ORACLE
CONSULTING

An Oracle White Paper
May 2014

BI Publisher 11g Custom Java Extensions

ORACLE

Disclaimer

Introduction

Oracle BI Publisher can make calls to custom functions written using Java extensions. Java extensions for BI Publisher is a powerful and existing feature prior to the release of 11g version. This white paper describes the process of building, deploying and making use of Java extensions for Oracle BI Publisher 11g (11.1.1.7.0).

In order for the BI Publisher to make use of these custom extensions, the code needs to be made available in the domain library directory or deployed into WebLogic server as a shared library through Java archive files (.jar files). Once made available, calls to these custom functions can be made from within the RTF templates. Java extensions provide handy solutions when neither the data model nor BI Publisher/XSL:FO language can support the business requirements. A variety of requirements such as, complex logic evaluation, complex date computation, mathematic calculations etc. can be fulfilled through the Java extensions.

At present the custom Java extensions for BI Publisher are not supported by Oracle Support.

This white paper describes the following in detail:

- Coding and compiling a sample Java extension
- Building the Java archive (.jar file) and updating the extension related information
- Deploying the Java extension as a WebLogic shared library
- Updating the extension related information into BI Publisher application (*xmlpserver.ear*)
- Updating (re-deploying) BI Publisher application into WebLogic
- Building the sample report
- BI Publisher Runtime Configuration parameters to accommodate Java extensions

About the Sample Used

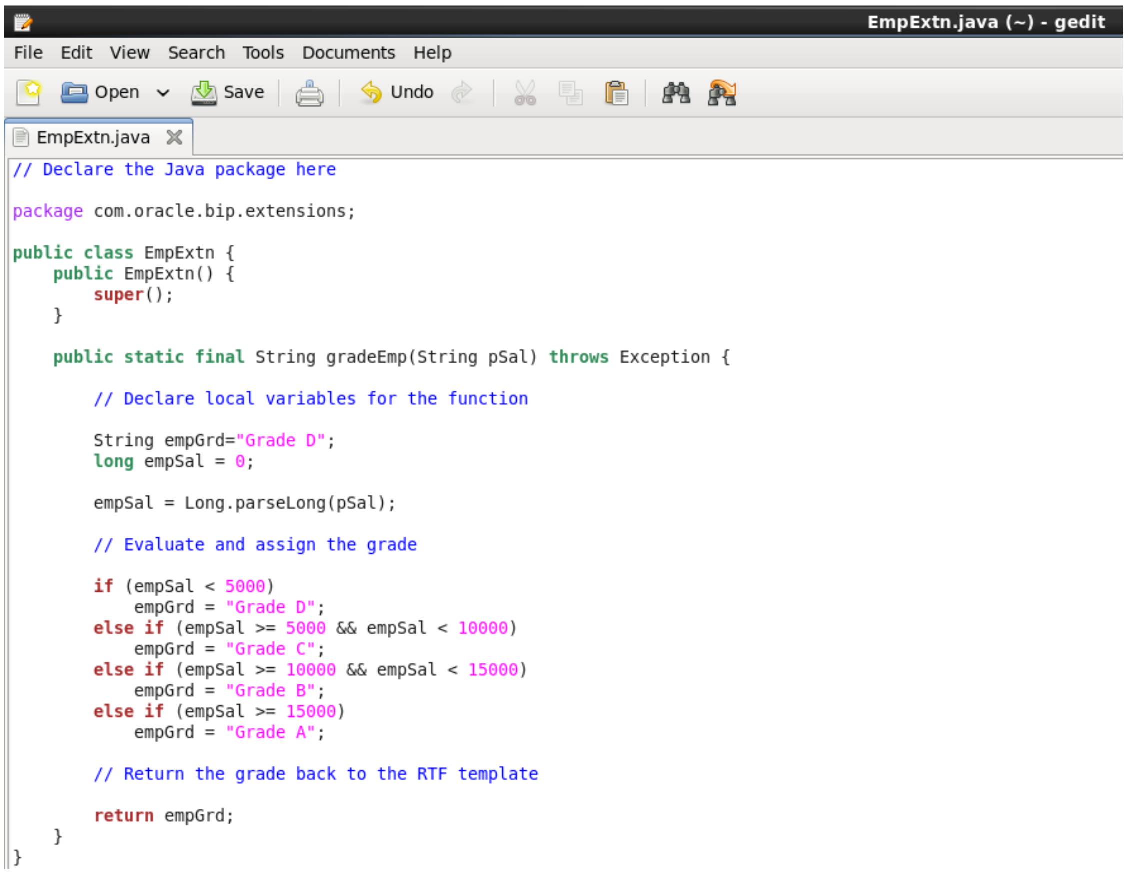
The sample used in this white paper makes use of Oracle Database sample HR schema. The OS used for Oracle Business Intelligence 11g (11.1.1.7.0) version is Oracle Linux 6 – 64 bit. For RTF template development, Microsoft® Word is used on Microsoft® Windows-7 desktop.

The sample report displays content from "Employees" table of HR schema along with employee grade. Grade is not part of the "Employee" table of HR schema. To determine the employee grade, Java extension is used. For this sample, employees earning less than $5,000 would be under grade "D", employees earning between $5,000 and $10,000 would be under grade "C", employees earning between $10,000 and $15,000 would be under grade "B" and the rest would be under grade "A".

While calling the Java extension, the RTF template would pass the salary of the respective employee record, and the Java extension would evaluate the same and return the grade.

Coding and Compiling the Java Extension

The following sample code is used for *EmpExtn.java* file:



```java
// Declare the Java package here

package com.oracle.bip.extensions;

public class EmpExtn {
    public EmpExtn() {
        super();
    }

    public static final String gradeEmp(String pSal) throws Exception {

        // Declare local variables for the function

        String empGrd="Grade D";
        long empSal = 0;

        empSal = Long.parseLong(pSal);

        // Evaluate and assign the grade

        if (empSal < 5000)
            empGrd = "Grade D";
        else if (empSal >= 5000 && empSal < 10000)
            empGrd = "Grade C";
        else if (empSal >= 10000 && empSal < 15000)
            empGrd = "Grade B";
        else if (empSal >= 15000)
            empGrd = "Grade A";

        // Return the grade back to the RTF template

        return empGrd;
    }
}
```

It is recommended to compile the Java code using the JDK supplied along with Oracle Business Intelligence 11g or a compatible JDK must be used.

Figure1. Code Used for Java Extension

The Java version that is supplied along with Oracle Business Intelligence 11g 11.1.1.7.0 is:

```
java version "1.6.0_35"
Java(TM) SE Runtime Environment (build 1.6.0_35-b52)
Java HotSpot(TM) 64-Bit Server VM (build 20.10-b01, mixed mode)
```

It is recommended to compile the above code through the JDK that is supplied along with Oracle Business Intelligence 11g 11.1.1.7.0 version. This code can be compiled using other versions of JDK as long as the compiled class files are compatible with the above mentioned JDK version.

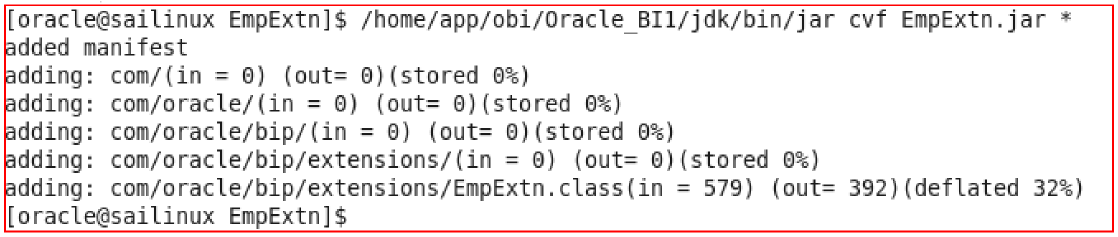
The location of this JDK is **<*MiddlewareHome*>/Oracle_BI1/jdk**. (Example: **/home/app/obi/Oracle_BI1/jdk**). Sample command to compile the code:

```
$ /home/app/obi/Oracle_BI1/jdk/bin/javac EmpExtn.java
```

The custom Java extension should always return a string as BI Publisher cannot handle any other data types. Once the file is successfully compiled, the class file needs to be archived into a Java archive using the "jar" command. The class file should be made available in the appropriate directory as per the package declaration inside the source code.

Sample "*jar*" command to archive the code:

$ /home/app/obi/Oracle_BI1/jdk/bin/jar cvf EmpExtn.jar *



```
[oracle@sailinux EmpExtn]$ /home/app/obi/Oracle_BI1/jdk/bin/jar cvf EmpExtn.jar *
added manifest
adding: com/(in = 0) (out= 0)(stored 0%)
adding: com/oracle/(in = 0) (out= 0)(stored 0%)
adding: com/oracle/bip/(in = 0) (out= 0)(stored 0%)
adding: com/oracle/bip/extensions/(in = 0) (out= 0)(stored 0%)
adding: com/oracle/bip/extensions/EmpExtn.class(in = 579) (out= 392)(deflated 32%)
[oracle@sailinux EmpExtn]$
```

Figure2. Archiving the Java Extension

The simplest way to deploy the archive is to make *EmpExtn.jar* file available in the WebLogic domain library directory i.e. ***<MiddleWare_Home>*/Oracle_BI1/user_projects/domains/bifoundation_domain/lib** (Example: **/home/app/obi/user_projects/domains/bifoundation_domain/lib**). In this method, once the file is copied the server needs to be restarted. Any changes to the archive in the future would also require the server to be restarted.

For those planning to follow the simple method, can copy the *EmpExtn.jar* as mentioned above and restart the server. Once the server is restarted, further steps can be followed from Building the Sample Report section of this white paper.

This white paper also describes the other method of deploying the archive as a WebLogic shared library in the following sections. The WebLogic documentation provided below describes in detail, about the usefulness of each of these methods:

Oracle® Fusion Middleware Developing Applications for Oracle WebLogic Server 11*g* Release 1 (10.3.5) Part Number E13706-05

Section "Adding JARs to the Domain /lib Directory" of
Chapter 8 Understanding WebLogic Server Application Classloading

http://docs.oracle.com/cd/E21764_01/web.1111/e13706/classloading.htm#i1096881

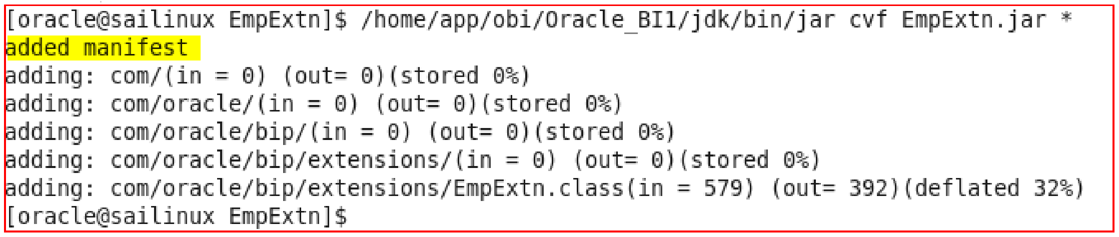
Chapter 9 Creating Shared Java EE Libraries and Optional Packages

http://docs.oracle.com/cd/E21764_01/web.1111/e13706/libraries.htm#g1092115

Updating the Extension Information in the Archive

This step is required only for deploying *EmpExtn.jar* as a WebLogic shared library.

