# Content

Contents

[1) Content 1](#_Toc380083974)

[2) Java 7](#_Toc380083975)

[1) Java developer code standard 7](#_Toc380083976)

[2) Java Project JSP EL 7](#_Toc380083977)

[3) Eclipse 8](#_Toc380083978)

[Note 8](#_Toc380083979)

[Download 8](#_Toc380083980)

[Hellos version 8](#_Toc380083981)

[Eclipse Plug-in 9](#_Toc380083982)

[Plug-in Manage 9](#_Toc380083983)

[SVN 11](#_Toc380083984)

[Aptana 11](#_Toc380083985)

[Maven 11](#_Toc380083986)

[FreeMarker 13](#_Toc380083987)

[Check Style 13](#_Toc380083988)

[Sonar 13](#_Toc380083989)

[WebLogic 13](#_Toc380083990)

[Fortify 13](#_Toc380083991)

[EclEmma 15](#_Toc380083992)

[Eclipse Cmd 15](#_Toc380083993)

[Git 15](#_Toc380083994)

[Code Structure plug-in 15](#_Toc380083995)

[Vaadin 18](#_Toc380083996)

[DevUtils 18](#_Toc380083997)

[4) Cygwin 19](#_Toc380083998)

[Install 19](#_Toc380083999)

[Tomcat Consumer 22](#_Toc380084000)

[Note: 25](#_Toc380084001)

[5) Producer 25](#_Toc380084002)

[1) Link 25](#_Toc380084003)

[2) Example 25](#_Toc380084004)

[6) Consumer sandbox 25](#_Toc380084005)

[1) Link 25](#_Toc380084006)

[2) Using 26](#_Toc380084007)

[7) UI 32](#_Toc380084008)

[Html 32](#_Toc380084009)

[Underline 32](#_Toc380084010)

[HTML5 32](#_Toc380084011)

[1) Tag 32](#_Toc380084012)

[2) Using 32](#_Toc380084013)

[3) Example 32](#_Toc380084014)

[4) Tools 32](#_Toc380084015)

[JavaScript 32](#_Toc380084016)

[1) Plug-in 32](#_Toc380084017)

[2) Using 32](#_Toc380084018)

[SCSS 33](#_Toc380084019)

[1) Overview 33](#_Toc380084020)

[8) Jmockit 33](#_Toc380084021)

[1、about link 33](#_Toc380084022)

[9) Java Coding Standard 34](#_Toc380084023)

[Overview 34](#_Toc380084024)

[Code Template 34](#_Toc380084025)

[Formatter Template 34](#_Toc380084026)

[CheckStyle Template 35](#_Toc380084027)

[10) Sonar 37](#_Toc380084028)

[1、Install 37](#_Toc380084029)

[2、Using 37](#_Toc380084030)

[Example: 37](#_Toc380084031)

[3、Sonar setting to Maven 39](#_Toc380084032)

[4、Sonar Plug in 40](#_Toc380084033)

[11) Fortify 40](#_Toc380084034)

[1. Install 40](#_Toc380084035)

[12) Java 41](#_Toc380084036)

[JMock 41](#_Toc380084037)

[Using steps 42](#_Toc380084038)

[Jmockit 42](#_Toc380084039)

[13) Sonar 44](#_Toc380084040)

[14) CMD 44](#_Toc380084041)

[15) 星座 45](#_Toc380084042)

[16) Office 46](#_Toc380084043)

[1) Word 46](#_Toc380084044)

[1) Overview 46](#_Toc380084045)

[2) Word Function 47](#_Toc380084046)

[2) Office Project 47](#_Toc380084047)

[18) Microsoft Sky Drive 48](#_Toc380084048)

[1) Sync OneNote 48](#_Toc380084049)

[19) Web Service 48](#_Toc380084050)

[20) IE 48](#_Toc380084051)

[1) 编码设置 48](#_Toc380084052)

[21) PC safe 49](#_Toc380084053)

[Tools 49](#_Toc380084054)

[Manage PC Tool 49](#_Toc380084055)

[22) Note 49](#_Toc380084056)

[1) IE和eclipse的编码不一致会导致页面乱码 49](#_Toc380084057)

[23) CSDN 49](#_Toc380084058)

[24) My 网盘 50](#_Toc380084059)

[25) Tomcat 50](#_Toc380084060)

[Remote Debug 50](#_Toc380084061)

[Set init parameter 50](#_Toc380084062)

[热部署 52](#_Toc380084063)

[应用服务器须知 52](#_Toc380084064)

[26) Git 53](#_Toc380084065)

[1) Git Data Flow 53](#_Toc380084066)

[2) Website 53](#_Toc380084067)

[3) Video 53](#_Toc380084068)

[4) Command 53](#_Toc380084069)

[5) Git Structure screenshot 55](#_Toc380084070)

[6) color summary 55](#_Toc380084071)

[7) Git create SSH key 55](#_Toc380084072)

[8) Submit local code to remote github.com 55](#_Toc380084073)

[9) Git-Scm 56](#_Toc380084074)

[10) Git summary 57](#_Toc380084075)

[11) Git command 58](#_Toc380084076)

[12) Git forking 58](#_Toc380084077)

[13) Git Merge conflicts 59](#_Toc380084078)

[14) Fetch and merge 59](#_Toc380084079)

[15) Fetch and rebase 60](#_Toc380084080)

[Demo as below: 61](#_Toc380084081)

[16) Git tag 61](#_Toc380084082)

[1) Command 61](#_Toc380084083)

[17) Demo – view a previous commit in Eclipse 61](#_Toc380084084)

[18) Set git proxy 63](#_Toc380084085)

[27) SiteMinder 64](#_Toc380084086)

[28) DB2 64](#_Toc380084087)

[1. 数据操作语言 (DML：select, delete, insert, update) 64](#_Toc380084088)

[2、数据定义语言 (DDL：create,alter) 66](#_Toc380084089)

[29) Hudson 68](#_Toc380084090)

[1. Overview 68](#_Toc380084091)

[2. Plugins 68](#_Toc380084092)

[30) Some Website 69](#_Toc380084093)

[31) 公关，简称PR 69](#_Toc380084094)

[32) Load Balancer 69](#_Toc380084095)

[1) GSLB 69](#_Toc380084096)

[2) LLB 69](#_Toc380084097)

[33) Work Tool 69](#_Toc380084098)

[34) IDC 70](#_Toc380084099)

[35) 规范 70](#_Toc380084100)

[36) Music Site 70](#_Toc380084101)

[37) 网络/镜像技术 70](#_Toc380084102)

[38) 网站性能分析工具 71](#_Toc380084103)

[1) Yslow 71](#_Toc380084104)

[Overview 71](#_Toc380084105)

[Using 71](#_Toc380084106)

[39) Keyword 71](#_Toc380084107)

[PV 71](#_Toc380084108)

[PII 71](#_Toc380084109)

[滤镜 72](#_Toc380084110)

[RJ-45 72](#_Toc380084111)

[MWC 72](#_Toc380084112)

[ETags 72](#_Toc380084113)

[HDMI 72](#_Toc380084114)

[公司职位简称表 72](#_Toc380084115)

[WWDC 75](#_Toc380084116)

[金砖四国 75](#_Toc380084117)

[金砖国家 75](#_Toc380084118)

[40) 电脑网络端口 76](#_Toc380084119)

[1) 360 76](#_Toc380084120)

[2) Cmd 76](#_Toc380084121)

[41) 截图快捷键 77](#_Toc380084122)

[42) JPDA 77](#_Toc380084123)

[43) Connection Pool 77](#_Toc380084124)

[连接池实现原理 77](#_Toc380084125)

[Example目标 77](#_Toc380084126)

[Technology 78](#_Toc380084127)

[44) Zookeeper 78](#_Toc380084128)

[45) Java Socket 78](#_Toc380084129)

# Java

## Java developer code standard

Code Coverage (test cases) **80%** + Comments 80% + Sonar (90%, not Blocker Critical Major Minor issues, and info not allow more than 30 issues)

## Java Project JSP EL

1. ${pageContext.request.contextPath} : 当前项目URL

Eg:

Document. Forms [0].action =

"${pageContext.request.contextPath}/tridionService.do?action=submitTridionServiceForm";

## 3） classpath

System.getProperty ("java.class.path")

Tomcat classpath=classpath+apache-tomcat-6.0.37\bin\bootstrap.jar

# Eclipse

## Note

32位eclipse只能使用32位 jdk

64位eclipse只能使用64位jdk

## Download

### Hellos version

Windows:

<http://www.eclipse.org/downloads/download.php?file=/technology/epp/downloads/release/helios/SR2/eclipse-jee-helios-SR2-win32-x86_64.zip>

<http://www.eclipse.org/downloads/download.php?file=/technology/epp/downloads/release/helios/SR2/eclipse-jee-helios-SR2-win32.zip>

Mac OS X

<http://www.eclipse.org/downloads/download.php?file=/technology/epp/downloads/release/helios/SR2/eclipse-jee-helios-SR2-macosx-cocoa.tar.gz>

<http://www.eclipse.org/downloads/download.php?file=/technology/epp/downloads/release/helios/SR2/eclipse-jee-helios-SR2-macosx-cocoa-x86_64.tar.gz>

Linux

<http://www.eclipse.org/downloads/download.php?file=/technology/epp/downloads/release/helios/SR2/eclipse-jee-helios-SR2-linux-gtk.tar.gz>

<http://www.eclipse.org/downloads/download.php?file=/technology/epp/downloads/release/helios/SR2/eclipse-jee-helios-SR2-linux-gtk-x86_64.tar.gz>

## Eclipse Plug-in

Some plug-in link is <http://renial.iteye.com/blog/661901>

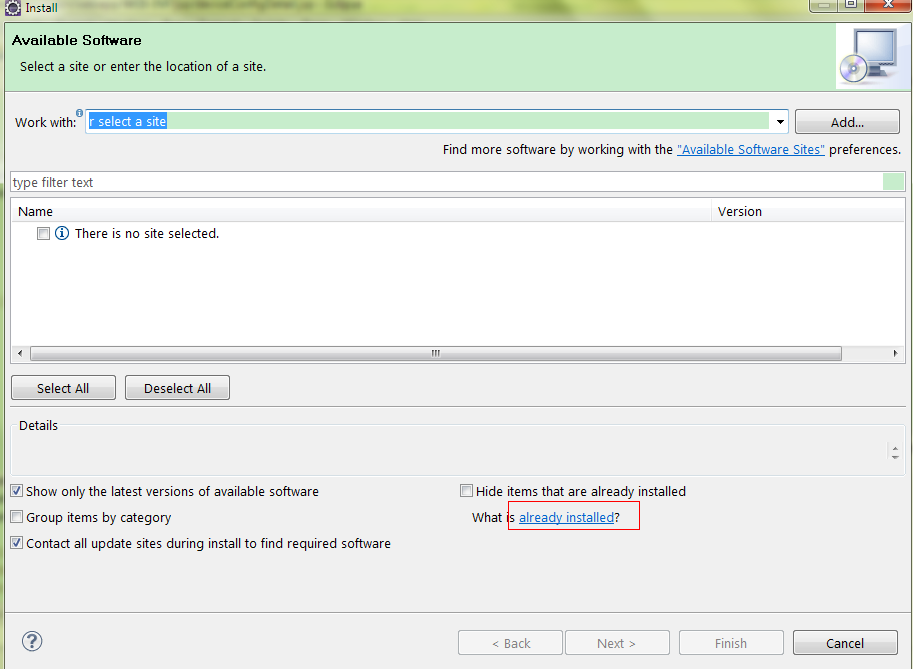
### Plug-in Manage

#### Install or uninstall

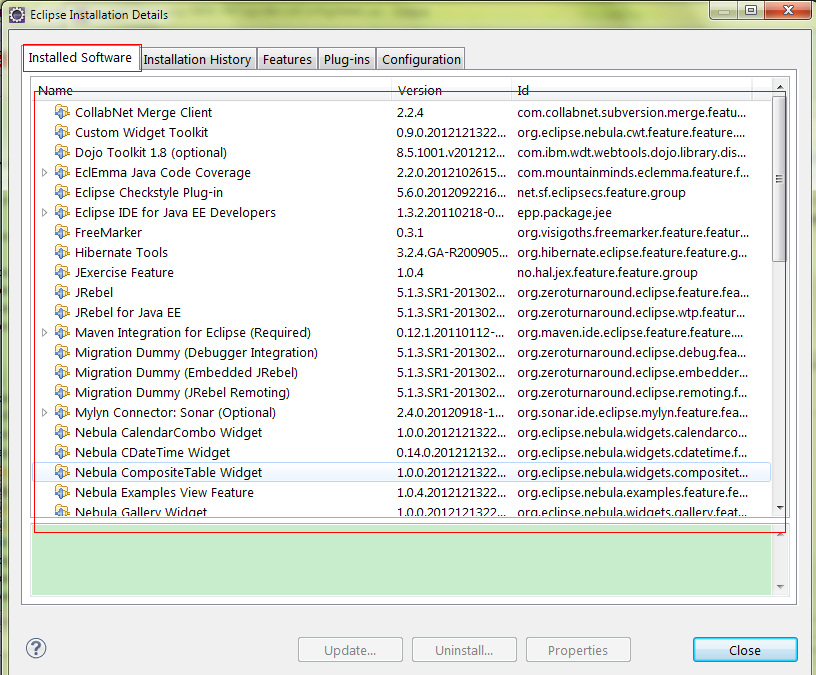
eclipse里有自动卸载插件的。3.5的话，help->install new software->already installed 会显示所有安装的插件，然后选中可以卸载，可以按ctrl同时选中几个再卸载

