Quick Start Guide – Time Machine AIX for WebSphere-ND

The following is a quick guide to evaluating Solution-Soft's Time Machine with IBM's WebSphere-ND (Network Deploy)   SOA Application Server. The Network-Deploy version behaves a bit differently than stand-alone WebSphere application servers and thus requires a different approach when used with Time Machine; specifically the default is that all instances of applications servers run in the context of root. While this can be changed in the administration console to some extent, some environments prefer to keep the context in the default mode. This paper will show how simple it is to integrate Time Machine to perform date & time (temporal) testing with WebSphere-ND running as 'root'.

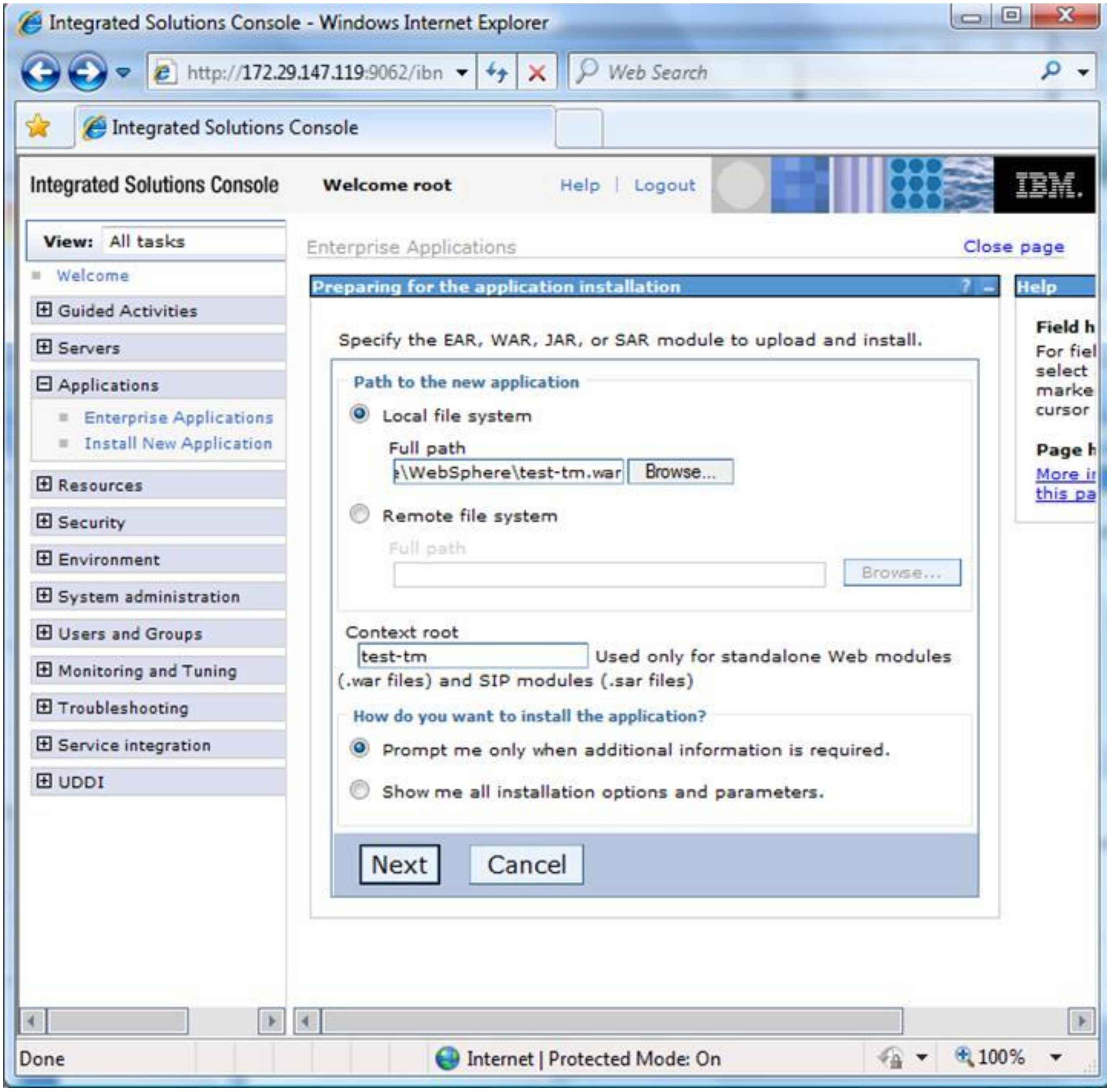
NOTE: This paper is specific to Time Machine for use with IBM UNIX AIX. For those running WebSphere-ND on other platforms please refer to the Time Machine manual or contact Solution-Soft support atsupport@solution-soft.com

Time Machine for WebSphere is supported on Windows, AIX, HP-UX, Linux, and Solaris

System Configuration used in creating this paper:
OS: AIX 5.3 Power 6, 2 CPUs, 4096 MB ram.
Time Machine: Version 4.01 R48 for IBM AIX 5.3.0. Build Date: 10/27/10
IBM WebSphere: ND, 6.1.0.0 Build Number: b0837.02 Build Date: 9/15/08

1) Logon to WebSphere Solutions Console as the Administrator. In this case WebSphere-ND is running on

http://172.29.147.119:9062/ibm/console/login.do



Specify the EAR, WAR, JAR, or SAR module to upload and install.

Path to the new application

[✓ Local file system]

Full path: \WebSphere\test-tm.war

Remote file system

Full path:

Context root: test-tm — Used only for standalone Web modules (.war files) and SIP modules (.sar files)

How do you want to install the application?

[✓ Prompt me only when additional information is required.]