Notice that when the jar file is created, jar tool automatically adds the manifest related information. The manifest information is available under folder **META-INF** with file name as **MANIFEST.MF**. The next step is to edit the MANIFEST.MF file to add the extension related information.



```
[oracle@sailinux EmpExtn]$ /home/app/obi/Oracle_BI1/jdk/bin/jar cvf EmpExtn.jar *
added manifest
adding: com/(in = 0) (out= 0)(stored 0%)
adding: com/oracle/(in = 0) (out= 0)(stored 0%)
adding: com/oracle/bip/(in = 0) (out= 0)(stored 0%)
adding: com/oracle/bip/extensions/(in = 0) (out= 0)(stored 0%)
adding: com/oracle/bip/extensions/EmpExtn.class(in = 579) (out= 392)(deflated 32%)
[oracle@sailinux EmpExtn]$
```

The extension name declared in the manifest file would be referenced inside the *xmlpserver.ear* file and in the RTF templates

Figure3. Manifest Information of the Archive

For editing the MANIFEST.MF file, **7-Zip** utility is used under Windows 7 environment. Any similar tool which allows the file to be modified inside the archive could also be used.

The archive can be opened by Right clicking on the file and by choosing 7-Zip – Open archive as shown below:

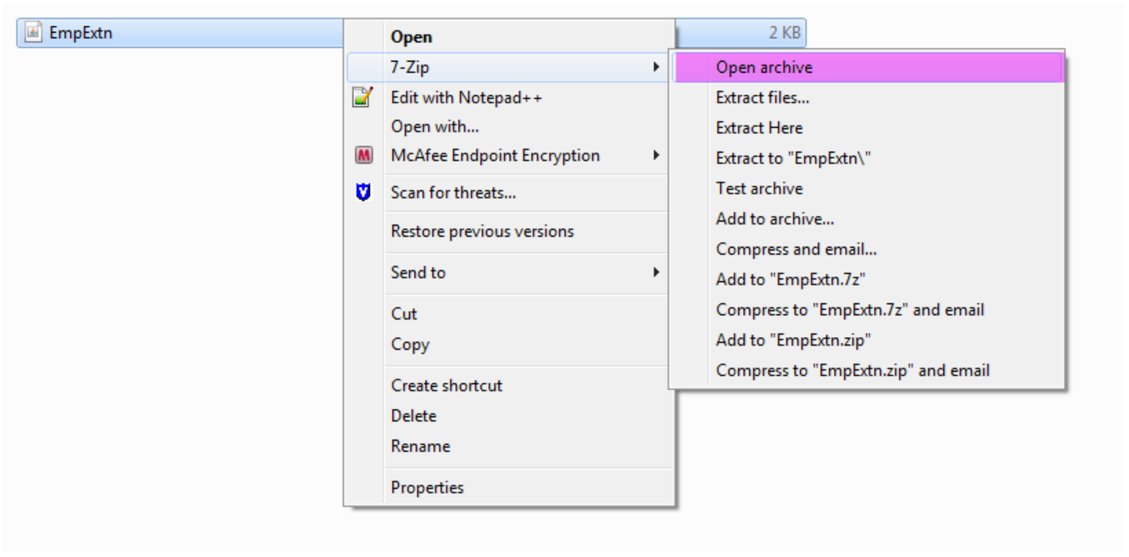


Figure 4. Manifest Information of the Archives

This will open the archive inside 7-Zip utility. The next step is to edit the **MANIFEST.MF** file, by double clicking on **MANIFEST** folder and then right click on **MANIFEST.MF** and choosing Edit as shown below:

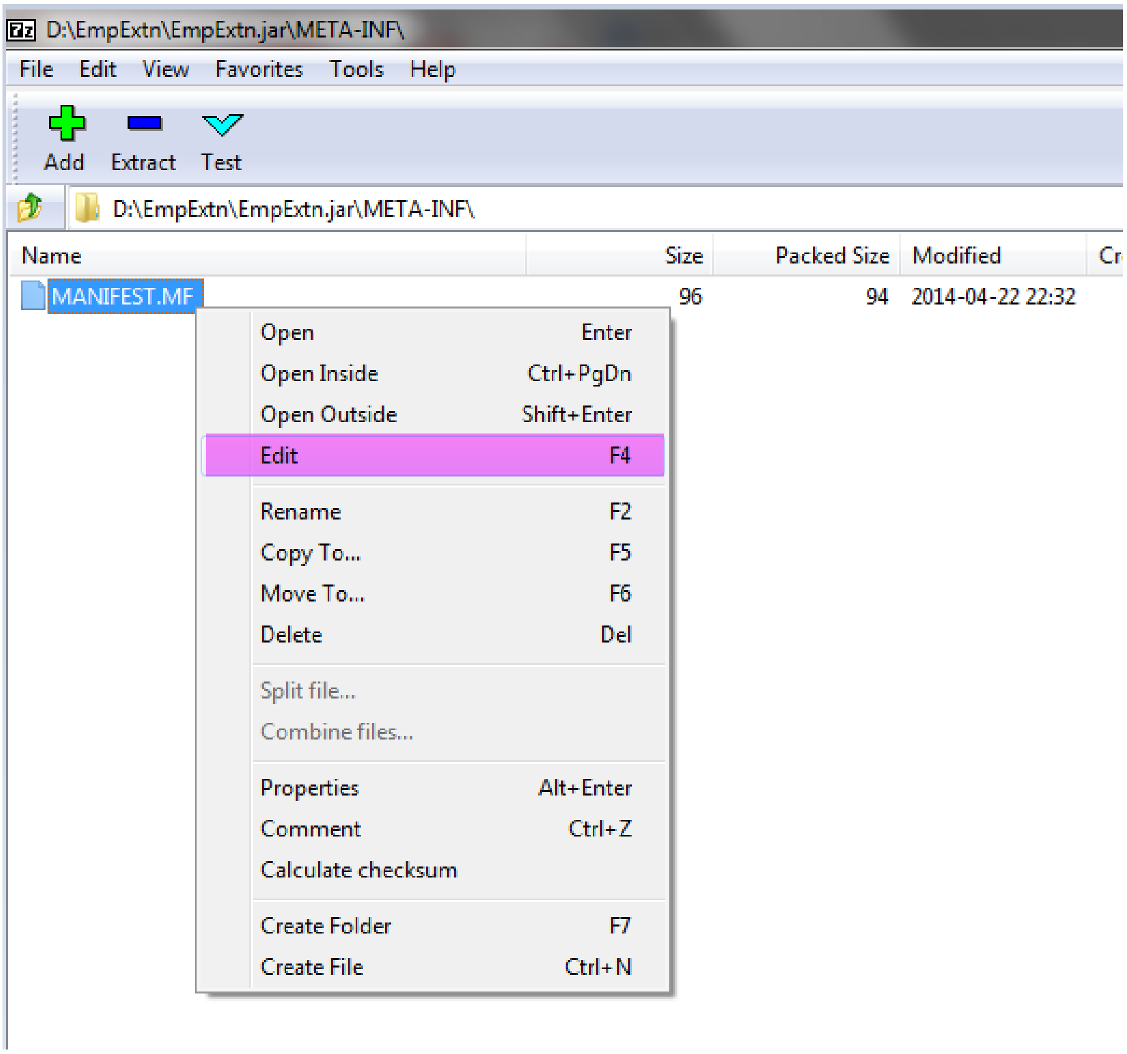
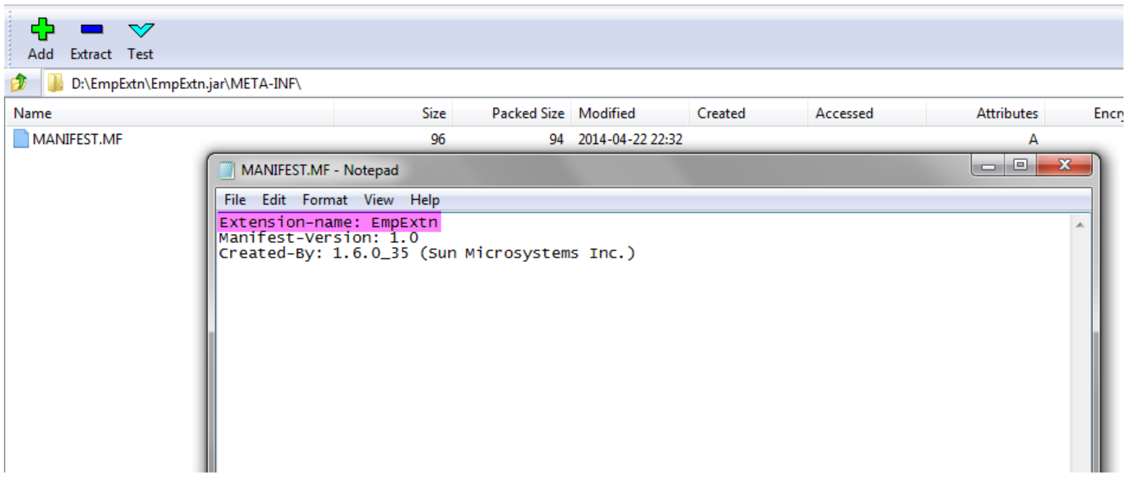


Figure 5. Manifest Information of the Archive

Once available for editing, the following line needs to be added to the MANIFEST.MF file:

Extension-name: EmpExtn

The extension name could be any user defined name (in this example it is "*EmpExtn*"). However, the same extension name needs to be added to the xmlpserver.ear file and the same extension name needs to be used as part of the namespace inside RTF template. This is explained in detail in the later part of this white paper.

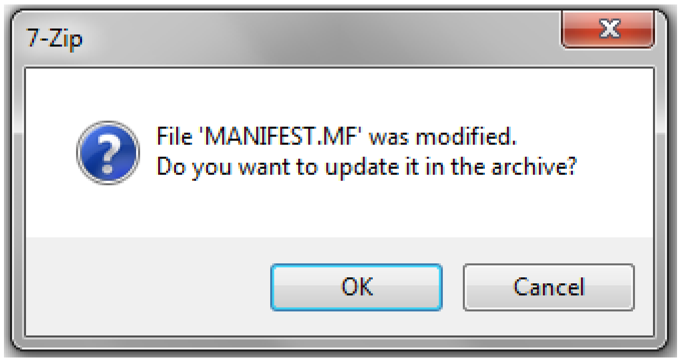


Name	Size	Packed Size	Modified	Created	Accessed	Attributes	Encr
MANIFEST.MF	96	94	2014-04-22 22:32			A	

```
Extension-name: EmpExtn
Manifest-Version: 1.0
Created-By: 1.6.0_35 (Sun Microsystems Inc.)
```

Figure 6. Extension Name in the Manifest File

After the “Extension-name” line is added, the file needs to be saved. After exiting the editor, 7-Zip would prompt to apply the changes into the archive:


7-Zip

File 'MANIFEST.MF' was modified.
Do you want to update it in the archive?

Figure 7. Saving the Manifest File

Once the “OK” button is pressed the archive is updated, it may need to be transferred to the server (as in this example the file is modified in Windows 7 desktop and needs to be transferred to the Linux server on which Oracle Business Intelligence 11.1.1.7.0 is installed)

Deploying the Java Extension as a WebLogic Shared Library

This step is required only if *EmpExtn.jar* is deployed as a WebLogic shared library.

