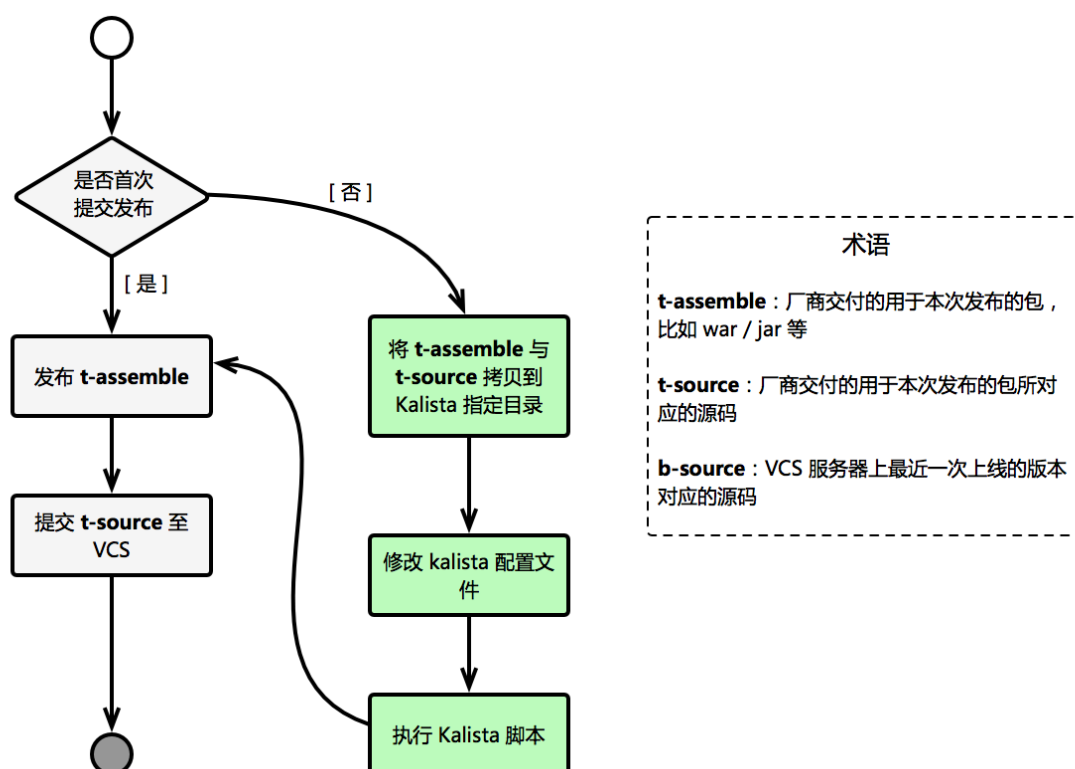


## User Manual for Kalista (v1.0.0)

Kalista 脚本用于扫描比对一个项目的两个版本之间的代码差异，与普通的文本比对工具的差别在于，它并不只是简单的以纯文本的方式进行差别比较（当然 Kalista 也支持这种方式的比较），它还能够针对不同的文件类型执行不同的扫描策略，比如对于 sqlmap 文件，Kalista 可以解析两个版本之间的 select, delete, insert 等元素的变化情况，并给出相应的报告。



### 执行环境

#### Kalista

通过 <https://raw.githubusercontent.com/watertao/devops/master/kalista/kalista.zip> 下载 Kalista，将下载的 Kalista 压缩包解压到任意路径，解压后的目录结构看起来如下（前缀 d 为目录，前缀 f 为文件）：

- [d] kalista
- [d] conf
- [d] docs
- [d] module
- [f] kalista.py

- conf

用于存放脚本执行时的配置文件

- docs

用于存放操作手册和开发手册等文档

- module

主脚本需要依赖的脚本库，包括第三方脚本库

- kalista.py

主脚本（即启动脚本）

在本手册的后文中，将使用 `$KALISTA_HOME` 描述 Kalista 解压的实际位置。

## Python

Kalista 采用 Python 3.6 编写，因此在执行该脚本前，请确保已安装 Python 3.6 (<https://www.python.org/>)。

