Web Testing Service User Guide

Version 1.0

Contents

1 Introduction

Web Testing Service is a sharing service that can perform customized Web tests and show test results on Web platforms (e.g. Phone, IVI), OS (e.g. Windows, Linux, Android, Tizen) and Web Browsers (e.g. Chrome, Firefox). It does so by connecting to backend Web Service such as Apache2 CGI, Web Socket Server, some test Media Stream, test APPs and performing Web tests from comprehensive Web test suites, including Web APIs defined by W3C specs, Khronos specs and so on. This testing service has a friendly user interface to display tests according to the classification of specs, and easy instructions for customizing web test and web test streams. Besides that, it also presents consolidate test results on Web Browsers as well as generate the results on target device. The Web Testing Service will extend a interface for W3C Testing Service which include all W3C CSS and Platform tests.

This guide is intended to help people prepare for deployment, serve as a resource during the deployment, and help navigate the challenges of maintenance.

2 Deployment Overview

In preparing for a deployment, the people need to understand exactly what this term entails. A better understanding of the aspects of deployment will help them know what to expect next.

2.1 Prior Knowledge

The platform of the service is based on Ubuntu Linux OS, the HTTP server which the service uses is Apache2 or Python base HTTP Server, which means you must know below basic Knowledge:

- Linux Basic Knowledge: http://www.tldp.org/guides.html and http://www.tldp.org/LDP/sag/html/index.html
- Installation of Ubuntu Desktop or Server: http://www.ubuntu.com/
- Configurations of Apache2.*: http://httpd.apache.org/docs/
- CGI supports of Apache2.*: http://httpd.apache.org/docs/2.2/howto/cgi.html
- Python Runtime: http://docs.python.org/ Ubuntu also built-in those related packages already

- Shell Basic Knowledge: https://help.ubuntu.com/community/UsingTheTerminal and http://www.tldp.org/LDP/abs/html/index.html

2.2 Reference Environment

This reference environment also is the default environment which the following sections talk about if no special instruction.

- Platform environment
  - Ubuntu 12.04 LTS Server 64-bit
  - Apache2.2
  - Python2.7.3
- Network environment

  Client need access the below ports of Web Testing Service Server

  - 80 port
    - ✧ Home Page: http://server-ip:80
    - ✧ Web Testing Service Runner: http://server-ip:80/webtests
    - ✧ W3C Testing Service Tests Repo: http:// server-ip/w3c/
  - 8080 port: For tests which need cross domain checking
  - 800* port: For W3C Testing Service
    - ✧ http://server-ip:8000/tools/runner/index.html: W3C Testing Service Runner
    - ✧ TBD

3 Deployment Phase

3.1 Install Ubuntu OS

Please should follow http://www.ubuntu.com/download/server/install-ubuntu-server to install the Ubuntu OS.

After the installation, please use Ubuntu terminal tools to continue following phases.

3.2 Install Apache

```
sudo apt-get install apache2
```

3.3 Install Service Deploy Package

Before the deployment, besides this document, a deployment package named wts.version.zip also provided. Basically, this package contains all tests resources, e.g. test case html files, CGI scripts, media stream. Here are the steps of package deployment:

- su root
- unzip path/to/wts.zip –d /opt/
- mkdir /opt/webtestingservice
- unzip /opt/wts/wts_tests.zip -d /opt/webtestingservice
- Update the variables which depends on the server network
  - Get the server's IP and save the result to var ipaddr:

    ```
    ipaddr=`ifconfig eth0 | awk '/inet/ {split($2,x,":");print x[2]}'`
    ```

    The result likes: 192.168.1.2, then using the following commands to replace the string 'MACRO_IPADDR' of the files under /opt/webtestingservice:

    ```
    replace_records=/tmp/replace_ipaddr.txt

    grep -rn "MACRO_IPADDR" /opt/webtestingservice | cut -d ':' -f1 | sort |
    uniq > ${replace_records}

    for replace_file in `cat ${replace_records}`;do sed -i
    "s/MACRO_IPADDR/${ipaddr}/g" ${replace_file};done
    ```

  - The deployment package includes a script tool "/opt/wts/update_tests.sh" which will help to execute above command lines in batch.
- Deploy CGI binaries

