Oracle Technical Note

Oracle9*i* Application Server의 설정과 관리를 위한 노하우

오라클에서는 인터넷 서버 제품군으로 Oracle Application Server 제품을 공급하고 있다. 그 동안, 수 차례의 릴리즈를 통하여 현재 OAS V4.0.8.2에 이르는 이 제품을 통해 고객들은 인터넷 기반의 업무를 구축할 수 있었다.

이 글에서는 e-business 고객 및 개발자들이 최근 새롭게 릴리즈된 Oracle9*i* Application Server(Oracle9*i*AS)를 사용하여 더욱 신속하게 응답하고 더욱 뛰어난 확대성을 발휘함으로써 경쟁 우위를 유지할 수 있도록 Oracle9*i*AS의 설정과 관리 팁들을 몇 가지 소개한다. 이 팁들은 일반적으로 많이 사용하는 컴포넌트들(PL/SQL Gateway, Servlet, JSP)을 설정하고, 웹 서버로서 Oracle9*i*AS를 관리하는 데 도움이 될 것이다.

참고로, 현재 Oracle9*i*AS의 버전은 플랫폼마다 약간의 차이는 있을 수 있지만, V1.0.2.0, V1.0.2.0.1, V1.0.2.1까지 릴리즈되어 있다.

| 글 | 전민석 | 한국오라클 인터넷제품지원팀
| minseok.jeon@oracle.com |

ORACLE®

PL/SQL Gateway 설정

PL/SQL Web Toolkit의 설치

PL/SQL Web Toolkit은 Oracle9*i*AS에서 사용하는 패키지로서, 오라클 데이타베이스에 저장되어 있는 PL/SQL 프로시저를 고객들이 사용하는 브라우저에서 수행하기 위해 제공되는 것으로, HTML 태그들을 프로시저화하여 만들어 놓은 빌트인 패키지이다. 간단히 설명하면, 데이타베이스에서 수행되는 패키지나 프로시저를 브라우저에서 보여줄 수 있게끔 HTML 태그들로 변경해 주는 역할을 한다고 생각하면 된다. 물론, 개발자들이 PL/SQL Web Toolkit을 이용하여 인터넷 애플리케이션을 개발할 때, 개발할 패키지나 프로시저 등은 이 PL/SQL Web Toolkit을 이용하게 된다.

Reference : Oracle Internet Application Server V1.0.2
(Using the PL/SQL Gateway) - A8263-01
Oracle Internet Application Server V1.0.2.1
(Using the PL/SQL Gateway) - A87562-01

- *Oracle9iAS v1.0.2 아래 버전인 경우*

(1) 시스템 유저로 데이타베이스에 커넥트한 후 owaload.sql 스크립트를 실행한다. 이때 실행되는 스크립트에서 다음의 설치를 위한 인수(Argument)들을 요구하므로 적당한 값을 지정해 주어야 한다.

```
SQL>@owaload sys_passwd owa_user owa_passwd default_tablespace
    temporary_tablespace connect_string log_file
```

(2) WPG_DOCLOAD를 이용하기 위해서는 docload.sql 을 DBA 권한이 있는 유저로 인스톨한다. 그리고, Oracle9*i*AS 툴킷 유저인OWA_PUBLIC의 오브젝트들을 애플리케이션 유저가 실행할 수 있도록 권한을 준다.

```
SQL> CONNECT OWA_PUBLIC/OWA_PUBLIC@ <TNS ALIAS>
SQL> GRANT EXECUTE ON owa_public.wpg_docload TO public;
```

- *Oracle9iAS v1.0.2 버전 이상인 경우*

sqlplus 또는 svrmgrl에서 시스템 유저로 데이타베이스에 커넥트한 후 $ORACLE_HOME/Apache/modplsql/owa/owaload.sql 스크립트를 수행하면 된다.

PL/SQL Gateway에서 DAD 구성

1. PL/SQL Gateway Configuration 메뉴로 들어간다.
   http://〈host_name〉.〈domain_name〉:〈port〉로 접속하여 mod_plsql을 선택하거나 또는 http://〈host_name〉.〈domain_name〉:〈port〉/pls /admin_/gateway.htm 으로 접속한다.

2. DAD(Database Access Descriptor)를 만들기 위하여 Gateway Database Access Descriptor Settings를 선택한다.
   2.1. Add Database Access Descriptor 화면의 Add Default(blank configuration)를 선택한다.
   2.2. 다음의 내용들을 기술한다. 물론, Connect String은 위에서 PL/SQL Web Toolkit을 설치할 때 지정한 값으로 일반적으로 $TNS_ADMIN /tnsnames.ora에 설정되어 있는 접속하려는 오라클 데이타베이스를 가리

키고 있는 tns-alias를 의미한다.