安装完成后，请确保 python 可以正确在命令行中执行（部分操作系统可能需要设置环境变量）：

```
# python3 --version
Python 3.6.0
```

## VCS 客户端

对于一个项目而言，每次新版本发布之后，其版本对应的源代码需要上传到 VCS 服务器上，这可以方便脚本快速获取最近一次上线的版本代码作为比较基线。

目前 Kalista 只支持 Subversion 作为 VCS。因此我们需要确保在执行脚本的操作系统上已经安装了 Subversion 客户端。Windows 操作系统建议安装 SlikSvn (<http://www.sliksvn.com>)；Mac 一般已经预装 svn 客户端，否则可以通过 homebrew 进行安装；Linux 的各种桌面发行版一般也已经预装，否则可以通过其自带的包管理器安装。

安装完成后，请确保 svn 可以正确在命令行中执行（部分操作系统可能需要设置环境变量）：

```
# svn --version
svn, version 1.9.4 (r1740329)
```

```
compiled Nov 15 2016, 20:35:27 on x86_64-apple-darwin15.0.0
```

```
...
```

## 执行脚本

---

Kalista 脚本提供了两个执行逻辑： 初始化工作目录 和 扫描比对。

### 初始化工作目录

Kalista 脚本的执行需要一个工作目录（workspace）。目标版本的代码及发布物、VCS 上签出的基线代码、执行过程中的临时文件、输出的扫描比对报告等都存放在这个工作目录中。因此在执行扫描比对前我们需要对工作目录进行初始化。工作目录的位置可以在配置文件中设置，详情参考后面的配置文件章节。

用法：

```
> python3 kalista.py [-p <conf_file_name>] init
```

其中 **init** 参数表示这条命令用于初始化工作目录。选项 **p** 用于指定命令需要读取的配置文件，比如 **-p community**，表示读取 community.json 配置文件，若缺省该选项，则默认读取 config.json。

范例：

```
# cd $KALISTA_HOME
# python3 kalista.py init
Initialize workspace [ ./workspace ]
Done.
```

### 扫描比对

当我们将目标发布包及源代码存放到工作目录的指定位置后（参考工作目录章节），可以执行扫描比对命令。该命令会先检测目标发布包与源代码是否匹配（即该发布包是否由给定的源代码构建而出），接着从 VCS 上签出最后一次上线的源代码作为比对基线，然后扫描比对目标源代码和基线源代码，最终输出比对报告。

用法：

```
> python3 kalista.py [-p <conf_file_name>]
```

与初始化工作目录命令的区别在于它没有参数 **init**。选项 **p** 用于指定命令需要读取的配置文件，比如 **-p community**，表示读取 community.json 配置文件，若缺省该选项，则默认读取 config.json。