  Specially, based on the deploy Ubuntu version, you need copy the corresponding CGI binaries to Ubuntu system folder "/usr/bin", for Ubuntu 64-bit:

  ```
  cp /opt/wts/tools/64bits/* /usr/bin/
  ```

  Or for Ubuntu 32-bit:

  ```
  cp /opt/wts/tools/32bits/* /usr/bin/
  ```

- Copy the /opt/wts/webtestingservice to /etc/apache2/sites-available and create a soft-link in /etc/apache2/sites-enabled:

```
ln -s /etc/apache2/sites-available/webtestingservice
/etc/apache2/sites-enabled/000-webtestingservice
```

Specially, on non-reference platform, you may need update the content of webtestingservice file as the different version of apache2 may involve new config features. This document will create new section for descriptions of this config file.

- Some special steps of W3C Service:
  - Add below lines to /etc/hosts file as W3C Service need resolve the hostnames

    ```
    Ser-ip          web-platform.test
    Ser-ip          www.web-platform.test
    Ser-ip          www1.web-platform.test
    Ser-ip          www2.web-platform.test
    Ser-ip          xn--n8j6ds53lwwkrqhv28a.web-platform.test
    Ser-ip          xn--lve-6lad.web-platform.test
    ```

  - The “Ser-ip” variable is the Ubuntu Server IP and can get by

    ```
    ifconfig eth0 | awk '/inet/ {split($2,x,":");print x[2]}'
    ```

3.4 Launch the Service

- Launch W3C Testing Service

  ```
  cd /opt/webtestingservice/suites/w3c/web-platform-tests
  python serve.py&
  ```

- Launch Web Testing Service

  ```
  /etc/init.d/apache2 restart
  ```

4 Maintenance Phase

4.1 Uninstall Web Testing Service

- su root
- Stop the W3C Testing Service

  Get the processes of the Service by below command and kill them

```
ps -ax | grep python | grep “python\ serve\.py”
```

- Stop the Web Testing Service

```
/etc/init.d/apache2 stop
```

- Clean the Service data

```
rm –rf /opt/webtestingservice
rm –rf /etc/apache2/sites-available/webtestingservice
rm –rf /etc/apache2/sites-enabled/000-webtestingservice
```

- Clean CGI binaries

```
rm /usr/bin/cgi-getcookie    /usr/bin/cgi-getfield
```

- Clean the /etc/hosts

  Delete the below lines you added before

```
xxxxx              web-platform.test
xxxxx              www.web-platform.test
xxxxx              www1.web-platform.test
xxxxx              www2.web-platform.test
xxxxx              xn--n8j6ds53lwwkrqhv28a.web-platform.test
xxxxx              xn--lve-6lad.web-platform.test
```

You can use this command:

```
sed –i ‘/web-platform\.test/d’ /etc/hosts
```

- The deployment package includes a script tool “/opt/wts/uninstall.sh” which will help to execute above command lines in batch to uninstall all Web Testing Service data.

4.2 Upgrade the Service

You need uninstall the service then install the new version again for Service Upgrade

4.3 Service Apache2 Config Analysis

Some time, there will always be some people want to customize the Service Deployment due to the special environment, here, this section will talk about the details of Service apache2 config file.

- Web Testing Service use two “VirtualHost”, there are 80 and 8080, below is the example of Apache2 “VirtualHost” config lines:

  ```
  NameVirtualHost *:80
  Listen 80
  <VirtualHost *:80>
  xxxxxxxxxxxxxxxxxxxxxxxxx
  </VirtualHost>
  NameVirtualHost *:8080
  Listen 8080
  <VirtualHost *:8080>
  xxxxxxxxxxxxxxxxxxxxxxxxx
  </VirtualHost>
  ```

- Port 80 is the main interface of the Service, home page, runner UI and most of tests entry all host on this port:

  ```
  DocumentRoot /opt/webtestingservice/homepage
  <Directory /opt/webtestingservice/homepage/>
                  Options Indexes FollowSymLinks MultiViews
                  AllowOverride None
                  Order allow,deny
                  allow from all
  </Directory>
  ```

  - The “DocumentRoot” should point to the home page html files folder. E.g.

