

[솔루션 기능 테스트]
T-WAS(Resin 3.x_Apache 2.x)
기능 테스트 절차서

한국소프트웨어진흥원
공개SW기술지원센터

<Revision 정보>

| 일자 | VERSION | 변경내역 | 작성자 |
|-------------|---------|-------|-----|
| 2007. 7. 25 | 0.1 | 초기 작성 | 손명선 |
| | | | |

목 차

| | |
|--------------------------------|----|
| 1. 문서 개요 ----- | 4 |
| 가. 문서의 목적 ----- | 4 |
| 나. 본 문서의 사용방법 ----- | 4 |
| 2. 테스트 절차 내역 ----- | 5 |
| 가. JDK 설치 테스트 ----- | 5 |
| 나. Apache 설치 테스트 ----- | 7 |
| 다. Resin 설치 테스트 ----- | 9 |
| 라. Apache 구동 테스트 ----- | 11 |
| 마. Resin 구동 테스트 ----- | 13 |
| 바. Apache & Resin 연동 테스트 ----- | 14 |

1. 문서 개요

본 문서는 SSD기반의 WAS 어플라이언스 장비인 T-WAS의 구성요소인 Resin 3.x와 Apache 2.x의 호환성 및 기능성 검증을 중심으로 테스트하여 향후 관련 솔루션 업체의 참고자료 활용을 위해 제작되었다.

가. 문서의 목적

다음과 같은 세부적인 목적을 달성하기 위하여 작성되었다.

- 0 Resin 3.x 설치 결과
- 0 Apache 2.x 설치 결과
- 0 Resin 3.x와 Apache 2.x의 연동 기능 검증
- 0 진행 중 문제 발생 사항과 각각의 진행사항

나. 본 문서의 사용방법

다음과 같은 방법으로 사용할 수 있다.

- 0 Linux Platform에서 Resin 3.x + Apache 2.x 기능성 테스트 결과를 확인한다.

2. 테스트 절차 내역

가. JDK 설치 테스트


| 단계 | 항목/시험/결과 | |
|----|----------|---|
| 1 | 시험항목 | JDK 설치 |
| | 시험절차 | - JDK 설치 |
| | 시험결과 | <p>- JDK 설치</p> <p>1. Resin은 JVM상에서 동작하므로 JDK가 필수적 요소입니다. Resin 2.x에서는 JDK 1.2 이상이 필요하며 Resin 3.x 에서는 JDK 1.4이상이 필요 합니다. (Resin 3.x 에서는 JDK 1.5 Version이 권장 사항입니다)</p> <p>2. http://java.sun.com/javase/downloads/index_jdk5.jsp 에 접속하여 JDK 1.5를 다운로드 받습니다. (jdk-1_5_0_12-linux-i586.bin)</p> <p>3. JDK 설치 파일을 해당 서버에 업로드 합니다.</p> <p>4. 파일을 업로드 한 위치로 이동을 합니다.</p> <p>5. 설치 파일을 보게 되면 실행권한이 없으므로 실행 권한을 부여 합니다.</p> <pre>#chmod 755 jdk-1_5_0_12-linux-i586.bin</pre> <p>6. jdk 설치 파일을 실행 합니다</p> <pre>#!/jdk-1_5_0_12-linux-i586.bin</pre> <p>7. 아래와 같이 라이선스 동의에 관한 질문이 나오면 yes를 합니다.</p> <pre>Do you agree to the above license terms ? [yse or no] yes</pre> <p>8. JDK 설치가 완료 되었다면 아래와 같이 환경 설정 파일(/etc/profile)을 열어 JDK의 Path를 설정해줍니다.</p> <pre># vi /etc/profile PATH="\$PATH:/usr/jdk1.5.0_12/bin" export JAVA_HOME="/usr/jdk1.5.0_12"</pre> |

| | |
|--|---|
| | <p>9. 새로운 profile 내용을 적용 시켜 줍니다.</p> <pre>#source /etc/profile</pre> |
| | <p>10. java 라고 입력 하였을 때 사용할 수 있는 java 옵션 목록이 정상적으로 나오는지 테스트 합니다.</p> <pre># java Usage: java [-options] class [args...] (to execute a class) or java [-options] -jar jarfile [args...] (to execute a jar file) where options include: -d32 use a 32-bit data model if available -d64 use a 64-bit data model if available -client to select the "client" VM -server to select the "server" VM -hotspot is a synonym for the "client" VM [deprecated] The default VM is client. -cp <class search path of directories and zip/jar files> -classpath <class search path of directories and zip/jar files> A : separated list of directories, JAR archives, and ZIP archives to search for class files. -D<name>=<value> set a system property -verbose[:class gc jni] enable verbose output -version print product version and exit -version:<value> require the specified version to run -showversion print product version and continue -jre-restrict-search -jre-no-restrict-search include/exclude user private JREs in the version search -? -help print this help message -X print help on non-standard options -ea[:<packagename>... :<classname>] -enableassertions[:<packagename>... :<classname>] enable assertions</pre> |

