

[<< Back to Dashboard](#)

Contents

Refactoring ColdBox

- [Refactoring ColdBox](#)
 - [Introduction](#)
 - [Guide 1: ANT via Eclipse](#)
 - [Guide 2: Search and Replace](#)

Covers up to version 3.5.0

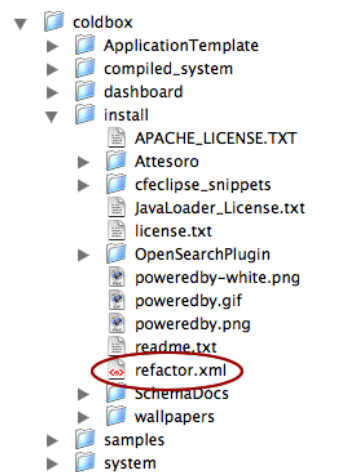
Introduction

This guide will show you how to refactor the ColdBox codebase in order for you to embed it in a distributable application or just install in another non-standard installation. In order to proceed with this guide you will need the following:

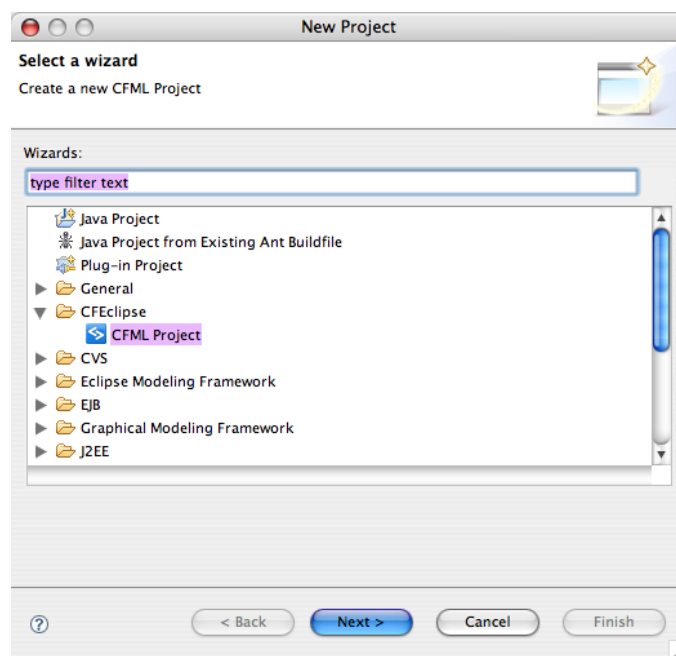
- Latest ColdBox [source code](#)
- [CFEclipse](#), [Adobe ColdFusion Builder](#), or any IDE that supports FIND/REPLACE
- [Apache ANT](#)

Guide 1: ANT via Eclipse

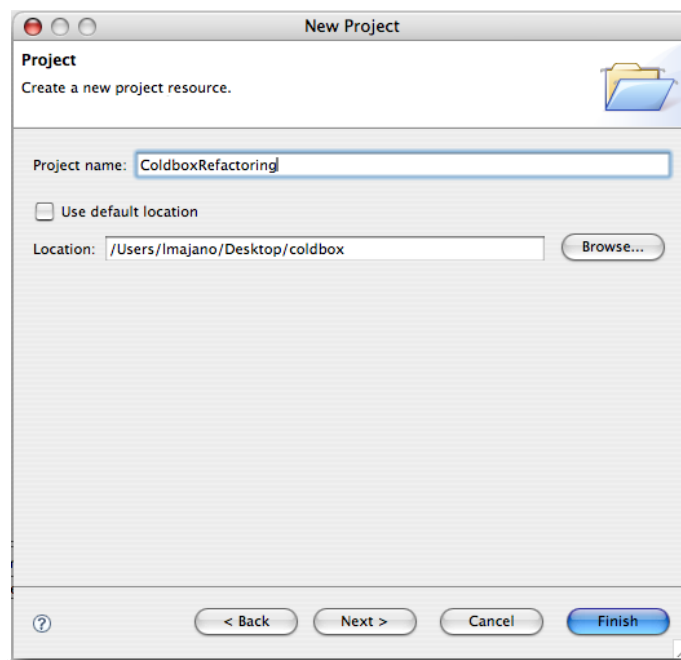
The ColdBox installation folder contains a file called **refactor.xml**. This file is an ANT task for use in refactoring the ColdBox source code. You should have a directory layout like the one shown below after downloading and extracting the latest ColdBox release:



In this sample, ColdBox has been downloaded and extracted onto the desktop, in a directory called **coldbox**. Once you have the **coldbox** folder, it's time to fire up [Eclipse](#) and create a new project. To do so, click on **File > New > Project > CFEclipse > CFML Project**



After selecting the CFML Project, choose a name for the project; in this sample it will be called **ColdBoxRefactoring**. Then select the location of the **coldbox** directory (on the desktop) and click on **Finish**.