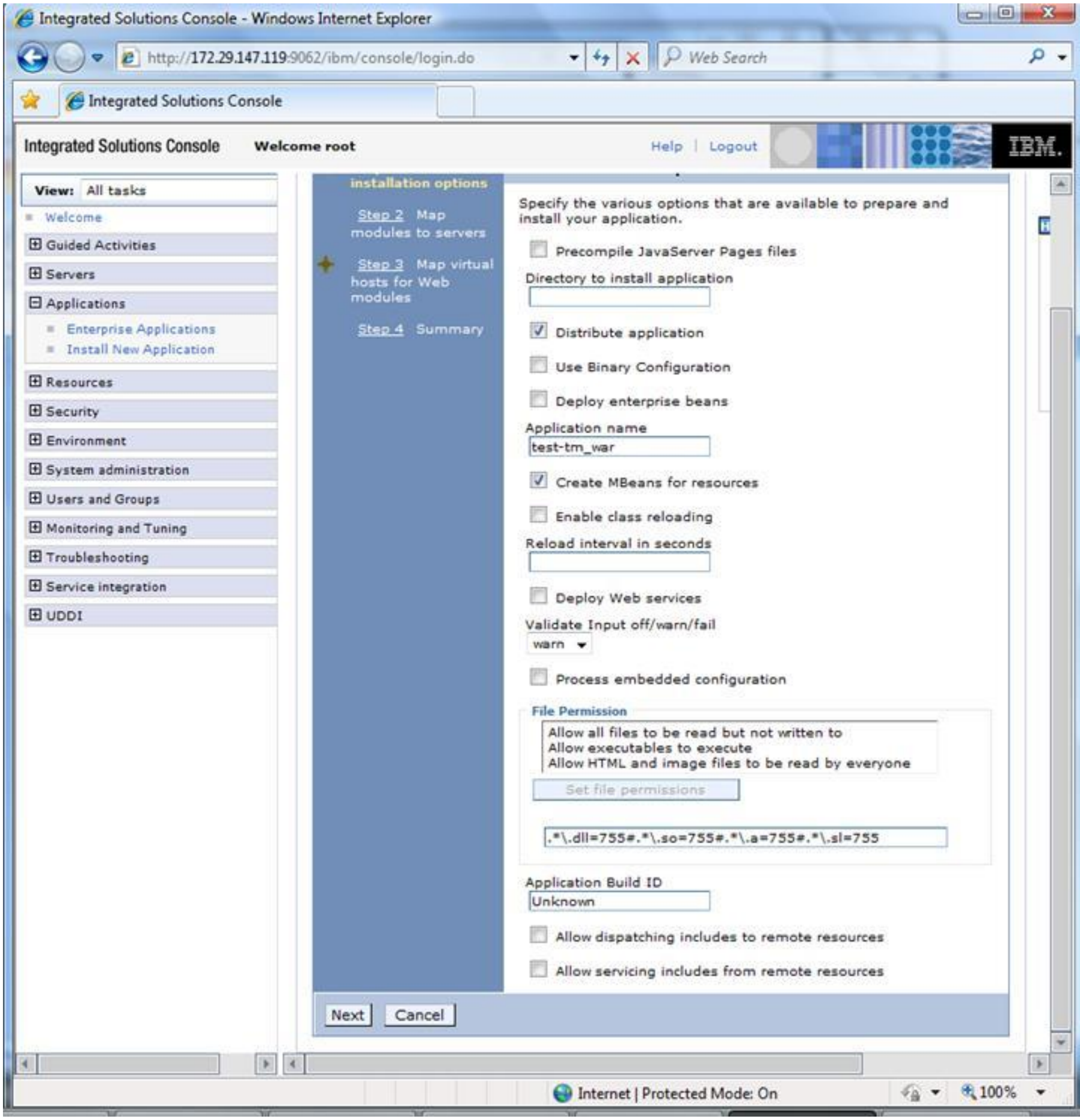
Show me all installation options and parameters.

Expand "Application" Then click on Install "New Application"

In the center pane click on "Local file system" then browse to your test WAR file.
NOTE: Solution-Soft has a sample WAR file located here: ftp://ftp.solution-soft.com/pub/tm/aix/530/war/test.war.
The source code used in this example is also available at the end of this document

In Context-root enter a name for the application. We'll use "test-tm".
Click [next]

2) On the next page (see image below) leave all the default setting as-is and click [Next]



Specify the various options that are available to prepare and install your application.

- [ ] Precompile JavaServer Pages files

Directory to install application:

- [✓] Distribute application
- [ ] Use Binary Configuration
- [ ] Deploy enterprise beans

Application name: test-tm_war

- [✓] Create MBeans for resources
- [ ] Enable class reloading

Reload interval in seconds:

- [ ] Deploy Web services

Validate Input off/warn/fail: warn

- [ ] Process embedded configuration

File Permission

Allow all files to be read but not written to
Allow executables to execute
Allow HTML and image files to be read by everyone

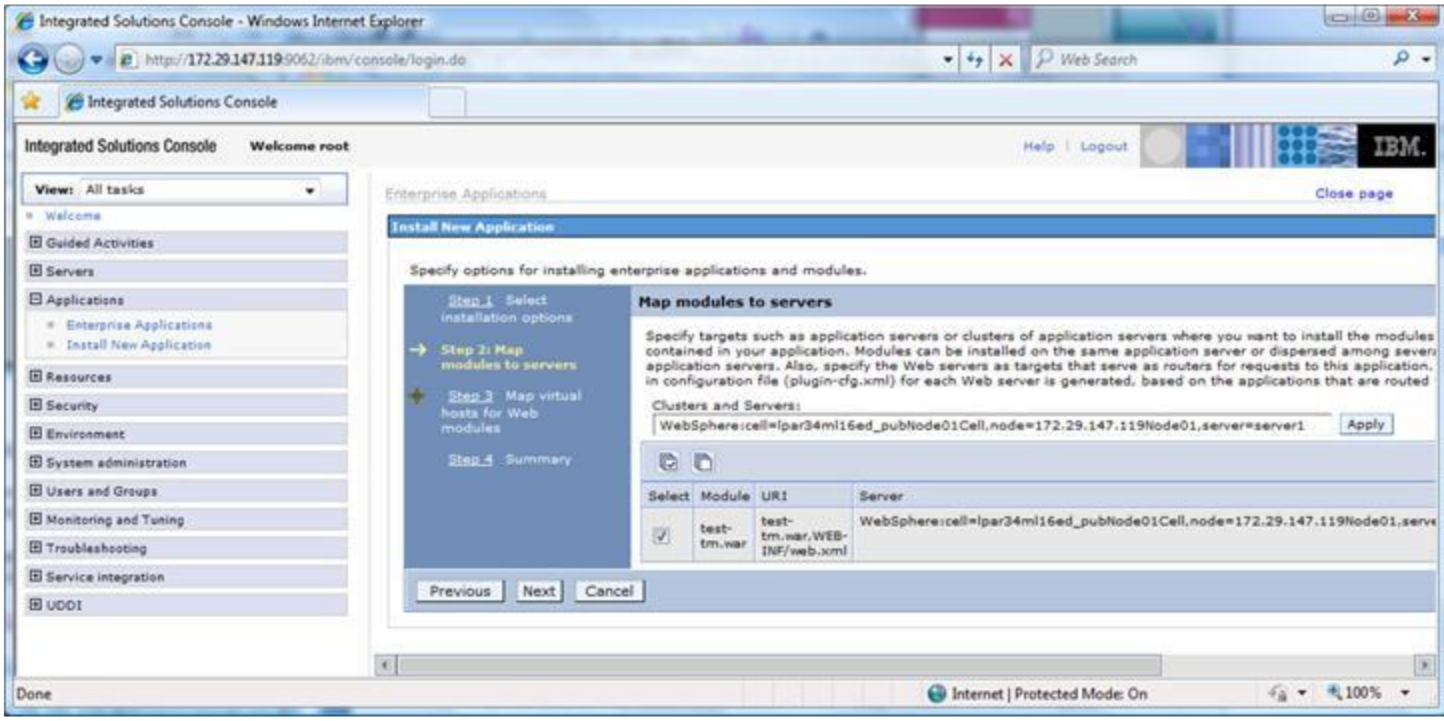
Set file permissions

.*\.dll=755#.*\.so=755#.*\.a=755#.*\.sl=755

Application Build ID: Unknown

- [ ] Allow dispatching includes to remote resources
- [ ] Allow servicing includes from remote resources

3) Now click on the select radio button in the “Map modules to servers” pane and click [Next]



Specify options for installing enterprise applications and modules.