    ```
    DocumentRoot /opt/webtestingservice/homepage
    <Directory /opt/webtestingservice/homepage/>
                    Options Indexes FollowSymLinks MultiViews
                    AllowOverride None
                    Order allow,deny
                    allow from all
    </Directory>
    ```

  - The URL “/opt” should map to the tests folder. E.g

```
ScriptAlias /opt "/opt/webtestingservice/suites/opt/"

<Directory "/opt/webtestingservice/suites/opt/">

                AllowOverride None

                Options +ExecCGI -MultiViews +SymLinksIfOwnerMatch
+FollowSymLinks +Indexes

                Order allow,deny

                Allow from all

                AddHandler
default-handler .html .htm .css .js .png .xml .gif .ogv .mp4 .ttf .jpg .jpeg .wgt .x
ht .xhtml .exe .mp3 .webm .json #Files having these name extension

</Directory>
```

This “/opt” folder should support CGI handler of *.cgi files

- The URL “/webtests” point to the test runner folder. E.g.

```
Alias /webtests "/opt/webtestingservice/harness"

<Directory "/opt/webtestingservice/harness">

               Options FollowSymLinks MultiViews

               AllowOverride None

               Order deny,allow

               Allow from all

</Directory>
```

- The Service also reserved a URL for W3C tests repo. E.g.

```
Alias /w3c "/opt/webtestingservice/suites/w3c/"

<Directory "/opt/webtestingservice/suites/w3c/">

               Options Indexes

               AllowOverride None

               Order deny,allow

               Allow from all

</Directory>
```

With this W3C URL supports, the end user can access W3C tests to view the tests files.

- Port 8000 is used for some cross domain tests and only involved URL “/opt” to point to the tests folder, the URL has the same config lines as Port 80 “/opt”
- Here is the whole config file example for deployment reference:

```
<VirtualHost *:80>
ServerAdmin webmaster@localhost
        DocumentRoot /opt/webtestingservice/homepage #Directory that forms the main document tree visible from the web, here is '/opt/webtestingservice',
        <Directory />
                Options FollowSymLinks
                AllowOverride None
        </Directory>
        <Directory /opt/webtestingservice/homepage/>
                Options Indexes FollowSymLinks MultiViews
                AllowOverride None
                Order allow,deny
                allow from all
        </Directory>
        ErrorLog ${APACHE_LOG_DIR}/error.log
        LogLevel warn
        CustomLog ${APACHE_LOG_DIR}/access.log combined
        ScriptAlias /opt "/opt/webtestingservice/suites/opt/"
        <Directory "/opt/webtestingservice/suites/opt/">
                AllowOverride None
                Options +ExecCGI -MultiViews +SymLinksIfOwnerMatch +FollowSymLinks +Indexes
                Order allow,deny
                Allow from all
                AddHandler default-handler .html .htm .css .js .png .xml .gif .ogv .mp4 .ttf .jpg .jpeg .wgt .xht .xhtml .exe .mp3 .webm .json #Files having these name extension
        </Directory>
        Alias /webtests "/opt/webtestingservice/harness"
        <Directory "/opt/webtestingservice/harness">
                Options FollowSymLinks MultiViews
                AllowOverride None
                Order deny,allow
                Allow from all
        </Directory>
        Alias /w3c "/opt/webtestingservice/suites/w3c/"
        <Directory "/opt/webtestingservice/suites/w3c/">
                Options Indexes
                AllowOverride None
                Order deny,allow
                Allow from all
        </Directory>
</VirtualHost>
NameVirtualHost *:8080
Listen 8080
<VirtualHost *:8080>
         ScriptAlias /opt "/opt/webtestingservice/suites/opt/"
         <Directory "/opt/webtestingservice/suites/opt/">
                 AllowOverride None
                 Options +ExecCGI -MultiViews +SymLinksIfOwnerMatch +FollowSymLinks +Indexes
                 Order allow,deny
                 Allow from all
                 AddHandler default-handler .html .htm .css .js .png .xml .gif .ogv .mp4 .ttf .jpg .jpeg .wgt .xht .xhtml .exe .mp3 .webm .json
         </Directory>
</VirtualHost>
```

5 Appendix A Terminology

Web Testing Service	The target Service of this document descripted
W3C Testing Service	The sub Service of the Web Testing Service which come from W3C upstream
Web Tests	Some HTML/JS/CSS files for Web Platforms or Web Browsers testing