Parameter	value
Database Access Descriptor Name	ora816
Schema Name	scott
Oracle User Name	scott
Oracle Password	tiger
Oracle Connect String	ora816
Default (Home) Page	cntsample.startup
Document Table	tab_upload
Document Access Path	docs
Document Access Procedure	cntsample.download
Extensions to be uploaded as LONG RAW	*

2.3. OK 버튼을 누른다.

샘플 프로시저와 패키지 작성

다음의 내용들을 SQL*Plus에서 실행한다. 물론, DAD 이름을 ora816 이 아닌 사용자가 지정한 값으로 사용한다면 스크립트의 소스 중에서 해당 이름을 적당히 변경해 주어야 한다.

```
SQL>  CONNECT scott/tiger

SQL>  SET DEFINE OFF

SQL>  CREATE TABLE TAB_UPLOAD (
      NAME VARCHAR2(256) NOT NULL,
      MIME_TYPE VARCHAR2(128) NULL,
      DOC_SIZE NUMBER NULL,
      DAD_CHARSET VARCHAR2(128) NULL,
      LAST_UPDATED DATE NULL,
      CONTENT_TYPE VARCHAR2(128) NULL,
      CONTENT LONG RAW NULL)
/

SQL>  create or replace procedure hella
      is
      begin
          htp.p('Hello ORA816!!');
      end;
/

SQL>  create or replace package cntsample as
          procedure startup;
          procedure input_form;
          procedure write_info (file_name   in  varchar2 default null);
```

```
        procedure download;
    end;
/

SQL>  create or replace package body cntsample as
    procedure startup is
    begin
        htp.p('<Your mod_plsql works fine!');
    end;

    procedure input_form is
    begin
        htp.p('<form enctype="multipart/form-data
            action="/pls/ora816/cntsample.write_info"
            method=""post">');
        htp.p('<input type=file" name="file_name">';
        htp.p('<input type="submit">');
    end;

    procedure write_info
        (file_name   in  varchar2 default null) as
    begin
        htp.header(1,'Upload Status');
        htp.print('Uploaded'|| file_name ||'successfully <p>');
        htp.p('<table border=1>');
        for x in ( select name,mime_type,doc_size,last_updated
        from tab_upload order by last_updated desc) loop
        htp.p('<tr><td>'||x.name|'<td>||x.mime_type||
            '<td>'||x.doc_size||
            '<td>'||to_char(x.last_updated,'mm/dd hh:mm?||
            '<td><a href="/pls/ora816/docs/'||x.name||
                囘>Download</a></tr>');
        end loop;
        htp.p('</table>');
    end;

    procedure download as
        v_filename varchar2(255);
    begin
        v_filename := owa_util.get_cgi_env('PATH_INFO');
        select name into v_filename from tab_upload
            where name = ltrim(v_filename,'/');
        wpg_docload.download_file(v_filename);
    exception
        when others then
```

```
                v_filename := null;
        end;

        end;
/
```

테스트

이제 웹 브라우저에서 설정이 잘 되었는지 살펴볼 차례다. 앞에서 작성한 샘플들은 화면에 간단히 메시지를 뿌리는 것과 파일을 웹에서 업로드/다운로드 하는 것이다.

```
http://<host_name>.<domain_name>:<port>/<path>/<DAD>/<user>.<object>
http://krint-5.kr.oracle.com/pls/ora816/hella
```

"Hello ORA816!!" 이 보이면 성공이다. 이제 다음 샘플을 실행해 보자.

```
http://krint-5.kr.oracle.com/pls/ora816/cntsample.input_form
```

만약, 수행이 되지 않는다면, 적절한 오류 메시지가 출력되므로 그 메시지의 내용에 따라 디버깅을 해야 한다. 여기서, 한 가지 참고 사항을 짚고 넘어가자. 기존에 OAS에서 PL/SQL Web Toolkit을 이용하여 개발된 사용자의 경우, 이를 Oracle9*i*AS로 전환하면서 접하는 문제는 OAS와 Oracle9*i*AS의 Directive가 다른 데서 오는 소스 수정의 부담일 것이다. 하지만 Oracle9*i*AS에서는 이를 소스의 수정 없이 쉽게 설정이 가능하다. Oracle9*i*AS에 있는 패키지나 프로시저를 PL/SQL Gateway를 통해서 브라우저에서 호출하려면 다음과 유사한 URL을 사용해야만 한다.

```
http(s)://[hostname]:[port]/[pls]/[DAD]/[schema].[package].[procedure]
```