The project is created and you can see it in your project explorer. The next step is to open the **refactor.xml** file and refer to the following screenshot, which describes two properties:

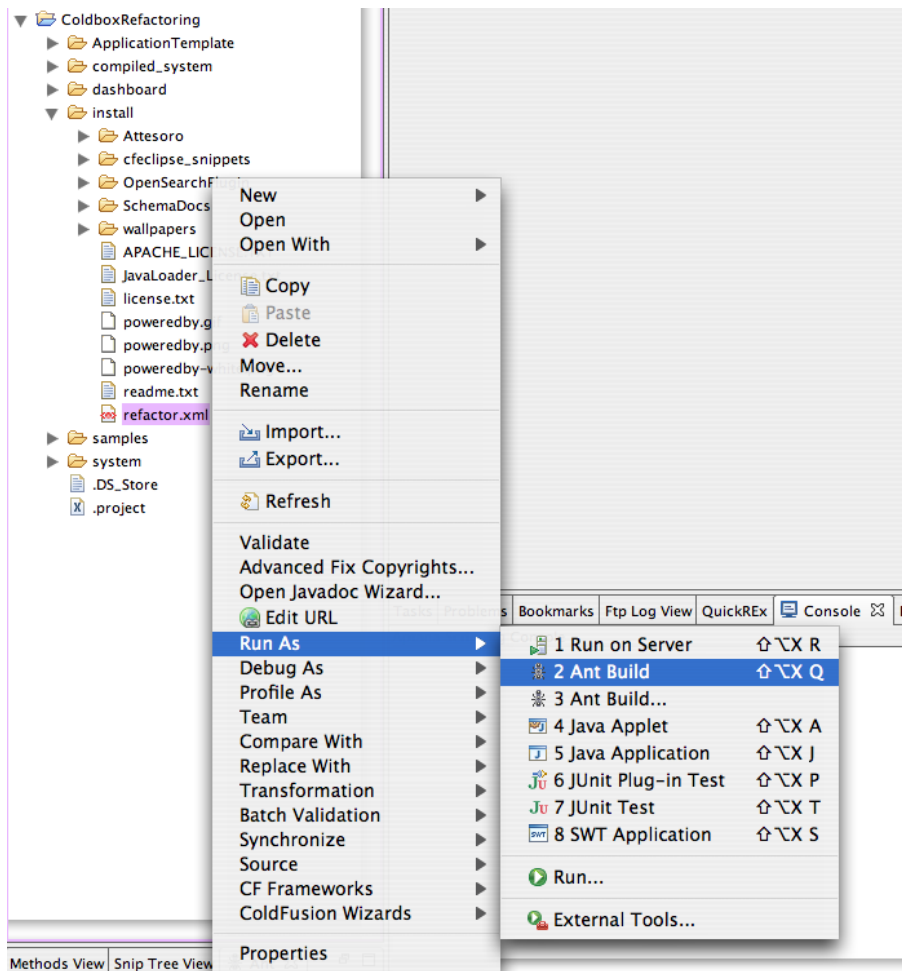
- **coldbox.system_path** : The location of the **system** folder
- **coldbox.destination** : Where to place the refactored code

```
<!-- Where the ColdBox system folder is -->
<property name="coldbox.system_path" value="${basedir}\src\system" />
<!-- Where to copy the refactored system folder -->
<property name="coldbox.destination" value="${basedir}\install\system" />
```

Now, you will need to change the properties to your liking. I mostly just change the **coldbox.system_path** to: **\${basedir}\system** which matches what you downloaded. You should have something similar to the screenshot below:

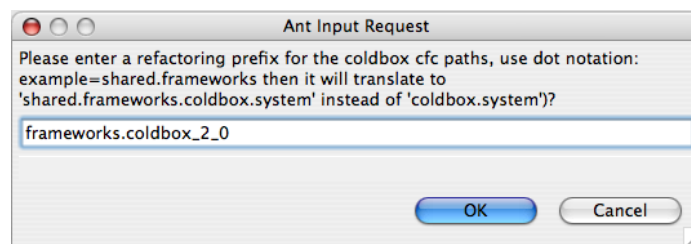
```
<!-- Where the ColdBox system folder is -->
<property name="coldbox.system_path" value="${basedir}\system" />
<!-- Where to copy the refactored system folder -->
<property name="coldbox.destination" value="${basedir}\install\system" />
```

The next step is to actually execute the **refactoring.xml** ANT Task. To do this, open the install folder and right click on the **refactor.xml** file. A context menu will appear with tons of options, look for the one called **Run As > Ant Build** Look below:

