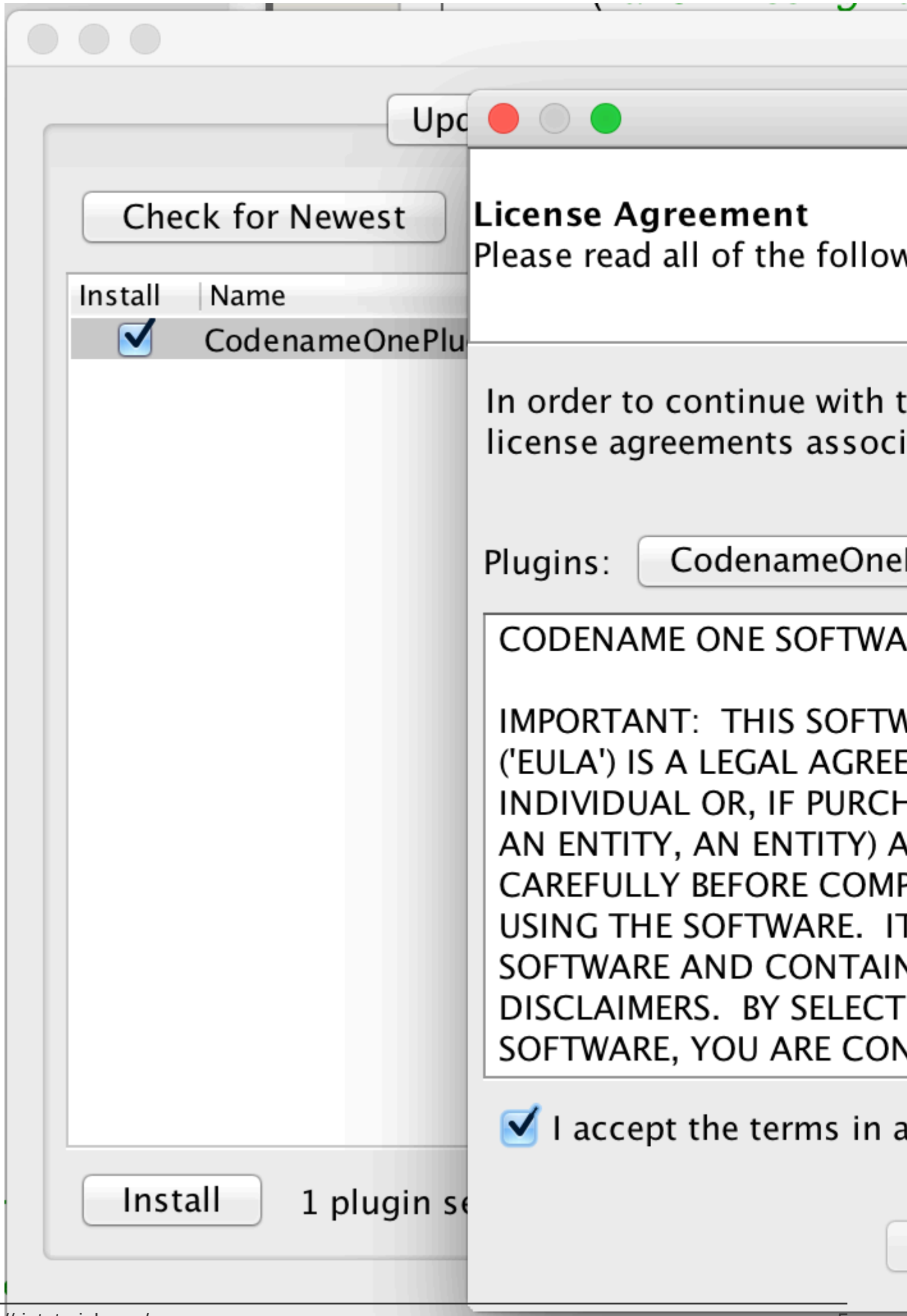
- Select the Tools->Plugins menu option



- Select the Available Plugins Tab
- Check The CodenameOne Plugin