- hostname : Oracle9*i*AS가 기동된 서버 이름이나 생성된 가상 호스트 이름
- port : 리스너가 기동되어 있는 포트 번호(default 80, 443 for https)
- pls : PL/SQL 요청이 접수되는 가상 경로(plsql.conf에 등록되어 있다.)
- DAD: 오라클 데이타베이스로의 접속을 위해 사용되는 DAD 이름 (PL/SQL Gateway Configuration에서 정의하며 wdbsvr.app에 등록되어 있다.)
- schema. package. procedure : 데이타베이스에 실행하고 싶은 오브젝트 이름

그런데, 만약 다음과 같이 다른 URL로 같은 프로시저를 호출하고 싶을 경우에는 어떻게 할까?

```
http://[hostname]:[port]/[someotherpath]/[schema].[package].[procedure]
```

PL/SQL Gateway Configuration 자체만으로는 이를 처리할 수 없기 때문에 몇몇 Apache Directives를 이용하여 같은 결과를 얻을 수 있다. 특히, 이전 버전인 WAS/OAS에서 이미 개발되어 있는 인터넷 애플리케이션 환경을 Oracle9*i*AS로 전환할 때 매우 유용할 것이다. 언급한 바와 같이 디폴트 PL/SQL Gateway는 '/pls'로 다음과 같이plsql.conf에 지정되어 있다.

```
[...]
        <Location /pls>
            SetHandler pls_handler
            ...
        </Location>
        [...]
```

그런데, 이 모듈 핸들러를 파일 시스템으로 바로 매핑할 수 없기 때문에 Alias Directives를 사용할 수 없다. 따라서, 가장 간단한 방법은 접수된 요청을 다른 URL로 바꾸는 것이다. plsql.conf에 다음과 유사하게 설정하면, 모든 http://{hostname}:{port}/myapp/... 요청을 http://{hostname}:{port}/pls/{DAD}/... 로 리디렉트해 줄 것이다

```
Redirect /myapp http://[hostname]:[port]/pls/[DAD]
```

또 다른 보다 유연한 방법은 리라이트(Rewrite) 엔진을 사용하는 것이다. (리라이트에 관한 보다 자세한 신택스 등은 URL(http://www.apache.org/docs/mod/mod_rewrite.html)을 참고하기 바란다.) 간단한 리라이트 샘플은 다음과 같다.

```
[...]
        <Location /pls>
            SetHandler pls_handler
            ...
        </Location>
        [...]

        RewriteEngine on
        RewriteRule ^/myapp/(.*)$ /pls/[DAD]/$1 [R,L]
```

여기서 주의할 점은 Windows NT의 경우는 리라이트 엔진의 기능이 디폴트로 설정되어 있지 않으므로 httpd.conf에서 LoadModule rewrite _module modules/ApacheModuleRewrite.dll에 리마크된 것을 풀어주고 AddModule mod_rewrite.c를 추가해 주어야 한다는 것이다.
위의 샘플은 기본적으로 어떤 /myapp/... 요청들을 /pls/{DAD}/ 로 바꿔 주는 역할을 한다. 그러나, 두 방법 모두 한 가지 단점이 있다. 첫 번째 요청이 다시 클라이언트로 돌아가서 새로운 URL로 처리되므로 별개의 두 리퀘스트가 요청되어졌음을 액세스 로그에서 확인할 수 있으며, 이는 물론 서버에 추가적인 부하를 주게 된다. 이에 대한 해결책은 Rewrite Directive의 새로운 버전을 사용하는 것이다.

```
[...]
        <Location /pls>
            SetHandler pls_handler
            ...
        </Location>
        [...]

        RewriteEngine on
        RewriteRule ^/myapp/(.*)$ /pls/[DAD]/$1 [PT]
```

PT(Pass-Through) Flag은 RewriteRule Directives의 아웃풋으로 서비스하기 위한 속성으로 보면 되고, 특히, URL의 명시적인 리디렉트가 이루어지지 않으며, 더군다나, 브라우저에서 Alias를 통해서 원래의 URL을 완전히 숨길 수 있으므로 필요한 경우 매우 효과적이다. 따라서, 애플리케이션 URL의 Alias 이름 사용시 위의 방법을 사용하기 바란다. 단, Virtual Hosting을 사용하는 경우는 상기의 Directives를 plsql.conf 대신 httpds.conf의 사용하는 〈VirtualHost〉 ... 〈/VirtualHost〉 directive 안에 넣어 주어야 한다.

Reference : http://www.apache.org/docs/mod/mod_rewrite.html

Servlet 설정

Oracle9*i*AS 설치 후 디폴트 포트의 index.html 화면에서 제공되는 Servlet 데모는 "Yes, It's working!"이라는 메시지를 보여주며 이는 다음의 URL로부터 서비스되는 것이다.

```
http://<host_name>.<domain_name>:7777/servlet/IsItWorking/
```

그런데, Servlet을 위해 미리 설정된 /servlet이나 /servlets 외에 다른 Virtual Alias를 사용하려면 간단히 httpd.conf에 원하는 Alias만 추가해서 되는 문제가 아니다. 왜냐하면, 사용자가 작성한 Servlet들을 특정 물리적 디렉토리에 위치시킨 후 이를 적당한 이름의 Virtual Path와 매핑시켜 주어야 하기 때문이다.