#### Steps

1. Click already Installed



1. Go to select you want to uninstall or update plug-in



1. Finally Restart your eclipse IDE

### SVN

### Aptana

### Maven

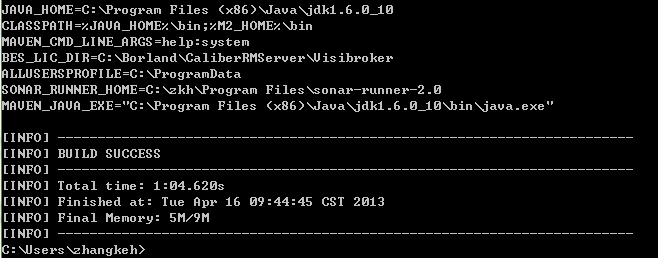
Maven help: system 用于查看所有的Java系统属性

POM: project object model

Eg result:







### FreeMarker

1. Setting
2. Parse

### Check Style

### Sonar

<http://docs.codehaus.org/display/SONAR/Installing+Sonar+in+Eclipse>

Plug-in Link:

<http://dist.sonar-ide.codehaus.org/eclipse/>

Update site:

<http://dist.sonar-ide.codehaus.org/eclipse-archives/2.4.0/>

### WebLogic

Oracle WebLogic Server Tools

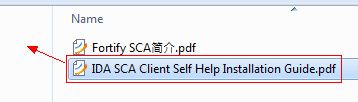
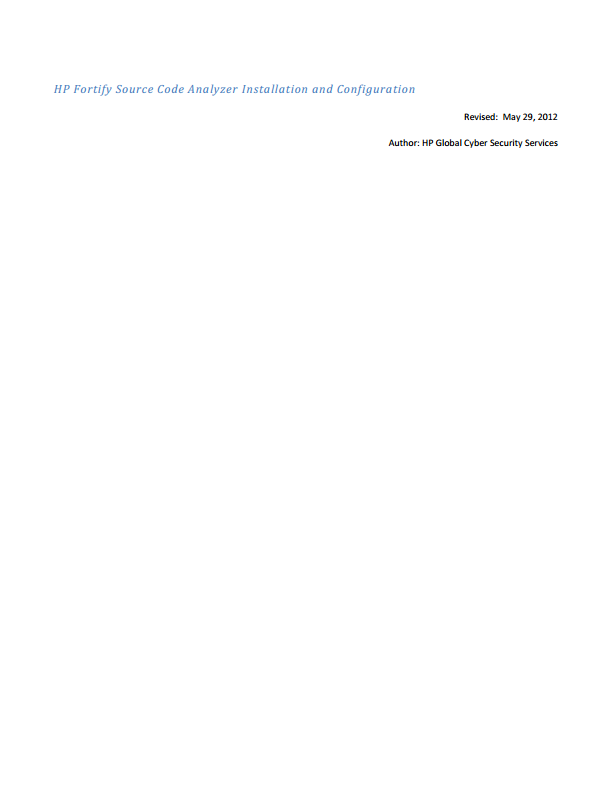
<http://download.oracle.com/otn_software/oepe/helios>

### Fortify

HP Fortify - file:/C:/Program Files/Fortify Software/HP Fortify v3.50/plugins/eclipse/

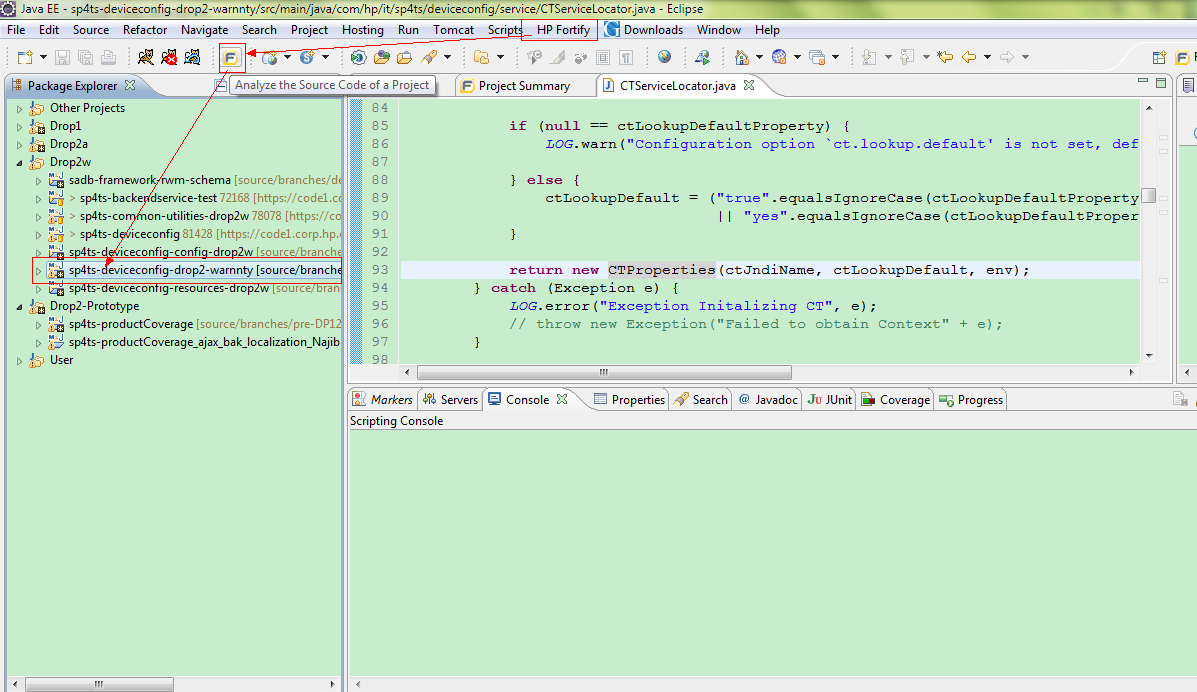
#### Readme Link

IDA SCA Client Self Help Installation Guide.pdf

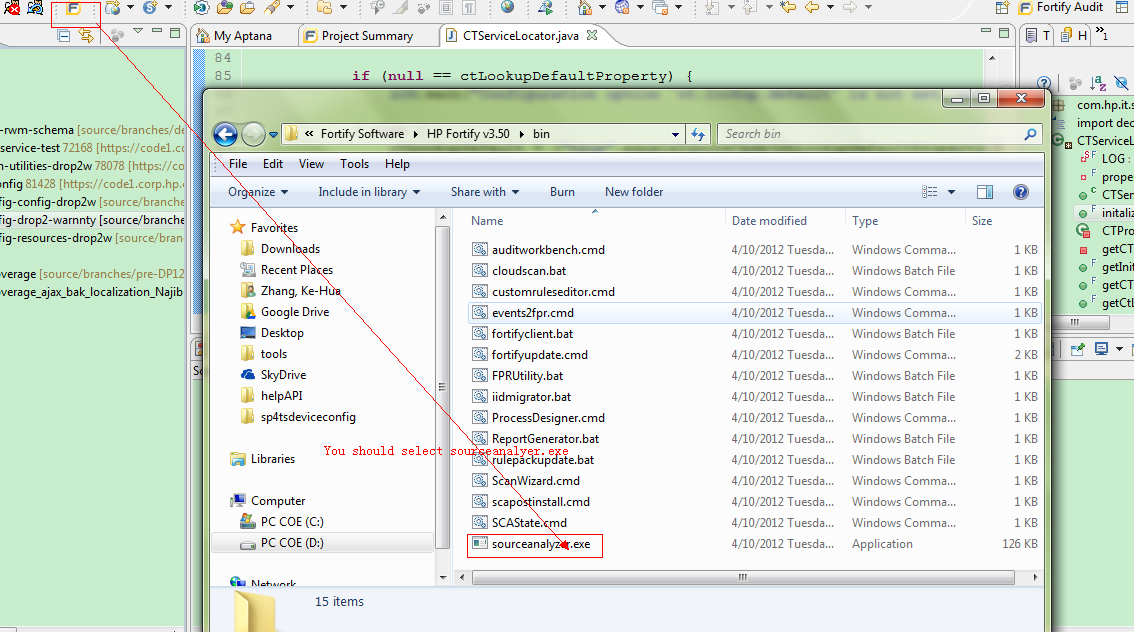


#### Using

1. *Select project to scan by fortify for need to check code issues*



1. You should select fortify scan exe file when First scan, such as



### EclEmma

Java Code Coverage plug in.

### Eclipse Cmd

1. 清理下插件缓存试试,如果你安装上了,这种情况说明插件没有在eclipse启动时被加载,建议在命令行中执行"ellipse -clean"命令试试

### Git

Name:

Link: <http://download.eclipse.org/egit/updates-2.3>

### Code Structure plug-in

#### STAN

1. Web Link

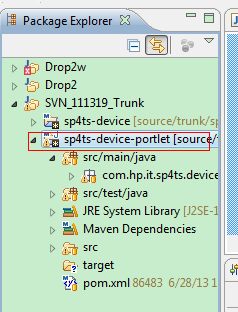
<http://stan4j.com/>

#### Metrics

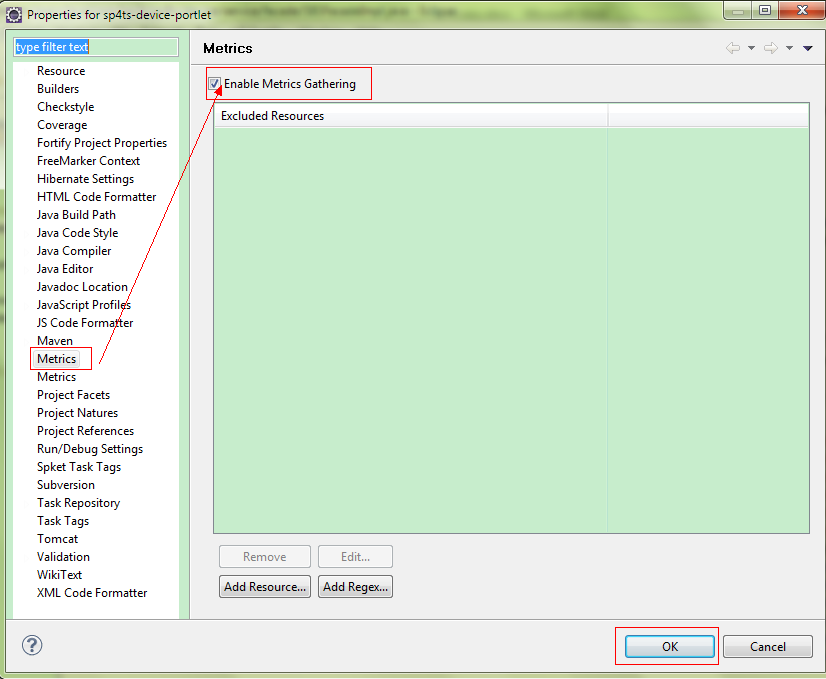
1. Web link:

<http://metrics2.sourceforge.net/>

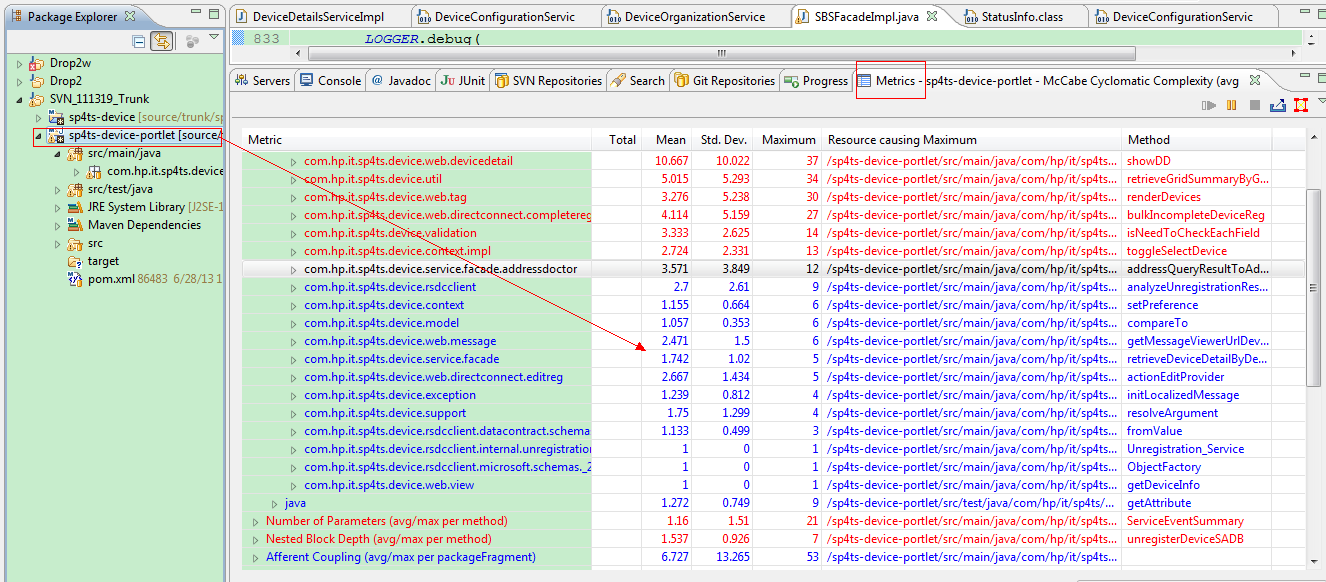
1. Using guide
2. Select a project on package explorer view



1. Right click this project to select properties
2. Select Metrics and enable Metrics Gathering



1. Gathering this project and get metrics analysis result such as below



#### Objectaid

1. Web link:

<http://www.objectaid.com/install-license>

### Vaadin

1. Website

<https://vaadin.com/learn>

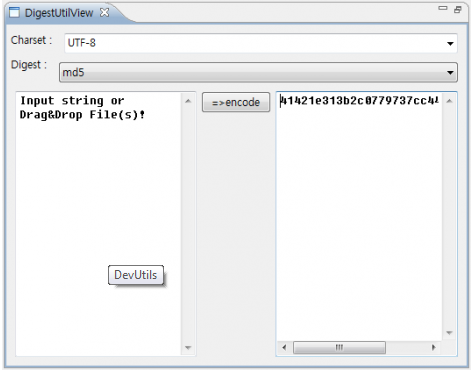
### DevUtils

Guideline: <http://marketplace.eclipse.org/content/devutils>

Plugin site: <http://dev-utils-plugin.googlecode.com/svn/trunk/update>

Overview:

对内容进行不同加密处理试图工具



Guideline:

<https://code.google.com/p/dev-utils-plugin/>

# Cygwin

## Install

Greetings HPSC developers,

If you are not now, or not soon to be working on DP10 or other post DP12 related development then you may immediately file/mark this message for later reference - otherwise read on!