范例：

---

```

# cd $KALISTA_HOME
# python3 kalista.py
Initialize workspace [ ./workspace ]
Done.
03-09 09:43 root      INFO      -----
03-09 09:43 root      INFO      - - BUILD CHECKER ( java web app ) #
check whether war is exactly build from specified source
03-09 09:43 root      INFO      -----
03-09 09:43 root      INFO      [ subprocess ] mvn -s
./workspace/artifact/conf/maven/settings.xml -f
./workspace/temp/cb/build/common/pom.xml clean install
03-09 09:43 root      INFO      [ checking ]
./workspace/temp/cb/chk/assemble/constant.jar -VS-
temp/cb/build/common/securities-bs-common-constant/target/securities-bs-
common-constant-0.0.1-SNAPSHOT.jar
03-09 09:43 root      INFO      -----
03-09 09:43 root      INFO      - - SCAN COMPARER ( raw file ) # scan the
files and compare as the raw text
03-09 09:43 root      INFO      -----
03-09 09:43 root      INFO      [U] common/securities-bs-common-
constant/pom.xml
03-09 09:43 root      INFO      [D] common/sqlmap/BaseSysUser_SqlMap.xml
03-09 09:43 root      INFO      [A] common/sqlmap/ArticleInfo_SqlMap.xml
03-09 09:43 root      INFO      [U]
common/sqlmap/MessSysNotice_SqlMap.xml
03-09 09:43 root      INFO      [U] common/sqlmap/UserBase_SqlMap.xml
03-09 09:43 root      INFO      Done.
03-09 09:43 root      INFO      -----
03-09 09:43 root      INFO      - - SCAN COMPARER ( i(my)batis sqlmap ) #
scan the sqlmap configuration files and find difference between 2 version
03-09 09:43 root      INFO      -----
03-09 09:43 root      INFO      [D] [ - sql - BaseSysUser.commonColumns ]
./workspace/base/common/sqlmap/BaseSysUser_SqlMap.xml
03-09 09:43 root      INFO      [D] [ - sql - BaseSysUser.dynamicUpdate ]
./workspace/base/common/sqlmap/BaseSysUser_SqlMap.xml
03-09 09:43 root      INFO      [D] [ - insert - BaseSysUser.insert ]
./workspace/base/common/sqlmap/BaseSysUser_SqlMap.xml
03-09 09:43 root      INFO      [D] [ - delete - BaseSysUser.delete ]
./workspace/base/common/sqlmap/BaseSysUser_SqlMap.xml
03-09 09:43 root      INFO      [D] [ - select - BaseSysUser.getById ]
./workspace/base/common/sqlmap/BaseSysUser_SqlMap.xml
03-09 09:43 root      INFO      [D] [ - select - BaseSysUser.getByOrgId ]
./workspace/base/common/sqlmap/BaseSysUser_SqlMap.xml
03-09 09:43 root      INFO      [D] [ - select - BaseSysUser.findAll ]
./workspace/base/common/sqlmap/BaseSysUser_SqlMap.xml
03-09 09:43 root      INFO      [D] [ - sql - BaseSysUser.dynamicWhere ]
./workspace/base/common/sqlmap/BaseSysUser_SqlMap.xml
03-09 09:43 root      INFO      [D] [ - select -
BaseSysUser.getByEmpNoAndOrgId_1 ]
./workspace/base/common/sqlmap/BaseSysUser_SqlMap.xml

```

```
03-09 09:43 root      INFO      [D] [ - select - BaseSysUser.getByEmpNo ]
./workspace/base/common/sqlmap/BaseSysUser_SqlMap.xml
03-09 09:43 root      INFO      [D] [ - select - BaseSysUser.getByEmpId ]
./workspace/base/common/sqlmap/BaseSysUser_SqlMap.xml
03-09 09:43 root      INFO      [D] [ - select - BaseSysUser.getByEmpName
] ./workspace/base/common/sqlmap/BaseSysUser_SqlMap.xml
03-09 09:43 root      INFO      [D] [ - select -
BaseSysUser.getByLikeEmpNo ]
./workspace/base/common/sqlmap/BaseSysUser_SqlMap.xml
03-09 09:43 root      INFO      Done.
03-09 09:43 root      INFO
```

## 工作目录

---

工作目录初始化后，看起来如下：

```
- [d] myworkspace
- [d] artifact
  - [d] assemble
  - [d] source
- [d] base
- [d] conf
- [d] output
- [d] temp
```

- artifact

用于存放目标发布包及源代码，所谓目标发布包和源代码就是本次准备要上线的版本

- assemble 存放发布包，比如 war 包
- source 存放源代码

- base

用于存放上一个成功发布版本的源代码，一般是从 VCS 服务器上签出，当然也支持直接拷贝到此目录

- conf

用于存放插件在运行过程中所需的配置文件，详情参考查件介绍章节

- output

用于存放扫描比对的结果报告

- temp

用于存放插件在执行过程中产生的临时文件

在本手册的后文中，将使用 `$WORKSPACE` 描述工作目录实际位置。

配置文件

Kalista 的配置文件使用了 JSON 格式，被保存在 `$KALISTA_HOME/conf` 目录下，在默认的分发包中包含了两个预定义的配置文件：

- `config.json`

包含了最基本的简单配置，用于检测 Kalista 脚本的基本运行环境是否安装正确。同时，这个文件也是 Kalista 的默认配置文件，也就是说当你在执行 Kalista 脚本时如果未指定 `-p` 选项，那么脚本将默认使用此配置文件，所以我们可以后续的使用过程中修改此文件作为己用。