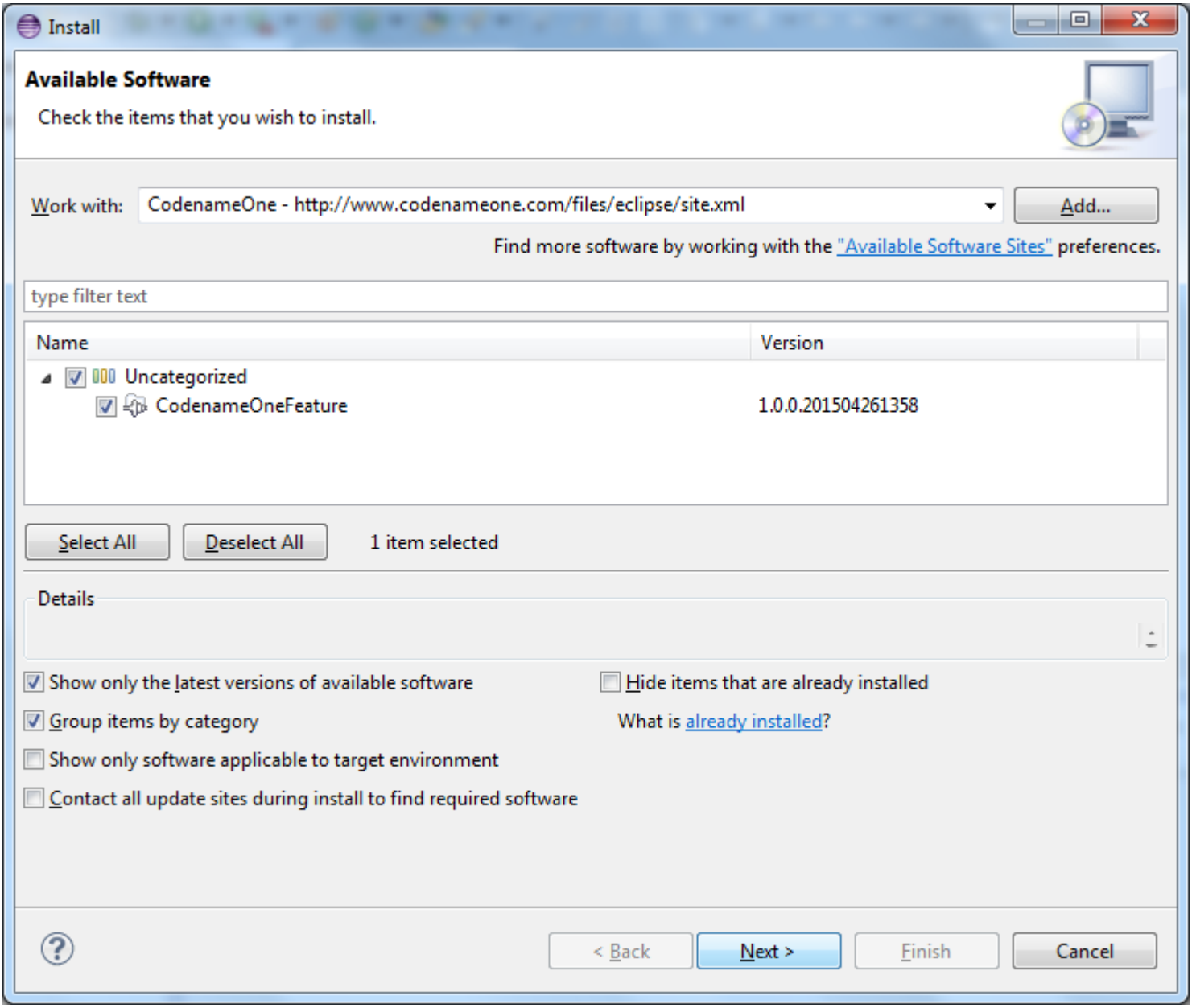


- click the `install` button below. Follow the Wizard instructions to install the plugin