If you know of someone who would be interested in this but is not included in the above distribution, please forward it along. Someday we may have a full HPSC development distribution list…

This message is to announce a new iteration, **V1.0.4**,  of the post-DP12 Vignette portal, (the consumer),  sandbox environments (Nothing is changing about your local producer – the current version of the producer sandbox is still V2.3.)

If you know all of this already, feel free to skip to the instructions below.

This sandbox template implements the current state of the DP10 HPSC page catalog and provides you with a  working local copy of the consumer portal.  This local copy of the portal uses remote portlets from an FUT environment, except for the ones that you are working on, which you wire in from the local OpenPortal producer that is also running on your workstation. This should not be a new concept to you.

Post DP12, the consumer sandbox runs on a different base environment and will be installed differently on your workstation.  Once installed, the usage and operation of the consumer portal will be the same as now.  A detailed, step by step set of instructions will eventually be created for the HPSC Developer web site.

There are a number of advantages to this new installer.  It is completely self-contained. It completely encapsulates Tomcat, Vignette and all other necessary component installations into a single package.  Each new version can be installed alongside the others without any danger of interference.  It includes a script for updating the clean sheet components in a single operation.

It does have a negative side – the installation itself takes quite a while to run – on the order of 20-30 minutes or so.  That is because it does, in fact, build the complete portal environment from scratch.

Kudos go to Liu Ye, of the GADSC China team for leading the adaptation of the SPF installation tool into this new SP4TS installation tool and creating the updates.

**The Environment**

The new consumer runs on the Tomcat servlet container instead of WebLogic.  It is installed using a new installation tool based on the CAS SPF portal installation tool.  The catch is that this new installation process is designed around Bash shell scripts, so the first order of business is to install [Cygwin](http://www.cygwin.com/) on your workstation. Cygwin is a collection of tools that provides a Linux like environment, including a Bash shell.

Installation of Cygwin is relatively simple.  Download the [setup.exe](http://cygwin.com/setup.exe) file and run it.  You should probably rename it to something a little more indicative of what it is, perhaps “Cygwin\_setup.exe”.  Hang on to this file because re-configuring or adding additional components is as simple as running this file again.  Cygwin is structured as a base environment upon which you can then choose from a variety of additional packages.  For our purposes, you only need the base environment plus one additional component. During the installation you will encounter an internet connection screen. The recommended proxy information to use is:

                Proxy Host: web-proxy.atl.hp.com

                Port: 8080

When you reach the “Select Packages” screen, the one additional component you need is “wget”.  The packages are presented as a tree and wget is located under the “Web” item. It is only necessary to select the “Bin” checkbox – unless you just want to see the source.

Upon completion, you will have a Cygwin icon, Description: Description: cid:image001.png@01CC9984.5558D680 , on your desktop.  Launching Cygwin will open a terminal window with a $ prompt. This is your bash shell. The default location for Cygwin is C:\cygwin.  Cygwin creates its own little world inside this directory structure. It is useful to note that you can access this structure from Windows Explorer as well as from the command line.  For example, when you need to download something and place it in your Cygwin home directory, you can manage that from Windows by copying the file to C:\cygwin\home\<Windows username>.

Cygwin will need access to a Java SDK.  For HPSC development this will need to be a 1.5 JDK.  If you already have your workstation java environment just the way you like it then you can skip this part and simply edit the  .bash\_profile file as described below to point to your Java installation.  If you would like to keep a self-contained environment, you can download a free-standing [jdk1.5.0\_22.zip](http://mhyde4.americas.hpqcorp.net/hpscdeveloper/downloads/jdk1.5.0_22.zip) to install inside Cygwin. Unzip this into C:\cygwin\usr\java\jdk1.5.0\_22.  Once this is done edit the file C:\cygwin\home\<Windows username>\.bash\_profile.  (Note the dot at the beginning of the file name!)  At the very end of this file add the following lines:

**# Set JAVA\_HOME to JDK**

**export JAVA\_HOME=/usr/java/jdk1.5.0\_22**

**export PATH=$PATH:/usr/java/jdk1.5.0\_22/bin**

Once this is complete you will have to “logout” of any open Cygwin windows and restart them.  Enter the “set” command and examine the listing of environment variables.  If you see **JAVA\_HOME** pointing to the right place and you see java in the **PATH** variable then you’re all set.

## Tomcat Consumer

**The Portal Installer**

The new portal installer is now something that is versioned and maintained in the [Nexus](http://repo1.corp.hp.com/nexus/index.html#nexus-search;sp4ts-portal-installer) repository. The artifact is named **sp4ts-portal-installer**. The current version is **1.0.4-SNAPSHOT**.  Note that this is a “cdi” file and that there are two versions.  ***The one you want has the “nojvm” classifier in the Nexus listing.***

***Step 1:  Download the installer***

You can download the file from your browser and copy it to your Cygwin home directory.

Or, if you like rocking the command line, you can use wget.  You will need the artifact URL from the repository.  Launch Cygwin and enter the wget command followed by the url inside double quotes. (Cut and paste is your friend.) As of this writing, the following command was accurate:

**$ wget "**[**http://repo1.corp.hp.com/nexus/service/local/artifact/maven/redirect?r=snapshots&g=com.hp.it.sp4ts.apps&a=sp4ts-portal-installer&v=1.0.4-SNAPSHOT&c=nojvm&p=cdi**](http://repo1.corp.hp.com/nexus/service/local/artifact/maven/redirect?r=snapshots&g=com.hp.it.sp4ts.apps&a=sp4ts-portal-installer&v=1.0.4-SNAPSHOT&c=nojvm&p=cdi)**"**

Either way, you wind up with a file in your Cygwin home directory with a name like:

sp4ts-portal-installer-1.0.4-20120105.065047-5-nojvm.cdi

***Step 2: Make it executable***

Remember you are in a unix mindset now, so you must make the downloaded cdi file executable:

**$ chmod 755 sp4ts-portal-installer-1.0.4-20120105.065047-5-nojvm.cdi**

***Step 3: Create the target directory***

The standard location for the portal installation is /opt/casfw, that would be C:\cygwin\opt\casfw.   Again, you can do this from the command line inside Cygwin, or from Windows.  If you are already running a previous version of this consumer then this is already done. By the way, [here’s a quick reference for Bash commands](http://ss64.com/bash/).

***Step 4: Run the installation***

Here’s where things are going to take a while.

From you Cygwin home directory, enter the following command:

**./sp4ts-portal-installer-1.0.4-20120105.065047-5-nojvm.cdi –d /opt/casfw**

Wait…

When the installation completes, the Vignette portal server will be running on port 8180, so your url is:

<http://localhost.hp.com:8180/portal/console>

The commands for stopping and starting the server are:

**/opt/casfw/portal-1.0.4-SNAPSHOT/bin/tomcat-portal.sh stop**

and

**/opt/casfw/portal-1.0.4-SNAPSHOT/bin/tomcat-portal.sh start**

Note that even though the start script completes somewhat quickly, it still takes a few minutes for Vignette to complete its startup.

Note also that the entire consumer environment is encapsulated into this new directory. If you’ve been using 1.0.3 then it’s directory should still be there and you can easily use either version of the portal.

**Local Portlets**

Just as before, you have to re-wire your local producer as a remote portlet server and import portlets etc.

**Web Resources**

The installation procedure will automatically deploy any of the portal side web resources that are available in the Nexus repository.  If your local development involves any web resources that are not yet available via Nexus then you will have to copy them manually to:

**/opt/casfw/portal-1.0.4-SNAPSHOT/software/web-resources**

**Clean Sheet Updates**

Your newly built portal environment includes a script for updating clean sheet components.

**/opt/casfw/portal-1.0.4-SNAPSHOT/bin/upgradeCleansheet.sh**

When new clean sheet components are released you can apply the consumer side bits by running this script.

**Log Files**

All log files are located under **/opt/casfw/portal-1.0.4-SNAPSHOT/var/log**.  Specifically, the Vignette log files are at **/opt/casfw/portal-1.0.4-SNAPSHOT/var/log/vignette-portal**.

Finally, you will note that this is not so neatly integrated into Eclipse as the WebLogic based consumer.  However, this lightweight Tomcat based portal seems to have much better performance than the ones built on WebLogic.  As always, I apologize in advance for any errors that may have crept into the above instructions – if you encounter a problem please let me know.

## Note:

Tomcat consumer web resources folder:

sp4ts-portal-1.1.1-SNAPSHOT\software\web-resources\resource3\hpsc\deviceconfig\css

Weblogic consumer web resources folder:

HPSC\_Consumer\_Sandbox\resources\resource3\hpsc\deviceconfig\js

# Producer

## Link

Portlet Link: <http://localhost:7002/portletdriver/>

Console link: <http://localhost:7002/console>

## Example

Test portlet link: <http://localhost:7001/portal/site/hpsc/it/dash/device>

# Consumer sandbox

## Link

Link: <http://localhost:7001/portal/console>

Console Link: <http://localhost:7001/console>

## Using

If you are not now or not soon to be working on, DP12.1 (Drop2) or later HPSC releases, then you may immediately file/mark this message for later reference - otherwise read on!

**If you know of someone who would be interested in this but is not included in the above distribution, please forward it along. Someday we may have a full HPSC development distribution list…**

This message is to announce a new iteration, **V1.1.1**,  of the post-DP10 Vignette portal (i.e. consumer) sandbox environment.  (Nothing is changing about your local producer – the current version of the producer sandbox is still V2.3.)

If you know all of this already, feel free to skip to the instructions below.

**This new version is based on the DP10 released components, but also includes all of Vignette portal configuration changes for Drop 2. It also includes the latest Cleansheet Components package (1.2.0-SNAPSHOT) and CleansheetWebResources package (2012.01.01).  (Cleansheet Web Resources is still at 2012.09.01 because there are still some problems with the latest version. Until further notice it is recommended that you stay with 2012.09.01.)**

Post DP12, the consumer sandbox runs on a different base environment (eg it is now Tomcat, not WebLogic) and will be installed differently on your workstation.  Once installed, the usage and operation of the consumer portal will be the same as now.  A detailed, step by step set of instructions will eventually be created for the HPSC Developer web site.

There are a number of advantages to this new installer.  It is completely self-contained. It completely encapsulates Tomcat, Vignette and all other necessary component installations into a single package.  Each new version can be installed alongside the others without any danger of interference.  It includes a script for updating the CleansheetWebResources in a single operation.

It does have a negative side – the installation itself takes quite a while to run – on the order of 20-30 minutes or so.  That is because it does, in fact, build the complete portal environment from scratch.

Many thanks go to Pei-Qing Zhou, of the China Team, for producing this update.  Please contact Pei-Qing for support questions.

**The Environment**

The new consumer runs on the Tomcat servlet container instead of WebLogic.  It is installed using a new installation tool based on the CAS SPF portal installation tool.  The catch is that this new installation process is designed around Bash shell scripts, so the first order of business is to install [Cygwin](http://www.cygwin.com/) on your workstation. Cygwin is a collection of tools that provides a Linux like environment, including a Bash shell.

Installation of Cygwin is relatively simple.  Download the [setup.exe](http://cygwin.com/setup.exe) file and run it.  You should probably rename it to something a little more indicative of what it is, perhaps “Cygwin\_setup.exe”.  Hang on to this file because re-configuring or adding additional components is as simple as running this file again.  Cygwin is structured as a base environment upon which you can then choose from a variety of additional packages.  For our purposes, you only need the base environment plus one additional component. During the installation you will encounter an internet connection screen. The recommended proxy information to use is:

                Proxy Host: web-proxy.atl.hp.com

                Port: 8080

When you reach the “Select Packages” screen, the one additional component you need is “wget”.  The packages are presented as a tree and wget is located under the “Web” item. It is only necessary to select the “Bin” checkbox – unless you just want to see the source.

Upon completion, you will have a Cygwin icon, Description: Description: Description: cid:image001.png@01CC9984.5558D680 , on your desktop.  Launching Cygwin will open a terminal window with a $ prompt. This is your bash shell. The default location for Cygwin is C:\cygwin.  Cygwin creates its own little world inside this directory structure. It is useful to note that you can access this structure from Windows Explorer as well as from the command line.  For example, when you need to download something and place it in your Cygwin home directory, you can manage that from Windows by copying the file to C:\cygwin\home\<Windows username>.

