

Installera och konfigurera Resin v. 3.0.x

- * Expander taggen ejb-server (4.5)
- * Säkerhetsrisk med /servlet/.
- * DataSource i servlet

Innehållsförteckning

1	INSTALLERA RESINS MJUKVARA	3
2	KÖRA RESIN ”SOM ETT PROGRAM”	4
3	INSTALLERA RESIN SOM EN TJÄNST.....	5
3.1	Starta och stoppa Resin-tjänsten.....	5
3.2	Konfigurera uppstart av tjänst.....	5
4	KONFIGURERA WEBBAPPLIKATIONER	7
4.1	Lägga till webbapplikation i resin.conf.....	7
4.2	Skapa resterande mappar för webbapplikation	8
4.3	Innan vi konfigurerar med filen web.xml... ..	8
4.4	Konfigurera för att kunna testa servlets	9
4.5	Konfigurera databas för EJB.....	10
4.6	Använda Oracle 9i istället för MySQL	11
5	KONFIGURERA PROJEKT I JCREATOR [ATT GÖRA]	12
5.1	Required libraries	12
5.2	Sökväg för kompillerade gränssnitt och klasser.....	12
6	TILL ”SIST”	12
7	INSTALLERA APACHE2 I WINDOWS.....	13
7.1	Om IIS är installerad.....	14
7.2	Konfigurera Apache.....	15
8	INSTALLERA RESIN I APACHE 2.....	17
8.1	Installera Resin i Apache 2	17
9	INSTALLERA PHP 4.3 I APACHE 2	18

Om denna sammanfattning

Denna sammanfattning beskriver hur man konfigurerar Resin v. 3.0.x (Windows-version) från Caucho (www.caucho.com) för att EJB ska fungera med databaser. Sist av allt beskrivs även hur man installerar Apache 2.0.x (för att serva statiska webbsurser – återigen Windows-version) samt hur man installerar Resin (och PHP) i Apache. Inställningar (i Apache och Resin) har gjorts efter mina egna tycken, d.v.s. inte någon välkänd standard. T.ex. så är roten för IIS som standard i C:\Inetpub\wwwroot\ och därför brukar jag installera roten för Apache och Resin i C:\Inetpub\webapps\ (standard är C:\Program\Apache Group\Apache2\htdocs\ respektive %RESIN_HOME%\doc\¹). På så sätt samlar jag alla hemsidor, m.m. på ett ställe och gör det lättare att göra säkerhetskopior.

För att skapa denna beskrivning har jag, utöver Resin, använt (gratisprogrammet) Crimson Editor (www.crimsoneditor.com) för att skärmdumpar med filstruktur och kod (bl.a. XML i konfigurationsfiler). Crimson Editor använder jag även för att redigera konfigurationsfiler (XML-filer), deployment descriptors (för EJB), HTML-/JSP-sidor och andra textbaserade filer. Som databashanterare använder jag bl.a. MySQL (också den gratis – www.mysql.com) och Oracle 9i (www.oracle.com). För att administrera MySQL använder jag programmet MySQL Front (som numera kostar pengar från v.3.0 – försök få tag på v.2.5 som är gratis) och, givetvis (?), SQL*Plus för Oracle. Jag har också använt Paint Shop Pro (www.jasc.com) för skärmdumpar och Microsoft Word XP för layout (*desktop publishing*, DTP).

Jag är givetvis tacksam för alla konstruktiva synpunkter på sammanfattningens utformning och innehåll.

Eskilstuna, juni 2004

Björn Persson, e-post: bjorn.persson@mdh.se

Mälardalens högskola

Institutionen för ekonomi och informatik

Personlig hemsida: <http://www.eki.mdh.se/personal/bpn01/>

¹ %RESIN_HOME% är mappen som Resin installerats i – mer om detta senare.

1 Installera Resins mjukvara

Att installera Resin är inte så svårt. ☺ Ladda ner senaste versionen av Resin från Caucho – denna beskrivning gjordes för Resin-CMP v. 3.0.6 men bör fungera för senare versioner. Det finns idag (2004-02-09) två versioner av Resin:

- Resin – med stöd för bl.a. servlets och JSP.
- Resin-CMP² (eller Resin-Enterprise, Resin-ee) – innehåller samma som Resin ovan samt stöd för EJB 2.0.

Det är den senare versionen vi vill ha (d.v.s. ladda ner) för att använda EJB.

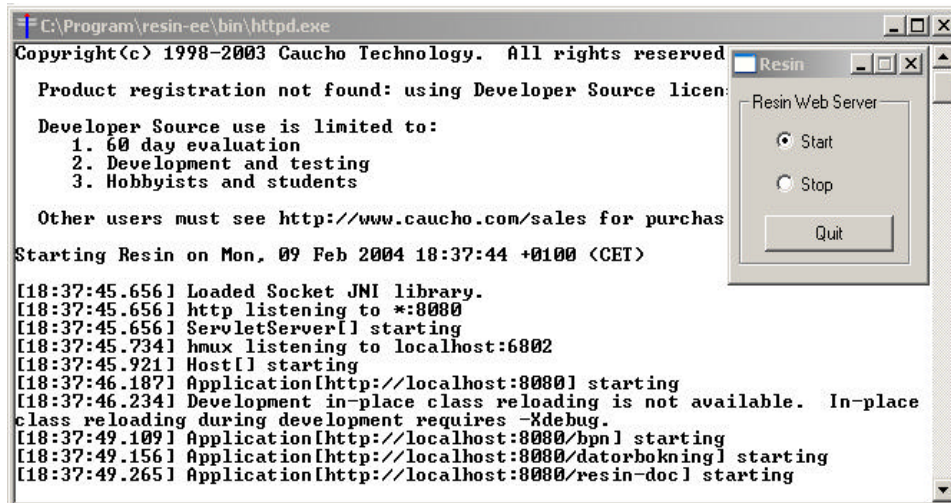
1. Packa upp ZIP-arkivet.
2. Kopiera mappen (den med mapparna `bin`, `cache`, `conf`, m.fl. i) till t.ex. `C:\Program`. Lämpligen döper vi om mappen till en aningen kortare sökväg, t.ex. `resin-ee` (d.v.s. så att sökvägen är `C:\Program\resin-ee\`). Denna mapp kommer refereras till som `%RESIN_HOME%` i fortsättningen.
3. Klart!

² Namnet försöker antagligen visa på att den stödjer Container-Managed Persistence (CMP).