For deploying the *EmpExtn.jar* as a shared library, this white paper used the location of this file as ***<MiddlewareHome>*/Oracle_BI1/user_projects/domains/bifoundation_domain/lib** (Example: **/home/app/obi/user_projects/domains/bifoundation_domain/lib**). Any other directory with appropriate permissions could be used in place of this.

The Java archive needs to be deployed as a shared library through WebLogic Admin Console. Once logged into the admin console (in this example using URL: **http://192.168.199.4:7001/console**), the "Lock & Edit" button needs to be clicked under "Change Center".

For deploying Java extension as a WebLogic shared library, the JAR file can be made available in any directory with appropriate permissions

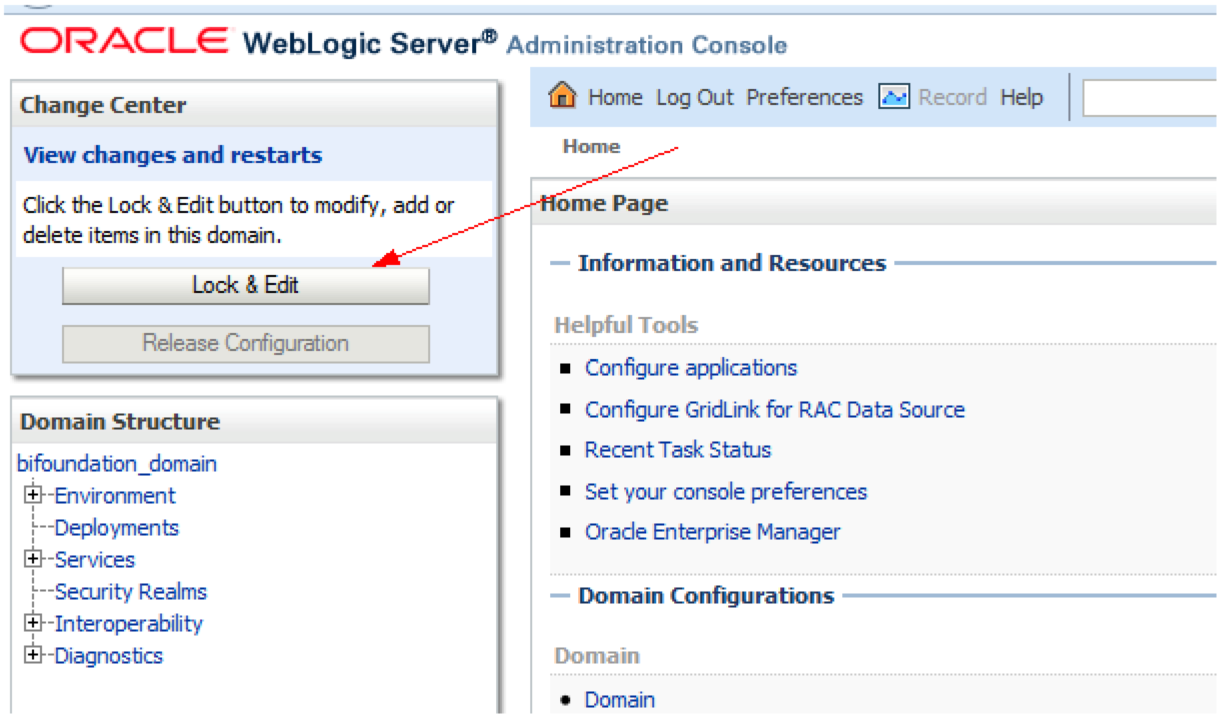
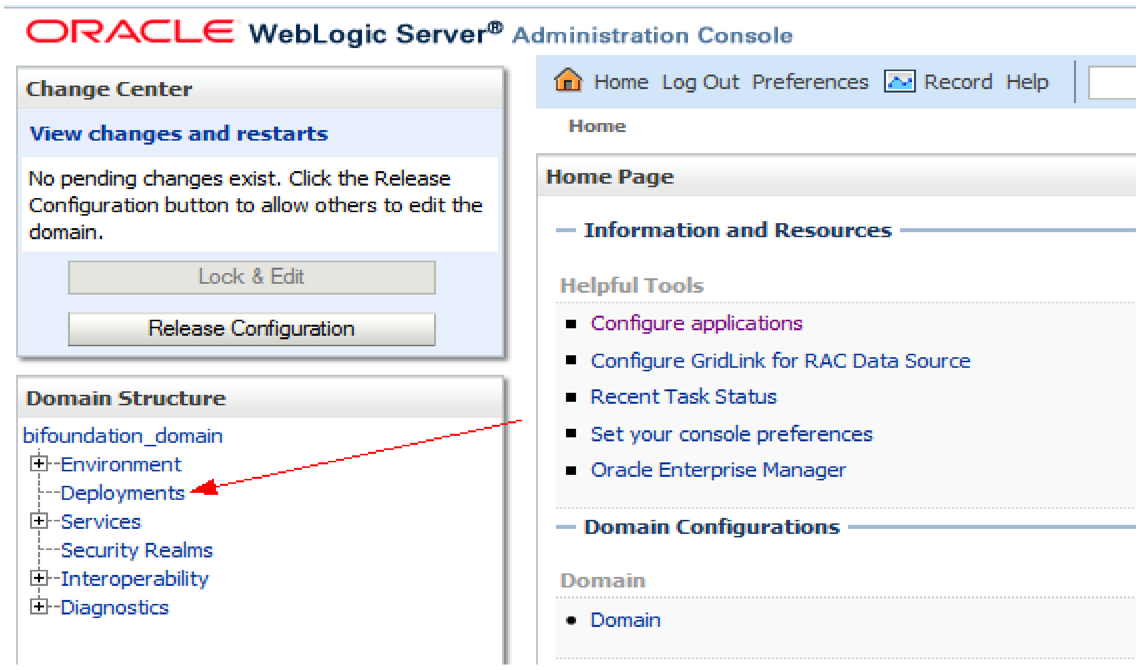

ORACLE WebLogic Server® Administration Console
Change Center
View changes and restarts
Click the Lock & Edit button to modify, add or delete items in this domain.
Lock & Edit
Release Configuration
Domain Structure
bifoundation_domain
Environment
Deployments
Services
Security Realms
Interoperability
Diagnostics
Home Log Out Preferences Record Help
Home
Home Page
Information and Resources
Helpful Tools
Configure applications
Configure GridLink for RAC Data Source
Recent Task Status
Set your console preferences
Oracle Enterprise Manager
Domain Configurations
Domain
Domain

Figure 8. WebLogic Admin Console – Lock & Edit

Once this is done, the library needs to be deployed by clicking on Deployments.



ORACLE WebLogic Server® Administration Console

Change Center

View changes and restarts

No pending changes exist. Click the Release Configuration button to allow others to edit the domain.

Domain Structure

bifoundation_domain
- Environment
- Deployments
- Services
- Security Realms
- Interoperability
- Diagnostics

Home Page

Information and Resources

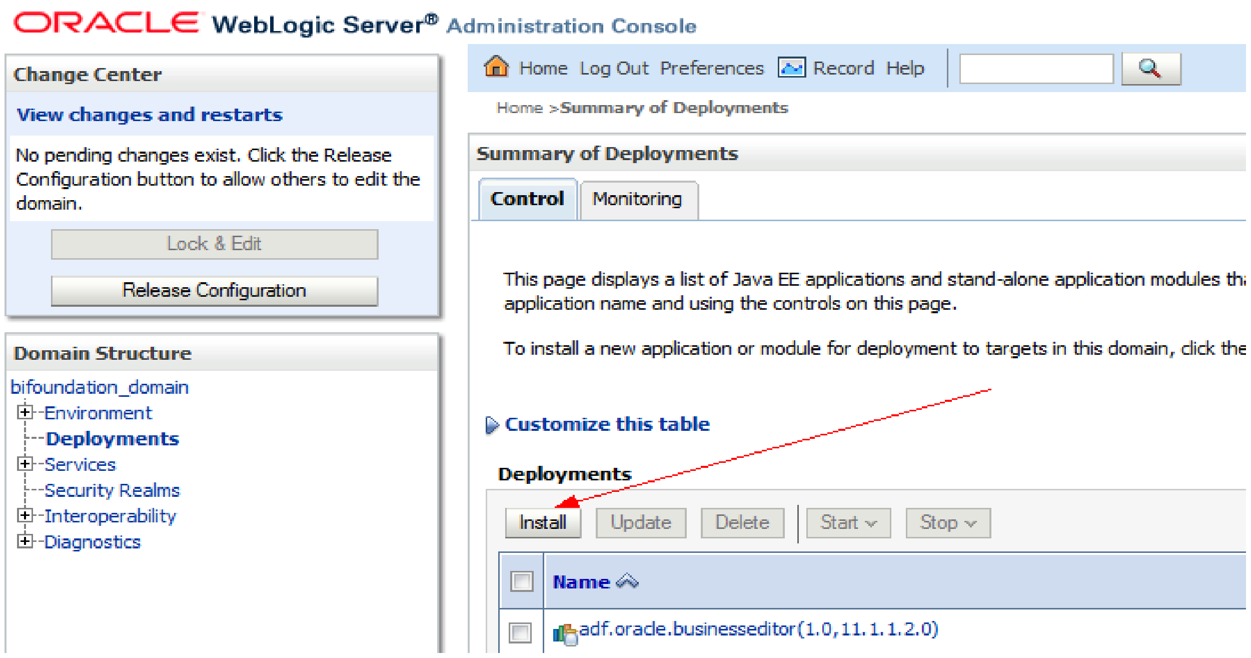
Helpful Tools
- Configure applications
- Configure GridLink for RAC Data Source
- Recent Task Status
- Set your console preferences
- Oracle Enterprise Manager

Domain Configurations

Domain
- Domain

Figure 9. WebLogic Admin Console Deployments

For installing the archive file as a library, “Install” button needs to be clicked.



ORACLE WebLogic Server® Administration Console

Change Center

View changes and restarts

No pending changes exist. Click the Release Configuration button to allow others to edit the domain.

Lock & Edit
Release Configuration
Home Log Out Preferences Record Help
Home >Summary of Deployments

Domain Structure

bifoundation_domain
- Environment
- Deployments
- Services
- Security Realms
- Interoperability
- Diagnostics

Summary of Deployments

This page displays a list of Java EE applications and stand-alone application modules thi
application name and using the controls on this page.