Cygwin will need access to a Java SDK.  For HPSC development this will need to be a 1.5 JDK.  If you already have your workstation java environment just the way you like it then you can skip this part and simply edit the  .bash\_profile file as described below to point to your Java installation.  If you would like to keep a self-contained environment, you can download a free-standing [jdk1.5.0\_22.zip](http://mhyde4.americas.hpqcorp.net/hpscdeveloper/downloads/jdk1.5.0_22.zip) to install inside Cygwin. Unzip this into C:\cygwin\usr\java\jdk1.5.0\_22.  Once this is done edit the file C:\cygwin\home\<Windows username>\.bash\_profile.  (Note the dot at the beginning of the file name!)  At the very end of this file add the following lines:

**# Set JAVA\_HOME to JDK**

**export JAVA\_HOME=/usr/java/jdk1.5.0\_22**

**export PATH=$PATH:/usr/java/jdk1.5.0\_22/bin**

Once this is complete you will have to “logout” of any open Cygwin windows and restart them.  Enter the “set” command and examine the listing of environment variables.  If you see **JAVA\_HOME** pointing to the right place and you see java in the **PATH** variable then you’re all set.

**The Portal Installer**

The new portal installer is now something that is versioned and maintained in the [Nexus](http://repo1.corp.hp.com/nexus/index.html#nexus-search;sp4ts-portal-installer) repository. The artifact is named **sp4ts-portal-installer**. The current version is **1.1.1-SNAPSHOT**.  Note that this is a “cdi” file and that there are two versions.  ***The one you want has the “nojvm” classifier in the Nexus listing.***

***Step 1:  Download the installer***

You can download the file from your browser and copy it to your Cygwin home directory.

Or, if you like rocking the command line, you can use wget.  You will need the artifact URL from the repository.  Launch Cygwin and enter the wget command followed by the url inside double quotes. (Cut and paste is your friend.) As of this writing, the following command was accurate:

**$ wget "**[**http://repo1.corp.hp.com/nexus/service/local/artifact/maven/redirect?r=snapshots&g=com.hp.it.sp4ts.apps&a=sp4ts-portal-installer&v=1.1.1-SNAPSHOT&c=nojvm&e=cdi**](http://repo1.corp.hp.com/nexus/service/local/artifact/maven/redirect?r=snapshots&g=com.hp.it.sp4ts.apps&a=sp4ts-portal-installer&v=1.1.1-SNAPSHOT&c=nojvm&e=cdi)**"**

Either way, you wind up with a file in your Cygwin home directory with a name like:

Sp4ts-portal-installer-1.1.1-20121210.084748-4-nojvm

***Step 2: Make it executable***

Remember you are in a unix mindset now, so you must make the downloaded cdi file executable:

**$ chmod 755 sp4ts-portal-installer-1.1.1-20121210.084748-4-nojvm**

***Step 3: Create the target directory***

The standard location for the portal installation is /opt/casfw, that would be C:\cygwin\opt\casfw.   Again, you can do this from the command line inside Cygwin, or from Windows.  If you are already running a previous version of this consumer then this is already done. By the way, [here’s a quick reference for Bash commands](http://ss64.com/bash/).

***Step 4: Run the installation***

Here’s where things are going to take a while.

From you Cygwin home directory, enter the following command:

**./sp4ts-portal-installer-1.1.1-20121210.084748-4-nojvm –d /opt/casfw**

Wait…

When the installation completes, the Vignette portal server will be running on port 8180, so your url is:

<http://localhost.hp.com:8180/portal/console>

The commands for stopping and starting the server are:

**/opt/casfw/portal-1.1.1-SNAPSHOT/bin/tomcat-portal.sh stop**

and

**/opt/casfw/portal-1.1.1-SNAPSHOT/bin/tomcat-portal.sh start**

Note that even though the start script completes somewhat quickly, it still takes a few minutes for Vignette to complete its startup.

Note also that the entire consumer environment is encapsulated into this new directory. If you’ve been using a previous version then it’s directory should still be there and you can easily use either version of the portal.

**Local Portlets**

Just as before, you have to re-wire your local producer as a remote portlet server and import Portlets etc.

**HPSC Web Resources**

The installation procedure will automatically deploy any of the portal side HPSC web resources that are available in the Nexus repository.  If your local development involves any web resources that are not yet available via Nexus then you will have to copy them manually to:

**/opt/casfw/portal-1.1.1-SNAPSHOT/software/web-resources**

**Cleansheet Web Resources**

Your newly built portal environment includes a script for updating CleansheetWebResources.

**/opt/casfw/portal-1.1.1-SNAPSHOT/bin/upgradeCleansheet.sh**

When new CleansheetWebResources are released you can apply the consumer side bits by running this script.

**Log Files**

All log files are located under **/opt/casfw/portal-1.1.1-SNAPSHOT/var/log**.  Specifically, the Vignette log files are at **/opt/casfw/portal-1.1.1-SNAPSHOT/var/log/vignette-portal**.

Finally, you will note that this is not so neatly integrated into Eclipse as the WebLogic based consumer.  However, this lightweight Tomcat based portal seems to have much better performance than the ones built on WebLogic.  As always, I apologize in advance for any errors that may have crept into the above instructions – if you encounter a problem please let us know.

# UI

## Html

### Underline

只使用于<a> text-decoration: underline

## HTML5

### Tag

### Using

### Example

### Tools

1. [PopClip](http://pilotmoon.com/popclip/)
2. [PopLine](http://kenshin54.github.io/popline/)

## JavaScript

### Plug-in

* 1. 一款效果很独特的页面内容缩放插件
     1. Zoom.js
     2. Link is : <http://www.cnblogs.com/lhb25/archive/2013/04/05/dom-zoom-in-javascript.html>
     3. source 

### Using

## SCSS

### Overview

SCSS 之类的预处理器本身就是为了生成 CSS 而设计的。

它的优势在于开发效率高。对于一些样式复杂的站点，用 SASS 之类的工具生成代码比手写 CSS 快得多。但是 SASS 不是为了取代 CSS 而生的，它是为了服务 CSS 而生。  
  
或者类比一下，机器不能直接执行 C，它执行的是编译后的机器码。浏览器也不能直接渲染 SCSS，它渲染的是编译出来的 CSS。  
  
不过 SCSS 与 CSS 的关系和所说的jquery永远不可能取代javascript」有一点区别。 JQuery 是一个 JavaScript 的库而非生成 JavaScript 的工具，它是用 JavaScript 开发出来的；而 CSS 是标记语言。CSS 的一些框架如 lessframework 或者 normalize，它们之于 CSS 的关系更像 jQuery 之于 JavaScript

# 8) Jmockit

## 1、about link

<http://wenku.baidu.com/view/4278d3d776eeaeaad1f3308c.html>

Arguments:

VM Arguments: -javaagent:C:/Users/mich/.m2/repository/mockit/jmockit/0.992/jmockit-0.992.jar

-javaagent:C:/Users/zhangkeh/.m2/repository/mockit/jmockit/0.992/jmockit-0.992.jar

# 9) Java Coding Standard

## Overview

This document is a collection of standards compiled by the ET DP team. It represents best practices we have adopted for framework and service construction.

## Code Template

Code template is for developers to generate the pre-defined comments for the source code. To import the template, please follow the below steps:

1. click on the Windows->Preferences
2. click on the Java from the left navigation tree
3. click on Code Templates
4. click the Import button to import the predefined template.
5. The template can be found at: [eclipse\_custom\_codetemplates.xml](http://bscserver:8082/products/community/modules/wiki/default.aspx?file=eclipse_custom_codetemplates.xml) as below:



## Formatter Template

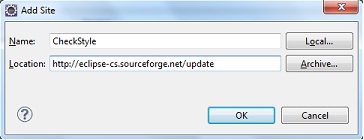
Formatter is for developers to format the source code with pre-defined code style. To import the template, please follow the below steps:

1. click on the Windows->Preferences
2. click on the Java from the leftnavigation tree
3. click on Formatter
4. click the Import button to import the predefined template.
5. The template can be found at:[eclipse\_custom\_formattemplate.xml](http://bscserver:8082/products/community/modules/wiki/default.aspx?file=eclipse_custom_formattemplate.xml) as below:

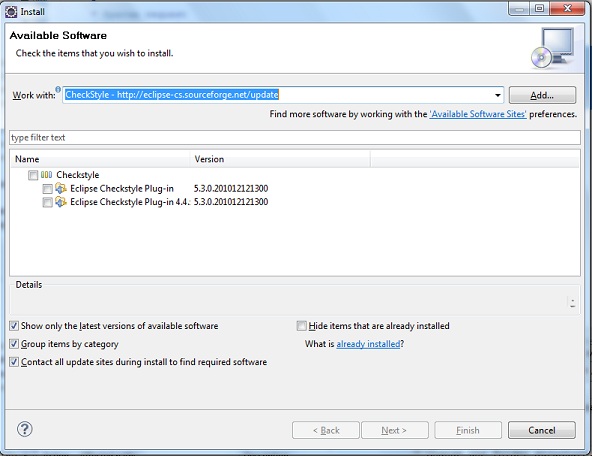
## CheckStyle Template

CheckStyle is a open-source tools for verifying the Java coding style and violation according to the Java coding standard. Follow below steps to install the CheckStyle eclipse plug in:

1. Launch eclipse (Note: Galileo is used for this illustration)
2. Click "Help" -> "Install New Software"
3. Click "Add" button beside "Work with" input box
4. Enter "<http://eclipse-cs.sourceforge.net/update>" in the Location input

[](http://bscserver:8082/products/community/modules/wiki/default.aspx?file=eclipse_checkstyle_install_1.jpg)

1. Click "Ok"

[](http://bscserver:8082/products/community/modules/wiki/default.aspx?file=eclipse_checkstyle_install_2.jpg)

1. Choose All CheckStyle plug-ins
2. Click "Next" button twice
3. Choose "Accept", then click "Finish" button

Once the CheckStyle has been successfully added, you will see the CheckStyle configuration from the "Window" -> "Preferences" settings.

Click the "Properties" button to configuration the "Location" input to point to the CheckStyle template: [eclipse\_custom\_checkstyletemplate.xml](http://bscserver:8082/products/community/modules/wiki/default.aspx?file=eclipse_custom_checkstyletemplate.xml) as below:



# 10) Sonar

Apache Derby 是*Sonar默认*安装的*数据库*, 并且不需要你安装。它能很好的用于Sonar的演示,但是在实际运用中我推荐你使用性能更好更强大的*数据库*。

## 1、Install

Link: <http://docs.codehaus.org/display/SONAR/Installing+Sonar>

## 2、Using

<http://rndwiki.atlanta.hp.com/confluence/display/SBS/Using+Sonar+Eclipse+to+Analyse+Code>

<http://16.49.149.148/wiki/index.php/Sonar>

### Example:

mvn install

mvn sonar:sonar



## 3、Sonar setting to Maven

 <Profiles>

        <Profile>

            <Id>sonar</id>

            <Activation>

                <ActiveByDefault>true</activeByDefault>

            </activation>

            <Properties>

                <!-- Example for MySQL-->

                <sonar.jdbc.url>

                  Jdbc: h2: tcp: //localhost:9092/sonar

                </sonar.jdbc.url>

                <sonar.jdbc.username>sonar</sonar.jdbc.username>

                <sonar.jdbc.password>sonar</sonar.jdbc.password>

                <!-- Optional URL to server. Default value is http://localhost:9000 -->

                <sonar.host.url>

                  http://localhost:9000

                </sonar.host.url>

            </properties>

        </profile>

     </profiles>

## 4、Sonar Plug in

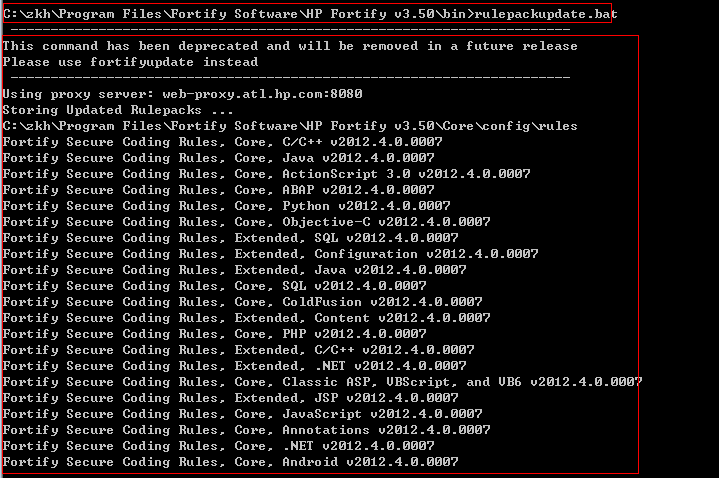
1) Quality Index Plug-in

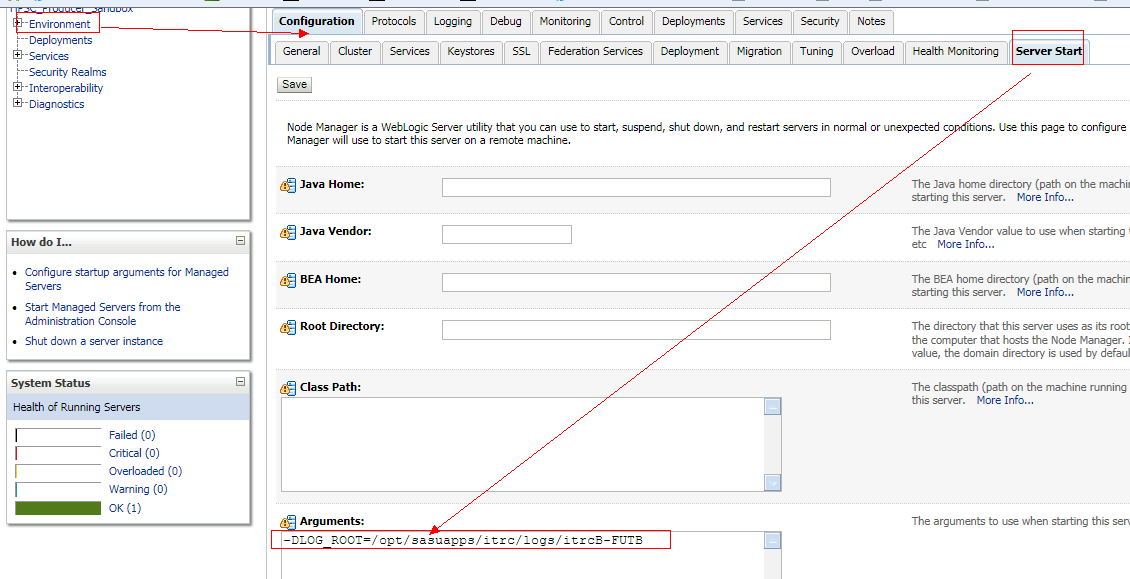
# 11) Fortify

## 1. Install

Install software---->update rule

Link:



JAVA

# 12) Java

Javadoc

Logic

Comments

Guidelines

Sequences

Data table’s relationship

## JMock

1. Introduction

JMock是一个[mock对象](http://www.mockobjects.com/)库，支持[测试驱动开发](http://www.c2.com/cgi/wiki?TestDrivenDevelopment)[的Java](http://java.sun.com/)代码

1. 作用

Mock对象设计和测试你的程序中的对象之间的相互作用

1. 特点
   1. 快速，轻松地定义mock对象，这样你就不会打破节目的节奏。
   2. 让您精确地指定对象之间的相互作用，降低了脆性的测试。
   3. 与您的IDE自动完成和重构功能
   4. 插入您最喜爱的测试框架
   5. 很容易扩展
2. Using steps

Install jMock context object

@Test(expected = JDOMException.class)

### Using steps

1. Install Jmock context object

Org.jmock.Mockery context=new Mockery ();

1. Mock using object

PortletRequest pr=context.mock(PortletRequest.class);

1. Ready run data

Install ready data

1. Checking by jMock Expectations

Expectations用于指定交互规则

context. checking(new Expectations() {{

one (subscriber).receive(message);

}});

## Jmockit

1. Using steps as below:
2. Ready install data (Preparatory stage)
3. Register mock class (Register mocked-up classes)
4. Jmockit stage
5. Implementation stage
6. Keywords
7. new mockit.Expectations() {{

}};

1. returns()
2. notStrict();表示当前行可执行也可不执行。

Eg: portletRequest.setAttribute("FeatureData.OID\_PORTALPARAM", deviceOID);notStrict();

1. returns(Object value);
2. throwsError(Error error);
3. throwsException(Exception exception);

@RunWith(PowerMockRunner.**class**)

PrepareForTest(CodeWithPrivateMethod.**class**)

**public** **class** CodeWithPrivateMethodTest {

@Test(expected = RuntimeException.**class**)

**public** **void** when\_gambling\_is\_true\_then\_always\_explode()

**throws** Exception {

CodeWthPrivateMethod spy =

PowerMockito.spy(**new** CodeWithPrivateMethod());

when(spy, method(

CodeWithPrivateMethod.**class**,

"doTheGamble",

String.**class**,

**int**.**class**

)

)

.withArguments(anyString(),anyInt())

.thenReturn(**true**);

spy.meaningfulPublicApi();

}

}

# 13) Sonar

Web site:

# 14) CMD

**CMD注释形式形式如下：**

**1、:: 注释内容（第一个冒号后也可以跟任何一个非字母数字的字符）  
2、rem 注释内容（不能出现**[**重定向**](http://wenwen.soso.com/z/Search.e?sp=S%E9%87%8D%E5%AE%9A%E5%90%91&ch=w.search.yjjlink&cid=w.search.yjjlink)**符号和管道符号）  
3、%注释内容%（可以用作行间注释，不能出现重定向符号和管道符号）  
4、:标签 注释内容（可以用作标签下方段的执行内容）**

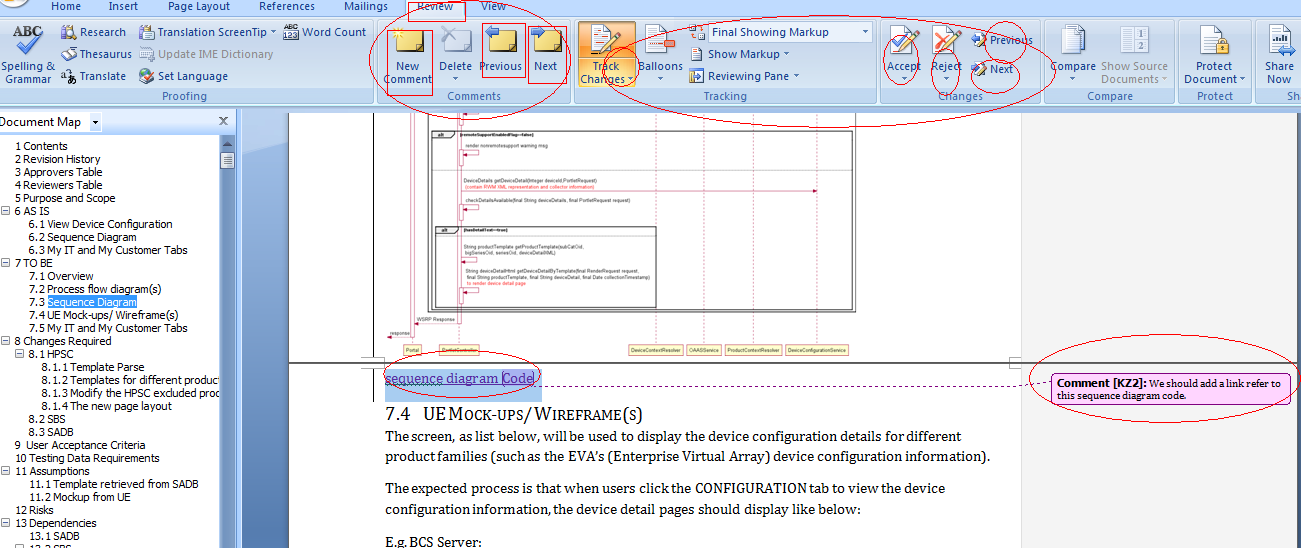
# **15) 星座**



# 16) Office

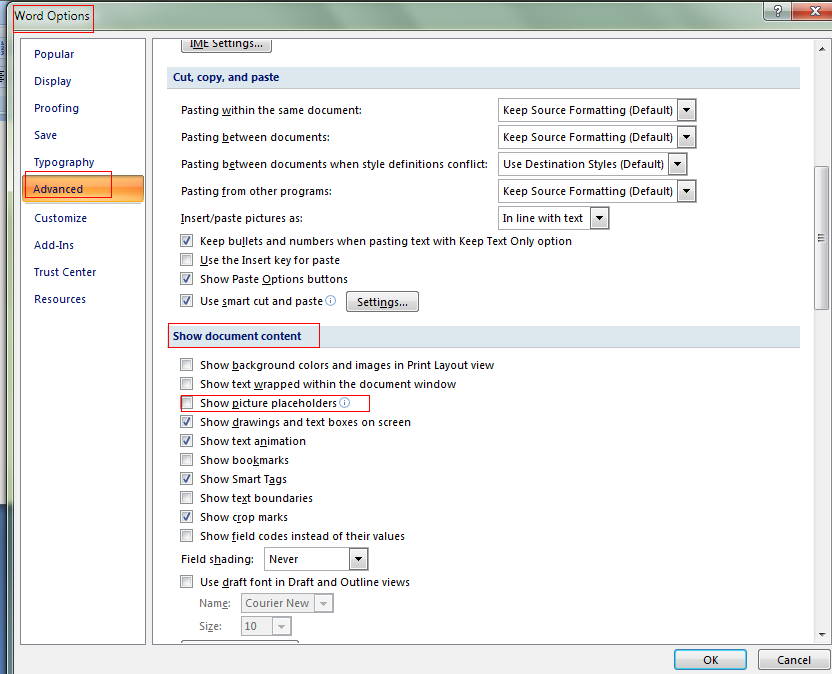
## 1) Word

### 1) Overview



### 2) Word Function

#### Set display picture on Word file



## 2) Office Project



Your Product Key is:

Y9Y6J-K3NVK-TFW9R-2Y8R9-MG7R8

# 18) Microsoft Sky Drive

## Sync OneNote

Microsoft Sky Drive to sync OneNote for EverNote

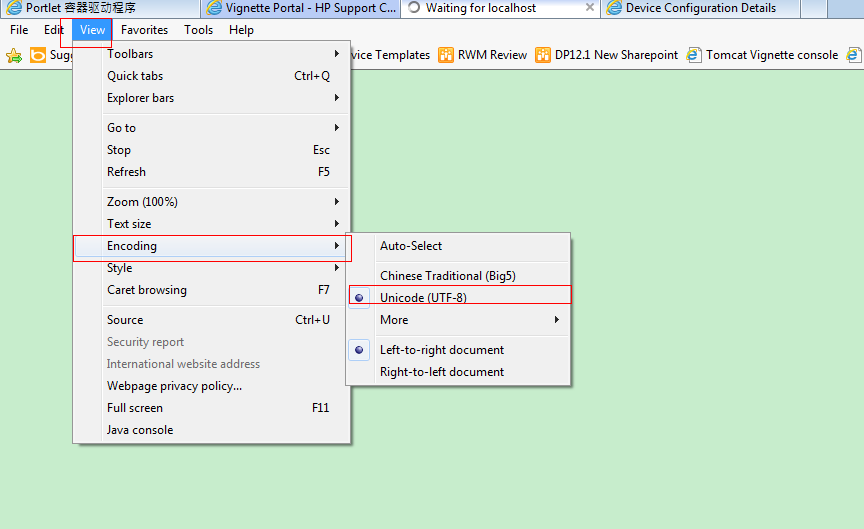


# 19) Web Service



# IE

## 编码设置



# PC safe

## Tools

OptimizerPro

PC performer

Note: 这是一些处理电脑系统问题的工具。有些是需要花钱买license

## Manage PC Tool

**Home:**<http://www.dextronet.com/swift-to-do-list-software>

**Download from Wupload**  
http://www.wupload.com/file/75912932/Swift...cracked-SND.zip  
  
**Download from FileSonic**  
http://www.filesonic.com/file/1585171524/S...cracked-SND.zip

# Note

## IE和eclipse的编码不一致会导致页面乱码

IE: view🡪encoding🡪setting specific encoding type

Eclipse: properties🡪encoding

# CSDN

Chinese software development net

# My 网盘

1. <https://mega.co.nz/>
2. <https://www.115.com/>
3. <http://dbank.vmall.com/netdisk/homepage.html>
4. <http://pan.baidu.com>
5. <http://www.kuaipan.cn>
6. <http://yunpan.360.cn/>

# Tomcat

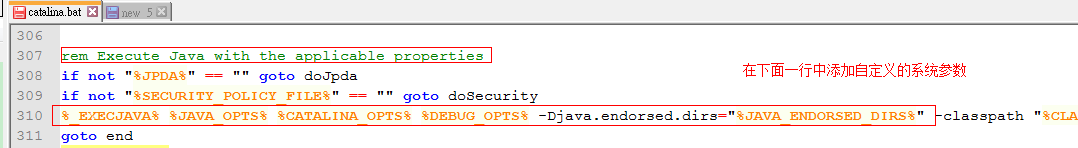
## Remote Debug

<http://nevenchen.iteye.com/blog/1830624>

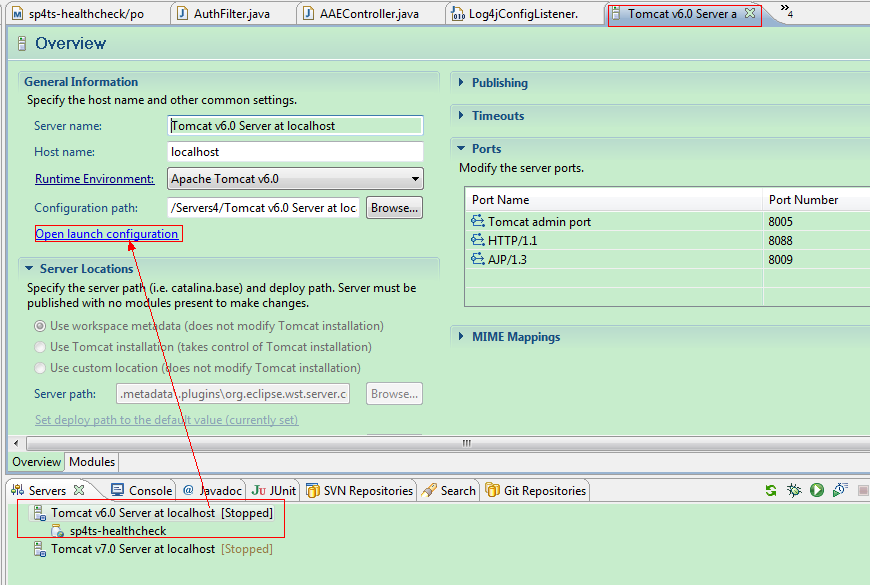
SET CATALINA\_OPTS=-server -Xdebug -Xnoagent -Djava.compiler=NONE -Xrunjdwp: transport=dt\_socket, server=y, suspend=n, address=8787