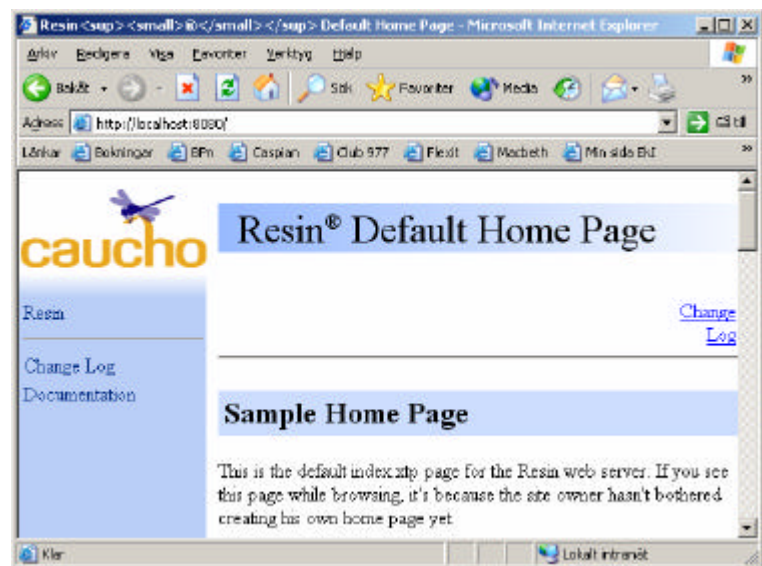
2 Köra Resin "som ett program"

Vi kan köra Resin som vilket annat program som helst – vilket vi även kan göra innan vi börjar installera Resin som en tjänst (om vi nu vill det ☺).

1. Kör Resin genom att köra filen HTTPD.EXE i mappen %RESIN_HOME%\bin. Detta öppnar två fönster: en kommandotolk med information om hur uppstart går samt ett enkelt formulär där vi kan starta/stoppa webbserver och stänga av Resin.



2. Vänta 1-5 minuter (beroende på hastighet på din dator) och öppna sen en webbläsare och surfa till `http://localhost:8080/`. Du bör se så startsidan med bl.a. länkar till dokumentationen (se bild till höger). Får du inte det så bör du vänta ett litet tag till och sen prova igen. Resin startar nämligen alla webbapplikationer som konfigurerats (desto fler desto längre tid tar det...☺).



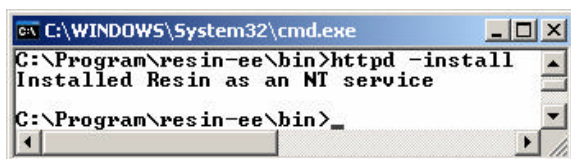
3. Klart!

Om du vill fortsätta köra Resin som ett program så kan du hoppa till avsnitt *Konfigurera webbapplikationer* längre ner. Fortsätt annars till nästa avsnitt för att installera Resin som en tjänst.

3 Installera Resin som en tjänst

Genom att installera Resin som en tjänst så kommer Resin startas varje gång din dator startas (eller när du själv väljer att starta den – se längre ner i detta avsnitt för mer information).

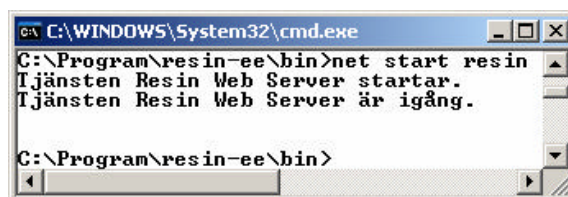
1. Stoppa Resin (om du inte redan gjort det) genom att klicka på knappen Quit i formulärfönstret.
2. Öppna en kommandotolk och gå till mappen %RESIN_HOME%\bin.
3. Skriv "httpd -install". Du bör få ett besked om att Resin nu installerats som en tjänst.



4. Klart!

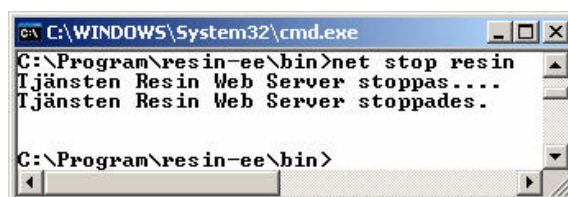
3.1 Starta och stoppa Resin-tjänsten

För att starta tjänsten skriver man "net start resin" i Kör-dialogen (välj Kör... från Start-menyn) eller skriv det i en kommandotolk (se bild nedan). Fördelen med det senare alternativet är att vi ser att tjänsten startas.



Som vi kan se av bilden ovan så ser vi inte när Resin laddat webbapplikationerna – vi får alltså prova genom att ladda startsidan för webbservern tills den visas.³

För att stoppa tjänsten skriver vi "net stop resin".



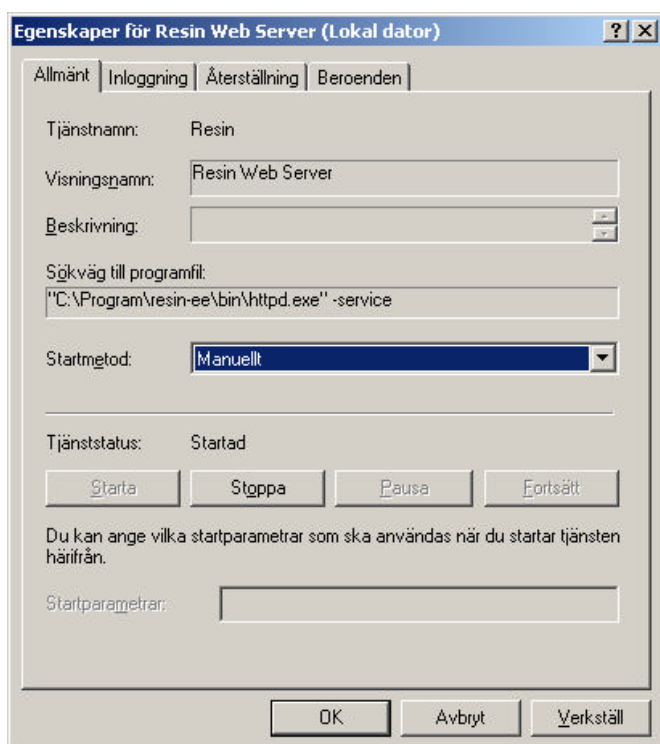
Du kan också använda Datorhantering (se hur man startar nedan) eller Tjänster i Kontrollpanelen för att starta och stoppa tjänsten. Tjänsten heter då *Resin Web Server*.

3.2 Konfigurera uppstart av tjänst

Desto fler tjänster vi startar när Windows startar, desto längre tid tar det att starta datorn. Och chansen att vi använder Resin dagligen är inte så stor...☺ Därför ska vi ställa om så att Resin inte startas vid start av Windows, d.v.s. vi måste starta den manuellt varje gång vi vill använda den.

³ Version 3 av Resin är **mycket** snabbare vad det gäller att ladda än vad version 2 var...☺

1. Högerklicka på ikonen Den här datorn på skrivbordet och välj Hantera från menyn som visas. Detta startar Datorhantering (eller vad som kallas Microsoft Management Console – MMC).
2. Expandera grenen Tjänster och program (antagligen längst ner i den vänstra delen av fönstret).
3. Markera grenen Tjänster. Detta visar alla tjänster som finns på datorn (de som är igång och inte).
4. Leta upp tjänsten som heter Resin Web Server⁴, högerklicka på den och välj Egenskaper från menyn som visas. Detta visar dialogrutan Egenskaper för Resin Web Server (se bild nedan).
5. Ändra listrutan Startmetod till Manuellt och klicka på knappen OK.
6. Klart!