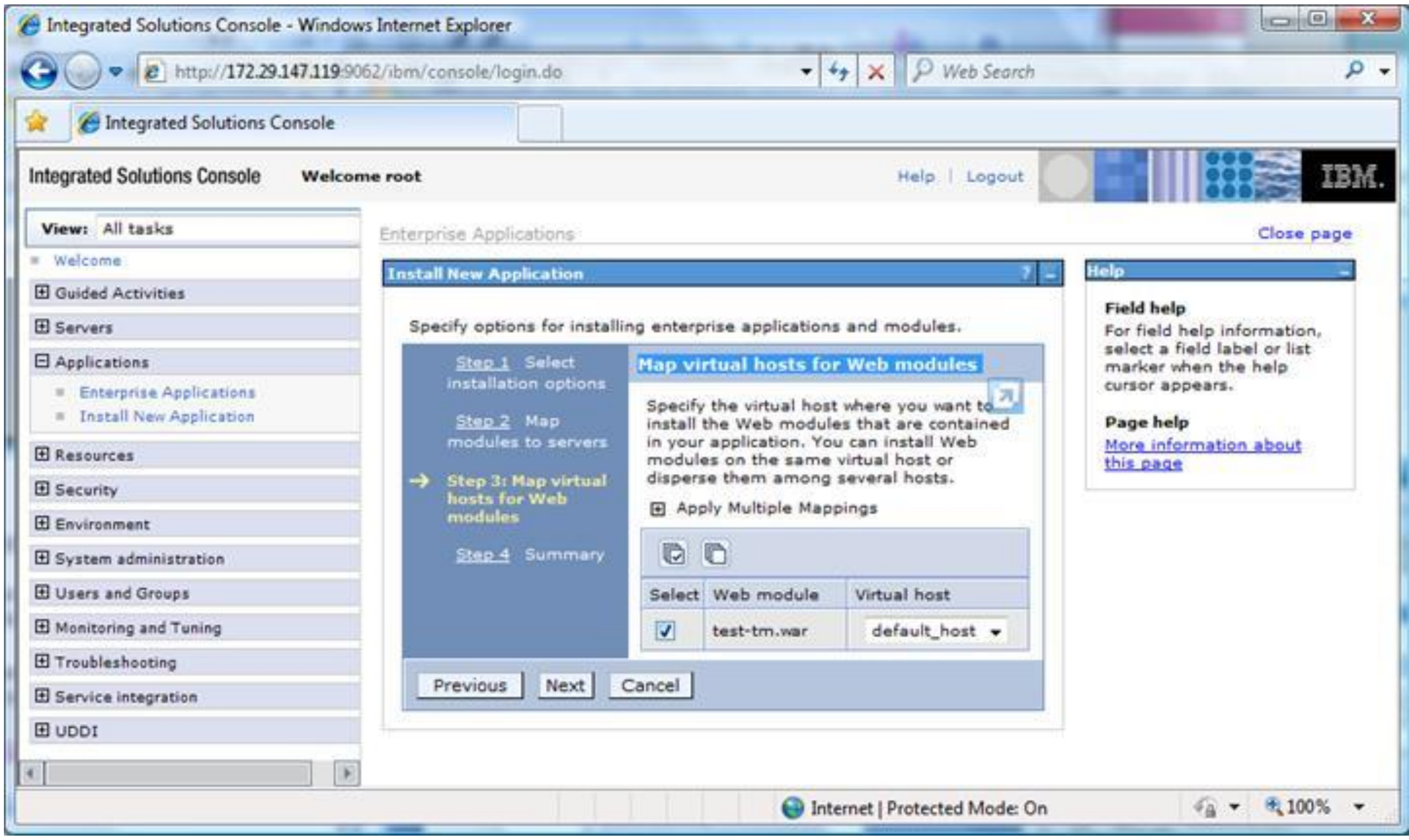
Map modules to servers

Specify targets such as application servers or clusters of application servers where you want to install the modules contained in your application. Modules can be installed on the same application server or dispersed among severa application servers. Also, specify the Web servers as targets that serve as routers for requests to this application. in configuration file (plugin-cfg.xml) for each Web server is generated, based on the applications that are routed

Clusters and Servers:
WebSphere:cell=lpar34ml16ed_pubNode01Cell,node=172.29.147.119Node01,server=server1

Select	Module	URI	Server
✓	test-tm.war	test-tm.war,WEB-INF/web.xml	WebSphere:cell=lpar34ml16ed_pubNode01Cell,node=172.29.147.119Node01,serve

4) In the “Map virtual hosts for Web modules” select the radio button for the new Web Module and click [Next].



Specify options for installing enterprise applications and modules.