Oracle9*i*AS의 JServ에서는 Zone을 사용해서 이를 처리하게 되는데, Zone 은 한마디로 여러 Servlet들을 특정 그룹의 이름으로 묶는 것이라고 보면 된다.

따라서, /servlet이나 /servlets 외에 나만의 Virtual Path를 사용하려면 JServ에 Zone을 새로 추가해 주어야 한다. 우선, 원하는 zone.properties와 Servlets을 위치시킬 디렉토리를 생성한다.

```
$ mkdir $ORACLE_HOME/Apache/Jserv/myservlets
$ cd $ORACLE_HOME/Apache/Jserv/myservlets
```

그리고, 기존의 zone.properties를 새로운 디렉토리로 복사한다.

```
$ cp $ORACLE_HOME/Apache/Jserv/servlets/zone.properties
```

그리고 $ORACLE_HOME/Apache/Jserv/etc/jserv.conf 파일을 열어서 다음의 항목을 기존의 ApjServMount 아래에 추가한다.

```
ApJServMount /myservlet /myzone
```

이제 마운트 포인트가 세 개가 되었을 것이다.

```
ApJServMount /servlets /root
ApJServMount /servlet /root
ApJServMount /myservlet /myzone
```

이것은 Virtual Directory /servlets이나 /servlet으로 들어오는 리퀘스트는 /root라는 Zone으로 서비스하고 새로 추가한 /myservlet은 /myzone이라는 Zone으로 처리하겠다는 설정이다. 이제 $ORACLE_HOME/Apache/Jserv/etc/jserv.properties에 기존 Zone 이외에 새로운 Zone과 Properties 파일을 등록하자.

```
zones=myzone
myzone.properties=/d02/ias/Apache/Jserv/myservlets/myzone.properties
```

물론, myzone.properties 파일은 기존의 $ORACLE_HOME/Apache /Jserv/etc/zone.properties 파일을 복사한 후 이름을 변경한다. 그리고, 기존의 Repositories Parameter 값 대신 새로운 디렉토리를 지정한다. 이 디렉토리는 이후 /myservlet에서 사용할 Servlets이 존재할 실제 파일 시스템의 디렉토리 이름이다.

```
repositories=/d02/ias/Apache/Jserv/myservlets
```

마지막으로, 새로운 설정이 잘 이루어졌는지 샘플을 통해서 확인해 보자. 위에서 지정한 Repositories Directory에서 다음의 SimpleServlet을 컴파일한 후 브라우저에서 다음의 URL 을 호출해 보자.

```
http://.:7777/myservlet/JdbcServlet
```

```
import java.io.*;
import java.sql.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class JdbcServlet extends HttpServlet {
        public void doGet (HttpServletRequest request, HttpServletResponse
            response)throws ServletException, IOException
        {

            response.setContentType("text/html;charset=EUC-KR?;

            PrintWriter out = response.getWriter();
            try {

            DriverManager.registerDriver (new oracle.jdbc.driver.
                OracleDriver());
            Connection conn =

DriverManager.getConnection("jdbc:oracle:thin:@152.69.42.28
                    :1521:ora816","scott",tiger");

            Statement stmt = conn.createStatement();
            String sqlstr = "select * from dept";
            ResultSet rset = stmt.executeQuery(sqlstr) ;

            out.println("<H1> Servlet Test for JDBC</H1><BODY>");
            out.println("<H3> Data Queried from Dept Table</H3>");
            out.println("<table border = 0 width=90% cellpadding=2
```

```
                cellspacing=2 align=center >");

            while ( rset.next() )


out.println("<tr><td>"+rset.getString(1)+"</td><td>"+rset.getString
                (2)+"</td><td>"+rset.getString(3)+"</td></tr><BR>");

            out.println("</table></BODY>");
            out.close();
    }

            catch (SQLException e) {
                out.println("Oracle Error Code :" + e.getErrorCode());
            }
        }
}
```

JSP에서 Beans 사용

Oracle9*i*AS에서는 Oracle JSP를 이용해서 JSP를 사용할 수 있다. 우선, Oracle9*i*AS의 JSP 설정이 잘 되어 있는지 샘플을 통해서 확인해 보자. 디폴트 포트의 index.html 화면에서 제공되는 JSP 데모를 선택한 후 Basic 샘플 다음의 Hellouser를 실행한다.

http://<host_name>.<domain_name>:<port>/demo/basic/hellouser/hellouser.jsp

잘 수행된다면 기본적인 JSP를 위한 환경은 이루어진 상태라고 보면 된다. 만약, 원하는 결과가 나오지 않는다면, 먼저 Servlet 쪽 설정이 올바로 되어 있는지 확인한 후 다시 수행해 보자. 그럼, 이제는 간단한 Beans를 사용하는 JSP를 직접 작성해 보자.

