# 自定义类说明

自定义类需要符合ant的结构要求，类开发完成后，将class类打包，放入ant\_home/lib下，在build.xml中增加自定义说明，即可完成自定义task的定义

如我新增了FileCombin.java用于使用文件清单处理，文件class放入lib目录下后，build文件中新增代码<taskdef name="fileCombin" classname="hayden.FileCombin" />即可

目前持续集成平台新增的自定义task说明如下

* hayden/TaskSvnRegGet.java
* hayden/BuildFail.java
* hayden/OutPutInfo.java
* hayden/FileCombin.java
* hayden/InnerClassFind.java
* hayden/SortContain.java
* hayden/SvnStatusLogParse.java
* hayden/UtilitySum.java

# TaskSvnRegGet

TaskSvnRegGet类为根据task号获取文件清单及版本号

参数如下

**private** String xmlFile=**null**; //解析的xml地址

**private** String listFile=**null**; //文件清单地址

**private** String propertyFile=**null**; //输出版本号文件路径

**private** String versionStr=**null**; //版本信息

**private** String jiraTask=**null**; //jira编号

**private** String xmlPath="/SourceCode/XML"; //xml文件目录

**private** String sqlPath="/Document/"; //sql文件目录

Boolean log=**false**; //是否输出日志

Boolean findPath=**false**; //是否找到jira对应的文件目录，未找到则进行输出

Boolean includeDir=**true**; //是否记录目录

Boolean xmlCheck = **false**; //是否是只检查xml规范性

其中，有默认值的项可以为空，其余参数必须录入

# BuildFail

BuildFail类，为异常类，主要用于构建过程中ant抛出异常，可以使整个构建失败。

String msg=**null**; //输出message

只有一个参数，即抛出异常的message，字符串类型

# OutPutInfo

OutPutInfo类，此类负责打印文本文件的内容

String srcFile=**null**; //输出文件路径

Boolean log=**true**; //是否输出日志

# FileCombin

FileCombin类，主要处理文件清单，包括对文件清单中的内容进行追加，添加目录等

**private** File srcFile=**null**; //原文件

**private** File destFile=**null**; //目标文件

**private** File finalFile=**null**; //处理文件中每行前增加字符串的目标文件

**private** String srcString=**null**; //需要增加的字符串

**private** String addString=**null**; //每行增加的字符串

此类有多个组合

其中，同时输入srcFile和destFile时，进行两个文件合并

同时输入destFile，finalFile和addString时，对destFile每行前增加一串字符串

同时输入srcSting和destFile时，在destFile文件末尾新增srcString字符串

# InnerClassFind

InnerClassFind类，检测文件清单中的内部类

**private** File listFile=**null**; //文件清单

**private** String srcDir=**null**; //文件目录

**private** Boolean log=**false**; //是否输出log，默认为否

其中srcDir是各系统根目录

主要是检测清单中的java文件，是否有内部类，内部类需要是文件名中包含$符号的，否则，无法检测出

# SortContain

对文件内容进行排序，主要用于sql文件清单的排序，以确定先后顺序

**private** String srcFile=**null**; //源文件

**private** String historyLog=**null**; //执行历史文件

**private** Boolean log=**false**; //是否输出日志

**private** Boolean checkHistory=**true**; //是否检查历史已经执行过

# SvnStatusLogParse

此类对svn目录中执行svn status的输出结果进行分析，可以输出某个状态的文件目录

**private** String statusInfoFile="";//svn status路径

**private** String listFile=""; //输出文件名路径的文件

**private** String jarDir="WEB-INF/lib/";//输出文添加路径，默认为lib文件夹

**private** String status="?";//要过滤的状态，默认的为无版本状态

**private** String excludeFile=".svn";//需要排除的文件名，默认为空

**private** Boolean log=**false**; //是否输出log，默认为否

# UtilitySum

此类主要分析文件清单涉及更新哪些系统，以及是否有java文件更新

**private** String srcFile=**null**; //文件清单

**private** String destFile=**null**; //输出文件

**private** Boolean log=**false**; //是否输出log，默认为否