Map virtual hosts for Web modules

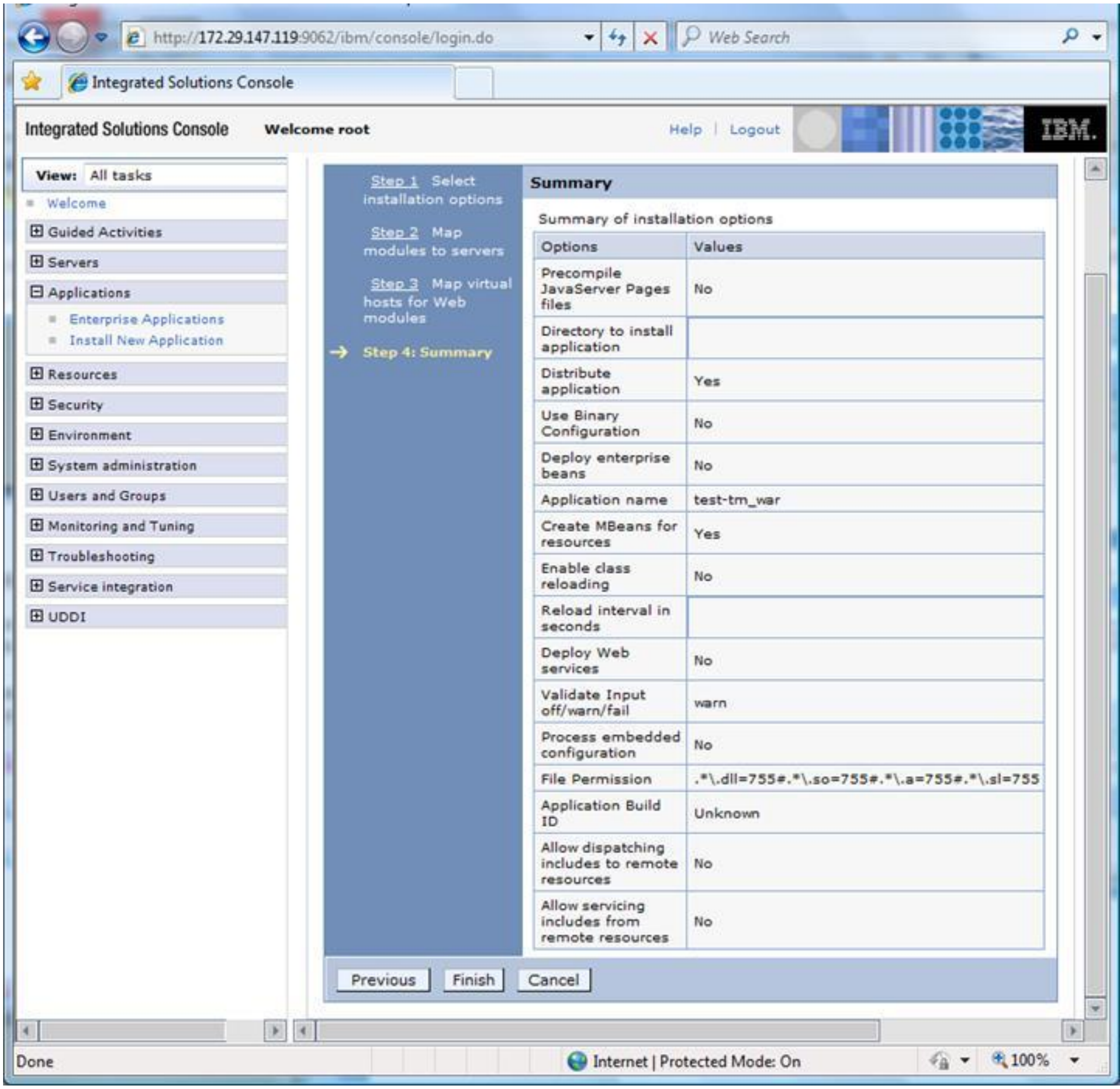
Specify the virtual host where you want to install the Web modules that are contained in your application. You can install Web modules on the same virtual host or disperse them among several hosts.

Select	Web module	Virtual host
✓	test-tm.war	default_host

Field help
For field help information, select a field label or list marker when the help cursor appears.

Page help
More information about this page

5) On the Summary page (step 4) keep all the defaults and click [Finish].


http://172.29.147.119:9062/ibm/console/login.do

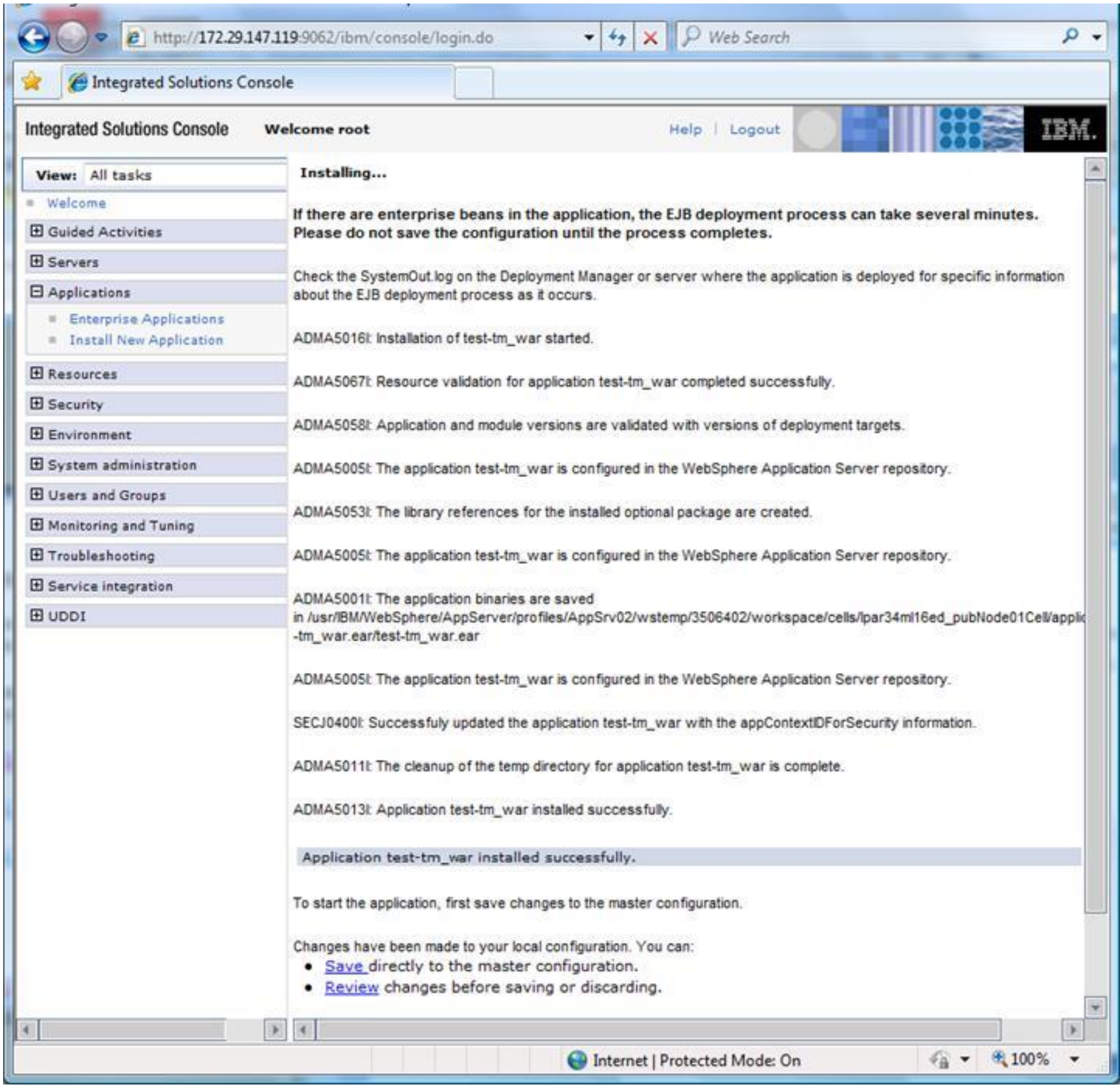
Integrated Solutions Console

Summary

Summary of installation options

Options	Values
Precompile JavaServer Pages files	No
Directory to install application	
Distribute application	Yes
Use Binary Configuration	No
Deploy enterprise beans	No
Application name	test-tm_war
Create MBeans for resources	Yes
Enable class reloading	No
Reload interval in seconds	
Deploy Web services	No
Validate Input off/warn/fail	warn
Process embedded configuration	No
File Permission	.*\.dll=755#.*\.so=755#.*\.a=755#.*\.sl=755
Application Build ID	Unknown
Allow dispatching includes to remote resources	No
Allow servicing includes from remote resources	No

6) Now you should see an Installation Summary and Success message as seen below.
Click on link named “Save” to save changes to the master configuration.