<myBean.java>

```
public class myBean {
        private String name = "World ...?

        public void setName(String name) {
            this.name = name;
        }

        public String getName() {
            return name;
        }
}
```

<bean.jsp>

```
<%@ page contentType="text/html;charset=EUC-KR"%>
<jsp:useBean id="hello" class="myBean" scope="page"/>

<html>
<head><title>빈 연습</title></head>
<body>
<center><h3>useBean 테스트</h3></center>
```

한글 테스트〈br〉

다음은 useBean 테스트입니다〈br〉

```
<hr>

<%
        out.println("Hello " + hello.getName());
%<

<hr>
</body>
</html>

$ cd $ORACLE_HOME/Apache/Jserv/myservlets
```

JSP에서 사용되는 Beans의 Classpath를 설정하는 방법은 크게 두 가지가 있다.

첫 번째, $ORACLE_HOME/Apache/Jserv/etc/jserv.properties 파일에 wrapper.classpath를 사용하는 방법이다. Java 클래스들이 위치한 디렉토리나 jar 파일을 직접 지정하면 된다. 그러나, 이 방법은 Beans가 자주 수정되는 경우라면 Beans의 변경사항을 브라우저에서 확인하기 위해서 매번 JServ를 재로드해야만 한다. 다시 말해, 일반 JSP를 처리할 때와는 달리 JSP 안에서 Beans를 사용할 때에는 Oracle9*i*AS를 기동한 상태에서 Beans 파일들의 내용을 변경하고 컴파일해도 이는 브라우저의 기존 JSP 내용에 변경사항이 적용되지 않는다. 물론, 재로드해도 마찬가지이다.

두 번째, $ORACLE_HOME/Apache/jsp/conf/ojsp.conf 파일의 Alias가 지정된 디렉토리 아래에 WEB-INF/classes 디렉토리를 생성해서 클래스들을 위치시킬 수 있다. 이 방법을 사용하면 수정된 Beans 내용이 바로 적용된다. 디폴트로는 $ORACLE_HOME/Apache/Apache/htdocs 아래에 WEB-INF 디렉토리가 생성된다. 예를 들어, 다음과 같은 내용이 ojsp.conf에 포함되어 있다고 가정하면,

```
# Set aliases for Oracle JSP
Alias /jspdocs/ ?d02/ias102/jsp/doc/"
#
# Associate OJSP extension with OJSP class to be called
ApJServAction .jsp /servlets/oracle.jsp.JspServlet
ApJServAction .sqljsp /servlets/oracle.jsp.JspServlet
#
```

jspdocs의 피지컬 디렉토리 아래에 WEB-INF/classes 디렉토리를 생성하고 사용하는 Beans들을 위치시키면 브라우저에서 단지 재로드만으로 변경되는 Beans의 사항들을 확인할 수 있다. 브라우저에서 http://〈host_name〉.〈domain_name〉:〈port〉/jspdocs /bean.jsp를 수행한 후 myBean.java의 네임을 변경해서 Bean을 재컴파일해 보면 JServ의 재기동 없이 변경된 사항을 계속 확인할 수 있다.

가상 호스팅 설정

웹 서버에서 사용되는 가상 호스트(Virtual Host)는 하나의 서버를 마치 여러 개의 다른 이름을 가진 서버가 운영되는 것처럼 보이게 하는 방법이다. 물론, 웹 호스팅을 하는 경우에는 유용한 방법이라 할 수 있다. 이를 위해서 기존 OAS에서는 Node Manager라는 Administration Listener 화면에서 설정하고자 하는 리스너(Listener)에 새로운 호스트 이름이나 로그 파일 정보 등을 등록할 수 있었으나, Oracle9*i*AS에서는 일반적인 환경 설정들이 httpd.conf 파일에 등록되어 있으므로 가상 호스트도 이 파일에 설정해 주어야 한다.