| | | |
|---|---|--|
| | | <pre> -da[:<packagename>... :<classname>] -disableassertions[:<packagename>... :<classname>] disable assertions -esa -enablesystemassertions enable system assertions -dsa -disablesystemassertions disable system assertions -agentlib:<libname>[=<options>] load native agent library <libname>, e.g. -agentlib:hprof see also, -agentlib:jdwp=help and -agentlib:hprof=help -agentpath:<pathname>[=<options>] load native agent library by full pathname -javaagent:<jarpath>[=<options>] load Java programming language agent, see java.lang.instrument </pre> |
| 비 | 고 | |

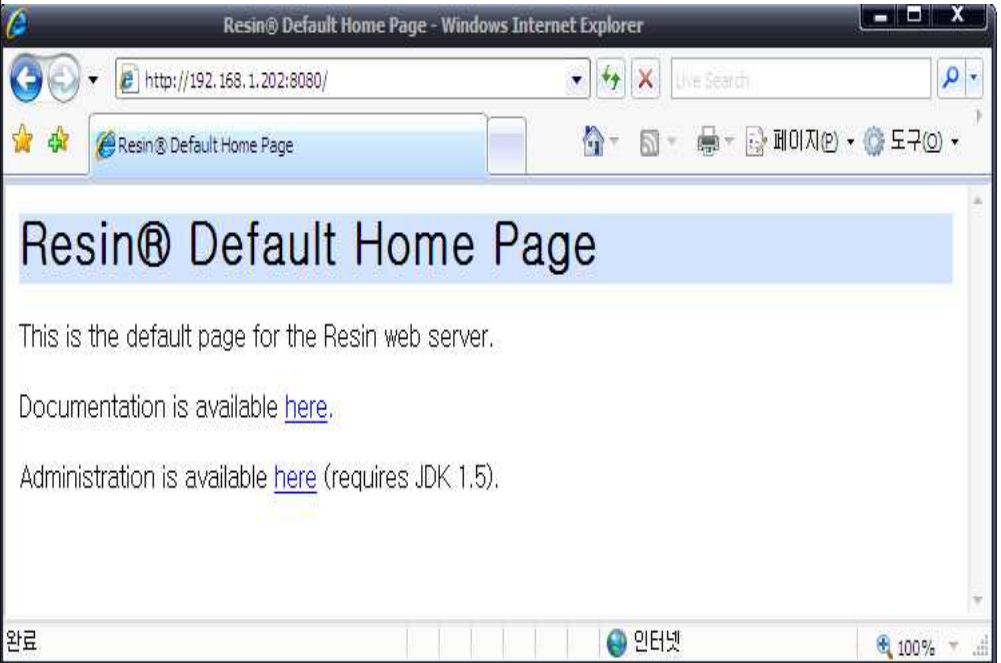
나. Apache 설치 테스트

| 단계 | 항목/시험/결과 | |
|----|----------|---|
| 1 | 시험항목 | Apache 설치 |
| | 시험절차 | - Apache 2.x 설치 |
| | 시험결과 | <p>- Apache 2.x 설치</p> <p>1. 웹사이트(http://httpd.apache.org)에서 Apache 2.0.59 버전을 다운로드 받습니다. (httpd-2.0.59.tar.gz)</p> <p>2. 다운로드 후 서버에 Apache 설치 파일을 업로드 합니다.</p> <p>3. 업로드한 위치로 이동 후 tar 명령어를 이용해 압축을 해제 합니다.</p> <pre># tar -zxvf httpd-2.0.59.tar.gz</pre> <p>4. 압축을 해제 하게 되면 httpd-2.0.59 라는 폴더가 생성이 됩니다. httpd-2.0.59 폴더로 이동을 합니다.</p> <pre># cd httpd-2.0.59</pre> <p>5. configure , make , make install 을 이용하여 Apache 를 컴파일 하고 설치 합니다. 컴파일시 DSO(Dynamic Shared Object)를 지원하도록 -enable-module=so 옵션을 주어 컴파일 하여 설치를 하여야 합니다.</p> <p>(DSO : 동적 모듈적재 방식이며,이 방식은 아파치를 먼저 컴파일 하고 다른 모듈을 추가로 설치할 때 아파치를 재 컴파일 하지 않고 추가적으로 모듈을 설치할수 있는 방식이다 아파치 설치후 Resin 과 연동시 mod_caucho.so라는 모듈을 추가해야 하므로 동적 모듈 적재 방식으로 컴파일을 합니다.)</p> <pre>#!/configure --prefix=/usr/apache --enable-module=so (prefix - 설치할 위치를 지정 해주는 옵션) #make #make install</pre> |