Glöm inte att du måste starta tjänsten varje gång du vill använda den och att det kan ta 1-5 minuter innan den är igång!

⁴ Tjänsten i Datorhantering heter alltså inte riktigt samma som den tjänst som vi startar genom att skriva "net start resin" i kommandotolken. Så är det oftast... (t.ex. så heter IIS *World Wide Web Publishing* men startas i kommandotolken med "net start w3svc").

4 Konfigurera webbapplikationer

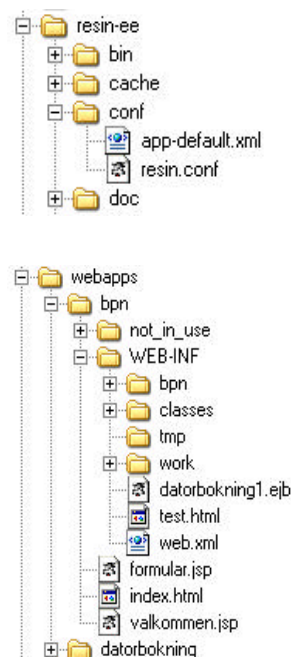
En webbapplikation är alla filer i en mapp samt dess undermapp (som inte ingår i en annan webbapplikation). Det som är speciellt med webbapplikationer är att de har sitt eget "minnesutrymme" (t.ex. för sessionshantering).

I Resin version 3 så styrs konfigurationer av webbapplikationer med tre filer:

- `resin.conf` – innehåller inställningar för webserver och vilka webbapplikationer som finns.
- `app-default.xml` – innehåller standardinställningar för webbapplikationer (bl.a. MIME-typer).
- `web.xml` – innehåller inställningar om en viss webbapplikation.

Det två första filerna (`resin.conf` och `app-default.xml`) finns i mappen `%RESIN_HOME%\conf` (se bild överst till höger) och finns endast en kopia av. Dessa filer redigerar vi en gång och "glömmer" sen. Den sista filen, `web.xml`, skapar vi en för varje webbapplikation vi vill ha och placeras i webbapplikationens mapp `WEB-INF` (se bild nederst till höger). Vissa konfigurationsalternativ kan placeras i både `resin.conf` och `web.xml`. Jag kommer dock lägga det mesta i `web.xml` (det som inte krävs i `resin.conf`).

På de flesta datorer jag jobbar med så har jag både IIS och Resin. Så för att "göra det lätt" för mig så brukar jag placeras mina webbapplikationer i Resin under mappen `C:\Inetpub\webapps` (i nedre bild ovan har jag två webbapplikationer – `bpn`, den expanderade, och `datorbokning`).



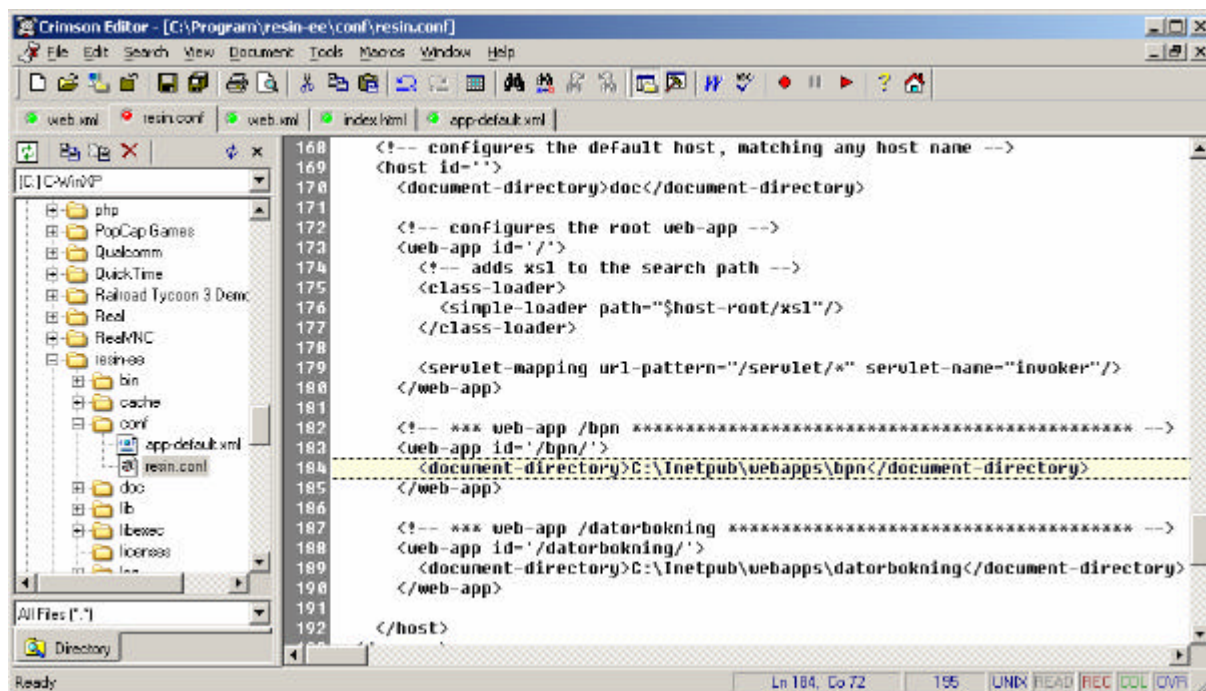
4.1 Lägga till webbapplikation i resin.conf

För att lägga till en webbapplikation så använder vi alltså filen `resin.conf`.

1. Öppna `resin.conf` (i mappen `%RESIN_HOME%\conf`) i t.ex. Crimson Editor (inte Internet Explorer! ☺).
2. Leta upp taggen `<host id=' '>` (nästan längst ner i filen).
3. Lägg till nedastående kod efter den avslutande `</web-app>` (ändra "bpn" i kod nedan till den sökväg du vill ha på din webserver – t.ex. din egen användaridentitet i högskolans nätverk).

```
<web-app id='/bpn/'>
  <document-directory>C:\Inetpub\webapps\bpn</document-directory>
</web-app>
```

I attributet `id` så anger vi sökvägen på webbservern och i taggen `document-directory` anger vi sökvägen i filsystemet till mapp med webbapplikation. Filen bör ha ett utseende som påminner om den i bilden nedan (här har jag konfigurerat **mina** två webbapplikationer `bpn` och `datorbokning` utöver den rotapplikation, d.v.s. `'/'`, som redan fanns).