## Set init parameter

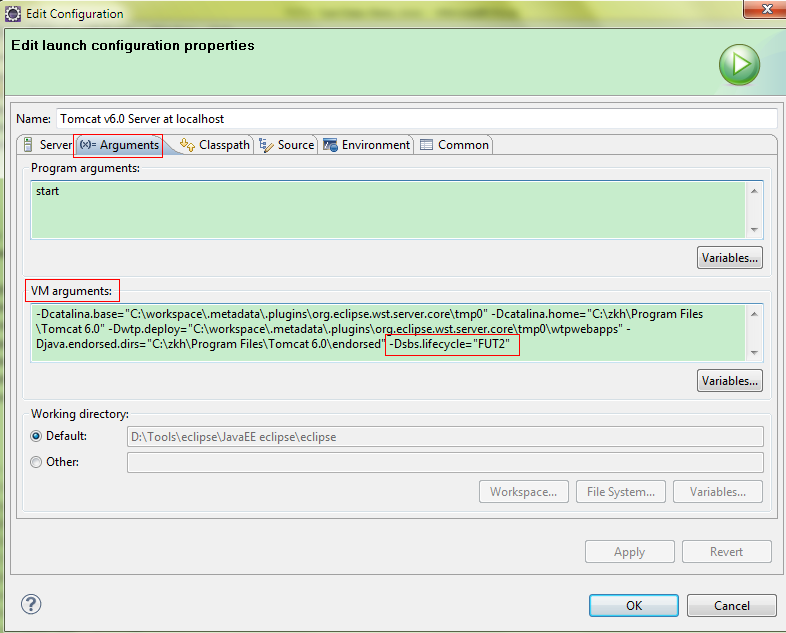
* 1. 服务启动时添加自定义系统参数
     1. 方式1 ： 在catalina.bat文件中添加自定义参数



* + 1. 方式2 ： 通过SDK来设置JVM启动时的初始化参数
       1. Open launch configuration



* + - 1. Set to add define parameter when JVM start



III方式3



右键设置启动参数

## 热部署

Reloadable="true"

<http://wenku.baidu.com/link?url=AeGo8ecBHgRWZaajgWNaDkfb1MzVsU3cpDlZzGAWX3tWWJTwq-2JjGD1VrMZ0quMWuz5BIRJdoSyxlMUSH_6hO0SBZmO2FY35JyUePa8Pbm>

## 应用服务器须知

1. JVM规范有关累加载器章节

<http://java.sun.com/docs/books/vmspec/2nd-edition/html/VMSpecTOC.doc.html>

1. Tomcat累加载机制

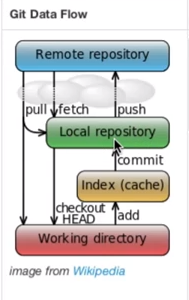
<http://www.huihoo.org/apache/tomcat/>

1. Java发射机制

<http://docs.oracle.com/javase/tutorial/reflect/>

# Git

## Git Data Flow



## Website

<http://git-scm.com>

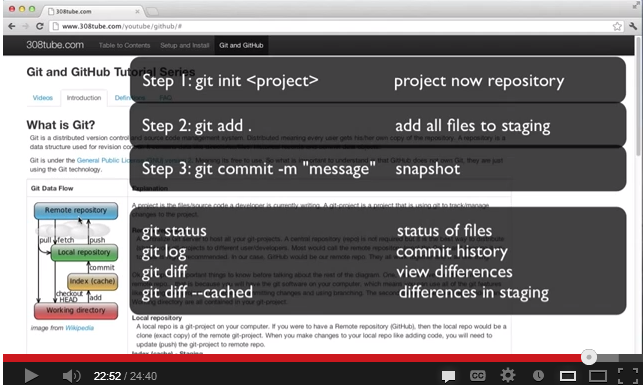
## Video

* 1. <http://www.youtube.com/watch?v=mYjZtU1-u9Y>
  2. <http://www.youtube.com/results?search_query=Git+and+GitHub+Version+Control+Tutorial+-+Part+2&oq=Git+and+GitHub+Version+Control+Tutorial+-+Part+2&gs_l=youtube.3..0.7017.8219.0.8567.3.3.0.0.0.0.764.1401.3-2j6-1.3.0...0.0...1ac.1.11.youtube.ShasHNEpvB4>

## Command

* 1. git config –global user.name “”
  2. git config –global user.email “”
  3. git init “project name”
  4. git status (查看项目状态)
     1. git status will tell you
        1. what branch your on
        2. what files have changed
        3. what files are not tracked
        4. hints on what to do next
        5. very help if you end up using the command line
  5. cd “project folder”
  6. Git add. (submit all files)
  7. git rm –cached “filename”
     1. eg: git rm –cached test.txt
  8. git commit –m “comments”
  9. git log
     1. 如果日志过多，此时余下的日志可以通过Enter回车显示，当全部显示完毕后，按Q 退出查看日志或者使用Shift+ZZ (Capital Z)。原因是因为操作过程等同于Linux下的操作，当操作结束时便可以通过Q 退出当前命令
  10. git log --oneline
  11. git diff
  12. git diff –cached
  13. git commit –a –m “msg”
  14. git commit –m “message”
  15. git status –s
  16. clear

## Git Structure screenshot



## Color summary

* 1. green submitted/modified files
  2. red removed files

## Git create SSH key

Ssh-keygen –t rsa –c “email”

Eg: Ssh-keygen -t rsa -C [ke-hua.zhang@hp.com](mailto:ke-hua.zhang@hp.com)

Ssh –T git@github.com

## Submit local code to remote github.com

1. Git remote add origin “name of remote repo”

Eg:



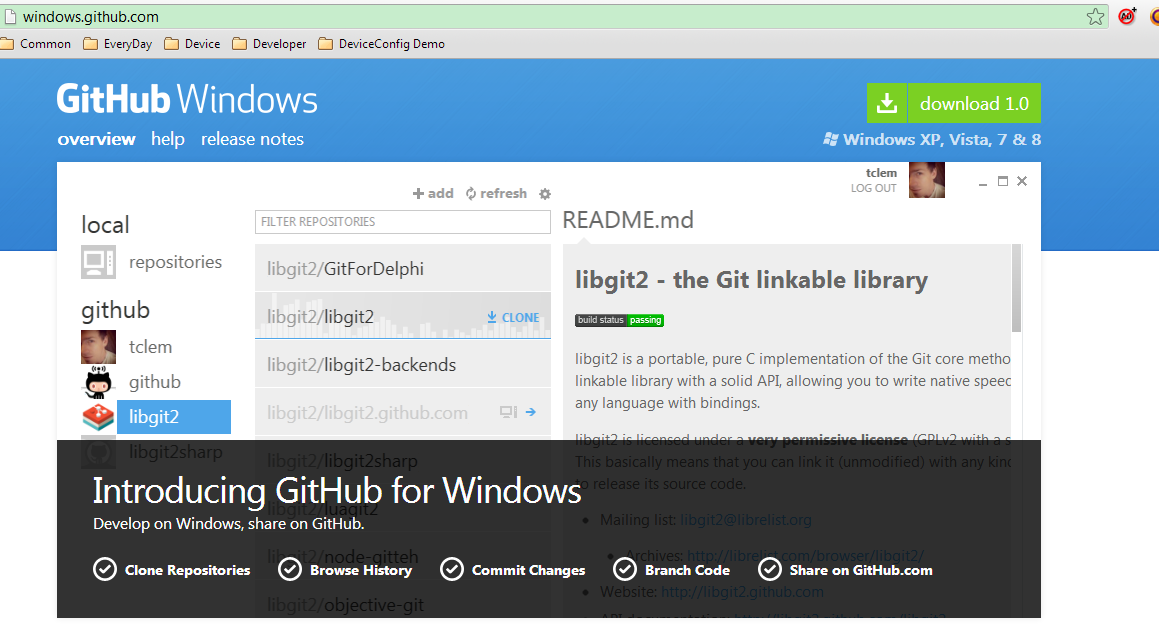


1. Git push origin master （提交代码到github.com网站的个人账户下具体项目中。）

## Git-Scm

* 1. Web link: <http://www.git-scm.com>
  2. Git hub windows client

<http://windows.github.com>



## Git summary

* 1. 本地项目提交到github
     1. Git remote add origin “project repository name”
     2. Git push origin master
  2. Github 上创建自身项目。然后通过本地git bash提交本地内容到Github repository
     1. Git add.
     2. Git commit –m “comment message”
     3. Git remote add origin “github.com repository ssh link”
     4. Git push origin master
  3. 创建branches
     1. Git branch (查看branch)
     2. Git branch “branch 名称” (创建新的branch)
     3. Git checkout “branch 名称” (选择具体的branch进行操作)
     4. Git clone command (we need to get the project down to this computer)
        1. Note: git clone [git@github.com:sanen/GitWebDynamicDemo.git](mailto:git@github.com:sanen/GitWebDynamicDemo.git)

## Git command

* 1. Command
     1. Git merge “branch name”

Note: merge current branch to “branch name”

* 1. Git pull github master

Git pull “name” “branch name” (获取Server具体branch最新代码)

It is similar to a MERGE but PULL synchronizes your local repository with the remote repository

Note: pull is using to sync a branch or master branch

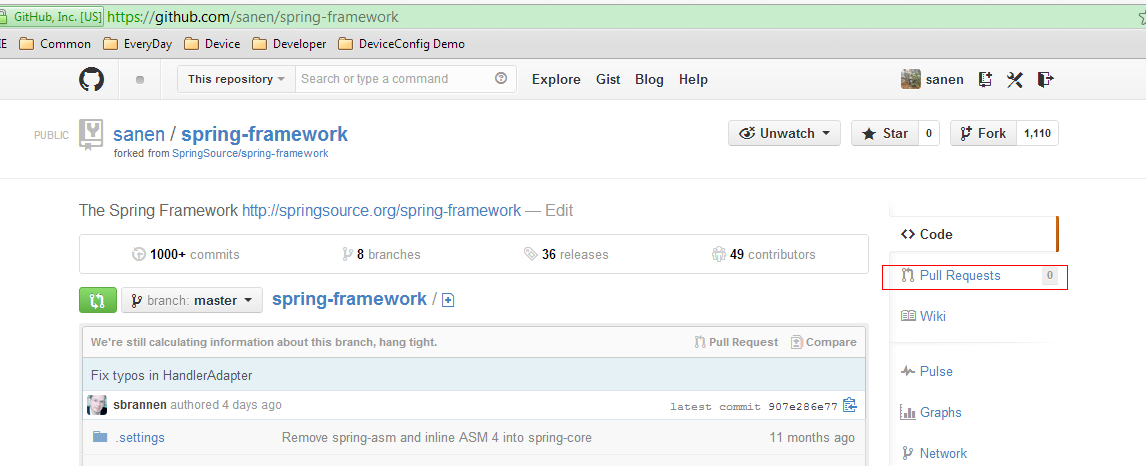
* 1. Git pull – sync with github
  2. Git push – upload
     1. Note: when working in a team, I would recommend always doing a pull before doing a push

## Git forking

1. Go to github
2. Search other account project
3. Click the fork button to clone other account project to myself account



1. Changes for this clone project
2. You should to pull request this project If you want to contribute to the original project, screenshot as below,



## Git Merge conflicts

1. Go to PowerShell
2. Git rebase –continue



## Fetch and merge

1. Git fetch github
2. Git merge github/master

Git merge “name/branch”

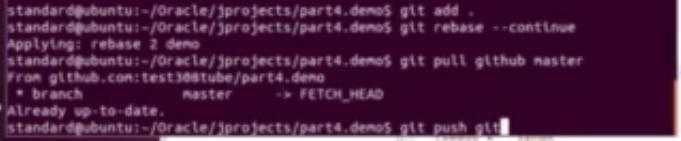
1. Change code then

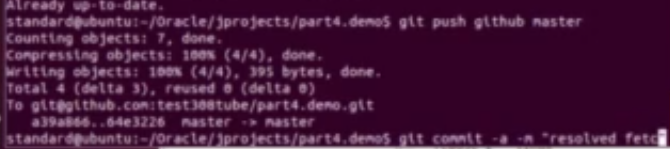
Git pull github master

1. Git push github master

Git push “name” ”branch”

## Fetch and rebase





Git add.