Installing...

If there are enterprise beans in the application, the EJB deployment process can take several minutes. Please do not save the configuration until the process completes.

Check the SystemOut.log on the Deployment Manager or server where the application is deployed for specific information about the EJB deployment process as it occurs.

```
ADMA5016I: Installation of test-tm_war started.

ADMA5067I: Resource validation for application test-tm_war completed successfully.

ADMA5058I: Application and module versions are validated with versions of deployment targets.

ADMA5005I: The application test-tm_war is configured in the WebSphere Application Server repository.

ADMA5053I: The library references for the installed optional package are created.

ADMA5005I: The application test-tm_war is configured in the WebSphere Application Server repository.

ADMA5001I: The application binaries are saved
in /usr/IBM/WebSphere/AppServer/profiles/AppSrv02/wstemp/3506402/workspace/cells/lpar34ml16ed_pubNode01Cell/applic
-tm_war.ear/test-tm_war.ear

ADMA5005I: The application test-tm_war is configured in the WebSphere Application Server repository.

SECJ0400I: Successfuly updated the application test-tm_war with the appContextIDForSecurity information.

ADMA5011I: The cleanup of the temp directory for application test-tm_war is complete.

ADMA5013I: Application test-tm_war installed successfully.
```

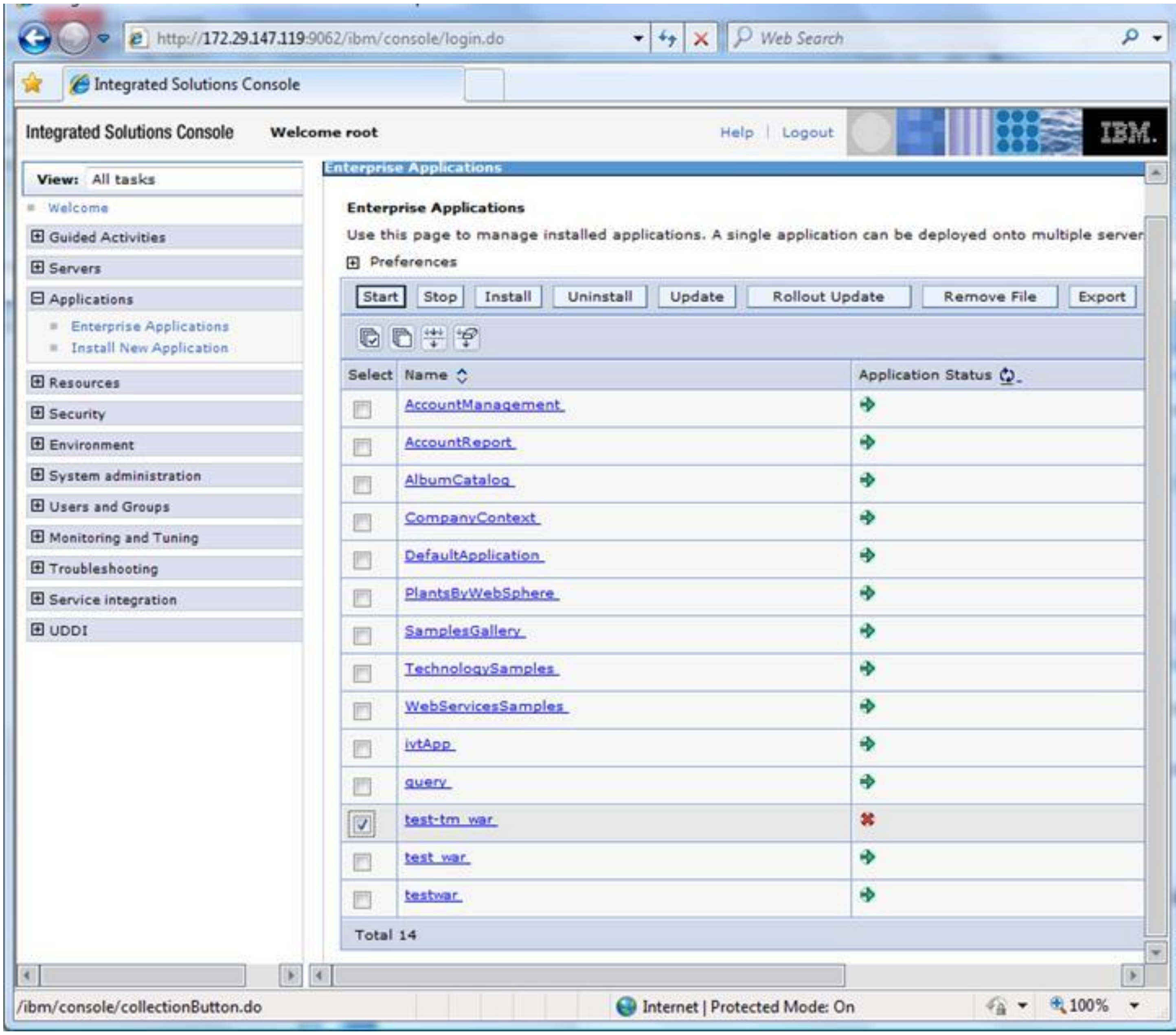
Application test-tm_war installed successfully.

To start the application, first save changes to the master configuration.

Changes have been made to your local configuration. You can:
- Save directly to the master configuration.
- Review changes before saving or discarding.

7) In the Management's left console expand the "Applications" tab again and click on the "Enterprise Applications" link.