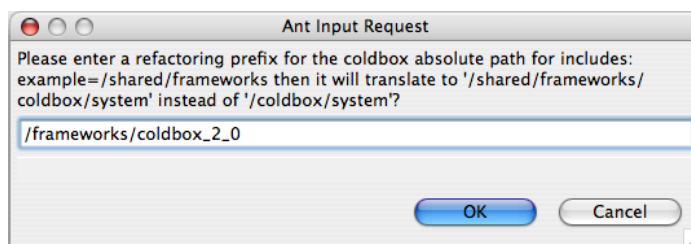


Once you click on the **Ant Build**, the task will execute and you will get a popup similar to the one below. In this popup, fill out the new prefix of the ColdBox path. The default one is **coldbox.system**, and lets say that I want the new location of the framework to be on a folder called frameworks off my root and in a directory called **coldbox_2_0**. Then I would type: **frameworks.coldbox_2_0**

Do not add a final **.** this is appended automatically.



After you click ok, then you will be asked for the absolute path of the new location. In my case I enter: **/frameworks/coldbox_3_0**



Once you click the final OK, the ant task will execute and you will see the following output on the eclipse console:

```

Buildfile: /Users/lmajano/Desktop/coldbox/install/refactor.xml
do_prefix:
do_copy:
[copy] Copying 55 files to /Users/lmajano/Desktop/coldbox/install/system
do_refactor:
[echo] Refactoring for coldbox absolute paths
[replace] Replaced 8 occurrences in 3 files.
[echo] Refactoring for coldbox instantiation and cfc paths
[replace] Replaced 50 occurrences in 36 files.
BUILD SUCCESSFUL
Total time: 13 seconds

```

This means the task executed correctly and you are ready to use the refactored code. The final screenshot below shows you the refactored system folder in the install directory and a code snapshot of some changes. ENJOY

```

161 </cfunction>
162
163 <!-- Plugin Factories --->
164 <cfunction name="getMyPlugin" access="Public" returnType="any" hint="I am the Custom Plugin cfc object factory." output="
165 <cfargument name="plugin" type="string" hint="The Custom Plugin object's name to instantiate" >
166 <cfreturn getPlugin(arguments.plugin,true)>
167 </cfunction>
168 <cfunction name="getPlugin" access="Public" returnType="any" hint="I am the Plugin cfc object factory." output="true">
169 <cfargument name="plugin" type="string" hint="The Plugin object's name to instantiate" >
170 <cfargument name="customPlugin" type="boolean" required="false" default="false" hint="Used internally to create custo
171 <cfargument name="newInstance" type="boolean" required="false" default="false" hint="If true, it will create and ret
172 <!-- *****>
173 <cfset var oPlugin = "">
174 <cfset var MetaData = structNew()>
175 <cfset var objTimeout = "">
176 <cfset var pluginKey = "plugin_" & arguments.plugin>
177 <cfset var pluginPath = "frameworks.coldbox_2_0.coldbox.system.plugins.#trim(arguments.plugin)#">
178
179 <!-- Custom Plugin Test --->
180 <cfif arguments.customPlugin>
181 <cfset pluginKey = "custom_plugin_" & arguments.plugin>
182 <cfset pluginPath = "#getSetting("MyPluginsLocation")#.trim(arguments.plugin)#">
183 </cfif>
184
185 <!-- Check For New Instance --->
186 <cfif arguments.newInstance>
187 <!-- Object not found, proceed to create and verify --->
188 <cfset oPlugin = CreateObject("component", pluginPath).init(this)>
189 <cfelse>
190 <!-- Lookup in Cache --->
191 <cfif instance.ColdboxOCM.lookup(pluginKey)>
192 <cfset oPlugin = instance.ColdboxOCM.get(pluginKey)>
193 <cfelse>
194 <!-- Object not found, proceed to create and verify --->
195 <cfset oPlugin = CreateObject("component", pluginPath).init(this)>
196 <!-- Get Object's MetaData --->
197 <cfset MetaData = getMetaData(oPlugin)>
198 <!-- Test for caching parameters --->
199 <cfif structKeyExists(MetaData, "cache") and isBoolean(MetaData["cache"]) and MetaData["cache"]>
200 <cfif structKeyExists(MetaData, "cachetimeout") >
201 <cfset objTimeout = MetaData["cachetimeout"]>
202 </cfif>
203 <cfset instance.ColdboxOCM.set(pluginKey,oPlugin,objTimeout)>
204 </cfif>
205 </cfif>
206 </cfif>
207

```

Guide 2: Search and Replace

This guide is just a basic use of search and replace features of your favorite IDE, either Eclipse, Dreamweaver, or whatever. All you need to do is the following:

1. Search for **coldbox.system** and replace it with **{Your Prefix}.coldbox.system**
2. Search for **/coldbox/system** and replace it with **/{Your Prefix}/coldbox/system**

Enjoy your refactored ColdBox recipe!

Categories:

- [level-advanced](#)
- [setup](#)