Git rebase –continue

Git pull github master

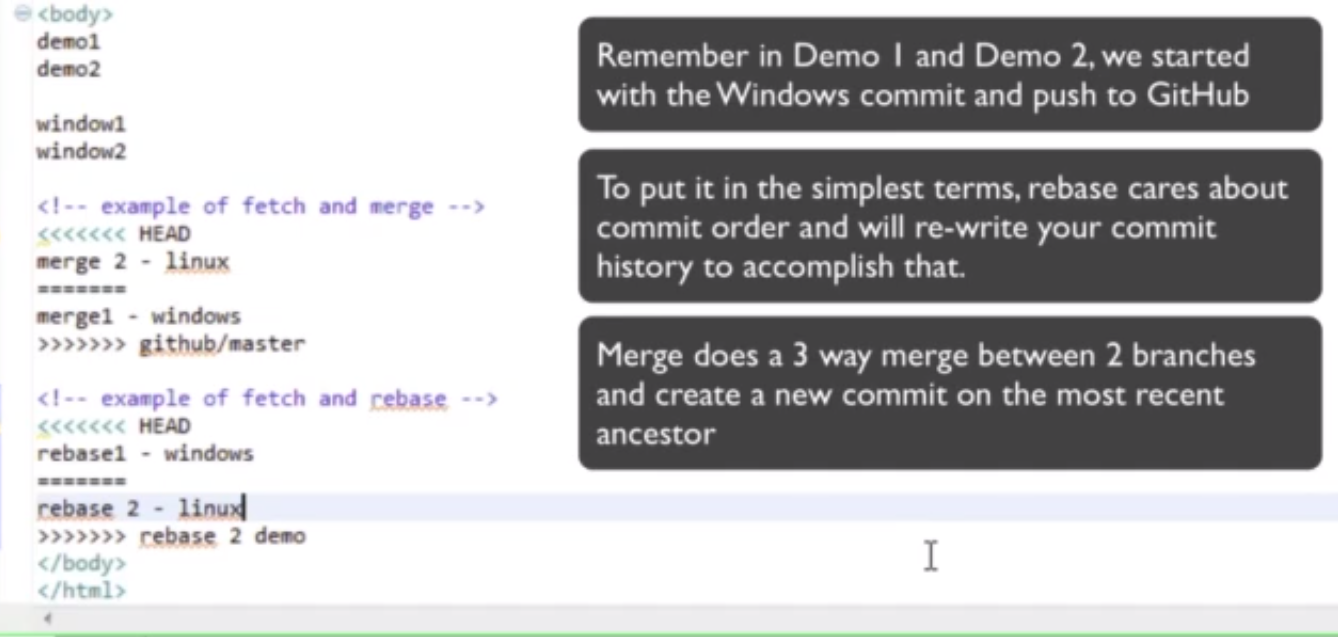
Git push github master

Git commit –a –m “msg”

Git pull github master

Git push github master

### Demo as below:



## Git tag

### Command

Git tag –a “annotation” –m “msg”

Git tag (查看tag)

Git show “tag annotation” (查看具体tag详细信息)

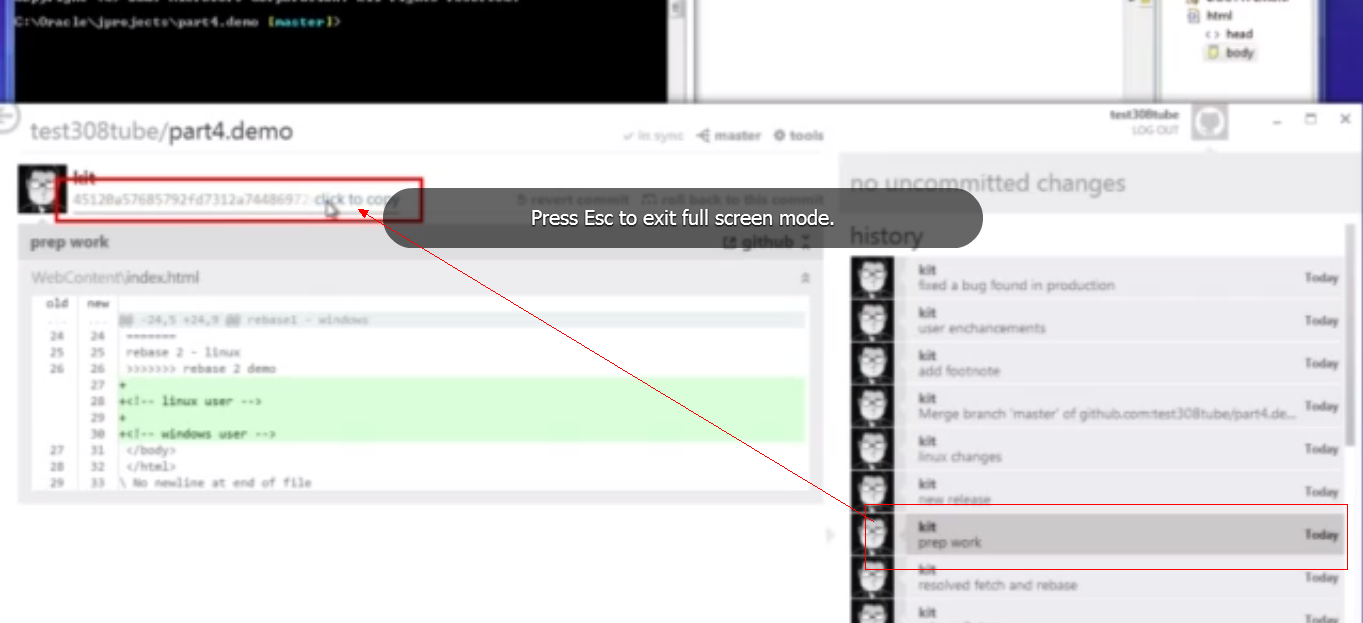
Git push “name” “tag annotation” (提交具体一个tag到git server)

Git push “name” –tag (提交所有的tag到git server)

Git checkout “tag annotation”

## Demo – view a previous commit in Eclipse

* 1. Open into PowerShell window
  2. Git checkout “commit hash”

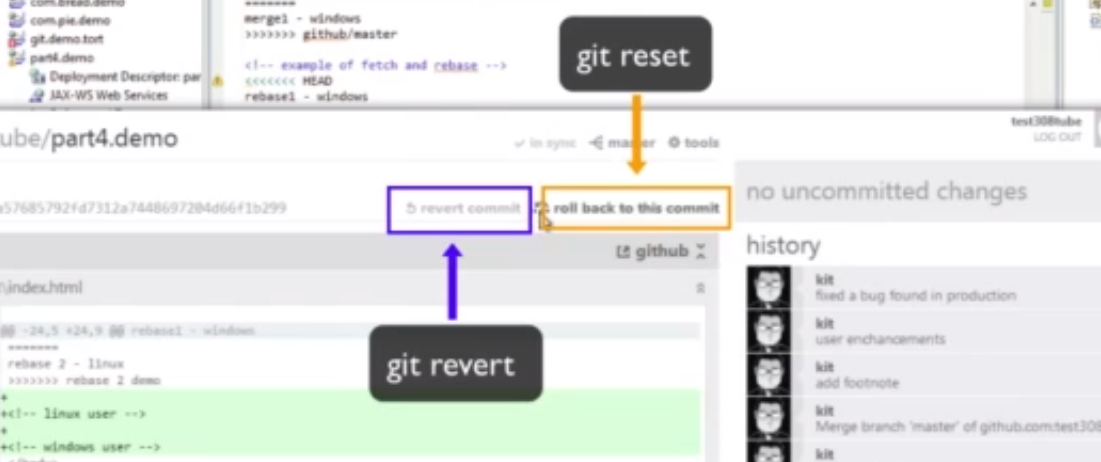
Eg: 

Permanently go back to a preivous commit/snapshot

You can use tow commands

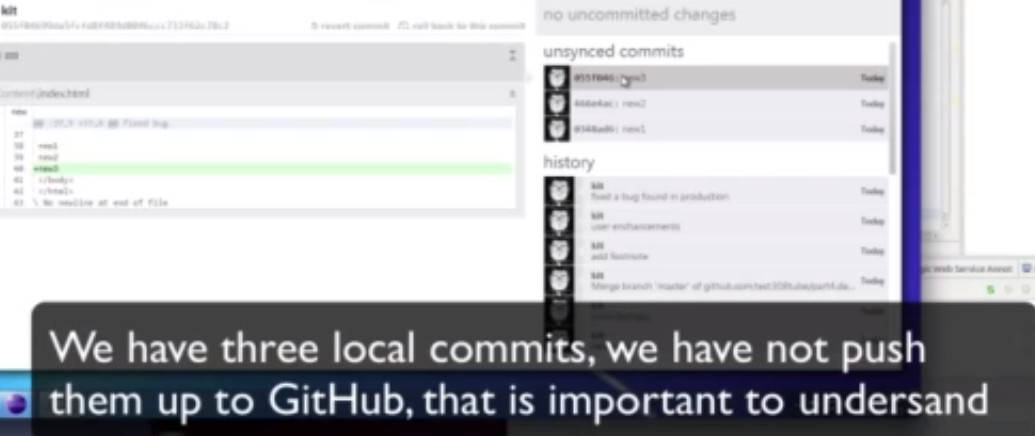
* + - 1. Reset
      2. Revert

Screenshot as below



Test steps:

1. Local commit



1. Git reset : only use reset if you have not shared your project

Git reset –hard “commit hash”

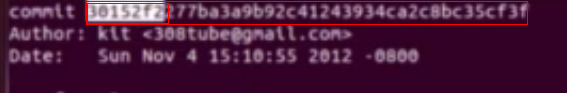
Git revert – when you have shared your project

Command:

Git revert “commit hash”

Ctrl+X

Enter

Note: commit has is below, 

## Set git proxy

Git config –global http.proxy http://user:password@proxy.yourname.com:8080

删除代理设置

Git config --system (? --global? --local) --unset http.proxy

# SiteMinder

1. Web address
   1. <http://www.siteminder.com/>

# DB2

[常用的DB2命令](http://www.iteye.com/topic/111812)

[DB2 CLP及系统命令](http://www.iteye.com/topic/611818)

[精妙SQL](http://www.iteye.com/topic/73854)

[Database圈子](http://database.group.iteye.com/)

## 1. 数据操作语言 (DML：select, delete, insert, update)

      <1> 查询数据库目录：  
             db2 list db directory  
    <2> 查询数据库中表  
           db2 list tables 当前用户  
           db2 list tables for all 所有表  
         db2 list tables for schema schemaname 指定模式的表  
    <3> 显示表结构  
           db2 describe table tablename  
    <4> 插入数据  
           db2 insert into tablename (字段名，字段名...) values (与字段名一一对应的值)  
           db2 insert into tablename1 (字段1，字段2，字段3...)   
         select 字段1，字段2，字段3...from tablename2 + 查询条件  
 <5> 更改表或视图数据  
           db2 update tablename/viewname set 字段名1='',字段2='',...+查询条件  
    <6> 删除数据  
             db2 delete from tablename where + 条件     
   <7> 导入数据  
          db2 "import from E:\name.txt of del insert into tableName"  
          db2 "import from E:\name.ixf of ixf commitcount 5000 insert /create/replace into tableName"  
          db2 "load client from D:\xx.txt of del insert/replace into tabName"(不需要写日志，但插入前表必须存在;不能create table)  
          db2 "load client from D:\xx.txt of del restart/terminate into tabName" 当导入数据出现问题被强行中断时，此表会被加锁，通过此命令可以解锁  
    <8>导出数据

db2 "export to E:\name.txt of del select \* from tableName"  
             db2 "export to E:\name.txt of del MODIFIED BY NOCHARDEL select \* from tableName"(导出不带分号的数据)  
         **导出表结构和数据**  
        db2 "export to E:\name.ixf of ixf MODIFIED BY NOCHARDEL select \* from tableName"  
        db2 "export to E:\name.ixf of ixf MODIFIED BY NOCHARDEL select \* from tableName fetch first (取数+UNM) rows only"(取固定条数)  
         **导出表结构**  
             db2look -d dbName -e -t tableName -o D:\xxx.sql(path) -i userName -w password  
             db2look -d dbName -z tabSchema -e -c -i userName -w password -o + 路径名  
         **导出存储过程结构**  
          db2 "export to xxx.sql of del select text from syscat.procedures where procname='大写存储过程名'"  
      <9> 查询表状态  
            db2 load query table + tableName  
      <10> 查询当前表数据量(数据入库时)  
          db2 select count(1) from tab with ur  
      <11> 修改当前表名、模式名  
 db2 rename table tab1 to tab2

## 2、数据定义语言 (DDL：create,alter)

       <1>创建或删除实例  
       db2icrt instance\_name/db2idrop -f instance\_name  
       linux:db2icrt -u user\_id instance\_name  
       <2>创建视图、表、模式  
     db2 create view/table/schema  
     创建指定用户的模式  
     db2 create schema schName AUTHORIZATION userName  
     db2 create schema AUTHORIZATION userName(没有指定模式名时，模式名隐含为用户名userName)  
      定义含有缺省值的表  
     db2 create table tableName(column1 数据类型,column2 数据类型 default '缺省值')  
     基于已存在的表  
     db2 create table clone\_tablename like tablename   
     db2 create table clone\_tablename as (select \* from tablename) definition only  
     创建物化查询表（MQT）   
     create table new\_table\_name as (select \* from table\_name) data initially deferred refresh deferred;  
     refresh table new\_table\_name;   
     注意：物化表类似一个查询，没有真正形成表，类型显示为Query。但它完全可以当表来用。  
                  创建表并指定其索引表空间  
                         db2 create table(.....) in userspace1 INDEX in userspace2  
                         (userspace1是表所在空间，userspace2是表上索引所在空间)   
    <3>创建视图  
           db2 create view viewname   
                as select 字段名1，字段名2...from table where + 条件  
         with check option 规定一种约束：通过视图插入或更新的每一行都必须符合视图的定义，如：  
         create view emp\_view2(empno,empname,deptno) as (select id,name,dept from employee where dept=10)with check option  
         当此视图用于更新数据或插入新值时，with check option 限制了dept列的输入值  
    <4>修改表(列，主键，唯一约束，检查约束)  
   1)添加新列 alter table tablename ADD COLUMN columnname 数据类型  
   2)添加约束   
   3)修改表中字段 alter table tablename alter columnname set data type 数据类型  
   4) 添加主键 alter table tablename add primary key(c1,c2)  
   <5>删除模式、表、视图  
      drop schema schName <CASCADE|RESTRICT>  
      CASCADE(级联)表示删除模式的同时删除该模式中所有的数据库对象  
      RESTRICT(限制)表示该模式下定义了数据库对象时，限制删除；没有任何数据库对象时才能删除  
   <6>重新组织表及其索引  
      重组表数据  reorg table tableName index indexName(根据索引)  
      重组表索引  reorg indexes all for table tableName  
   <7>重新收集表及其索引统计信息  
      runstats on table tableName for indexes all(跑批前重新收集所用表信息可以提高效率)  
   <8>DB2自动增长主键方法  
      IDENTITY列  
        generated always as identity(start with 1,increment by 1)将一个字段指定为自增长型字段，放在数据类型后。  
      SEQUENCE对象(序列)  
    3、数据控制语言(DCL：grant,revoke)