8) The center pane now becomes "Enterprise Applications". Locate your new application and select the button to the left of its name and then click [Start]. The Application Status field's Arrow Icon will change from red to green once it is started.



http://172.29.147.119:9062/ibm/console/login.do

Integrated Solutions Console — Welcome root

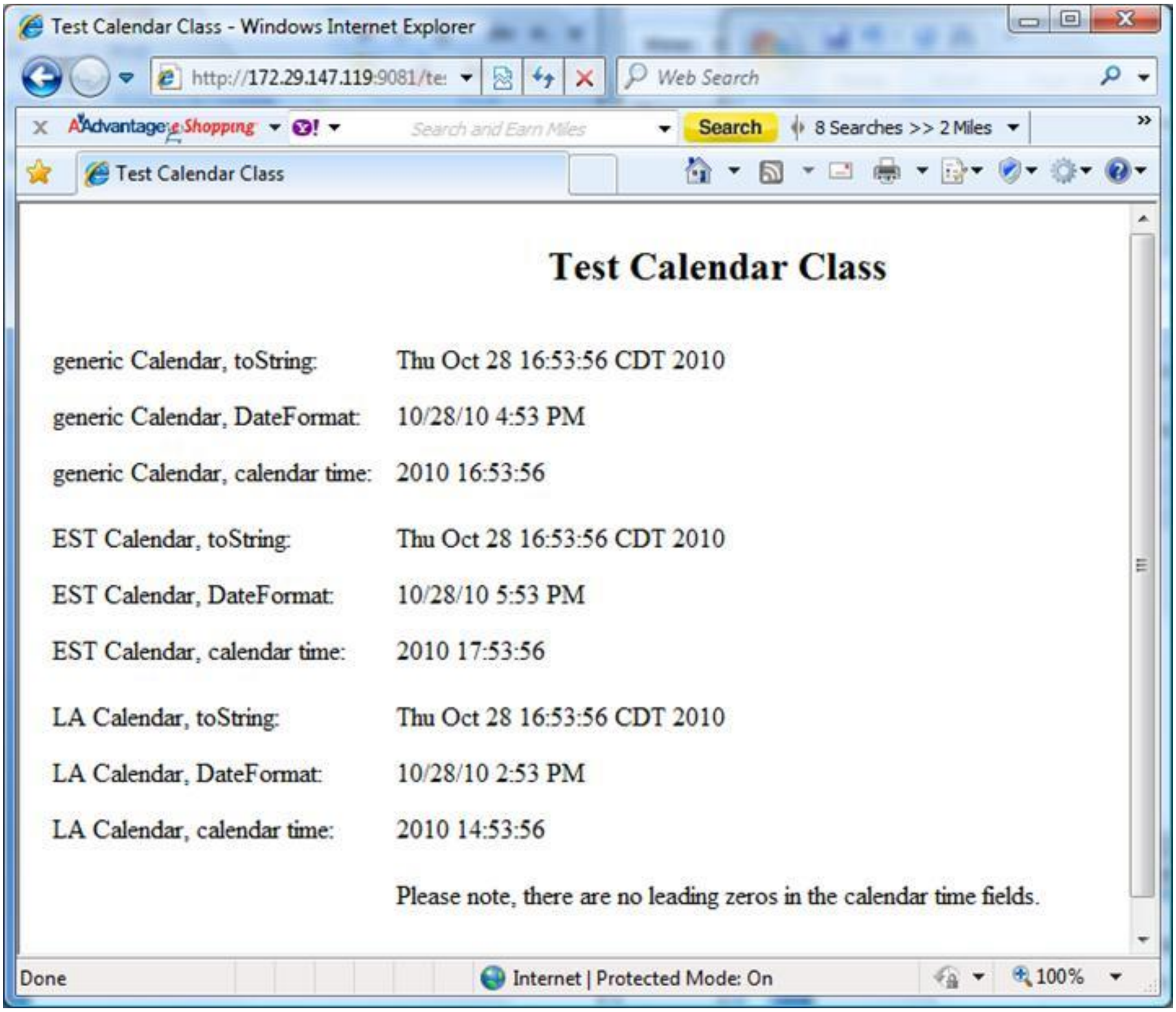
Enterprise Applications

Use this page to manage installed applications. A single application can be deployed onto multiple server

Select	Name	Application Status
☐	AccountManagement	➔
☐	AccountReport	➔
☐	AlbumCatalog	➔
☐	CompanyContext	➔
☐	DefaultApplication	➔
☐	PlantsByWebSphere	➔
☐	SamplesGallery	➔
☐	TechnologySamples	➔
☐	WebServicesSamples	➔
☐	ivtApp	➔
☐	query	➔
✓	test-tm_war	✖
☐	test_war	➔
☐	testwar	➔

Total 14

9) Now the new test application should be running on the machine in the default location.
In the case of the machine used for this paper it is http://172.29.147.119:9081/test-tm/. Start another Web browser window and launch the application to verify it shows current system time & date correctly.


Test Calendar Class - Windows Internet Explorer
http://172.29.147.119:9081/te
Web Search
Search
8 Searches >> 2 Miles
Test Calendar Class
Done
Internet | Protected Mode: On
100%

Test Calendar Class

generic Calendar, toString:	Thu Oct 28 16:53:56 CDT 2010
generic Calendar, DateFormat:	10/28/10 4:53 PM
generic Calendar, calendar time:	2010 16:53:56
EST Calendar, toString:	Thu Oct 28 16:53:56 CDT 2010
EST Calendar, DateFormat:	10/28/10 5:53 PM
EST Calendar, calendar time:	2010 17:53:56
LA Calendar, toString:	Thu Oct 28 16:53:56 CDT 2010
LA Calendar, DateFormat:	10/28/10 2:53 PM
LA Calendar, calendar time:	2010 14:53:56

Please note, there are no leading zeros in the calendar time fields.

Now assign a Time Machine virtual date to the WebSphere server running this test application.

11) From the AIX command line enter the following “ps -ef | grep server1”. This will show the process ID of the WebSphere application server, assuming it is called “server1”. We will use the process ID in Time Machine to assign the virtual date. You will see a result much like the condensed view below:

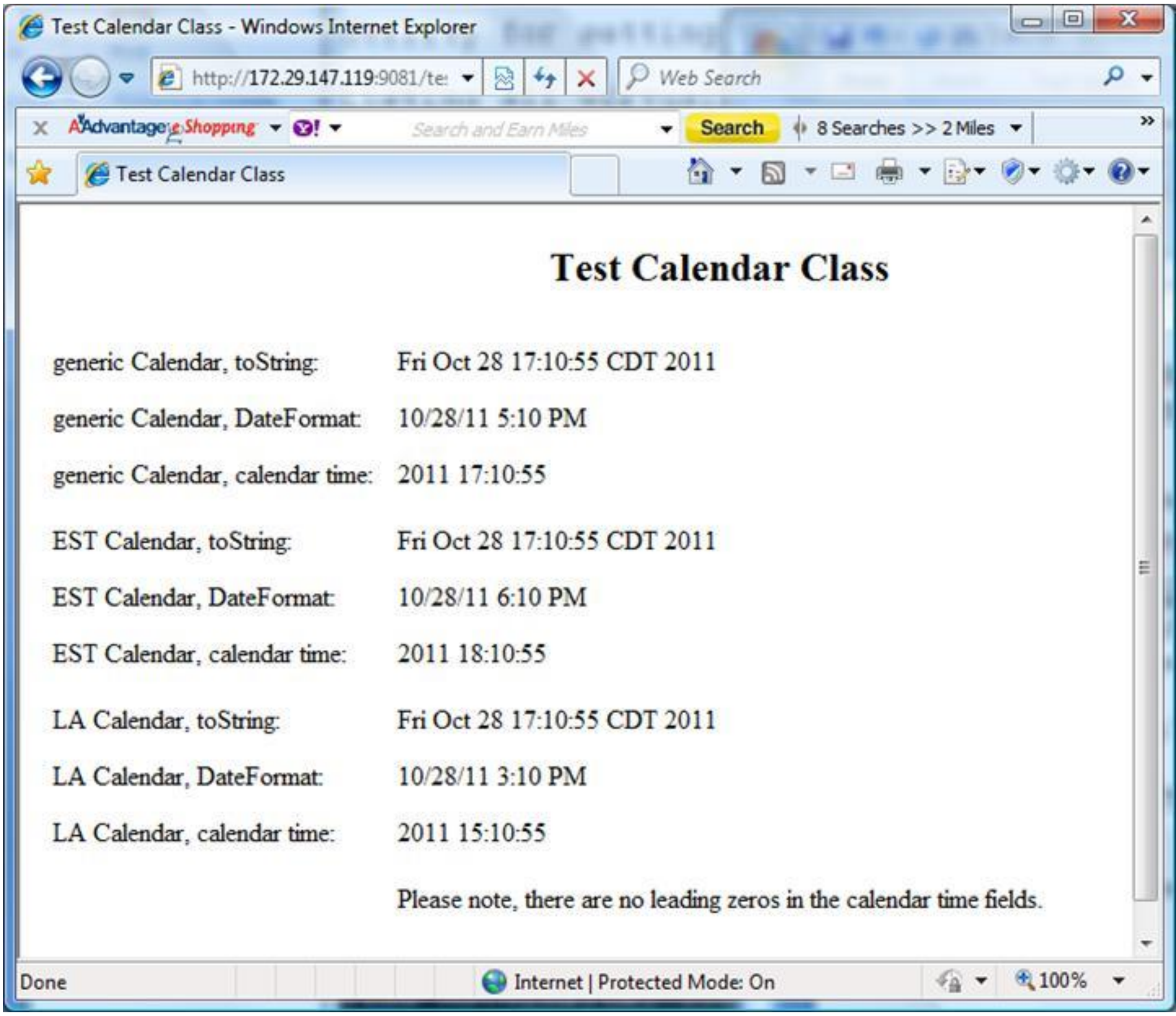
```
root@lpar34ml16ed_pub[/] > ps -ef | grep server1

root 352500    1   0   Oct 27  pts/3  5:08
/home/u0001507/WebSphere/AppServer/java/bin/java -
```

```
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

com.ibm.ws.runtime.WsServer
/usr/IBM/WebSphere/AppServer/profiles/AppSrv02/config
lpar34ml16ed_pubNode01Cell 172.29.147.119Node01 server1
```

12) From the command line set a virtual date exactly one year ahead of the system clock by entering "/etc/ssstm/tmproc -a -p 352500 -y 1". Now reload the test application to see the date one year ahead.

```
root@lpar34ml16ed_pub[/] > /etc/ssstm/tmproc -a -p 352500 -y 1

Copyright(c) 1997-2010 SolutionSoft Systems, Inc.  All Rights Reserved

Time Machine(tm) for IBM AIX 5.3.0.  Version 4.01 R48



Utility for setting TM kernel module configuration.



PID 352500 has been added with a running virtual clock [speed = 1]: Fri Oct 28
17:09:36 2011



Utility for setting TM kernel module configuration.



PID 352500 has been added with a running virtual clock [speed = 1]: Fri Oct 28
17:09:36 2011
```

13) Go back to your browser window for the new test application and hit the refresh button.
The date will now show one year in the future!


Test Calendar Class - Windows Internet Explorer
http://172.29.147.119:9081/te
Web Search
Search
8 Searches >> 2 Miles
Test Calendar Class
Done
Internet | Protected Mode: On
100%

Test Calendar Class

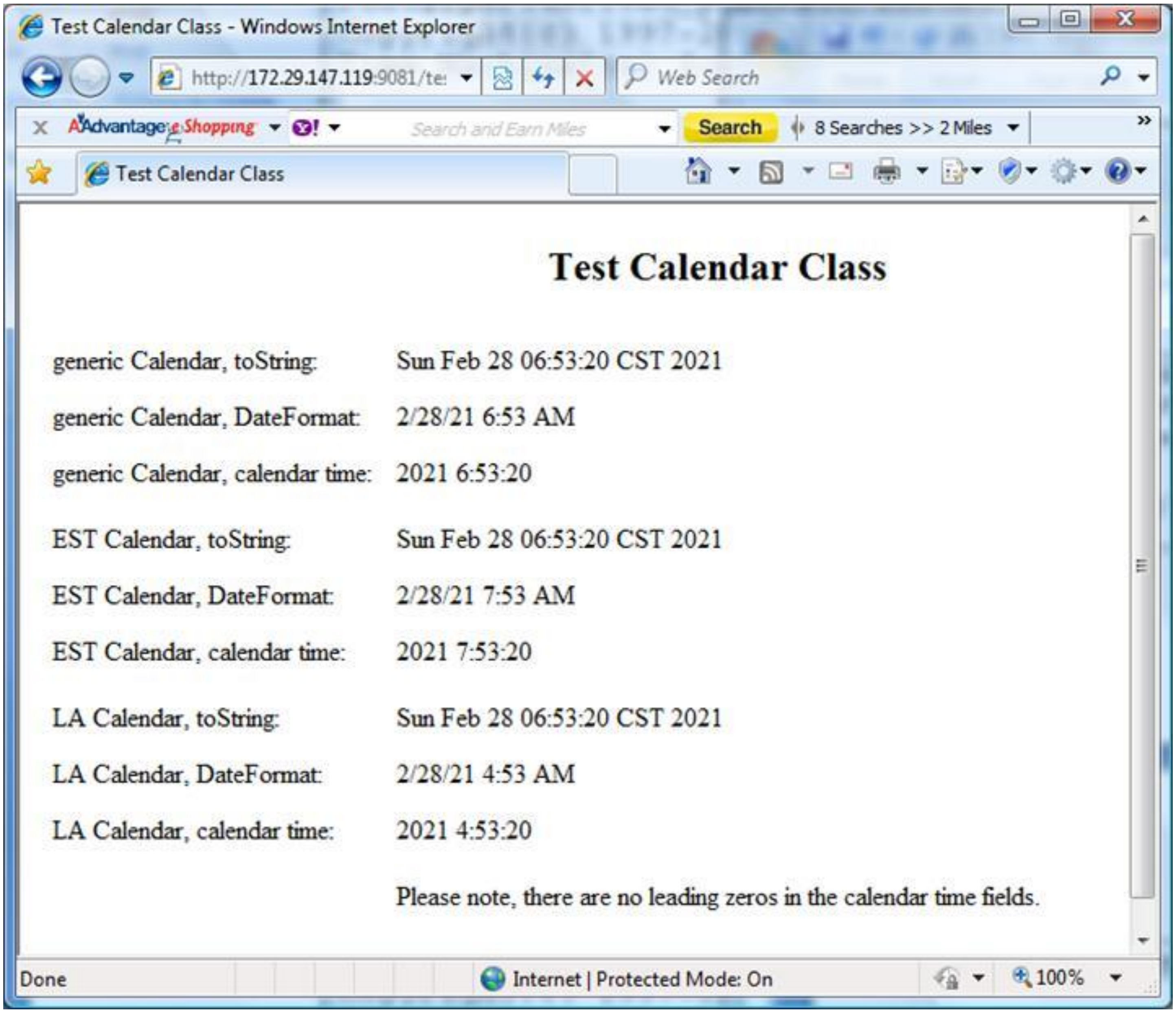
```
generic Calendar, toString:        Fri Oct 28 17:10:55 CDT 2011
generic Calendar, DateFormat:      10/28/11 5:10 PM
generic Calendar, calendar time:   2011 17:10:55
EST Calendar, toString:            Fri Oct 28 17:10:55 CDT 2011
EST Calendar, DateFormat:          10/28/11 6:10 PM
EST Calendar, calendar time:       2011 18:10:55
LA Calendar, toString:             Fri Oct 28 17:10:55 CDT 2011
LA Calendar, DateFormat:           10/28/11 3:10 PM
LA Calendar, calendar time:        2011 15:10:55
                                   Please note, there are no leading zeros in the calendar time fields.
```