가상 호스트 설정도 주소 기반의 가상 호스트(IP Address Based Virtual Host)와 이름 기반의 가상 호스트(Name Based Virtual Host) 등으로 구분할 수 있으나, 기본적으로 사용하고자 하는 OS상에서 여러 호스트 이름이 같은 IP 주소로 DNS에 설정되어 있다는 가정하에 이름 기반의 가상 호스트 설정 방법을 살펴보겠다. 이에 대한 확인 방법은 UNIX 프롬프트에서 /usr/sbin/nslookup 명령을 사용하면 알 수 있다.

```
ora816: /ora> nslookup o200
Server:  krsun3.kr.oracle.com  ---> DNS 서버의 정보
Address:      152.69.40.33

Name:   o200.kr.oracle.com    ---> 확인하고자 하는 호스트의 정보
Address:      152.69.40.75
```

예를 들어, www.hostname.com과 www.myhostname.com이 같은 IP 주소인 111.22.33.44로 설정되어 있다면, 다음과 같은 내용을 $ORACLE_HOME/Apache/Apache/conf/httpd.conf에 추가해 줌으로써 간단하게 이름 기반의 가상 호스트로 작동할 수 있다.

```
NameVirtualHost -> 111.22.33.44
위 부분은 Apache 1.3.x에서 새롭게 추가된 Directive이다.
<VirtualHost 111.22.33.44>
ServerName www.hostname.com
DocumentRoot /www/hostname
</VirtualHost>

<VirtualHost 111.22.33.44>
ServerName www.myhostname.com
DocumentRoot /www/myhostname
</VirtualHost>
```

물론, 기존에 사용하던 다른 Directive들도 위에서 지정한 가상 호스트 이름에서 사

용하고 싶다면 위의 VirtualHost Directive 안에 원하는 것들을 설정하면 된다.

예를 들어, www.myhostname.com에서 my-cgi라는 cgi 가상 디렉토리를 사용하면서 따로 로그 파일 등을 생성하려면 다음과 같이 설정하면 된다.

```
<VirtualHost 111.22.33.44>
ServerName www.myhostname.com
DocumentRoot /www/myhostname

TransferLog /www/myhostname/logs/access_log
ErrorLog /www/myhostname/logs/error_log
AgentLog /www/myhostname/logs/agent_log

ScriptAlias /my-cgi/ /www/myhostname/my-cgi/
</VirtualHost>
```

새로운 인스턴스(리스너)의 등록

설치 때 사용했던 포트 대신 다른 포트(일반적으로 디폴트 80 포트)로 서비스하고 싶다면, 다음의 작업에서 구성 파일인 httpd.conf에서 포트 번호만 바꾸어 서비스하면 된다. 하지만, 여러 개의 리스너를 만들어 각기 다른 포트와 다른 컴포넌트들을 설정하여 따로 관리하고자 한다면, 각각의 애플리케이션 및 업무에 맞게 따로 리스너를 만들어 관리하면 효율적이다. Oracle9*i*AS가 다음과 같은 환경으로 설치되어 있을 경우, 기존 인스턴스 외에 따로 다른 포트로 리스너를 생성하는 방법에 대해 알아보자. 아래의 예제는 Oracle9*i*AS 1.0.2 버전을 기준으로 작성되었다.

```
ORACLE_HOME=/ias
APACHE_HOME=/ias/Apache/Apache
```

$APACHE_HOME/conf에 새로 8080 포트를 이용하는 myApache이라는 이름의 리스너를 생성해보자.

1. 리스너의 디렉토리 생성

```
/ias/Apache/Apache/conf] mkdir myApache
/ias/Apache/Apache/conf] cd myApache
/ias/Apache/Apache/conf/myApache] mkdir logs
```

2. 리스너의 구성 파일 생성 및 수정

```
/ias/Apache/Apache/conf] cp httpd.conf ./myApache/myApache.conf
/ias/Apache/Apache/conf] cp $ORACLE_HOME/Apache/Jserv/etc/
        jserv.conf ./myApache/jserv.conf
/ias/Apache/Apache/conf] cp $ORACLE_HOME/Apache/Jserv/etc/
        jserv.properties ./myApache/jserv.properties
/ias/Apache/Apache/conf] cp $ORACLE_HOME/Apache/Jserv/etc/
        zone.properties ./myApache/zone.properties
/ias/Apache/Apache/conf] cp $APACHE_HOME/conf/oracle_apache.conf ./
        myApache/oracle_apache.conf
/ias/Apache/Apache/conf] cp $APACHE_HOME/conf/mod__ose.conf ./
        myApache/mod__ose.conf
```

/ias/Apache/Apache/conf] cp $ORACLE_HOME/Apache/modplsql/cfg/
plsql.conf ./myApache/plsql.conf
/ias/Apache/Apache/conf] cp $ORACLE_HOME/Apache/jsp/conf/
ojsp.conf ./myApache/ojsp.conf
/ias/Apache/Apache/conf] cp $ORACLE_HOME/xdk/admin/xml.conf ./
myApache/xml.conf
/ias/Apache/Apache/conf] cd myApache
/ias/Apache/Apache/conf/myApache] vi myApache.conf

2.1. PidFile 수정

PidFile /ias/Apache/Apache/conf/myApache/logs/httpd.pid

2.2. ScoreBoardFile 수정

ScoreBoardFile /ias/Apache/Apache/conf/myApache/logs/
httpd.scoreboard

2.3. ErrorLog 수정

ErrorLog /ias/Apache/Apache/conf/myApache/logs/httpd_error_log

2.4. CustomLog 수정

CustomLog /ias/Apache/Apache/conf/myApache/logs/httpd_access_log
common

2.5. include 수정

include?ias/Apache/Apache/conf/myApache/jserv.conf?
include?ias/Apache/Apache/conf/myApache/oracle_apache.conf?