To install a new application or module for deployment to targets in this domain, click the

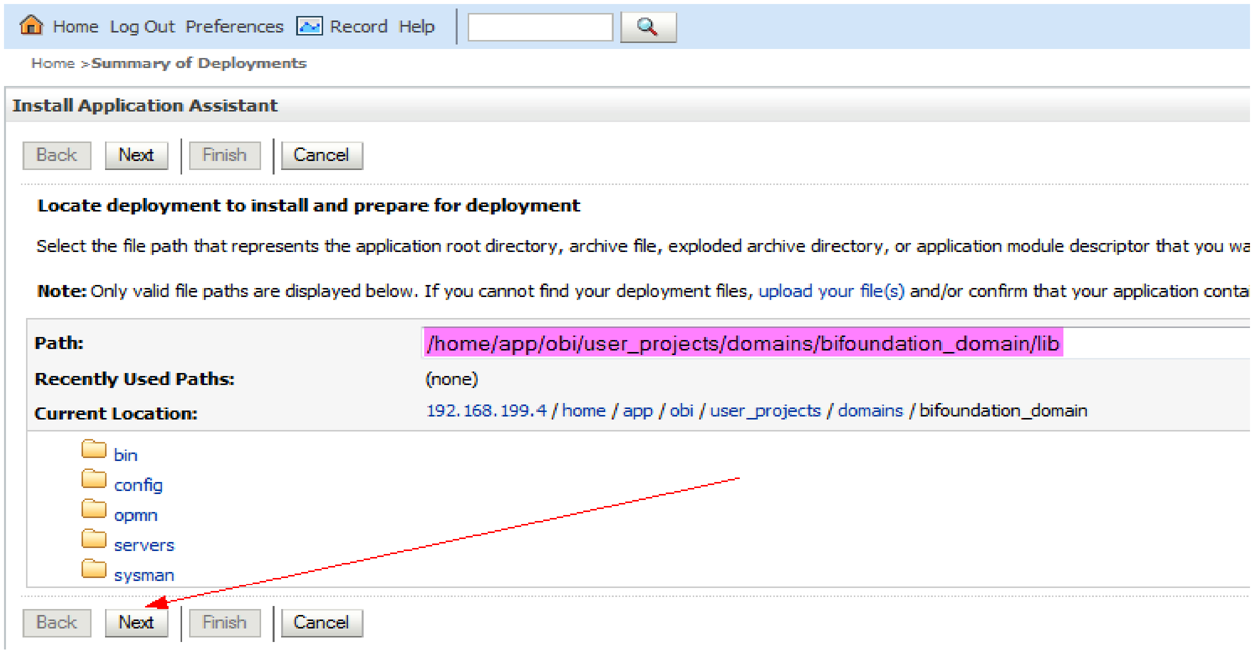
Customize this table

Deployments

	Name
☐	adf.oracle.businesseditor(1.0,11.1.1.2.0)

Figure 10. WebLogic Admin Console Installing Shared Library

The path for the Java archive file needs to be entered inside the "Path" text field (for this example it is: **/home/app/obi/user_projects/domains/bifoundation_domain/lib**) and "Next" button needs to be clicked. Alternatively, a directory or an archive file can be selected as well, if the path is already pointing to the desired location.


Home Log Out Preferences Record Help
Home >Summary of Deployments
Back
Next
Finish
Cancel

Install Application Assistant

Locate deployment to install and prepare for deployment

Select the file path that represents the application root directory, archive file, exploded archive directory, or application module descriptor that you wa

Note: Only valid file paths are displayed below. If you cannot find your deployment files, upload your file(s) and/or confirm that your application conta

Path: /home/app/obi/user_projects/domains/bifoundation_domain/lib

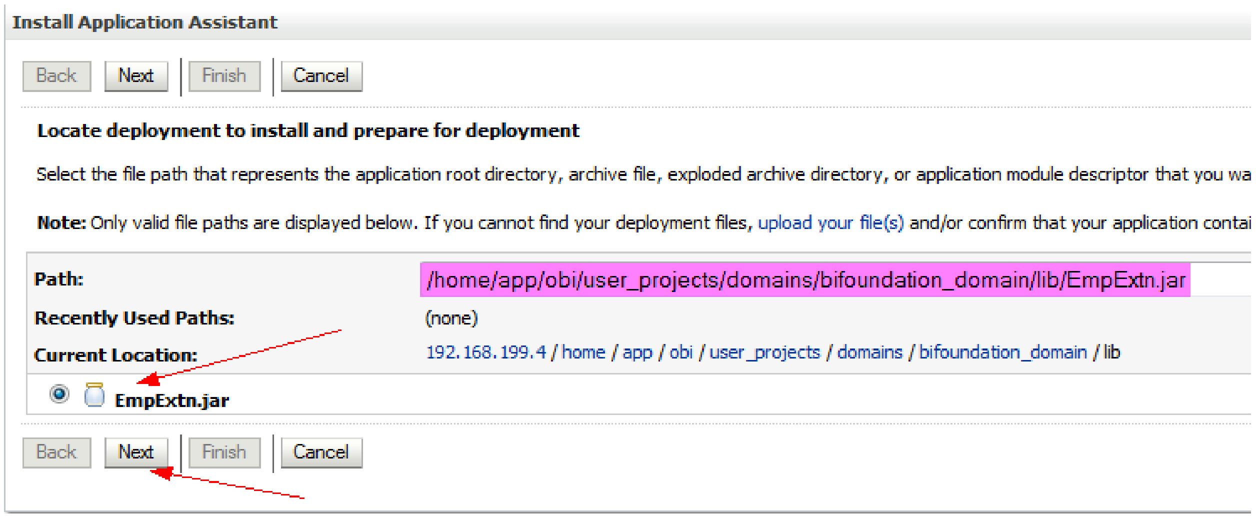
Recently Used Paths: (none)

Current Location: 192.168.199.4 / home / app / obi / user_projects / domains / bifoundation_domain

bin
config
opmn
servers
sysman

Figure 11. WebLogic Shared Library Location

The archive files needs to be selected and "Next" button needs to be clicked to move forward.



Install Application Assistant

Locate deployment to install and prepare for deployment

Select the file path that represents the application root directory, archive file, exploded archive directory, or application module descriptor that you wa

Note: Only valid file paths are displayed below. If you cannot find your deployment files, upload your file(s) and/or confirm that your application contai

Path: /home/app/obi/user_projects/domains/bifoundation_domain/lib/EmpExtn.jar

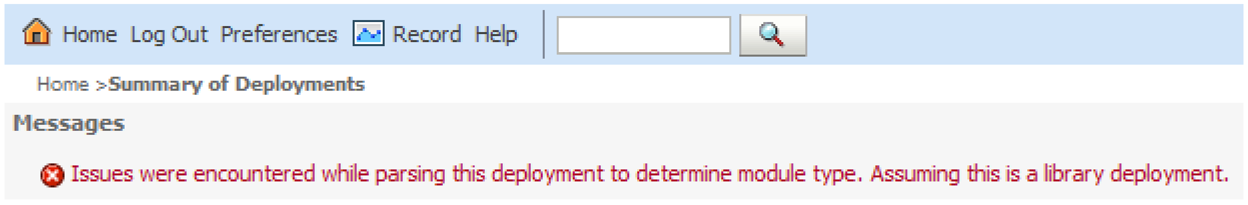
Recently Used Paths: (none)

Current Location: 192.168.199.4 / home / app / obi / user_projects / domains / bifoundation_domain / lib

● EmpExtn.jar

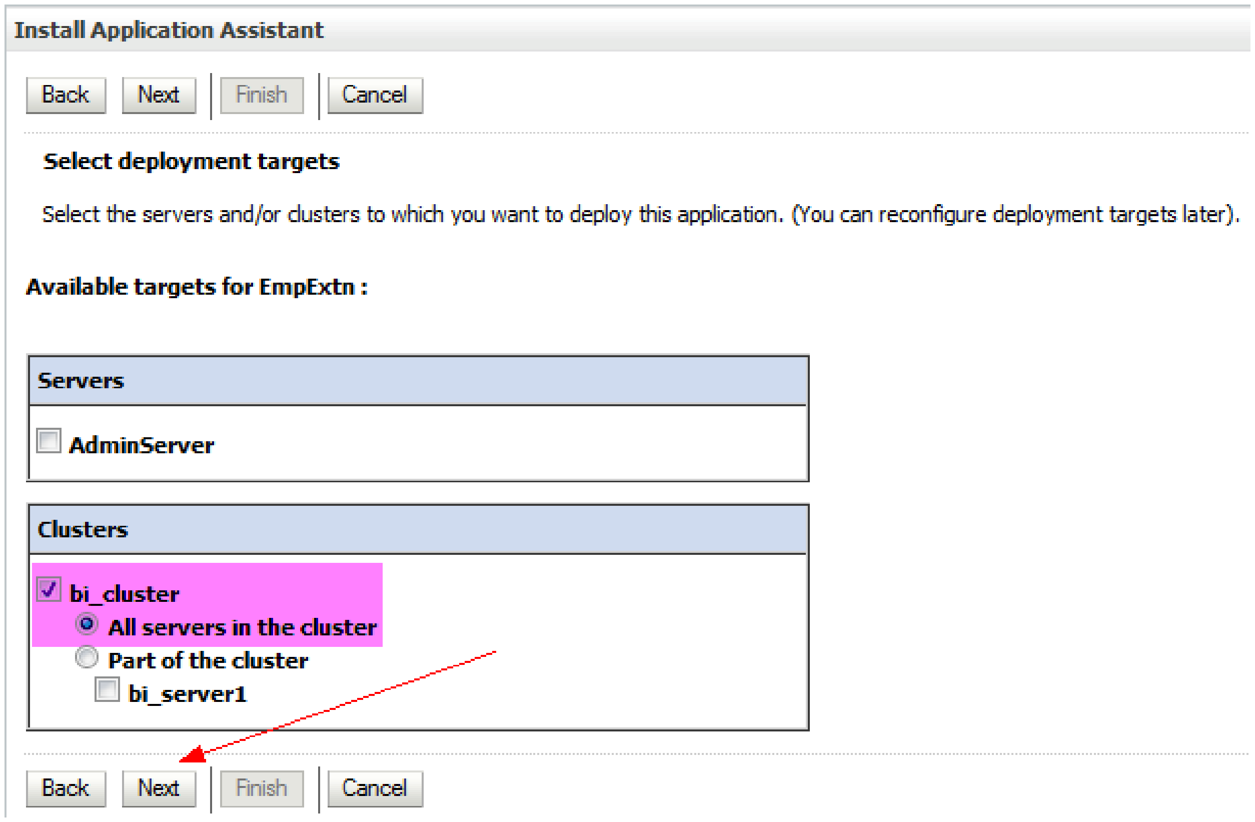
Figure 12. WebLogic Shared Library Selection

The next step is to choose the deployment target. The following error message might be displayed in this screen and this message can be ignored, by clicking on “Next” button after choosing the deployment targets.


Home Log Out Preferences Record Help
Home >Summary of Deployments
Messages

Issues were encountered while parsing this deployment to determine module type. Assuming this is a library deployment.