Upd
Check for Newest
Install
Name
CodenameOnePlu
Install
1 plugin se
License Agreement
Please read all of the follow
In order to continue with t
license agreements associ
Plugins:
CodenameOnel
CODENAME ONE SOFTWA
IMPORTANT: THIS SOFTW
('EULA') IS A LEGAL AGREE
INDIVIDUAL OR, IF PURCH
AN ENTITY, AN ENTITY) A
CAREFULLY BEFORE COMP
USING THE SOFTWARE. IT
SOFTWARE AND CONTAIN
DISCLAIMERS. BY SELECT
SOFTWARE, YOU ARE CON
I accept the terms in a

for the location to Work with and press Enter.

Select the entries & follow the wizard to install



Installing Codename One In IntelliJ IDEA

Download & install IntelliJ/IDEA. **Notice that Android Studio will not work.**

Install the plugin using The Plugin Center

Use the search functionality in the plugin center to find and install the Codename One plugin.

What is Codename One & How Does it Work?

Codename One is a set of tools for mobile application development that derive a great deal of its architecture from Java.

Codename One's mission statement is:

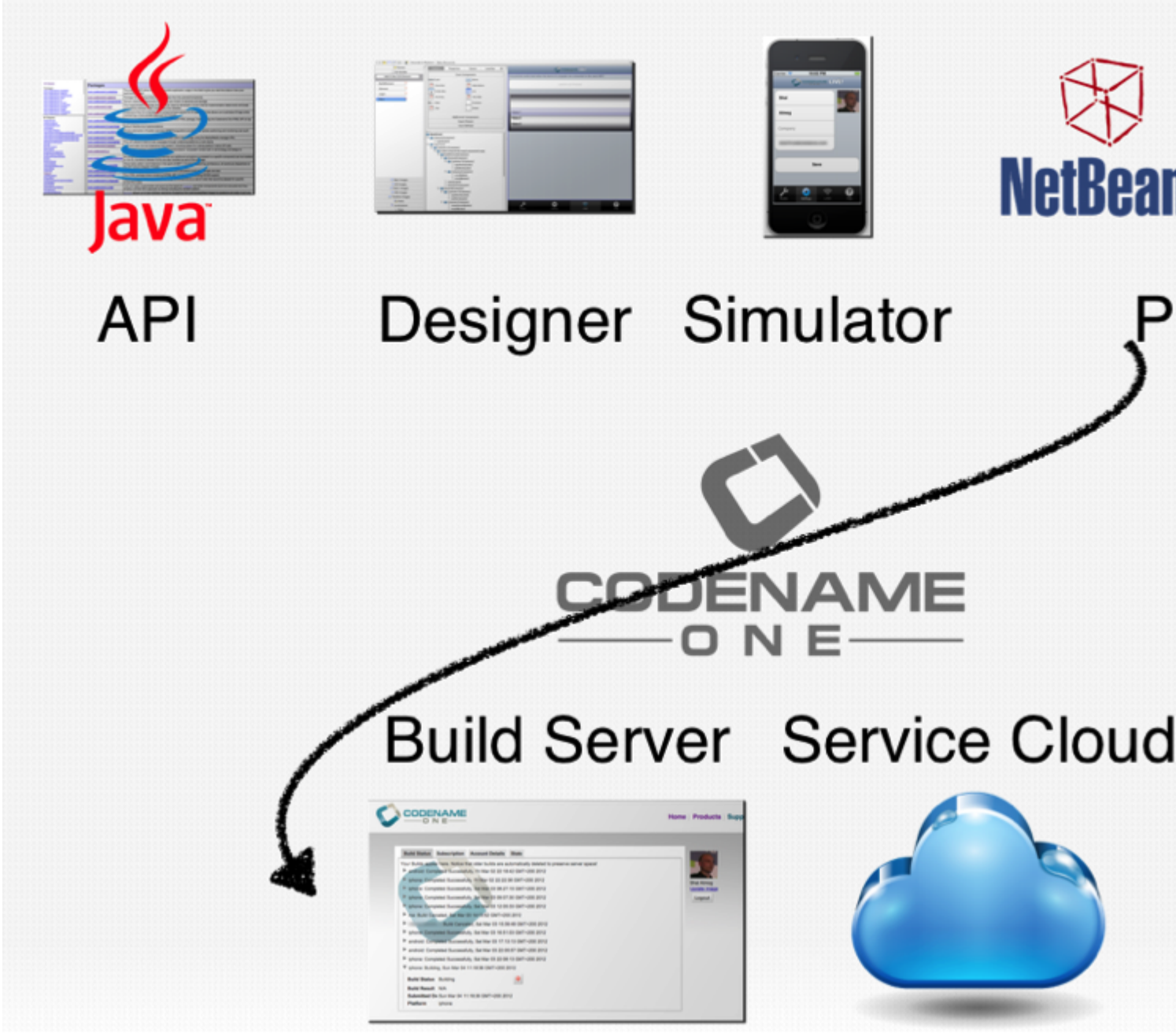
> Unify the complex and fragmented task of mobile device programming into a single set of tools, APIs & services. As a result create a more manageable approach to mobile application development without sacrificing the power/control given to developers.