2.6. Port 수정

Port 8080
Listen 8080

2.7. ServerRoot 수정

ServerRoot ?ias/Apache/Apache/conf/myApache?

2.8. LoadModule 부분의 상대 Path를 절대 Path로 변경 vi editor에서 다음을 수행한다.

:1,$s/libexec/\/ias\/Apache\/Apache\/libexec/g

3. jserv.conf의 수정

3.1. ApJServProperties

ApJServProperties /ias/Apache/Apache/conf/myApache/jserv.properties

3.2. ApJServLogFile

ApJServLogFile /ias/Apache/Apache/conf/myApache/logs/jserv.log

3.3. ApJServDefaultPort

ApJServDefaultPort 8010

4. jserv.properties

4.1. log.file

log.file=/ias/Apache/Apache/conf/myApache/logs/jserv.log

4.2. port

port=8010

4.3. repositories

root.properties=/ias/Apache/Apache/conf/myApache/zone.properties

5. oracle_apache.conf 수정

5.1. include 수정

include ?ias/Apache/Apache/conf/myApache/mod__ose.conf"
include ?ias/Apache/Apache/conf/myApache/plsql.conf"
include ?ias/Apache/Apache/conf/myApache/ojsp.conf"
include ?ias/Apache/Apache/conf/myApache/xml.conf"

6. mod_ose.conf 수정

LoadModule 부분의 상대 Path를 절대 Path로 변경한다.

LoadModule ose_module /ias/Apache/Apache/libexec/libjipa8i.so

7. start shell program 생성

```
/ias/Apache/Apache/conf/myApache] vi myApache_start
#!/bin/sh
ulimit -n 1024
ORACLE_HOME=/ias; export ORACLE_HOME
PERL5LIB=$ORACLE_HOME/Apache/perl/lib/5.00503:$ORACLE_HOME
/
        Apache/perl/lib/site_
perl/5.005; export PERL5LIB
LD_LIBRARY_PATH=$ORACLE_HOME/lib:$LD_LIBRARY_PATH;
        export LD_LIBRARY_PATH
WV_GATEWAY_CFG=$ORACLE_HOME/Apache/modplsql/cfg/
        wdbsvr.app ; export WV_GATEWAY_CFG
$ORACLE_HOME/Apache/Apache/bin/httpd -f $ORACLE_HOME/Apache/
        Apache/conf/myApache/myApache.conf


/ias/Apache/Apache/conf/myApache] chmod 755 myApache_start
```

8. listener start

```
/ias/Apache/Apache/conf/myApache] ./myApache_start
```

9. listener stop

```
/ias/Apache/Apache/conf/myApache/logs] kill 'at httpd.pid'
```

Basic Authentication 설정

Oracle9*i*AS의 HTTP 리스너 관리자는 인증된 사용자들이 서버의 일부분 또는 서버에 제한적으로 접근을 허용할 필요가 있다 그러면, 패스워드 파일을 사용하여 Basic Authentication을 어떻게 설정하는지에 대해 알아보자.

a. 유저네임과 패스워드를 패스워드 파일에 생성한다.
b. Oracle9*i*AS의 HTTP 리스너에서 사용자가 원하는 디렉토리에 대해 제한적으로 접근이 가능하도록 설정한다.

서버에서 패스워드 파일을 생성할 수 있도록 Oracle9*i*AS에서는 $ORACLE_HOME/Apache/Apache/bin/htpasswd라는 유틸리티를 제공한다.