4. Spara filen.
5. Klart!

Efter några sekunder (eller max någon minut ☺) så kommer Resin att skapa mappen som du angav i taggen `document-directory`. I denna mapp så har även Resin skapat en mapp `WEB-INF` (och eventuell fler undermappar) – ta **inte** bort denna (dessa) mapp(-ar). `WEB-INF` används för att lägga konfigurationsfilen `web.xml`, *deployment descriptors* (med filändelsen `.ejb`!) samt en mapp `classes` där vi kommer lägga klassfiler för servlets och EJB.

4.2 Skapa resterande mappar för webbapplikation

Resin skapar som sagt några mappar automatiskt. Men för att vi inte ska "glömma" det så skapar vi några egna mappar (som kan behövas senare i sammanfattning).

1. Gå till webbapplikationens mapp `WEB-INF`.
2. Skapa en mapp `classes`. Denna mapp kommer vi att lägga klassfilerna för våra servlets och EJB (se *Konfigurera projekt i JCreator* nedan för hur vi kan kompilera direkt till denna mapp).
3. Skapa en mapp `lib`. Denna mapp kan vi lägga JAR-filer i, t.ex. drivrutinerna för databas eller andra JAR-filer som vi behöver.

4.3 Innan vi konfigurerar med filen `web.xml`...

Observera att ett litet fel i filen `web.xml` (eller en *deployment descriptor*) för webbapplikation kommer innebära att webbapplikation inte "fungerar". Med fungerar menar jag att Resin inte kommer "tjäna" med att svara på webbläsares begäran om webbsidor, JSP eller servlets (eller EJB för den delen) som finns i webbapplikationen. Är ni flera som jobbar mot samma webbapplikation så kan ni alltså "förstöra" för varandra!

Skulle ni inte lösa ett fel i `web.xml` eller en *deployment descriptor* så bör ni alltså avlägsna koden som är orsaken till problemet (tills ni löst problemet eller försöker att lösa det igen)!

4.4 Konfigurera för att kunna testa servlets

Innan vi gör för mycket så är det alltid bra om man kan testa sina servlets. Vi vill alltså ha ett enkelt sätt att testa vilken servlet som helst. (När vi väl testat våra servlets så bör vi skapa "alias" för varje servlet – mer om detta i framtida versioner av denna sammanfattning hoppas jag ☺)

1. Skapa en fil (eller öppna existerande) med namnet `web.xml` i webbapplikationens mapp `WEB-INF`.
2. Lägg till nedanstående kod (du behöver **inte** lägga till de taggar som redan finns om filen existerar!).

```
<web-app>
  <!-- servlet-mapping for /servlet/* -->
  <servlet-mapping>
    <url-pattern id="/servlet/*" />
    <servlet-name id="invoker" />
  </servlet-mapping>
</web-app>
```

Genom att lägga till ovanstående kod så kan vi testköra våra servlets med URL:er så som `http://localhost/bpn/servlet/ServletKlass` (ersätt "bpn" med namnet på din webbapplikation och "ServletKlass" med namnet på din servlets klass – **utan** filändelsen `.class`!).⁵ Ovanstående kod i `web.xml` **bör** tas bort i en produktionsmiljö!

3. Spara filen.
4. Klart!

Som alltid när ni ändrar i `web.xml` (eller `resin.conf`) så kan det ta ett litet tag innan Resin läst ändringarna och laddat om webbapplikationen. Ha tålamod! ☺

4.4.1 Exempel med enkel servlet för test

Nedan visas koden för en enkel (HTTP-)servlet som skriver ut enklaste möjliga HTML-koden för att skriva ut texten Hello world! som en "hemsida".

```
package bpn;           //Namn på paket - bör vara unikt (t.ex. com.mdh.eki.bpn01)

import javax.servlet.*;           //Importerera paket för servlet
import javax.servlet.http.*;      //Importerera paket för HTTP-servlet
import java.io.*;                 //Importerera paket för I/O (PrintWriter)

public class EnkelServlet extends HttpServlet
{
    //Metod som anropas då webbläsare begär webbresurs med HTTP GET
    public void doGet(HttpServletRequest req, HttpServletResponse res)
        throws ServletException, IOException
    {
        PrintWriter out = res.getWriter();           //Hämta utström (out stream)

        out.print("<html>\n<body>\n");               //Skriv ut HTML-kod för hemsida
        out.print("<p>Hello world!\n");
        out.print("</body>\n</html>");

        out.close();                               //Stäng utström
    } //service()
}
```

⁵ Om servlet finns i ett paket måste klassens fullständiga namn användas, t.ex. `paket1.paket2.ServletKlass`.

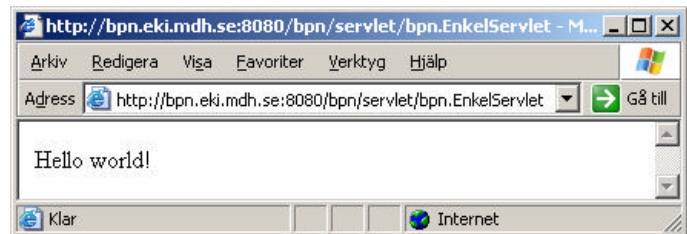
```
} //class EnkelServlet
```

För att kompilera servlet ovan så krävs tillgång till filen `J2EE.JAR` (d.v.s. att denna fil finns i miljövariabeln `CLASSPATH` – kan lösas genom att lägga till ett Required Library i JCreator, se kapitel *Konfigurera projekt i JCreator*).

Kompilera klassen och kopiera filen till webbapplikationens mapp `WEB-INF\classes`. För att testa servlet i webbläsare så använder vi en URL i stil med

`http://localhost:8080/bpn/servlet/bpn.EnkelServlet` (ersätt eventuellt `localhost` med adressen till webbserver – t.ex.

`kompis.eki.mdh.se`, `/bpn/` med namnet på din webbapplikation samt `bpn.EnkelServlet` med namnet på ditt paket respektive namn på servlet-klass).



4.5 Konfigurera databas för EJB

Om vi vill använda EJB så vill vi oftast använda Container-Managed Persistence (CMP). För detta behöver vi konfigurera en datakälla (oftast en relationsdatabas). I nedanstående exempel kommer jag använda databashanteraren MySQL och efteråt kommer jag förklara vad som behöver ändras för att använda Oracle 9i.⁶

Observera att databashanteraren **måste** vara tillgänglig! Om den **inte** är det så kommer du inte kunna surfa till någon webbsida, JSP eller servlet i webbapplikationen!

Observera också att man endast kan ha **en datakälla** per webbapplikation!