14) Now let’s try something really complex. Assign the WebSphere server a Time Machine virtual date of February 28th, 2021 at 3:15 am and to top that off we will also give the application an accelerated clock so that the clock speed is 100 times faster than real time.
Enter “etc/ssstm/tmproc -a -p 352500 –x 022803152021 –s 100”

```
root@lpar34ml16ed_pub[/] > etc/ssstm/tmproc -a -p 352500 -x 022803152021 -s 100

Copyright(c) 1997-2010 SolutionSoft Systems, Inc.  All Rights Reserved

Time Machine(tm) for IBM AIX 5.3.0.  Version 4.01 R48
```

```
Utility for setting TM kernel module configuration.

PID 352500 has been added with a running virtual clock [speed = 100]: Sun Feb 28 03:15:00 2021
```

15) Go back to the Application view in your browser and repeatedly hit your refresh button. You will see the clock moving forward, starting from the year 2021, but at a very fast rate.



Test Calendar Class

```
generic Calendar, toString:       Sun Feb 28 06:53:20 CST 2021
generic Calendar, DateFormat:     2/28/21 6:53 AM
generic Calendar, calendar time:  2021 6:53:20
EST Calendar, toString:           Sun Feb 28 06:53:20 CST 2021
EST Calendar, DateFormat:         2/28/21 7:53 AM
EST Calendar, calendar time:      2021 7:53:20
LA Calendar, toString:            Sun Feb 28 06:53:20 CST 2021
LA Calendar, DateFormat:          2/28/21 4:53 AM
LA Calendar, calendar time:       2021 4:53:20
                                  Please note, there are no leading zeros in the calendar time fields.
```

16) To set the WebSphere server back to real time enter "/etc/ssstm/tmproc -d -p 352500" to delete the virtual date assigned to this process. If you are running a number of instances of application servers and they all have virtual clocks assigned to them you can reset them all back to the system time by running "/etc/ssstm/tmproc reset".

The following is the source code for the test application, JavaServerPage-testTime1.jsp.

```
<%@ page import="java.io.*" %>
<%@ page import="java.util.*" %>
<%@ page import="java.text.DateFormat" %>
<HTML>
<HEAD><TITLE> Test Calendar Class </TITLE></HEAD> <BODY>
<% Calendar cal  = Calendar.getInstance();
   Calendar cal2 = Calendar.getInstance(TimeZone.getTimeZone("America/New_York"),
Locale.US);

   Calendar cal3 =
Calendar.getInstance(TimeZone.getTimeZone("America/Los_Angeles"), Locale.US);
      Date dt = new Date(); %>

  <table cellpadding=5 cellspacing=5>
    <tr>
      <td valign><h2> <center> Test Calendar Class  </center></h2> </td>      </tr>
    <tr>
       <td>generic Calendar, toString:  </td>
         <td> <%  out.print(cal.getTime().toString());  %></td>         </tr>
    <tr>

        <td> generic Calendar, DateFormat: </td>

        <td> <%  DateFormat df = DateFormat.getInstance();

             df.setCalendar(cal);

             out.print(df.format(cal.getTime()));  %>  </td> </tr>

    <tr>

      <td>generic Calendar, calendar time: </td>

        <td> <% out.print(cal.get(Calendar.YEAR) +"   "+
cal.get(Calendar.HOUR_OF_DAY) +":"+
                  cal.get(Calendar.MINUTE) +":"+  cal.get(Calendar.SECOND)); %> </td>

      </tr>

    <tr>

        <td> EST Calendar, toString:</td>

        <td> <%  out.print(cal2.getTime().toString()); %> </td> </tr>

    <tr>
```

```
        <td> EST Calendar, DateFormat: </td>

        <td> <%  DateFormat df2 = DateFormat.getInstance();

             df2.setCalendar(cal2);

             out.print(df2.format(cal2.getTime()));  %>  </td> </tr>

    <tr>

        <td> EST Calendar, calendar time:  </td>

        <td> <%  out.print(cal2.get(Calendar.YEAR) +"   "+
cal2.get(Calendar.HOUR_OF_DAY) +":"+ cal2.get(Calendar.MINUTE)
+":"+  cal2.get(Calendar.SECOND));  %></td>

    </tr>

    <tr>

      <tr>
      <td>LA Calendar, toString: </td>

      <td> <%  out.print(cal3.getTime().toString());  %></td> </tr>

    <tr>

        <td> LA Calendar, DateFormat:</td>

        <td> <%  DateFormat df3 = DateFormat.getInstance();

             df3.setCalendar(cal3);

             out.print(df3.format(cal3.getTime()));  %></td> </tr>

    <tr>

        <td> LA Calendar, calendar time: </td>

        <td> <%  out.print(cal3.get(Calendar.YEAR) +"   "+
cal3.get(Calendar.HOUR_OF_DAY) +":"+ cal3.get(Calendar.MINUTE)
+":"+  cal3.get(Calendar.SECOND));  %></td>

    </tr>

    <tr>
```

```
        <td> Please note, there are no leading zeros in the calendar time fields. </td>
    </tr>

  </table>

 <br>

</body>

</html>
```