| | | |
|---|---|--|
| | | <p>6. Apache 설치가 정상적으로 끝났다면 정상적으로 구동이 되는지 테스트를 합니다. Apache_Home/bin 디렉토리로 이동을 합니다</p> <pre>#cd /usr/apache/bin</pre> <p>7. Apache 구동후 아래와 같은 페이지가 정상적으로 나오는지 테스트를 합니다.</p> <pre>#!/apachectl start</pre>  |
| 비 | 고 | |

다. Resin 설치 테스트

| 단계 | 항목/시험/결과 | |
|----|----------|---|
| 1 | 시험항목 | Resin 설치 |
| | 시험절차 | - Resin 설치 |
| | 시험결과 | <p>- Resin 설치</p> <p>1. 웹 사이트(http://www.caucho.com/download/index.xtp)에서 tar.gz 의 형식으로 된 Resin Pro 3.0.23 Version을 다운로드 합니다.(resin-pro-3.0.23.tar.gz) (Resin Pro Version은 별도의 라이선스가 필요합니다)</p> <p>2. 다운로드한 Resin 설치 파일을 해당 서버에 업로드 합니다.</p> <p>3. 설치 파일을 업로드 한 위치로 이동을 합니다.</p> <p>4. tar 명령어를 이용하여 Resin 설치 파일의 압축을 해제 합니다</p> <pre>#tar -zxvf resin-pro-3.0.23.tar.gz</pre> <p>5. 압축을 해제 하게되면 resin-pro-3.0.23 폴더가 생성이 됩니다. resin-pro-3.0.23 폴더로 이동을 합니다</p> <pre>#cd resin-pro-3.0.23</pre> <p>6. Apache 와 마찬가지로 configure , make , make install 을 이용하여 컴파일 하고 설치를 합니다.</p> <pre>#!/configure --prefix=/usr/resin //resin 설치할 위치 지정 #make #make install</pre> |

| | |
|-----|--|
| | <p>7. Resin 설치가 되었다면 정상적으로 구동이 되는지 테스트를 합니다. Resin_Home/bin 디렉토리 로 이동을 합니다</p> <pre>#cd /usr/resin/bin</pre> <p>8. Resin을 구동하여 아래와 같은 페이지가 정상적으로 나오는지 테스트 합니다. (http://서버 IP:8080 - resin default port :8080)</p> <pre>#!/httpd.sh start # Resin httpd start at 2007. 07. 04. (수) 13:37:58 KST</pre>  |
| 비 고 | |