1. Öppna filen (eller skapa, om inte redan gjort, en ny fil med namnet) `web.xml` i webbapplikationens mapp `WEB-INF`.
2. Lägg till nedanstående kod (du behöver **inte** lägga till de taggar som redan finns om filen existerar!).

```
<web-app>

  <database>
    <jndi-name>jdbc/test</jndi-name>
    <driver>
      <type>org.gjt.mm.mysql.Driver</type>
      <url>jdbc:mysql://localhost:3306/test</url>
      <user>username</user>
      <password>password</password>
    </driver>
  </database>

  <ejb-server jndi-name="ejb">
    <config-directory>WEB-INF</config-directory>
    <data-source>jdbc/test</data-source>
  </ejb-server>
```

⁶ Varför jag använder MySQL för min beskrivning är för att det är en enkel databashanterare som är gratis, d.v.s. större chans att kunna installera och använda på egen dator. Tillsammans med verktyg som MySQL Front (använd version 2.5 – är gratis, version 3 kostar pengar) eller MySQL:s egna verktyg MySQL Control Center så är MySQL lättare att använda än t.ex. Oracle. **Observera** att MySQL (före version 5) "som standard" inte stödjer referensintegritet!

<web-app>

3. Spara filen.

4. Klart!

Ovan så talar vi om att vi skapar en datakälla (med taggen `database`) med namnet `test` (i JNDI-kontext `jdbc`). Datakällan använder följande taggar för inställningen:

- `driver` – information om datakälla (beskrivs av inneslutna taggar).
- `type` – drivrutinen som ska användas. Är det MySQL så måste ni ladda ner deras Java Connector, packa upp ZIP-filen och leta upp JAR-filen (med ett långt namn...☺). Placera denna JAR-fil i mappen `lib` under webbapplikationens mapp `WEB-INF`.⁷
- `url` – adress till databashanterare och databas. I exempel ovan så används en lokalt installerad MySQL-databas med standardport och databasen `test`.
- `user` – användaridentitet som ska användas för att logga på databas.
- `password` – lösenord till användaridentitet ovan.

Sen talar vi om för eventuella EJB:s som ska använda CMP att de ska använda datakällan i taggen `database` genom att lägga till taggen `ejb-server`. (Som standard så är JNDI-namnet `"cmp"` – här har jag dock kallat den för `"ejb"` för att passa mina exempel. ☺) Det viktiga är att taggen `data-source` matchar `jndi-name` i `database`-taggen!

Som alltid när ni ändrar i `web.xml` (eller `resin.conf`) så kan det ta ett litet tag innan Resin läst ändringarna och laddat om webbapplikationen. Ha tålamod! ☺

4.6 Använda Oracle 9i istället för MySQL

För att använda Oracle istället för MySQL så behöver vi ändra på två saker i taggen `database` (markerat med fet stil) – värdet på taggen `type` samt värdet på taggen `url`. Precis som med MySQL så behöver vi JAR-filen med drivrutinerna – dom finns⁸ någonstans under mappen där Oracle installerats⁹ och heter `ojdbc14.jar`. Filen kan placeras i mappen `lib` under webbapplikationens mapp `WEB-INF`.

```
<database>
  <jndi-name>jdbc/test</jndi-name>
  <driver>
    <type>oracle.jdbc.pool.OracleConnectionPoolDataSource</type>
    <url>jdbc:oracle:thin:@julia:1521:orcl</url>
    <user>username</user>
    <password>password</password>
  </driver>
</database>
```

⁷ Istället för att lägga filen i `WEB-INF\lib` för varje webbapplikation så kan denna placeras i mappen `%RESIN_HOME%\lib\local` (mappen `local` måste skapas) – då är JAR-filen, d.v.s. drivrutinerna, tillgänglig för alla webbapplikationer.

⁸ Vi (jag och andra kollegor) har märkt att den inte alltid finns med när man bara installerar Oracle-klienten...

⁹ På Ekonomihögskolan så har Oracle installerats till `C:\Program\Oracle`.

5 Konfigurera projekt i JCreator [ATT GÖRA]

OK, JCreator är inte den bästa utvecklingsmiljön, men den är gratis och den är enklare än många andra gratis-versioner (bl.a. så är den inte så seg som andra miljöer ☺). Och är vi beredda att betala (ett antal dollar mindre än kommersiella verktyg) så får vi i alla fall en av dom mest intressanta funktionerna: "intellisense"¹⁰ – d.v.s. att lista med möjliga attribut och metoder visas när vi t.ex. sätter en punkt efter en variabel med objekt i.

Nåja, om vi nu ska använda JCreator så är det en sak vi måste (eller bör) göra – lägga till *required libraries* för gränssnitt och klasser i J2EE och Resins klasser – samt en sak som underlättar test av servlets/EJB – konfigurera sökväg att kompilera gränssnitt och klasser till. Sökväg för kompilerade gränssnitt och klasser gör att vi kan kompilera direkt till mappstruktur i webbapplikation, d.v.s. vi slipper kopiera filer hela tiden. Denna beskrivning utgår från JCreator version 3.xx¹¹.

5.1 *Required libraries*

5.1.1 Skapa required libraries

Innan vi kan skapa ett required library måste vi skapa ett nytt projekt (och därmed workspace).

5.1.2 Använda required libraries

För att vi ska kunna använda biblioteken

5.2 Sökväg för kompilerade gränssnitt och klasser

6 Till "sist"...

Vi kan även installera Resin så att den samarbetar med Apache (eller IIS). Apache (eller andra webbservrar) är med största sannolikhet mer effektiva på att hantera statiska webbresurser är Resin – huvudsyftet med Resin är att exekvera servlets/JSP. Vi slipper även skriva numret på porten (d.v.s. 8080) hela tiden (vilket bl.a. underlättar användandet av relativa URL:er i webbapplikationer). Resten av denna sammanfattning behandlar hur man installerar Apache 2 och konfigurerar så att den samarbetar med Resin.

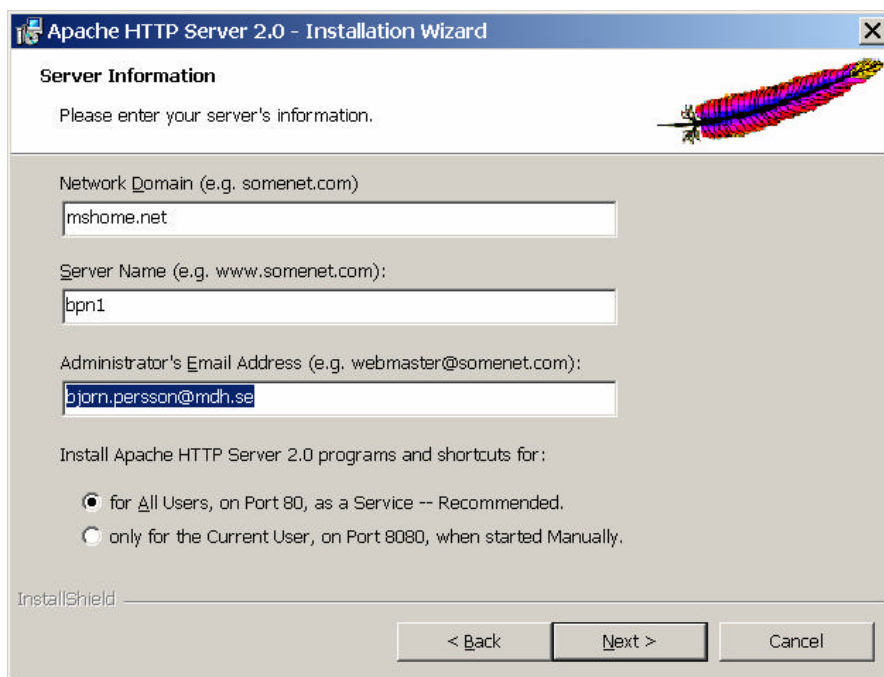
På Ekonomihögskolan har servern Kompis (för komponentkurserna) inte konfigurerats så – på Kompis används (i skrivande stund) Resin för att serva även statiska resurser.