    将表的特权授予用户  
    grant select,update,delete on table tableName to user userName with grant option  
    将包特权授予同组  
    grant control on package packageName on group groupName with grant option

# 29) Hudson

## 1. Overview

Hudson 拥有的特性包括：

* 易于安装，只要把hudson.war部署到servlet容器，不需要数据库支持。
* 抑郁配置-所有配置都是通过其提供的web界面实现。
* 集成RSS/E-mail/IM-通过RSS发布构建结果或当构建失败时通过e-mail实时通知。
* 生成JUnit/TestNG测试报告。
* 分布式构建支持-Hudson能够让多台计算机一起构建/测试。
* 文件识别-Hudson能够跟踪哪次构建生成那些jar,哪次构建使用哪个版本的jar等。
* 插件支持-Hudson可以通过插件扩展，你可以开发适合自己团队使用的工具。

## 2. Plugins

Hudson通过网络自动下载插件，而且可以通过“更新”进行插件更新。

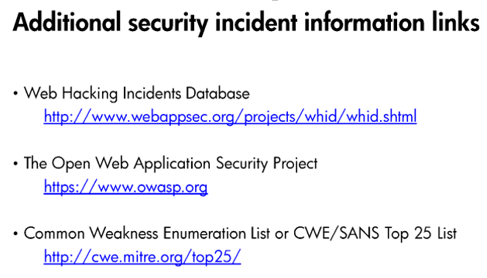
安装：

1. 如果不能连接外网，也可以从一家安装过的插件的hudson中奖插件copy出来，hudson插件的后缀是hpi, 然后可以通过页面“高级”-“上传插件”进行安装.

2. Hudson默认的插件保持目录为：HUDSON\_HOME/plugins, 所以也可以直接将插件copy到该目录当中。重启hudson即可。

HUDSON\_HOME：hudson的主目录，默认为当前用户目录下的.hudson

# 30) Some Website



# 31) 公关，简称PR

# 32) Load Balancer

## 1) GSLB

Global Service Load Balancing (GSLB)-Global& Site

## LLB

Local Load – Balancing (LLB)

# 33) Work Tool

1) NetBeans

2) Myeclipse

3) Eclipse

4) XmlSpy

5) SQL Server

6) Oracle

7) Notepad++

8) Xmind

Overview:

Xmind 是思维图软件，非常好用的记录工具，功能非常强大。

<http://www.xmind.net/download/win/>

# 34) IDC

IDC（Internet Data Center) ，即互联网数据中心，可以为用户提供包括：申请域名、租用虚拟主机空间、主机托管等服务。此外，还有国际数据公司、初始直接费用等多种含义.

# 35) 规范

ISO8583规范

<http://wenku.baidu.com/link?url=MxgVa3GYavetX18JLBy5lb5wdD2BCVCsOWsJ3ZkahTgOnYHe5DhTsMFyr38Lwk8QJt9Q30bI6yXXsAvMiZeITwKu4l3LQL6ejRJ1YOj5LuC>

# 36) Music Site

<https://soundcloud.com>

# 37) 网络/镜像技术

1. 镜像技术

Overview:

<http://baike.baidu.com/link?url=vGHx87Vis0IaDTS_qMOW3na89fHAWp5XaBxfPOnoeqKwlgZJa4U4Bo0GPZZEgPPD>

分类：镜像技术包括三种方式：本地端口镜像; 远程端口镜像; 流镜像

1. 集群技术

<http://baike.baidu.com/view/4804677.htm>

1. 网络设备

<http://baike.baidu.com/view/1158081.htm>

1. ACL

<http://baike.baidu.com/link?url=0gjRv0zrX1d-xHl9VzboBO9dZqwesj8rFT1wt69FZGsZlbcxM4oaJMFq1m1ItUGM>

# 38) 网站性能分析工具

## 1) Yslow

### Overview

Yslow 是雅虎开发的机遇网页性能分析的浏览器插件

<http://developer.yahoo.com/yslow/>

### Using

# 39) Keyword

## PV

Page view 页面浏览量

## PII

个人身份信息

## 滤镜

主要是用来实现图像的各种特殊效果

<http://baike.baidu.com/link?url=xaXTBj84MgNZIEXlFfMxoOcvMuoqtA7Q0BOdc_bo2L-mXb5LKxVlsKtpdyI-bXQw>

## RJ-45

网络端口

## MWC

Mobile world congress 世界移动通信大会，MWC的前身是3GSM

<http://baike.baidu.com/link?url=uESWlODOsyqaaqlGLxps8xh2UaadfEe2rdsjspvZN5rcO6JrfVR-6o100MRNM1SL>

## ETags

在windows下常用的源代码查看工具是Source Inside,在linux下我习惯用etags, gtags, grep来查看源代码

Guide: <http://blog.csdn.net/wuyao721/article/details/3059242>

## HDMI

高清晰度多媒体接口（英文：High Definition Multimedia Interface，HDMI）是一种数字化视频/音频接口技术，是适合影像传输的专用型数字化接口，其可同时传送音频和影音信号，最高数据传输速度为5Gbps

## 公司职位简称表

CEO，CFO，COO，CBO，CCO，CDO，

CHO，CIO，CKO，CMO，CNO，CPO

公司职位简称表.

首席品牌官【CBO】 chief brand officer

首席文化官【CCO】 Chief Cultural Officer

开发总监【CDO】 chief Development officer

首席执行官【CEO】 Chief Executive officer

首席财务官【CFO】 Chief finance officer

人事总监 【CHO】 Chief Human resource officer

首席知识官 知识主管 【CKO 】 Chief Knowledge

首席信息官【CIO】 chief information officer

首席知识官【CKO】 chief knowledge officer

首席市场官【CMO】 chief Marketing officer

首席谈判官【CNO】 chief Negotiation officer

首席营运官【COO】 chief Operation officer

公关总监【CPO】 chief Public relation officer

质量总监【CQO】 chief Quality officer

销售总监【CSO】 chief Sales officer

首席技术官【CTO】 chief Technology officer

评估总监【CVO】 chief valuation officer

首席执行官【CEO】（Chief Executive Officer），

CEO是美国人在20世纪60年代进行公司治理结构改革创新时的产物，它的出现在某种意义上代表着将原来董事会手中的一些决策权过渡到经营层手中。在我国，CEO这个概念最早出现在一些网络企业中。在那里，CEO往往是自封的，也很少有人去研究这一称谓对企业到底意味着什么。但是，当“CEO”在中国叫得越来越响的时候，我们应该认识到，高层人员称谓的改变不是一件小事，设立CEO职位不应仅仅是对时尚的追赶。 CFO（Chief Financial Officer）意指公司首席财政官或

CFO财务总监，是现代公司中最重要、最有价值的顶尖管理职位之一，是掌握着企业的神经系统（财务信息）和血液系统（现金资源）灵魂人物。做一名成功的CFO需要具备丰富的金融理论知识和实务经验。公司理财与金融市场交互、项目估价、

风险管理、产品研发、战略规划、企业核心竞争力的识别与建立以及洞悉信息技术及电子商务对企业的冲击等自然都是CFO职责范围内的事。在一个大型公司运作中，CFO是一个穿插在金融市场操作和公司内部财务管理之间的角色。担当CFO的人才大多是拥有多年在金融市场驰骋经验的人。在美国，优秀的CFO常常在华尔街做过成功的基金经理人。

COO（Chief Operation Officer）首席营运官的职责主要是负责公司的日常营运，辅助CEO的工作。一般来讲，COO负责公司职能管理组织体系的建设，并代表CEO处理企业的日常职能事务。如果公司未设有总裁职务，则COO还要承担整体业务管理的职能，主管企业营销与综合业务拓展，负责建立公司整个的销售策略与政策，组织生产经营，协助CEO制定公司的业务发展计划，并对公司的经营绩效进行考核。

CAO：Answerer 首席答辩人，专门负责解答媒体、债权人和用户等有关网站倒闭问题的询问。

CBO：Business Plan 首席商业计划官，是首席财务官的助理之一，专门针对不同的投资人制订相应的BP。

CCO：Cost Control 首席成本控制官，凡超过100元以上的支出必须由CC0批准。

CDO：Domain name 首席域名官，负责公司域名注册、网站清盘时域名的拍卖、域名法律纠纷等相关问题。

CEO：Exchange 首席交换官，一般由国际CEO自由联盟随时更换，是一个常设的短期职能岗位，类似足球教练。

CFO：Financial 首席财务官，公司最重要的领导人，决定公司命运的主要人物。 CGO：Guideline 首席方针制订官，规划公司的宏伟蓝图，一般是5年以后的目标。

CHO：Harmony 首席协调官，调解投资者和经营者之间的冲突，并确保公司内部矛盾不要泄露。

CIO：Inspector 首席检查官，检查公司内部工作状况，监督员工工作态度。

## WWDC

苹果电脑全球研发者大会Worldwide Developers Conference

## 金砖四国

BRIC: 巴西、俄罗斯、印度和中国

## 金砖国家

BRICS: 巴西、俄罗斯、印度、中国和南非

[BRICS](http://baike.baidu.com/view/117741.htm)

# 40) 电脑网络端口

## 1) 360

使用360的网络连接查看电脑所有端口使用情况，截图如下

## 2) Cmd

命令格式：Netstat -a -e -n -o -s－an  
  
-a 表示显示所有活动的TCP连接以及计算机监听的TCP和UDP端口。  
  
-e 表示显示以太网发送和接收的字节数、数据包数等。  
  
-n 表示只以数字形式显示所有活动的TCP连接的地址和[端口号](http://zhidao.baidu.com/search?word=%E7%AB%AF%E5%8F%A3%E5%8F%B7&fr=qb_search_exp&ie=utf8)。  
  
-o 表示显示活动的TCP连接并包括每个连接的进程ID（PID）。  
  
-s 表示按协议显示各种连接的统计信息，包括[端口号](http://zhidao.baidu.com/search?word=%E7%AB%AF%E5%8F%A3%E5%8F%B7&fr=qb_search_exp&ie=utf8)。  
  
-an 查看所有开放的端口

# 41) 截图快捷键

PrintScreen (整个屏幕截屏)

Alt+PrintScreen (窗口截屏)

QQ: Ctrl+Alt+A

# 42) JPDA

Java 平台调试架构 (Java Platform Debugger Architecture)

# 43) Connection Pool

## 连接池实现原理

1.用户给servlet发送请求，请求Dao要Connection

2. Dao从“连接池”中取出Connection资源，与DB的通讯

3.当用户离开之后，释放该Connection, 那么该Connection被释放到连接池中，等待下一个用户来

## Example目标

通过简单的增删改查来做到下面几个关于连接池的方式，让我们更了解几种优化的方式

1.自定义一个Pool, 来实现类似于现在开源连接池为我们做的一些操作

2.使用Tomcat内置的连接池（apache dbcp）

3.使用DBCP数据库连接池

4.使用C3P0数据库连接池（推荐）

## Technology

* + 1. Commons-pool
    2. Commons-pool2

<http://commons.apache.org/proper/commons-pool/>

# 44) Zookeeper

## Site

<http://zookeeper.apache.org/releases.html>

## Guide

<http://blog.csdn.net/y_xianjun/article/details/8190047>

# 45) Java Socket

<http://blog.csdn.net/kongxx/article/details/7288896#comments>

# 46) Phone

2012&2013智能机销量

<http://www.ifanr.com/news/401210>

# 47) 人生格言

<http://hi.baidu.com/yanhuahawaii/item/3b69983fbf1061dba984283d>

<http://baike.soso.com/v50273180.htm>

# 48）2014 适合结婚的日期

<http://www.artx.cn/nongli/2014-yi-jiehun.html>

# 49） Person

## Email

### Yeah

[zhangkehuait@yeah.net](mailto:zhangkehuait@yeah.net)

cap.cat.sun-

### AOL

[zhangkehuait@aol.com](mailto:zhangkehuait@aol.com)

cap.cat.sun-

### Hotmail

zhangkehuait@hotmail.com/

Cap.cat.sun-619

### Outlook

[zhangkehuait@outlook.com](mailto:zhangkehuait@outlook.com)

cap.cat.sun-

### 126

[Zhangkehuait@126.com](mailto:Zhangkehuait@126.com)

Capc.at.sun-126

Gmail

zhangkehuait@gmail.com

猎聘网

Zkh811