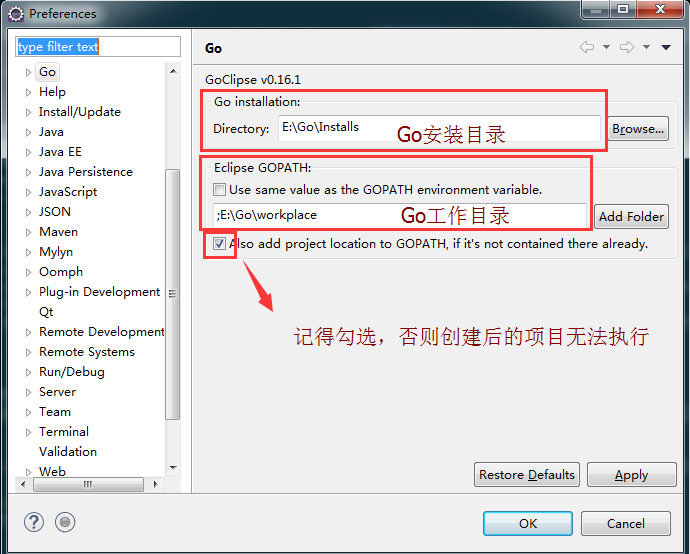
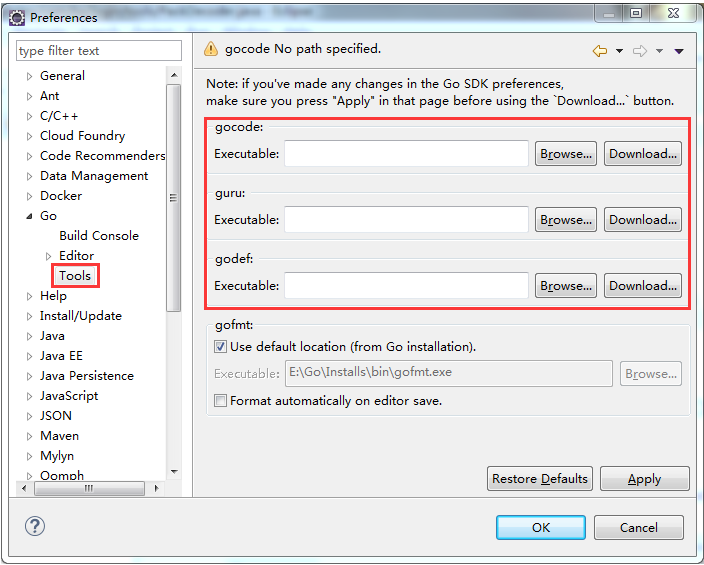
在Eclipse中依次点击菜单：Windows -> Preferences -> Go，进入到Go插件的配置界面，然后进行 Go Dictionary 和 GOPATH 两项关键配置项的填写（记得把最后一项勾选上）：



填写完毕后点击 Apply 应用设置内容，然后切换到 Go 页签的子页签 Tools，这里需要配置三个工具的目录，分别是：gocode、guru 和 godef：



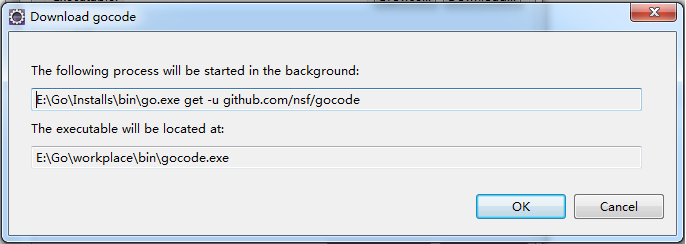
这里可以使用两种方式来完成此三项的配置：

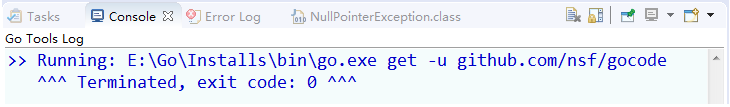
- 方式一：比较简单，直接点击对应项后面得 Download 按钮，进行在线安装，下载完成后会自动编译并设置对应项的配置参数；