¹⁰ Intellisense kan vara ett Microsoft-begrepp, men jag känner inte till något annat ord för detta annat än det närliggande *autocompletion*. ☺

¹¹ I skrivande stund finns version 3.10 ute.

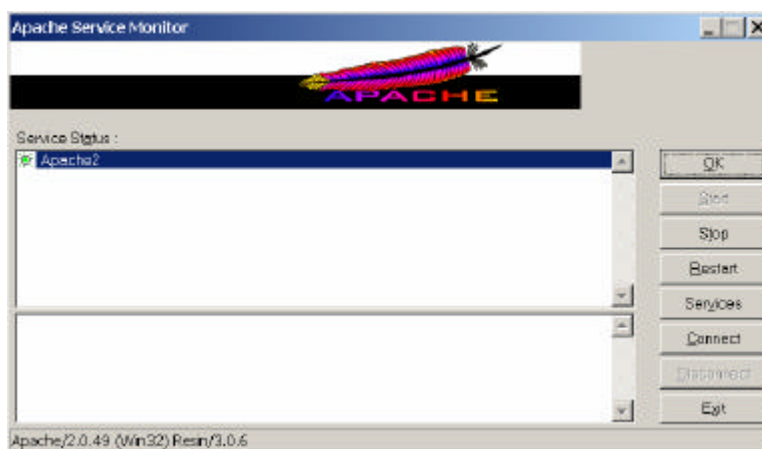
7 Installera Apache2 i Windows

1. Ladda ner version med Windows-installation (.MSI-fil).
2. Dubbelklicka på MSI-filen för att starta installationsprogrammet (detta kan kräva att du laddar ner Windows Installation 2.0 från Microsoft). Klicka på Nästa>/Next>.
3. Godkänn licensavtal (genom att klicka på radioknappen I accept...) och klicka på Nästa>/Next>.
4. Klicka på Nästa>/Next> (i skärmen Read This First).
5. Fyll i nätverksdomän (d.v.s. DNS-domän, om någon), datoradress (inkl. nätverksdomän om någon – annars fyll bara i namnet på datorn, i mitt fall bpn1), din e-postadress (ej obligatorisk ☺) samt radioknappen for All Users...¹². Klicka på Nästa>/Next>.



6. Välj Typical som Setup Type och klicka på Nästa>/Next>.
7. Klicka på Nästa>/Next> för att installera under C:\Program\Apache Group\Apache2.
8. Klicka på Install för att börja installation (d.v.s. börja kopiera filer).
9. Klicka på Finish för att avsluta installation.

Installationsprogrammet startar automatiskt Apache Service Monitor – ett program som kan användas för att stoppa och starta Apache2-tjänsten. Om



¹² Detta installerar Apache som en tjänst i Windows (d.v.s. den startar, eller kan starta, med datorn). Om du har IIS installerat så kommer installationen att misslyckas då den är installerad på port 80 (läs mer under Om IIS är installerad nedan).

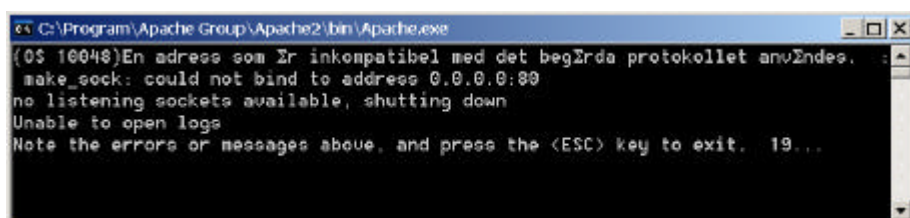
programmet inte ska startas med datorn så ta bort den i Autostart (den kan startas manuellt under Apache HTTP Server-programgruppen).

För att testa installationen så öppnar man URL:en `http://<datornamn>/` – detta bör visa Apaches startsida (med meddelande om att installation lyckades). På hemsidan finns bl.a. en länk till dokumentationen för Apache. Dokumentationen kan även nås på adressen `http://<datornamn>/manual/` vid ett senare tillfälle.

7.1 Om IIS är installerad

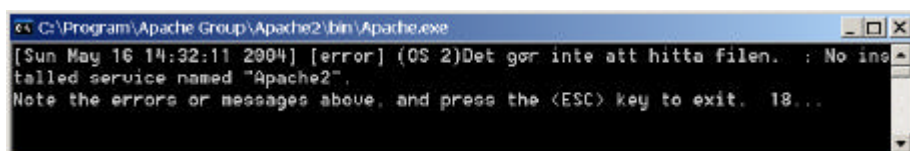
Om IIS är installerad på datorn (vilket den är på alla ”mina”¹³ datorer ☺) så använder den redan port 80. Därför kommer installationsprogrammet för Apache att misslyckas när vi försöker installera Apache som en tjänst.

Hur vet man att installationsprogrammet misslyckas? Genom att en kommandotolk med nedanstående (eller liknande) felmeddelande visas. Detta fel uppstår då IIS använder port 80.