- `example.json`

包含了比较完整的配置范例，在编写自己的配置文件时可以参考此配置文件。不建议修改此配置文件。

Kalista 的配置可以分为两大部分：主配置 和 插件配置。  
主配置参考：

配置项	描述
<code>workspace.path</code>	工作目录的位置，建议使用绝对路径。如果使用相对路径，则需要切记，其相对的路径为脚本执行时的当前位置（并非脚本所在位置）。  范例： "/home/watertao/Work/myworkspace"
<code>vcs.ignore</code>	是否需要从 VCS 签出代码。若为 <code>true</code> ，则不需从 VCS 签出代码，直接手动拷贝基线源码到 <code>\$WORKSPACE/base</code> 下即可。若为 <code>false</code> ，则需要从指定的 VCS 签出基线源码。
<code>vcs.working_copies[]</code>	脚本可以从多个 VCS URL 签出源码，配置时可以将其写在一个列表中
<code>vcs.working_copies[].remote</code>	VCS URL。  范例： "http://10.10.10.176:88/svn/accountcenter/Code/accountcenter/branches/2.9.7.4/common"

vcs.working_copies[].local	从 VCS 签出的代码所保存的路径。此处是相对路径，相对于工作目录的 base 文件夹。  范例： "./" 代表将代码保存到 \$WORKSPACE/base 下 "./test" 代表将代码保存到 \$WORKSPACE/base/test 下
build_checkers[]	用于检查目标发布包与目标源码之间是否匹配，即目标发布包是否由目标源码构建而来。不同的项目类型使用不同的 build checker 插件进行检查，它们的配置也会不同，具体参考插件介绍章节。
scan_comparers[]	用于扫描比对目标源码与基线源码之间的差别。不同的比对插件有不同的配置，具体参考插件介绍章节。

build checker 插件

build checker 插件用于检查给到的目标发布包（位于 \$WORKSPACE/artifact/assemble 中的发布包，如 war 包）是否是由给到的目标源码（位于 \$WORKSPACE/artifact/source 中的源码）所构建而成。其检测的步骤一般是将目标源码进行构建，然后将此构建包与给到的目标发布包进行递归解压 checksum 比对，如果完全一致则认为目标发布包与目标源码是匹配的。

<b>javawebapp</b>	
该插件用于检查基于 Maven 构建的 java webapp 项目。	
<b>执行环境</b>	
JDK 1.6+ Apache Maven 3.0+	
<b>配置参数</b>	
ignore	若为 true 则忽略此插件的执行；若为 false 则执行此插件。
processor	插件标识，固定为 'javawebapp'
	指定 maven 执行时的 settings 文件位置，需要存放在 \$WORKSPACE/conf 下。该路径是一个相对路径，相对于

configuration.mvn_setting	<p>\$WORKSPACE/conf 目录。</p> <p>范例： 'maven/settings.xml' 代表 \$WORKSPACE/conf/maven/settings.xml</p>
configuration.buils[]	<p>有时候一个项目的源码可能由多个 POM 项目构成，那就需要写多个构建配置</p>
configuration.buils[].pom	<p>pom.xml 的位置。相对路径，相对于 \$WORKSPACE/artifact/source 目录。</p> <p>范例： 'common/pom.xml' 代表 \$WORKSPACE/artifact/source/common/pom.xml</p>
configuration.checks[]	<p>指定需要进行检查的两个压缩文件，比如 war 或 jar。可以指定多组。</p>
configuration.checks[].assemble	<p>指定进行检查目标压缩包。相对路径，相对于 \$WORKSPACE/artifact/assemble 目录。</p> <p>范例： 'test.war' 代表 \$WORKSPACE/artifact/assemble/test.war</p>
configuration.checks[].build	<p>指定插件构建出的检查参考压缩包。相对路径，相对于 \$WORKSPACE/artifact/source 目录。</p> <p>范例： 'common/securities-bs-common-constant/target/securities-bs-common-constant-0.0.1-SNAPSHOT.jar'</p>

### 配置文件

settings.xml	<p>Maven 的执行时配置文件，需要放在 \$WORKSPACE/conf 目录下，与配置参数中的 <code>configuration.mvn_setting</code> 对应。</p>
--------------	--