Figure 13. WebLogic Shared Library Ignore Warning



Select deployment targets

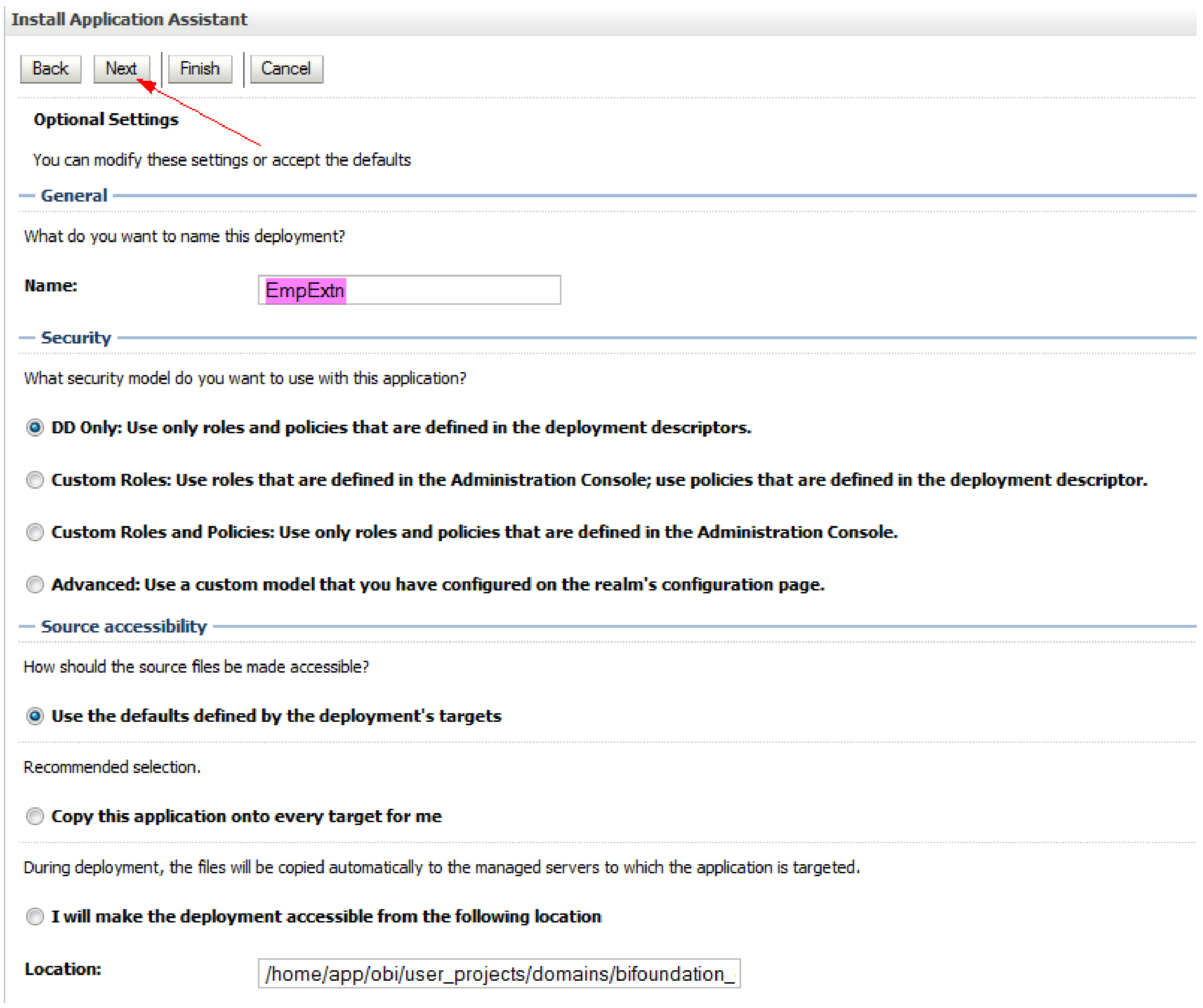
Select the servers and/or clusters to which you want to deploy this application. (You can reconfigure deployment targets later).

Available targets for EmpExtn :

- [ ] AdminServer
- [✓] bi_cluster
  - [✓ All servers in the cluster]
  - Part of the cluster
    - [ ] bi_server1

Figure 14. WebLogic Shared Library Deployment Target

For this example, all the default values are retained in the resulting screen, by ensuring the “Name” in the “General” section is “*EmpExtn*”. The “Next” button needs to be clicked to move forward.



Optional Settings

You can modify these settings or accept the defaults

General

What do you want to name this deployment?

Name: EmpExtn

Security

What security model do you want to use with this application?

- [✓] **DD Only: Use only roles and policies that are defined in the deployment descriptors.**
- **Custom Roles: Use roles that are defined in the Administration Console; use policies that are defined in the deployment descriptor.**
- **Custom Roles and Policies: Use only roles and policies that are defined in the Administration Console.**
- **Advanced: Use a custom model that you have configured on the realm's configuration page.**

Source accessibility

How should the source files be made accessible?

- [✓] **Use the defaults defined by the deployment's targets**

Recommended selection.

- **Copy this application onto every target for me**

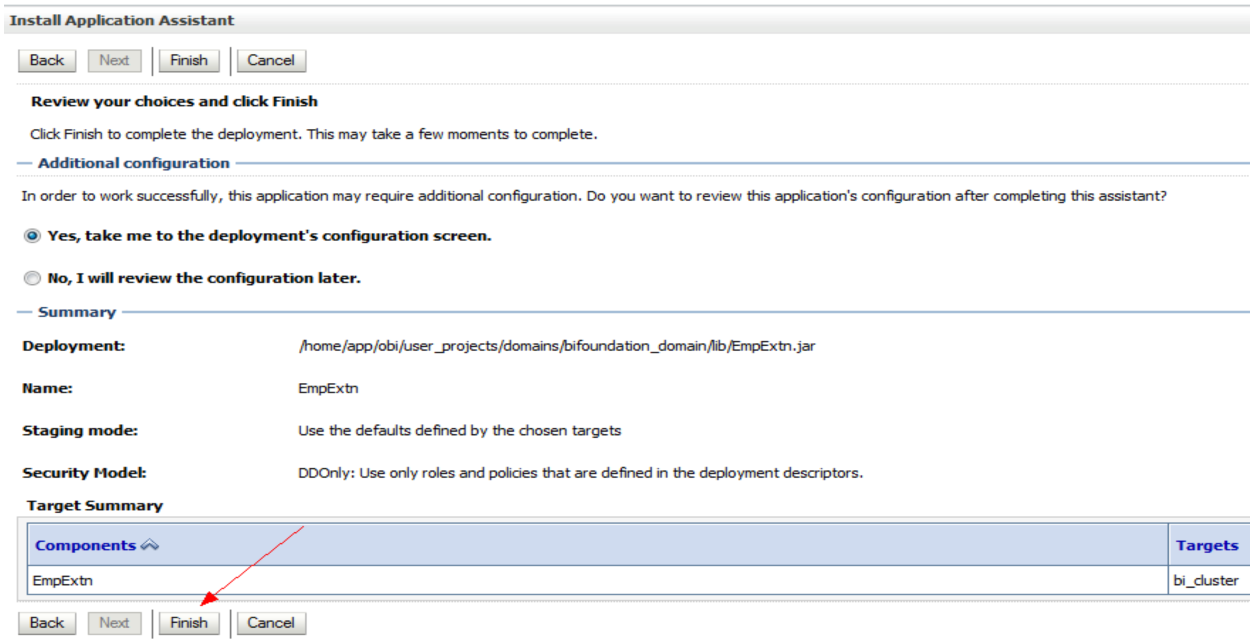
During deployment, the files will be copied automatically to the managed servers to which the application is targeted.

- **I will make the deployment accessible from the following location**

Location: /home/app/obi/user_projects/domains/bifoundation_

Figure 15. WebLogic Shared Library Extension/Library Name

Again default values are retained in the resulting screen and "Finish" button is clicked.


Install Application Assistant

Review your choices and click Finish

Click Finish to complete the deployment. This may take a few moments to complete.

Additional configuration

In order to work successfully, this application may require additional configuration. Do you want to review this application's configuration after completing this assistant?

[✓ Yes, take me to the deployment's configuration screen.]

No, I will review the configuration later.

Summary

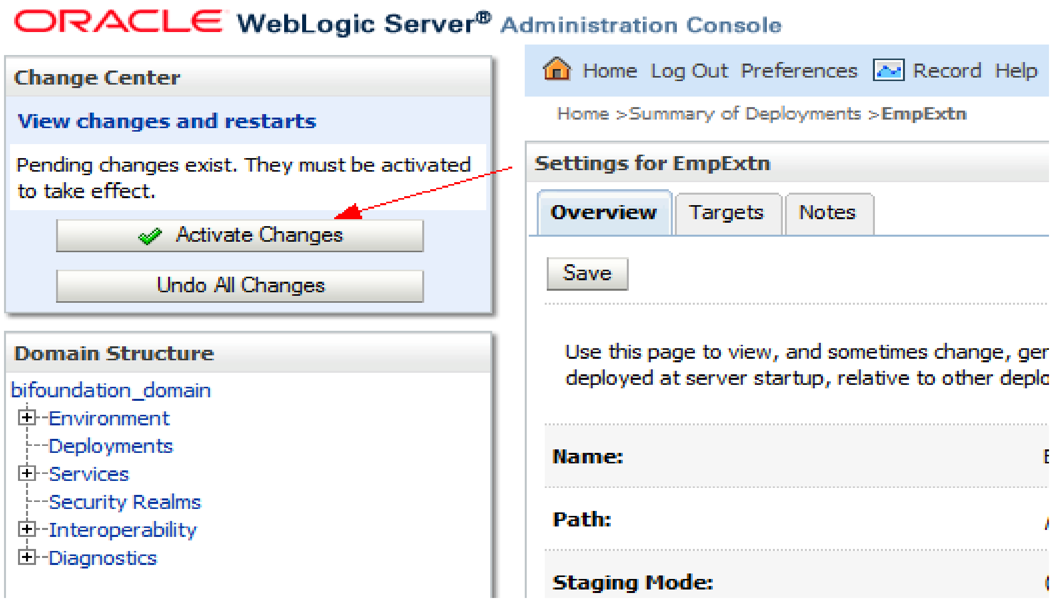
Deployment:	/home/app/obi/user_projects/domains/bifoundation_domain/lib/EmpExtn.jar
Name:	EmpExtn
Staging mode:	Use the defaults defined by the chosen targets
Security Model:	DDOnly: Use only roles and policies that are defined in the deployment descriptors.

Target Summary

Components	Targets
EmpExtn	bi_cluster

Figure 15. WebLogic Shared Library Completion

Once the deployment is completed, “Activate Changes” button needs to be clicked under “Change Center” area, to complete the library deployment.


ORACLE WebLogic Server® Administration Console
Change Center
View changes and restarts

Pending changes exist. They must be activated to take effect.

Use this page to view, and sometimes change, ger deployed at server startup, relative to other deplc

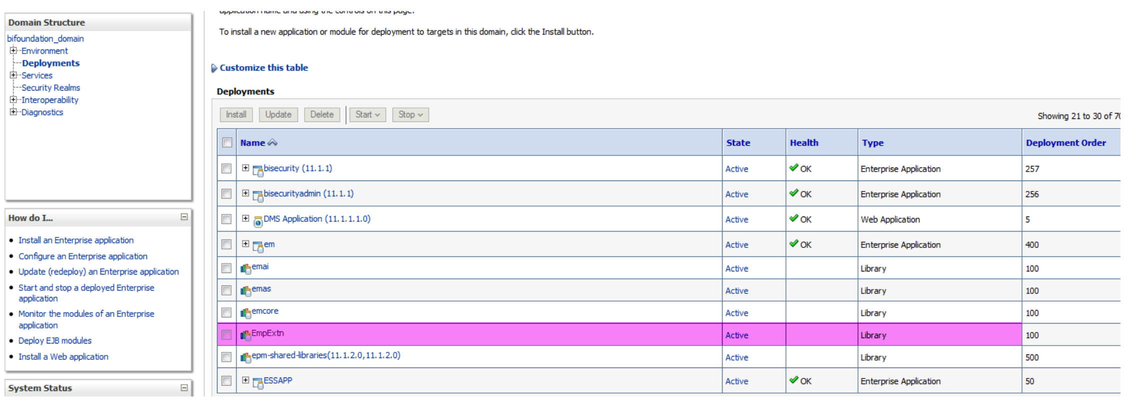
Name:

Path:

Staging Mode:

Figure 16. WebLogic Shared Library Activate Changes

Once successfully deployed the extension would be listed as a library under deployments as shown below:



Domain Structure

bifoundation_domain
- Environment
- Deployments
- Services
- Security Realms
- Interoperability
- Diagnostics

How do I...