This effectively means bringing that old "Write Once Run Anywhere" (WORA) Java mantra to mobile devices without "dumbing it down" to the lowest common denominator.

How Does Codename One Work?

Codename One unifies several technologies and concepts into a single facade:

- API - abstracts the differences between the various devices.
- Plugin - the only piece of software installed on client machines, it includes the following features:
  - IDE integration - preferences, completion, the ability to send a native build
  - Simulator - native device simulator that runs locally and allows debugging the application
  - Designer/GUI Builder - high level tools
- Build Servers - The build servers accept native device builds sent by the plugin and convert the binaries (JAR's, not sources) to native applications as explained below.
- Cloud Servers - The cloud servers provide features such as push notification, cloud logging etc.



Why Build Servers?

The build servers allow building native iOS Apps without a Mac and native Windows apps without a Windows machine. They remove the need to install/update complex toolchains and simplify the process of building a native app to a right click.

E.g.: Since building native iOS applications requires a Mac OS X machine with a recent version of xcode Codename One maintains such machines in the cloud. When developers send an iOS build such a Mac will be used to generate C source code using ParparVM and it will then compile the C source code using xcode & sign the resulting binary using xcode. You can install the binary to your device or build a distribution binary for the appstore. Since C code is generated it also means that your app will be "future proof" in a case of changes from Apple. You can also inject Objective-C native code into the app while keeping it 100% portable thanks to the "native interfaces" capability of Codename One.

Subscribers can receive the C source code back using the include sources feature of Codename One and use those sources for benchmarking, debugging on devices etc.

The same is true for most other platforms. For the Android, J2ME & Blackberry the standard Java code is executed as is.

Java 8 syntax is supported thru retrolambda installed on the Codename One servers. This is used to convert bytecode seamlessly down to Java 5 syntax levels. Java 5 syntax is translated to the JDK 1.3 cldc subset on J2ME/Blackberry to provide those language capabilities and API's across all devices. This is done using a server based bytecode processor based on retroweaver and a great deal of custom code. Notice that this architecture is transparent to developers as the build servers abstract most of the painful differences between devices.

Why ParparVM

On iOS, Codename One uses ParparVM which translates Java bytecode to C code and boasts a non-blocking GC as well as 64 bit/bitcode support. This VM is fully open source in the Codename One git repository. In the past Codename One used XMLVM to generate native code in a very similar way but the XMLVM solution was too generic for the needs of Codename One. ParparVM boasts a unique architecture of translating code to C (similarly to XMLVM), because of that Codename One is the only solution of its kind that can **guarantee** future iOS compatibility since the officially supported iOS toolchain is always used instead of undocumented behaviors.

> **NOTE:** XMLVM could guarantee that in theory but it is no longer maintained.