Scan Comparer 插件用于扫描比对两个版本的源代码，并根据插件的规则找出源代码间的差异，最终输出插件特有的报告。两个版本的源代码分别为 \$WORKSPACE/artifact/source 下的 和 \$WORKSPACE/base 下的。

raw

该插件对纯文本进行比对，并输出报告。

执行环境

N/A

配置参数

ignore	若为 true 则忽略此插件的执行；若为 false 则执行此插件。
processor	插件标识，固定为 'raw'
configuration.filter []	<p>在进行比对的时候，有些文件可能无需进行比较，那么可以在此指定过滤。目前只支持完整的文件名加扩展名，未来会支持采用通配符等方式。</p> <p>范例： 'DS_Store' 在进行比对的时候，如果发现这个文件，则忽略比较</p>

配置文件

N/A

报告

变化类型	文件
UPDATE	<a href="#">common/securities-bs-common-constant/pom.xml</a>
DELETE	<a href="#">common/sqlmap/BaseSysUser_SqlMap.xml</a>
ADD	<a href="#">common/sqlmap/ArticleInfo_SqlMap.xml</a>
UPDATE	<a href="#">common/sqlmap/MessSysNotice_SqlMap.xml</a>
UPDATE	<a href="#">common/sqlmap/UserBase_SqlMap.xml</a>

```
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
<modelVersion>4.0.0</modelVersion>

<parent>
<groupId>com.kingstar.securities</groupId>
<artifactId>common-build</artifactId>
<version>1.0.0</version>
</parent>

<groupId>com.kingstar.securities</groupId>
<artifactId>securities-bs-common-constant</artifactId>
<version>0.0.1-SNAPSHOT</version>
<name>员工工作平台常量包</name>
<description>员工工作平台常量包</description>

<dependencies>

</dependencies>

</project>
```

sqlmap

该插件对 Mybatis 的 sqlmap 文件进行扫描比对，找出文件中 select, delete, insert 等元素的变化。

执行环境

N/A

配置参数

ignore	若为 true 则忽略此插件的执行；若为 false 则执行此插件。
processor	插件标识，固定为 'sqlmap'
configuration.conf_path[]	通过这个参数可以指定多个 sqlmap 文件所在的目录。相对路径，相对于 \$WORKSPACE/artifact/source 目录  范例： 'common/sqlmap' 代表扫描 \$WORKSPACE/artifact/source/common/sqlmap 下的所有 xml 文件。