- Install an Enterprise application
- Configure an Enterprise application
- Update (redeploy) an Enterprise application
- Start and stop a deployed Enterprise application
- Monitor the modules of an Enterprise application
- Deploy EJB modules
- Install a Web application

System Status

To install a new application or module for deployment to targets in this domain, click the Install button.

Customize this table

Deployments

Install | Update | Delete | Start | Stop

Showing 21 to 30 of 7

Name	State	Health	Type	Deployment Order
bisecurity (11.1.1)	Active	✓ OK	Enterprise Application	257
bisecurityadmin (11.1.1)	Active	✓ OK	Enterprise Application	256
DMS Application (11.1.1.1.0)	Active	✓ OK	Web Application	5
em	Active	✓ OK	Enterprise Application	400
emai	Active		Library	100
emas	Active		Library	100
emcore	Active		Library	100
EmpExtn	Active		Library	100
epm-shared-libraries(11.1.2.0,11.1.2.0)	Active		Library	500
ESSAPP	Active	✓ OK	Enterprise Application	50

Figure 17. WebLogic Shared Library Deployment Listing

Updating Extension Information for BI Publisher Application

This step is required only if *EmpExtn.jar* is deployed as a WebLogic shared library. Also, this process needs to be repeated every time a new patch is applied which updates the version of *xmlpserver.ear*.

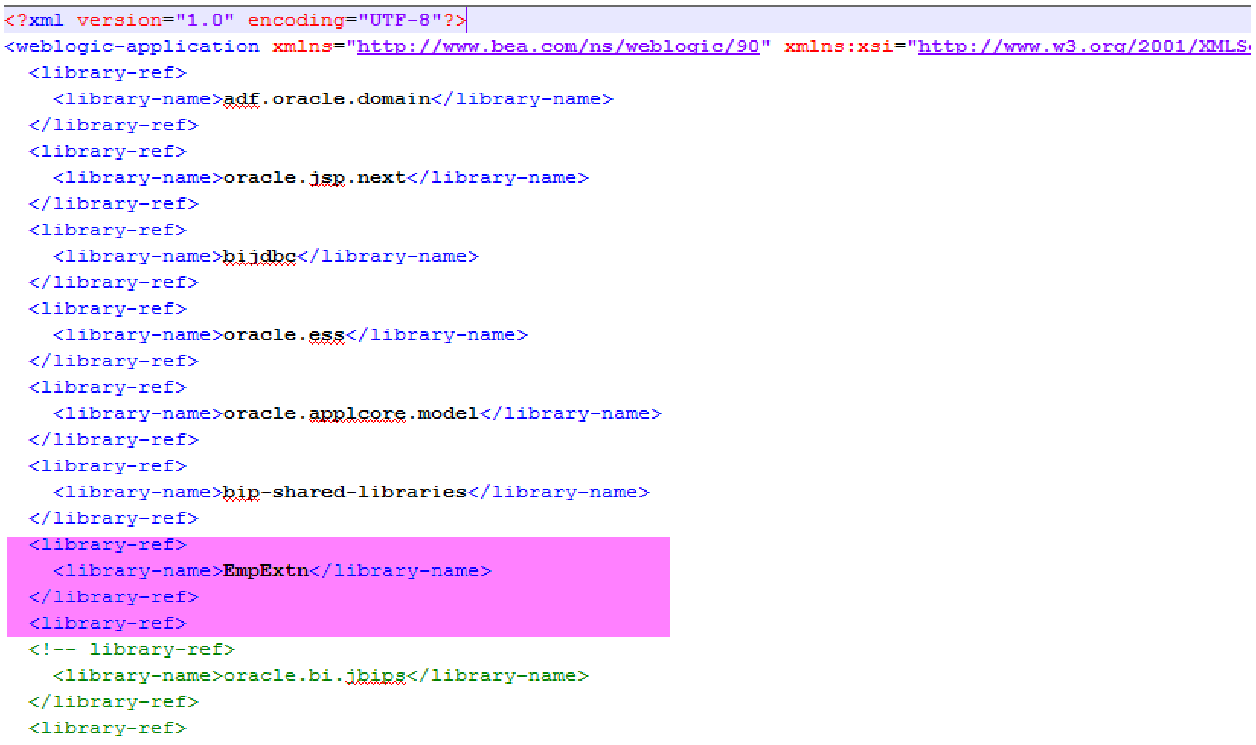
To make the BI Publisher application access the shared library, the extension related information needs to be entered for *xmlpserver.ear* file. This is similar to the way *EmpExtn.jar* is modified in the above steps. For this example 7-Zip utility is used to open the archive to edit the files inside.

The *xmlpserver.ear* file available under ***<Middleware_Home>*/Oracle_BI1/bifoundation/jee** (example: **/home/app/obi/Oracle_BI1/bifoundation/jee**) needs to be backed up first. This is to ensure that in case of any severe errors during deployment, we can always revert to the out of the box *xmlpserver.ear.*

Once the archive is opened, *weblogic-application.xml* which resides in *META-INF* directory of the *xmlpserver.ear* archive needs to be modified.

For *weblogic-application.xml*, it is better to extract the file, edit the file and then copy (drag back into the archive. Once the file is extracted, following lines need to be added, so that the library reference is made:

<library-ref>
<library-name>EmpExtn</library-name>
</library-ref>



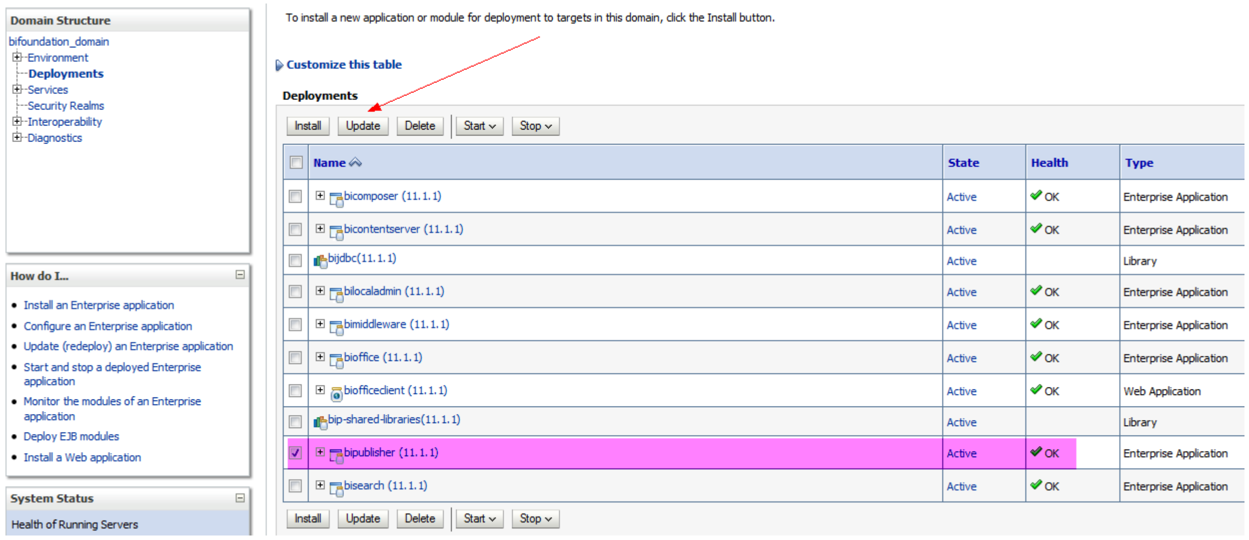
```
<?xml version="1.0" encoding="UTF-8"?>
<weblogic-application xmlns="http://www.bea.com/ns/weblogic/90" xmlns:xsi="http://www.w3.org/2001/XMLS
  <library-ref>
    <library-name>adf.oracle.domain</library-name>
  </library-ref>
  <library-ref>
    <library-name>oracle.jsp.next</library-name>
  </library-ref>
  <library-ref>
    <library-name>bijdbc</library-name>
  </library-ref>
  <library-ref>
    <library-name>oracle.ess</library-name>
  </library-ref>
  <library-ref>
    <library-name>oracle.applcore.model</library-name>
  </library-ref>
  <library-ref>
    <library-name>bip-shared-libraries</library-name>
  </library-ref>
  <library-ref>
    <library-name>EmpExtn</library-name>
  </library-ref>
  <library-ref>
  <!-- library-ref>
    <library-name>oracle.bi.jbips</library-name>
  </library-ref>
  <library-ref>
```

Figure 18. Adding the Shared Library (*EmpExtn*) to Library Reference Listing

Updating (re-deploying) BI Publisher Application

This step is required only if *EmpExtn.jar* is deployed as a WebLogic shared library.

The BI Publisher application needs to be updated through WebLogic Admin Console. Once logged into the admin console, "Lock & Edit" button needs to be clicked under "Change Center" area. After this "bipublisher" application needs to be selected by clicking on "Deployments" under "Domain Structure". After this "Update" button needs to be clicked to re-deploy the application.


Domain Structure
bifoundation_domain
Environment
Deployments
Services
Security Realms
Interoperability
Diagnostics
How do I...
Install an Enterprise application
Configure an Enterprise application
Update (redeploy) an Enterprise application
Start and stop a deployed Enterprise application
Monitor the modules of an Enterprise application
Deploy EJB modules
Install a Web application
System Status
Health of Running Servers
Customize this table
Install
Update
Delete
Start
Stop

To install a new application or module for deployment to targets in this domain, click the Install button.

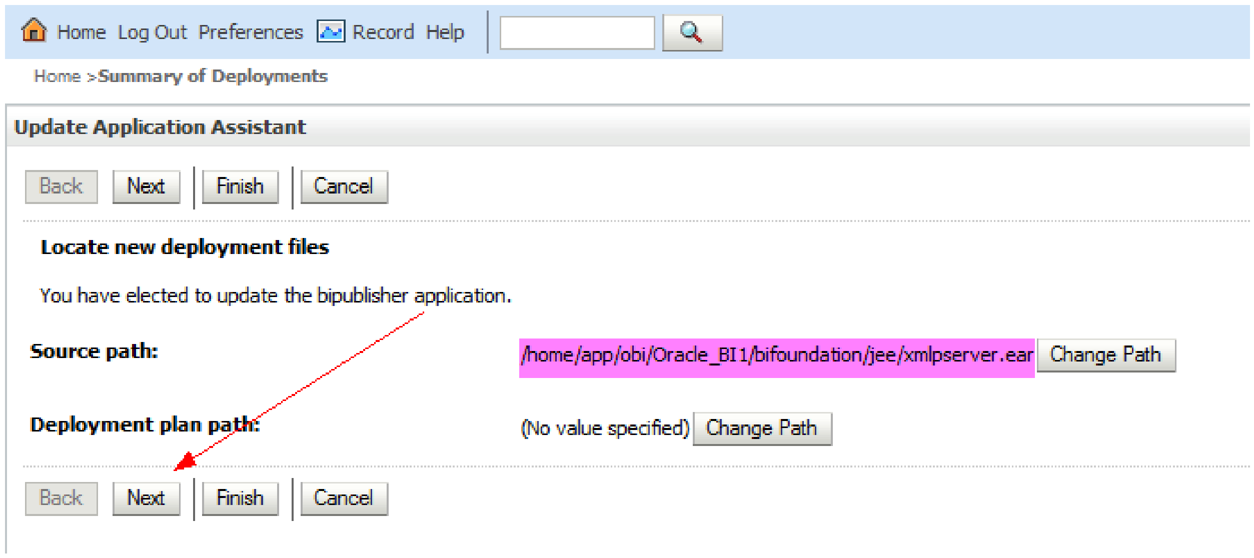
Deployments

	Name	State	Health	Type
	bicomposer (11.1.1)	Active	✓ OK	Enterprise Application
	bicontentserver (11.1.1)	Active	✓ OK	Enterprise Application
	bijdbc(11.1.1)	Active		Library
	bilocaladmin (11.1.1)	Active	✓ OK	Enterprise Application
	bimiddleware (11.1.1)	Active	✓ OK	Enterprise Application
	bioffice (11.1.1)	Active	✓ OK	Enterprise Application
	biofficeclient (11.1.1)	Active	✓ OK	Web Application
	bip-shared-libraries(11.1.1)	Active		Library
✓	bipublisher (11.1.1)	Active	✓ OK	Enterprise Application
	bisearch (11.1.1)	Active	✓ OK	Enterprise Application

Figure 19. Update BI Publisher Application

In the resulting screen, "Source path" field needs to point to :

***<Middleware_Home>*/Oracle_BI1/bifoundation/jee/xmlpserver.ear** (example: **/home/app/obi/Oracle_BI1/bifoundation/jee/xmlpserver.ear**) and Next button needs to be clicked. If the "Source path" is not correct then this can be changed by clicking on "Change Path" button.