라. Apache 구동 테스트

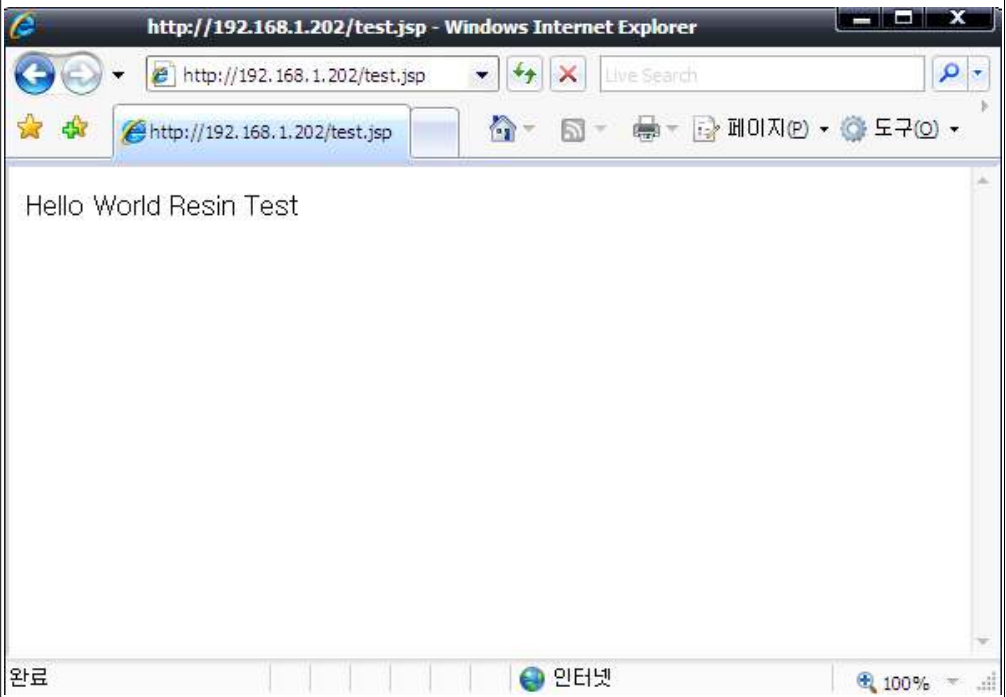
| 단계 | 항목/시험/결과 | |
|----|----------|---|
| 1 | 시험항목 | Apache Start |
| | 시험절차 | ./apachectl start |
| | 시험결과 | <pre># cd /usr/apache/bin # ./apachectl start</pre> |
| | 비 고 | |
| 2 | 시험항목 | Apache Stop |
| | 시험절차 | ./apachectl stop |
| | 시험결과 | <pre># cd /usr/apche/bin # ./apachectl stop</pre> |
| | 비 고 | |

마. Resin 구동 테스트

| 단계 | 항목/시험/결과 | |
|----|----------|---|
| 1 | 시험항목 | Resin Start |
| | 시험절차 | ./httpd.sh start |
| | 시험결과 | <pre># cd /usr/resin/bin # ./httpd.sh start # Resin httpd start at 2007. 07. 04. (수) 13:37:58 KST</pre> |
| | 비고 | |
| | | |
| 2 | 시험항목 | Resin Stop |
| | 시험절차 | ./httpd.sh stop |
| | 시험결과 | <pre># cd /usr/resin/bin # ./httpd.sh stop Stopping httpd</pre> |
| | 비고 | |
| | | |

바. Apache & Resin 연동 테스트

| 단계 | 항목/시험/결과 | |
|----|----------|--|
| 1 | 시험항목 | Apache & Resin 연동 테스트 |
| | 시험절차 | - Apache & Resin 연동 테스트 |
| | 시험결과 | <p>- Apache & Resin 연동 테스트</p> <p>1 Resin 설치 파일이 있는 곳으로 이동을 합니다</p> <pre>#cd /home/resin-pro-3.0.23</pre> <p>2.mod_caucho.so 모듈을 생성하기 위해 --with-apxs=/Apache_Home/bin/apxs 옵션을 주어 Resin을 재 컴파일 합니다.</p> <pre>#!/configure --with-apxs=/usr/apache/bin/apxs # make #make install</pre> <p>3. 정상적으로 컴파일이 되었다면 mod_caucho.so 모듈이 생성이 되었는지 확인을 합니다.</p> <pre>#cd /usr/apache/modules # ls httpd.exp mod_caucho.so</pre> <p>4. Apahce 와 Resin의 Document Root 위치를 동일하게 해줍니다.</p> <ol style="list-style-type: none"> 1) Resin_home/conf 위치로 이동을 합니다. (cd /usr/resin/conf) 2) vi 편집기로 resin.conf 파일을 엽니다. (vi resin.conf) 3) 아래와 같이 Document-directory 위치를 수정합니다 (라인번호 : 256) <pre><resin> <server> <host id="" root-directory="."> <web-app id="/" document-directory="/usr/apache/htdocs" /> </resin></pre> |

| | | |
|---|---|--|
| | | <p>5. Document-directory 위치에 test.jsp 파일을 만들어 정상적으로 페이지가 나오는지 테스트 합니다. (http://서버 IP/test.jsp)</p> <pre>#cd /usr/apahce/htdocs #vi test.jsp <% out.println("Hello World Resin Test"); %> #cd /usr/apache/bin #./apachecl start // Apache start #cd /usr/resin/bin #./httpd.sh start // Resin start</pre>  |
| 비 | 고 | |