配置文件

N/A

报告

变化类型	命名空间	元素类型	元素标识	文件
DELETE	BaseSysUser	sql	commonColumns	<a href="#">./workspace/base/common/sqlmap/BaseSysUser_SqlMap.xml</a>
DELETE	BaseSysUser	sql	dynamicUpdate	<a href="#">./workspace/base/common/sqlmap/BaseSysUser_SqlMap.xml</a>
DELETE	BaseSysUser	insert	insert	<a href="#">./workspace/base/common/sqlmap/BaseSysUser_SqlMap.xml</a>
DELETE	BaseSysUser	delete	delete	<a href="#">./workspace/base/common/sqlmap/BaseSysUser_SqlMap.xml</a>
DELETE	BaseSysUser	select	getById	<a href="#">./workspace/base/common/sqlmap/BaseSysUser_SqlMap.xml</a>
DELETE	BaseSysUser	select	getByOrgId	<a href="#">./workspace/base/common/sqlmap/BaseSysUser_SqlMap.xml</a>
DELETE	BaseSysUser	select	findAll	<a href="#">./workspace/base/common/sqlmap/BaseSysUser_SqlMap.xml</a>
DELETE	BaseSysUser	sql	dynamicWhere	<a href="#">./workspace/base/common/sqlmap/BaseSysUser_SqlMap.xml</a>
DELETE	BaseSysUser	select	getByEmpNoAndOrgId_1	<a href="#">./workspace/base/common/sqlmap/BaseSysUser_SqlMap.xml</a>
DELETE	BaseSysUser	select	getByEmpNo	<a href="#">./workspace/base/common/sqlmap/BaseSysUser_SqlMap.xml</a>
DELETE	BaseSysUser	select	getByEmpId	<a href="#">./workspace/base/common/sqlmap/BaseSysUser_SqlMap.xml</a>
DELETE	BaseSysUser	select	getByEmpName	<a href="#">./workspace/base/common/sqlmap/BaseSysUser_SqlMap.xml</a>
DELETE	BaseSysUser	select	getByLikeEmpNo	<a href="#">./workspace/base/common/sqlmap/BaseSysUser_SqlMap.xml</a>
DELETE	BaseSysUser	select	pageSelect_count	<a href="#">./workspace/base/common/sqlmap/BaseSysUser_SqlMap.xml</a>
DELETE	BaseSysUser	select	pageSelect	<a href="#">./workspace/base/common/sqlmap/BaseSysUser_SqlMap.xml</a>
DELETE	BaseSysUser	select	pageSelectLike_count	<a href="#">./workspace/base/common/sqlmap/BaseSysUser_SqlMap.xml</a>
DELETE	BaseSysUser	select	pageSelectLike	<a href="#">./workspace/base/common/sqlmap/BaseSysUser_SqlMap.xml</a>
ADD	ArticleInfo	sql	commonColumns	<a href="#">./workspace/artifact/source/common/sqlmap/ArticleInfo_SqlMap.xml</a>
ADD	ArticleInfo	sql	dynamicUpdate	<a href="#">./workspace/artifact/source/common/sqlmap/ArticleInfo_SqlMap.xml</a>
ADD	ArticleInfo	insert	insert	<a href="#">./workspace/artifact/source/common/sqlmap/ArticleInfo_SqlMap.xml</a>
ADD	ArticleInfo	delete	delete	<a href="#">./workspace/artifact/source/common/sqlmap/ArticleInfo_SqlMap.xml</a>
ADD	ArticleInfo	select	getById	<a href="#">./workspace/artifact/source/common/sqlmap/ArticleInfo_SqlMap.xml</a>
ADD	ArticleInfo	sql	dynamicWhere	<a href="#">./workspace/artifact/source/common/sqlmap/ArticleInfo_SqlMap.xml</a>
ADD	ArticleInfo	select	getByProperties	<a href="#">./workspace/artifact/source/common/sqlmap/ArticleInfo_SqlMap.xml</a>
ADD	ArticleInfo	select	pageSelect_count	<a href="#">./workspace/artifact/source/common/sqlmap/ArticleInfo_SqlMap.xml</a>
ADD	ArticleInfo	select	pageSelect	<a href="#">./workspace/artifact/source/common/sqlmap/ArticleInfo_SqlMap.xml</a>
UPDATE	MessSysNotice	sql	commonColumns	<a href="#">./workspace/base/common/sqlmap/MessSysNotice_SqlMap.xml</a>
UPDATE	UserBase	sql	commonColumns	<a href="#">./workspace/base/common/sqlmap/UserBase_SqlMap.xml</a>
UPDATE	UserBase	insert	insert	<a href="#">./workspace/base/common/sqlmap/UserBase_SqlMap.xml</a>

```
<sql id="commonColumns">
    USER_ID ,
    ACCOUNT_ID ,
    USER_NAME ,
    MOBILE ,
    NICK_NAME ,
    IMAGE ,
    SIGN ,
    RANK ,
    TEST ,
    STATUS ,
    REG_TIME ,
    UPD_TIME ,
    FORBID_DAYS ,
    FORBID_TIME
</sql>
```

## struts

该插件对 Apache Struts2 的配置文件进行扫描比对，找出文件中 action 元素的变化。

### 执行环境

N/A

配置参数

ignore	若为 true 则忽略此插件的执行；若为 false 则执行此插件。
processor	插件标识，固定为 'struts'
configuration.conf _path[]	通过这个参数可以指定多个 struts 文件。相对路径，相对于 \$WORKSPACE/artifact/source 目录  范例： 'common/src/main/resources/struts.xml' 代表扫描 \$WORKSPACE/artifact/source/common/src/main/resources/struts.xml 文件。

配置文件

N/A

报告

[ 暂无截图 ]