Home >Summary of Deployments

Update Application Assistant

Locate new deployment files

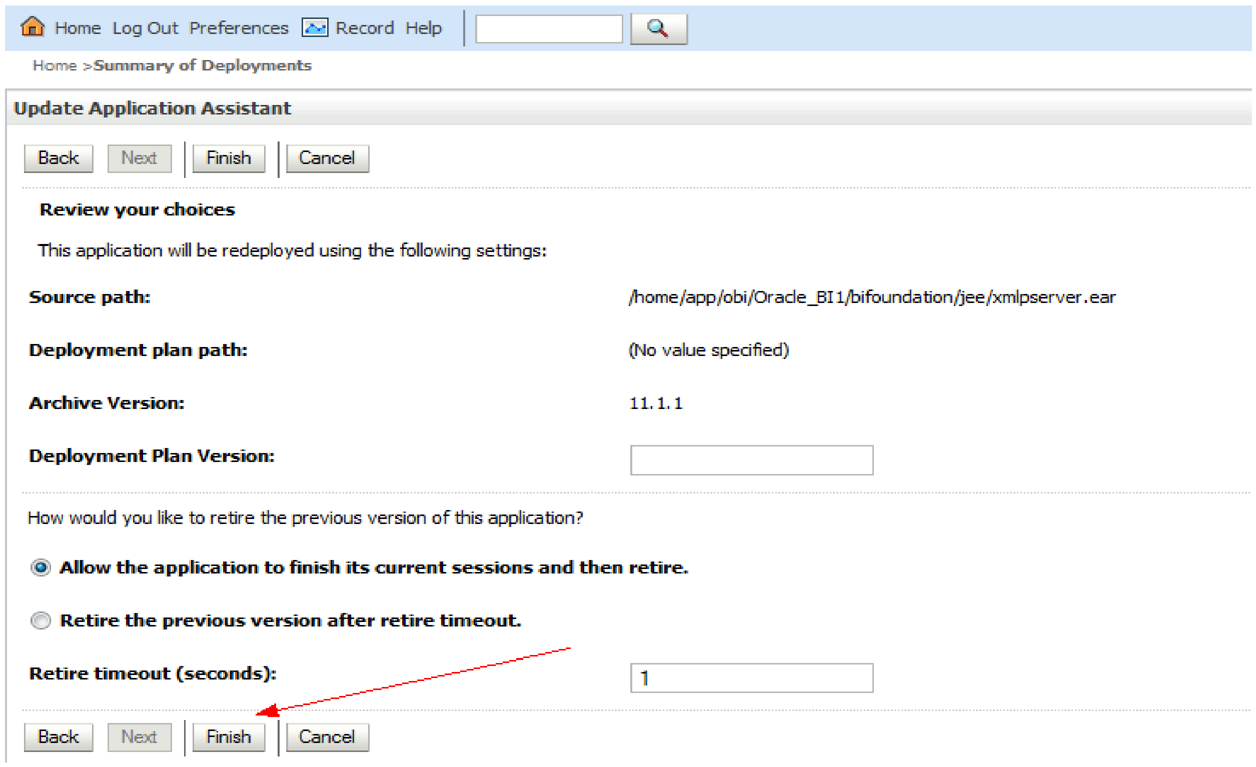
You have elected to update the bipublisher application.

Source path: /home/app/obi/Oracle_BI1/bifoundation/jee/xmlpserver.ear

Deployment plan path: (No value specified)

Figure 20. Select xmlpserver.ear File

The last step is to leave the default values and click on "Finish" button.



Home >Summary of Deployments

Update Application Assistant

Review your choices

This application will be redeployed using the following settings:

Source path: /home/app/obi/Oracle_BI1/bifoundation/jee/xmlpserver.ear

Deployment plan path: (No value specified)

Archive Version: 11.1.1

Deployment Plan Version:

How would you like to retire the previous version of this application?

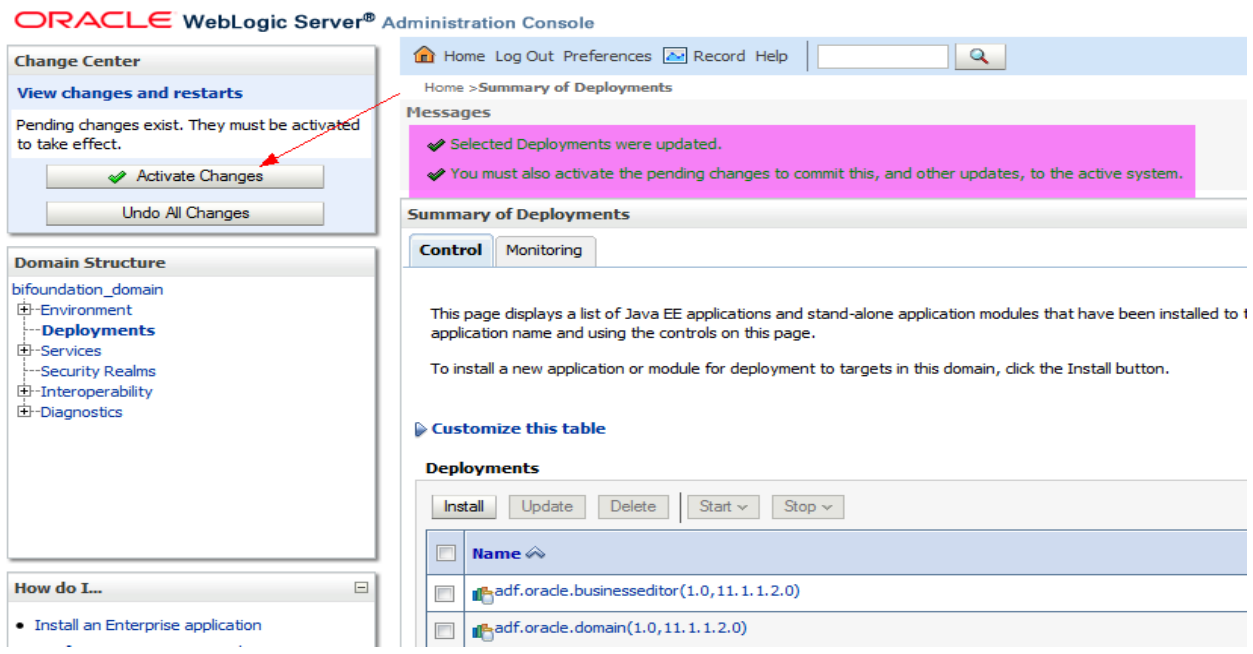
[✓ Allow the application to finish its current sessions and then retire.]

Retire the previous version after retire timeout.

Retire timeout (seconds): 1

Figure 22. Finish Updating BI Publisher Application

If there are no errors, then the deployment would complete and displays the screen similar to the following. To complete the deployment "Activate Changes" button needs to be clicked.



Pending changes exist. They must be activated to take effect.

✓ Selected Deployments were updated.

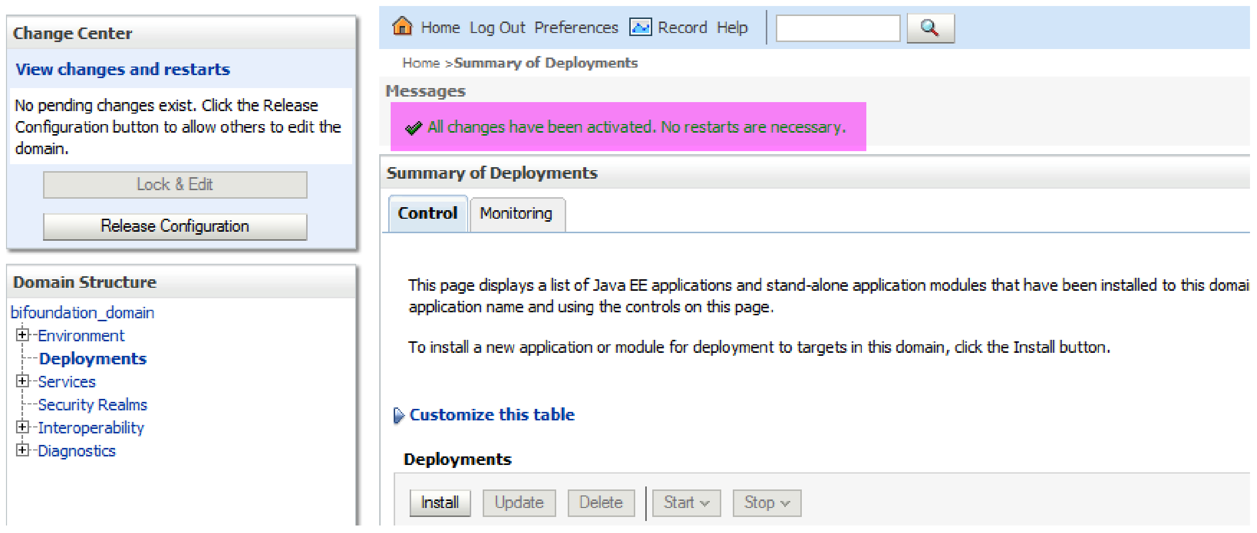
✓ You must also activate the pending changes to commit this, and other updates, to the active system.

This page displays a list of Java EE applications and stand-alone application modules that have been installed to t application name and using the controls on this page.

To install a new application or module for deployment to targets in this domain, click the Install button.

Figure 23. Activate the Updated BI Publisher Application

If all goes well the following screen would be displayed and this would complete the modification and re-deployment of "bipublisher" application (*xmlpserver.ear*).



No pending changes exist. Click the Release Configuration button to allow others to edit the domain.

✓ All changes have been activated. No restarts are necessary.