The key advantages of ParparVM over other approaches are:

- Truly native - since code is translated to C rather than directly to ARM or LLVM code the app is "more native". It uses the official tools and approaches from Apple and can benefit from their advancements e.g. latest bitcode or profiling capabilities.
- Smaller class library - ParparVM includes a very small segment of the full JavaAPI's resulting in final binaries that are smaller than the alternatives by orders of magnitude. This maps directly to performance and memory overhead.
- Simple & extensible - to work with ParparVM you need a basic understanding of C. This is crucial for the fast moving world of mobile development, as Apple changes things left and right we need a more agile VM.

Windows Phone/UWP

Codename One has 2 major Windows VM ports and 3 or 4 rendering pipelines within those ports.

The old Windows Phone port used XMLVM to translate the Java bytecode to C#. Notice that the XMLVM backend that translates to C# is very different from the one that was used in the past to translates code for iOS.

Codename One now targets UWP by leveraging a modified version of iKVM to build native

Windows Universal Applications.

JavaScript Port

The JavaScript port of Codename One is based on the amazing work of the TeaVM project. The team behind TeaVM effectively built a JVM that translates Java bytecode into JavaScript source code while maintaining threading semantics using a very imaginative approach.

The JavaScript port allows unmodified Codename One applications to run within a desktop or mobile browser. The port itself is based on the HTML5 Canvas API to provide a pixel perfect implementation of the Codename One API's.

> **NOTE:** The JavaScript port is only available for Enterprise grade subscribers of Codename One.

Desktop, Android, RIM & J2ME

The other ports of Codename One use the VM's available on the host machines/environments to execute the runtime. https://github.com/orfjackal/retrolambda[Retrolambda] is used to provide Java 8 language features in a portable way, for older devices retroweaver is used to bring Java 5 features.

The Android port uses the native Android tools including the gradle build environment in the latest versions.

The desktop port creates a standard JavaSE application which is packaged with the JRE and an installer.

> **NOTE:** The Desktop port is only available to pro grade subscribers of Codename One.

Lightweight Components

What makes Codename One stand out is the approach it takes to UI where it uses a "lightweight architecture" thus allowing the UI to work seamlessly across all platforms. As a result most of the UI is developed in Java and is thus remarkably portable and debuggable. The lightweight architecture still includes the ability to embed "heavyweight" widgets into place among the "lightweights".

Lightweight Architecture Origin

Lightweight components date back to Smalltalk frameworks, this notion was popularized in the Java world by Swing. Swing was the main source of inspiration to Codename One's predecessor LWUIT. Many frameworks took this approach over the years including JavaFX & most recently Ionic in the JavaScript world.

In Codename One

A Lightweight component is a component that is written entirely in Java, it draws its own interface and handles its own events/states. This has huge portability advantages since the same code executes on all platforms, but it carries many additional advantages.

Lightweight components are infinitely customizable by using standard inheritance and overriding paint/event handling. Since a lightweight component is written entirely in Java, developers can preview the application accurately in the simulators & GUI builder. This avoids many common pitfalls of other WORA solutions where platform specific behavior foiled any saved effort. Hence all the effort saved in coding was lost in debugging esoteric device only oddities.

Codename One achieves fast performance by drawing using the native gaming API's of most platforms e.g. OpenGL ES on iOS.

Versions In Codename One

One of the confusing things about Codename One is the versions. Since Codename One is a SaaS product versioning isn't as simple as a 2.x or 3.x moniker. However, to conform to this convention Codename One does make versioned releases which contribute to the general confusion.

When a version of Codename One is released the version number refers to the libraries at the time of the release. These libraries are then frozen and are made available to developers who use the Versioned Builds feature. The plugin, which includes the designer as well as all development that is unrelated to versioned builds continues with its regular updates immediately after release. The same is true for the build servers that move directly to their standard update cycle.

Read Getting started with Codename One online:
https://riptutorial.com/codenameone/topic/1077/getting-started-with-codename-one

Credits

S. No	Chapters	Contributors
1	Getting started with Codename One	Community, kaya, Shai Almog