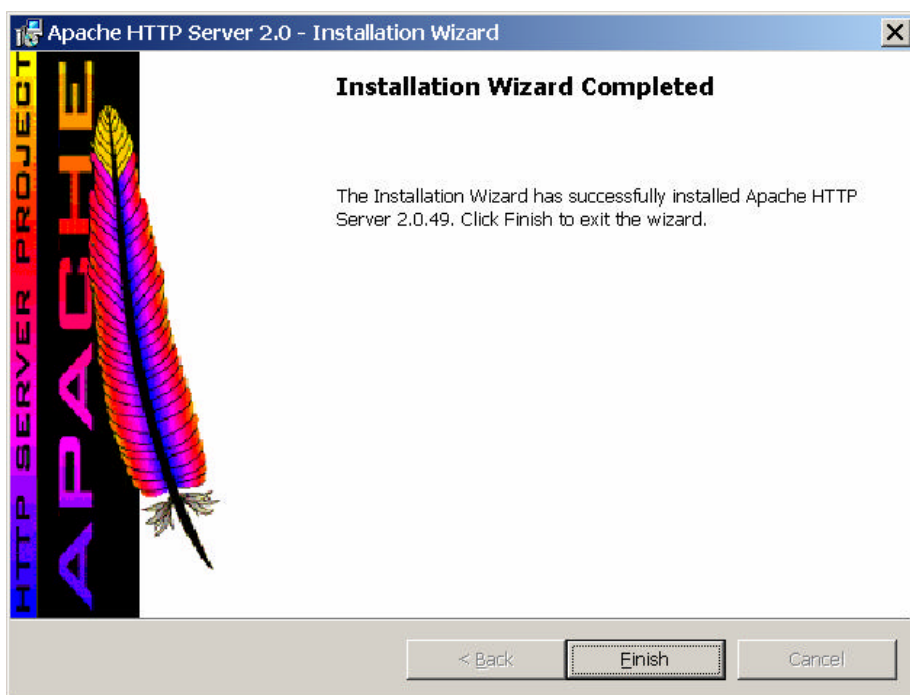
```
C:\Program\Apache Group\Apache2\bin\Apache.exe
(OS 10048)En adress som är inkompatibel med det begärda protokollet användes.
make_sock: could not bind to address 0.0.0.0:80
no listening sockets available, shutting down
Unable to open logs
Note the errors or messages above, and press the <ESC> key to exit. 19...
```

Eftersom det inte gick att starta Apache så gick det inte att installera Apache som en tjänst i datorn, vilket leder till att även nedanstående felmeddelande visas.



```
C:\Program\Apache Group\Apache2\bin\Apache.exe
[Sun May 16 14:32:11 2004] [error] (OS 2)Det går inte att hitta filen. : No in-
stalled service named "Apache2".
Note the errors or messages above, and press the <ESC> key to exit. 18...
```

Intressant nog så kommer ändå installationsprogrammet att visa att installationen gick bra...



¹³ D.v.s. alla datorer som jag installerat och använder för webbutveckling. ☺

För att fixa detta så gör följande **efter** att installationsprogrammet har körts första gången.

1. Öppna filen `C:\Program\Apache Group\Apache2\conf\httpd.conf` (om Apache installerats på standardplatsen) i t.ex. Anteckningar.
2. Sök upp raden `"Listen 80"` och ersätt med `"Listen 8080"`.
3. Sök sen upp raden `"ServerName <ditt datornamn>:80"` och ersätt med `"ServerName <ditt datornamn>:8080"` (ersätt `<ditt datornamn>` med din dators namn, bpn1 i mitt fall).
4. Spara och stäng filen `httpd.conf`.
5. Kör installationsprogrammet igen genom att dubbelklicka på MSI-filen. Klicka på Nästa>/Next>.
6. Välj radioknappen Repair och klicka på Nästa>/Next>.
7. Klicka på Install för att börja installation (d.v.s. kopiera filer igen).
8. Under installationen så kan felet till höger visas. Detta är inte ett allvarligt fel. ☺ Stäng dialogrutan genom att klicka på OK.
9. Klicka på Finish för att avsluta installationen.



Om installationen gick bra, och programmet Apache Service Monitor inte avslutats, så ska en ikon med grön pil visas i *System Tray* nere till höger (se bild till höger). (Det rödaktiga i ikonen är Apache-fjädern. ☺)



För att testa installationen så öppnar man URL:en `http://<datornamn>:8080/` – detta bör visa Apaches startsida (med meddelande om att installation lyckades). På hemsidan finns bl.a. en länk till dokumentationen för Apache. Dokumentationen kan även nås på adressen `http://<datornamn>:8080/manual/` vid ett senare tillfälle.

Skälet till att andra installationen av Apache fungerar beror på att vi ändrat på filen `httpd.conf` (eller rent utav att den redan existerar) och att installationsprogrammet därför inte kommer att ersätta denna fil.

7.2 Konfigurera Apache

Nedanstående konfigurationer är (oftast) inte nödvändiga, men är (oftast) praktiska att göra.

- Mapp som ska innehålla roten för webbserver.
- Ändra filer som laddas som standard om ingen fil anges i URL
- Ändra port som Apache svarar på (se avsnitt *Om IIS är installerad* ovan).

7.2.1 Mapp som ska innehålla roten för webbserver

Genom att flytta roten för webbserver så kan vi placera den på ett ställe som gör det "lättare" att göra säkerhetskopior (t.ex. om dator installeras om). Med lättare menar jag att alla webbapplikationer (i IIS, Resin och Apache) placeras på ett och samma ställe. Av detta skäl brukar jag placera Resins och Apaches rötter¹⁴ i `C:\Inetpub\webapps`.

1. Öppna filen `C:\Program\Apache Group\Apache2\conf\httpd.conf`.
2. Leta upp raden `"DocumentRoot "C:/Program/ Apache Group/ Apache2/htdocs""` ändra till `"DocumentRoot "C:/Inetpub/webapps""`.

¹⁴ Måste erkänna att jag ville skriva "rotar" (*rots*) här...☺

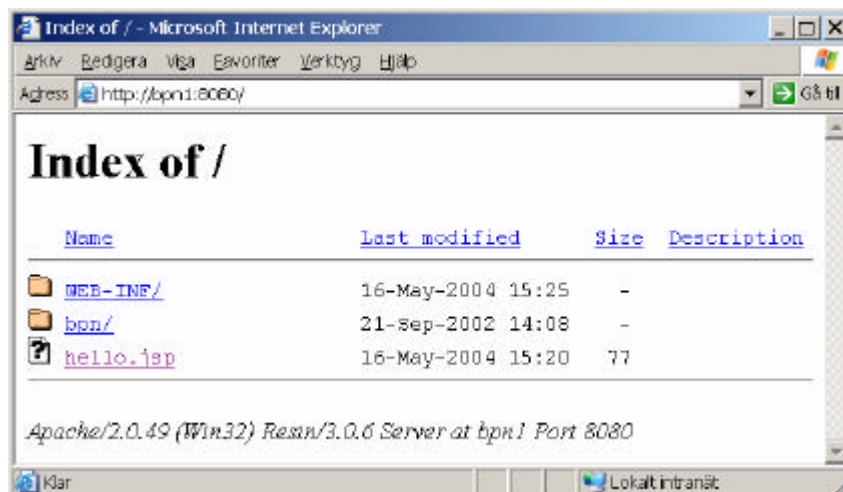
3. Leta även upp raden "<Directory "C:/Program/Apache Group/Apache2/htdocs">" (och alla andra rader som "pekar" på mappen htdocs) och ersätt med "<Directory "C:/Inetpub/webapps">" (respektive sökväg till mappen webapps).
4. Spara och stäng filen httpd.conf.
5. Stoppa Apache och starta igen.

Testa Apache genom att öppna URL:en `http://<datornamn>:8080/` – startsidan för Apache bör inte visas. Istället bör något i stil med det i bilden till höger (vilka mappar och filer beror på vilka mappar och filer som finns i

`C:\Inetpub\webapps\`.
Lägg en HTML-fil (t.ex. `hello.html`) i

`C:\Inetpub\webapps\`
och testa att öppna URL:en

`http://<datornamn>:8080/hello.html` (eller vad filen kallades).