- 方式二：通过分别下载这三项的Go源码，然后使用 go build 指令编译得到对应的 .exe 文件，并将其目录填写到对应栏中即可。

安装 gocode ：

这里以 gocode为例，使用方式一，点击之后出现弹窗：





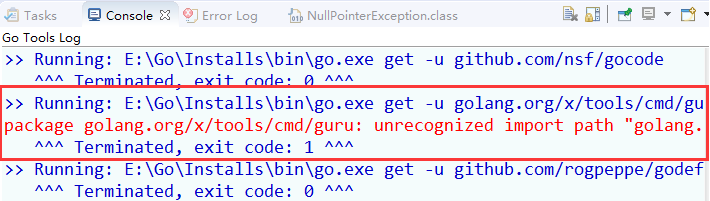
并且在之前配置的Go工作目录 GOPATH 的 bin 目录下会多出一个 gocode.exe 文件。

安装 godef ：

以相同的方式完成第三项的安装：

安装 guru ：

这种方式无法安装第二项，因为第二项获取资源的路径是 golang.org/x/tools/cmd/guru，但是不翻墙的话无法访问 golang.org 这个网址，安装会报错：



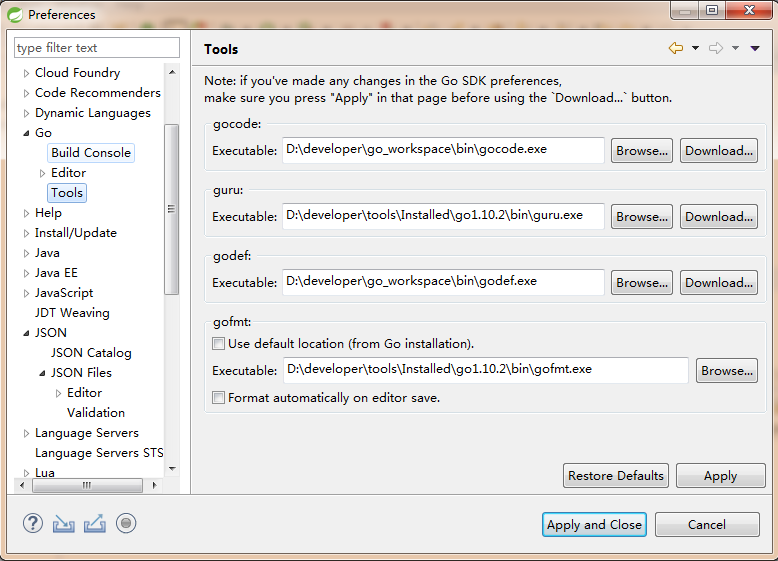
所以这里我们尝试使用方式二来完成 guru 的配置：

先从Github上获取 guru 的源码(https://github.com/golang/tools.git)，当然通常的做法就是把 整个Git工程 给下载下来；将项目压缩包解压，改名为 tools ，然后在之前配置的Go安装目录的 src 目录下新建一个文件夹命名为 golang.org ，再在此文件夹下新建一个 x 文件夹，将解压后的文件放到 golang.org/x/ 文件夹下；

在命令行中定位到 GOPATH 配置的目录下的 bin 目录下，执行 go build golang.org/x/tools/cmd/guru，执行成功发现在此目录生成了一个 guru.exe 文件；

配置Eclipse中 Tools 的 guru 目录为上述生成 guru.exe 文件的路径。

点击 Apply 保存配置信息，至此，配置操作就已经完成了：



测试:

在工程的 scr 目录下新建一个文件夹 main，然后新建一个 helloworld.go 文件，内容如下：

package main

import (

"fmt"

)

func main(){

fmt.Println("Hello world!")

}