〈〈 사용법 〉〉

```
$ htpassword
usage:
        htpasswd [-cmdps] passwordfile username
        htpasswd -b[cmdps] passwordfile username password


-c  Create a new file.
-m  Force MD5 encryption of the password.
-d  Force CRYPT encryption of the password (default).
-p  Do not encrypt the password (plaintext).
-s  Force SHA encryption of the password.
-b  Use the password from the command line rather than prompting for it.
On Windows and TPF systems the '-m' flag is used by default.
On all other systems, the '-p' flag will probably not work.
```

1. 새로운 패스워드 파일에 Username이 'demo'이고 Password가 'demo'인 유저를 생성해 본다.

```
$ htpasswd -c users.pw demo
        New password:
        Re-type new password:
        Adding password for user demo
```

그리고 패스워드를 입력한다

2. 이미 만들어진 users.pw 파일에 새로운 유저인 'guest'를 등록하는 경우

```
$ htpasswd users.pw guest
        New password:
        Re-type new password:
        Adding password for user guest
```

그리고 패스워드를 입력한다 패스워드 파일인 users.pw는 물리적인 디렉토리에 존재하므로 사용자들이 브라우저를 통해 패스워드 파일을 다운로드할 수 없게 어떠한 가상 디렉토리에도 매핑을 하지 않는다.

3. 어느 곳을 접근 제한할 것인지 결정한다.

http://krint-5.kr.oracle.com/internal/에 대해 Basic Authentication을 설정할 것이라고 가정한다.

4. $ORACLE_HOME/Apache/Apache/httpd.conf에 /internal/이라는 가상 디렉토리를 생성한다.

```
Alias /internal/ "/ias/internal/"
```

즉, 어떠한 클라이언트의 브라우저에서라도 http://krint-5.kr.oracle.com/internal/을 접근하게 되면 Oracle9*i*AS의 HTTP 리스너에서는 이와 같이 매핑되어 있는 물리적 디렉토리인 '/ias/internal/'를 액세스하게 된다.

5. 이곳에 대해 일반적인 접근을 제한하기 위하여 httpd.conf에 다음과 같은 Directive를 추가한다.

```
<Location /internal/>
AuthName ?he internal network?AuthType Basic
AuthUserFile ?u01/oracle/offline/user.pw?Require valid-user
Order Deny,Allow
Deny From All
Allow From .oracle.com
Satisfy Any
</Location>
```

AuthName은 사용자들이 URL을 통해 접근시 사용자가 로그인 다이얼로그에서 볼 수 있는 Authentication Realm의 이름이다 AuthType은 사용하고 있는 Authentication의 종류가 무엇인지를 보여주는 상태로 위의 경우는 Basic이다. AuthUserFile은 패스워드 파일이 위치한 Full Path를 기술한다.

Require valid-user는 사용자의 AuthUserFile에 접근이 허용되어 있다면 username/password가 반드시 존재하는지에 대한 상태이다. Order/deny/all directives는 kr.oracle.com domain내에 있는 호스트에 대해서 제한적으로 접근을 한다. Satisfy Any는 만일 사용자들이 kr.oracle.com domain 안에 있거나 또는 Basic Authentication의 요구를 만족한다면 접근이 허용된다는 것을 의미한다.
kr.oracle.com domain에 있는 사용자라면 즉시 로그인 다이얼로그가 나오지 않는다.

6. Basic Authentication의 테스트 변경된 내용을 반영하기 위해서 Oracle9*i*AS의 HTTP 리스너를 재시동한다.

클라이언트에서 http://ktint-5.kr.oracle.com/internal/과 같이 HTTP 리퀘스트를 하게 되면, 즉시 다음과 같은 로그인 다이얼로그가 나타난다.

```
"Enter username for the internal network at
 <machine>. <domain>. <suffix>: <port>"
```

또한 접근이 허용된 사용자에 대해 username과 password를 공지하여주는 것은 당연할 것이다.

한국오라클(주)
서울특별시 강남구 삼성동 144-17
삼화빌딩
대표전화 : 2194-8000
FAX : 2194-8001

한국오라클교육센타
서울특별시 영등포구 여의도동 28-1
전경련회관 5층, 7층
대표전화 : 3779-4242~4
FAX : 3779-4100~1

대전사무소
대전광역시 서구 둔산동 929번지
대전둔산사학연금회관 18층
대표전화 : (042)483-4131~2
FAX : (042)483-4133

대구사무소
대구광역시 동구 신천동 111번지
영남타워빌딩 9층
대표전화 : (053)741-4513~4
FAX : (053)741-4515

부산사무소
부산광역시 동구 초량동 1211~7
정암빌딩 8층
대표전화 : (051)465-9996
FAX : (051)465-9958

울산사무소
울산광역시 남구 달동 1319-15번지
정우빌딩 3층
대표전화 : (052)267-4262
FAX : (052)267-4267

광주사무소
광주광역시 서구 양동 60-37
금호생명빌딩 8층
대표전화 : (062)350-0131
FAX : (062)350-0130

고객에게 완전하고 효과적인
정보관리 솔루션을 제공하기 위하여
오라클사는 전 세계 145개국에서
제품, 기술지원, 교육 및
컨설팅 서비스를
제공하고 있습니다.

http://www.oracle.com/
http://www.oracle.com/kr