7.2.2 Ändra filer som laddas som standard om ingen fil anges i URL

Om vi öppnar en URL utan filnamn (d.v.s. som avslutas med ett snedstreck "/") på en webbserver så laddas som standard en fil med ett visst namn (om den existerar i vald mapp). På de flesta webbservrar heter filen `index.html` (eller `.htm`).¹⁵ Om vi ska använda JSP så är det praktiskt om vi även lägger till `index.jsp` (och `index.php` om vi även ska använda PHP). En fördel med detta är att vi kan använda URL:er så som `"/` för att ladda t.ex. huvudmenyer (i t.ex. `index.html`) i webbapplikation (eller någon av dess undermappar).

1. Öppna filen `C:\Program\Apache Group\Apache2\conf\httpd.conf`.
2. Leta upp raden `DirectoryIndex index.html index.html.var` (eller liknande) ändra till `DirectoryIndex index.html index.html.var index.jsp index.php` (om inte PHP ska installeras så kan sista filnamnet tas bort).¹⁶
3. Spara och stäng filen `httpd.conf`.
4. Stoppa Apache och starta igen.

¹⁵ Givetvis heter den något annat i IIS – `default.htm`, ett namn som på sitt sätt är mer givande än `index.htm`. ☺

¹⁶ `index.html.var` är Apaches speciella format som gör det möjligt att ha olika filer för olika språk – inget vi behöver bry oss om om vi ska använda servlets/JSP och/eller EJB.

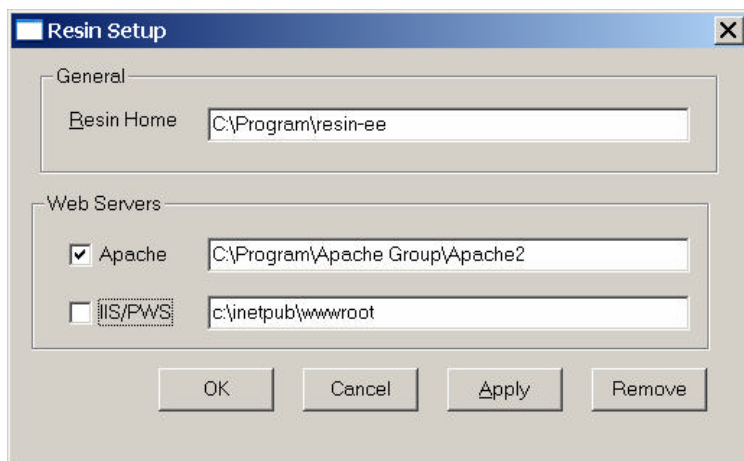
8 Installera Resin i Apache 2

Ladda ner Resin och packa upp ZIP-filen enligt instruktioner ovan. Innan Resin installeras så stoppas lämpligen tjänsten för Apache så att det inte uppstår konflikter mellan Resin och Apache.

8.1 Installera Resin i Apache 2

Innan vi installerar Resin i Apache så måste vi ändra port som Resin svarar på.¹⁷

1. Öppna filen %RESIN_HOME%\conf\resin.conf.
2. Leta upp raden "<http id="" host="" port="8080"/>" och ändra porten (d.v.s. 8080) till t.ex. 8008.
3. Kör programmet %RESIN_HOME%\bin\setup.exe. Kontrollera att sökvägen i textrutan Resin Home stämmer. Markera kryssrutan Apache samt fyll i sökvägen till installationsmappen för Apache (C:\Program\Apache Group\Apache2 om Apache installerades i standardmappen). Avmarkera kryssrutan IIS/PWS (om förbockad). Klicka sen på Apply för att spara inställning och klicka sen på OK för att stänga dialogruta.¹⁸



4. Leta upp raden "<document-directory>doc</document-directory>" och ersätt "doc" med "C:\Inetpub\webapps" (eller den mapp som du valt som rot för Apache).
5. Stoppa tjänsterna Resin och Apache2 (om de är igång). Starta tjänsterna (igen om de var igång tidigare ☺).

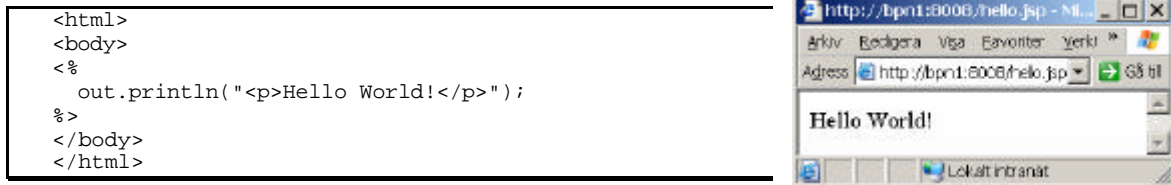
För att testa Apache öppnar man URL:en
http://<datornamn>:8080/ och Resin
http://<datornamn>:8008/ (ersätt
<datornamn> med namnet på din dator – bpn1 i
mitt fall). Om fel i bild till höger fås när Resin
testas kan en fil hello.jsp med nedanstående
kod skapas och placeras i C:\Inetpub\webapps\



¹⁷ Detta är endast nödvändigt om Apache installerats på port 8080.

¹⁸ Jag har märkt att det **inte** går att bara klicka på OK för att spara inställningen. Men detta kan ha ändrats eller har ändrats med senare versioner av Resin.

samt URL ändras till `http://<datornamn>:8008/hello.jsp`.



Resins program `setup.exe` lägger till raden `"LoadModule caucho_module
"C:/Program/resin-ee/libexec/apache-2.0/mod_caucho.dll"` i Apaches konfigurationsfil (`httpd.conf`).

8.1.1 Om Resin inte fungerar

Om du får problem med att Resin inte fungerar (d.v.s. servlets eller JSP inte fungerar) när du installerat den i Apache, prova att göra följande.

1. Stoppa tjänsterna för Resin och Apache.
2. Öppna en kommandotolk och byt till mappen `%RESIN_HOME%\bin`.
3. Skriv `"httpd -remove"` (för att ta bort Resin-tjänsten).
4. Skriv `"httpd -install"` (för att installera Resin-tjänsten igen – observera att vi måste ändra startmetod till Manuellt igen om vi inte vill att tjänsten ska starta med datorn – se beskrivning ovan).
5. Starta tjänsterna för Resin och Apache igen.

Prova även universal lösningen om inget annat verkar fungera – starta om datorn. ☺

9 Installera PHP 4.3 i Apache 2

PHP är ofta förknippad med Apache och MySQL (samt Linux som LAMP – Linux, Apache, MySQL och PHP). Därför beskrivs även hur man installerar PHP i Apache (och för att jag brukar variera mellan Java och PHP för utveckling av webbapplikationer ☺).