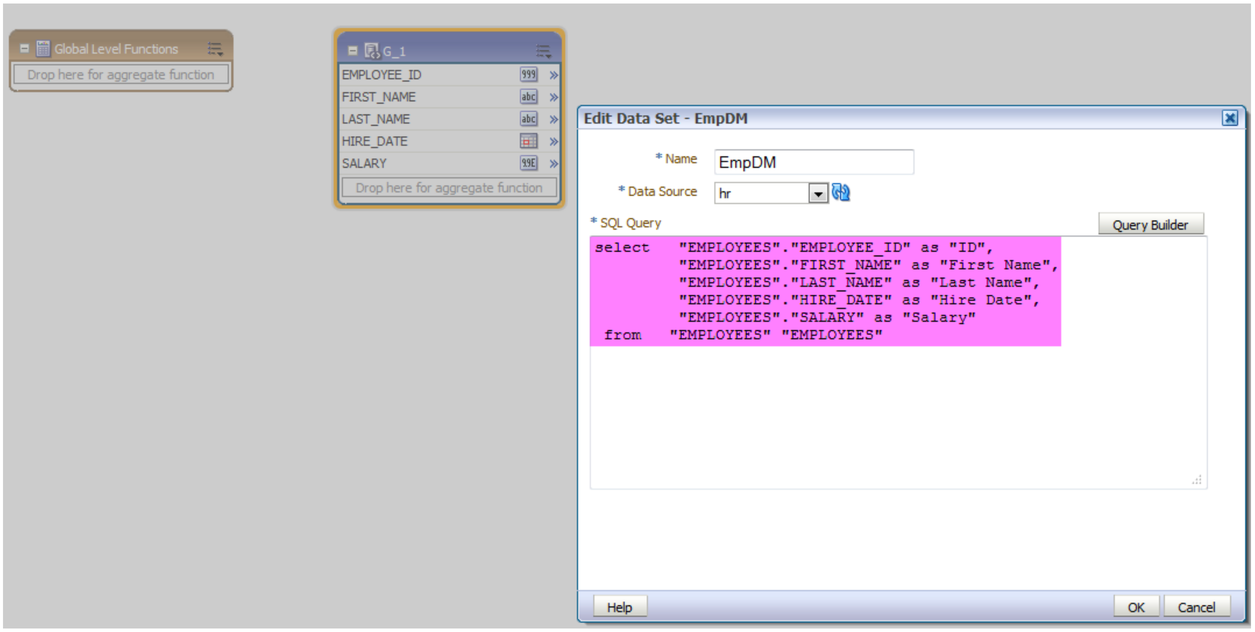
This page displays a list of Java EE applications and stand-alone application modules that have been installed to this domai application name and using the controls on this page.

To install a new application or module for deployment to targets in this domain, click the Install button.

Figure 24. Activation Complete

Building the Sample Report

As mentioned in the About the Sample Used section, for the data model Oracle Database sample schema "HR" is used. The BI Publisher data model is built using the following query:



Name: EmpDM
Data Source: hr

```
select    "EMPLOYEES"."EMPLOYEE_ID" as "ID",
          "EMPLOYEES"."FIRST_NAME" as "First Name",
          "EMPLOYEES"."LAST_NAME" as "Last Name",
          "EMPLOYEES"."HIRE_DATE" as "Hire Date",
          "EMPLOYEES"."SALARY" as "Salary"
 from    "EMPLOYEES" "EMPLOYEES"
```

Figure 25. BI Publisher Data Model

Once the above data model is built and sample data is saved, the next step is to build the RTF template.

Before making reference to the custom Java function in the RTF template, the XML namespace needs to be declared. In this sample the namespace is declared in the form field with the following tag:

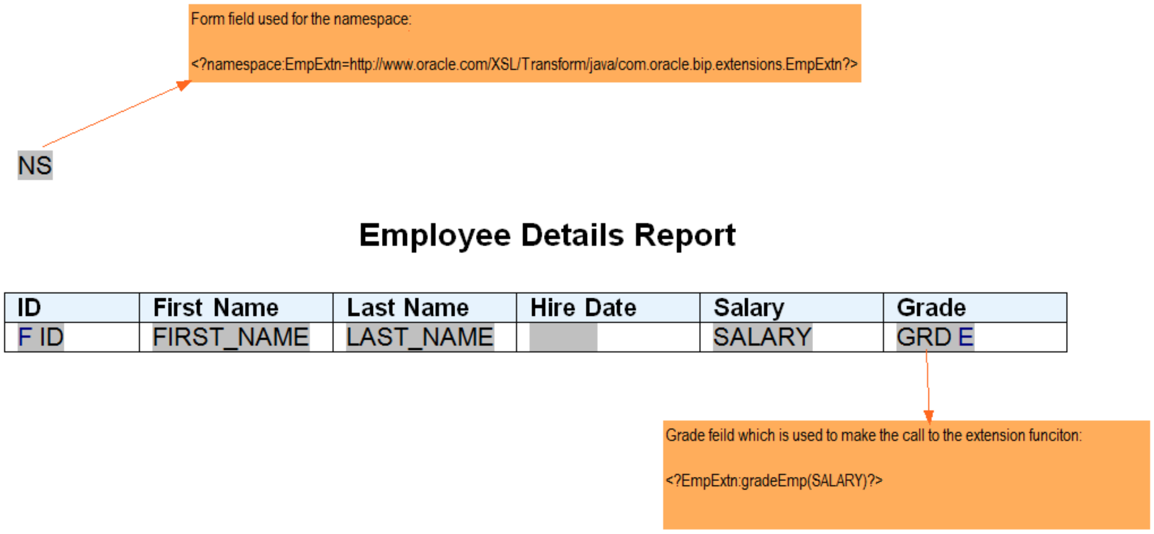
<?namespace:EmpExtn=http://www.oracle.com/XSL/Transform/java/com.oracle.bip.extensions.EmpExtn?>

Note that the extension name needs to be declared right after "namespace:" (for this sample it is "EmpExtn"). This is followed by **http://www.oracle.com/XSL/Transform/java/** and appended with package name with class name at the end. In this sample the package name is "*com.oracle.bip.extensions*" and the Java class name is "*EmpExtn*".

Once the namespace is declared, calls can be made to the respective functions. For this sample, apart from the columns selected in the data model, an additional field called "Grade" is added. The "Grade" field is used for making the call to the "*gradeEmp*" function. The syntax to call the extended function is:

namespace:function_name([Parameter1],[Parameter 2],….[Parameter n])

Actual tag used in this sample: **<?EmpExtn:gradeEmp(SALARY)?>**



Form field used for the namespace:

<?namespace:EmpExtn=http://www.oracle.com/XSL/Transform/java/com.oracle.bip.extensions.EmpExtn?>

NS

Employee Details Report

ID	First Name	Last Name	Hire Date	Salary	Grade
F ID	FIRST_NAME	LAST_NAME		SALARY	GRD E

Grade feild which is used to make the call to the extension funciton:

<?EmpExtn:gradeEmp(SALARY)?>

Figure 26. Bi Publisher Sample Template

Once the template is tested and ready, the report needs to be created in the BI Publisher and the tested template needs to be uploaded to the report. Before executing the report, the run time property of ***Disable External References*** needs to be set to false. Setting "Disable External References" to "False" allows sub-templates and Java extensions to be called through the templates.

The first step is to click on Properties as shown below:

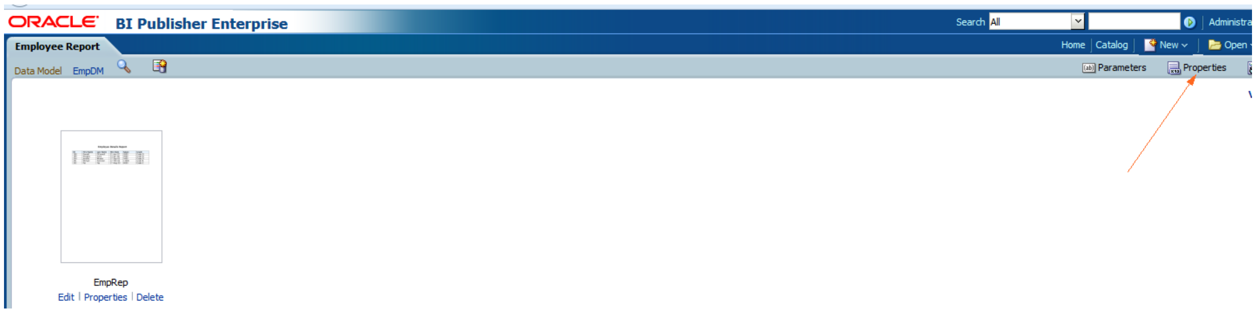
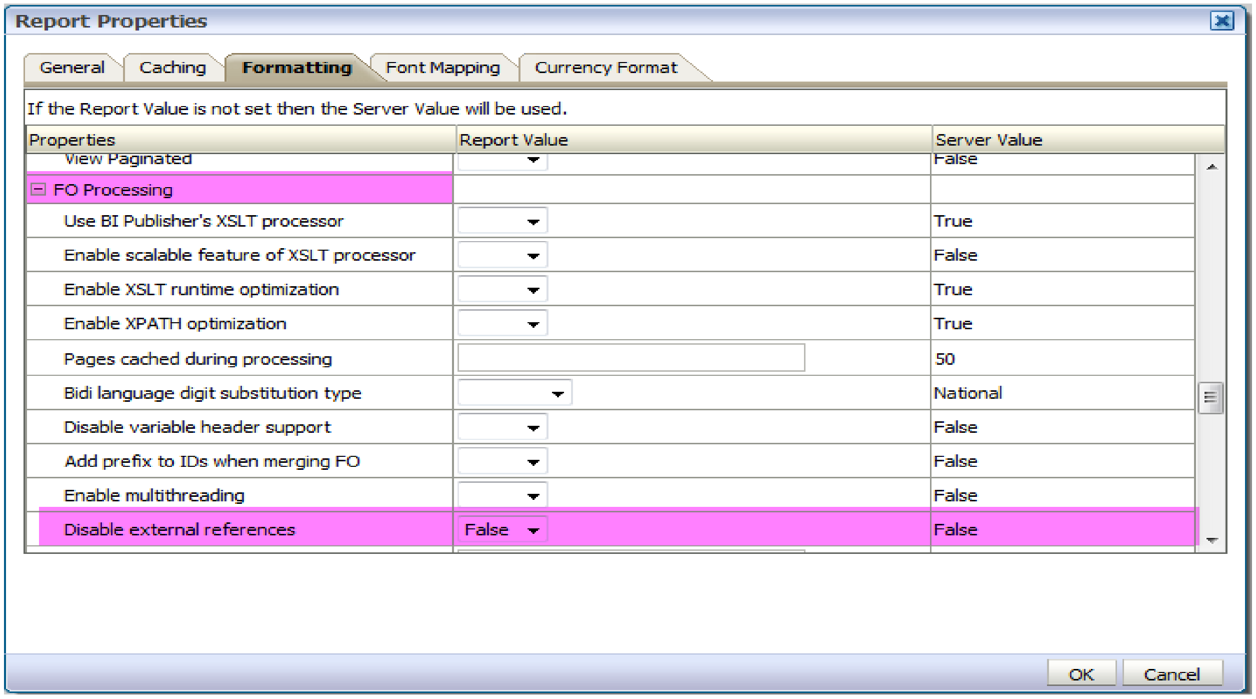

ORACLE BI Publisher Enterprise
Search All
Administra
Employee Report
Home | Catalog | New | Open
Data Model EmpDM
Parameters
Properties
EmpRep
Edit | Properties | Delete

Figure 27. Report Properties

The next step is to click on "Formatting" tab and change the value to "False" (for "Disable External References")under "FO Processing" section as shown below:

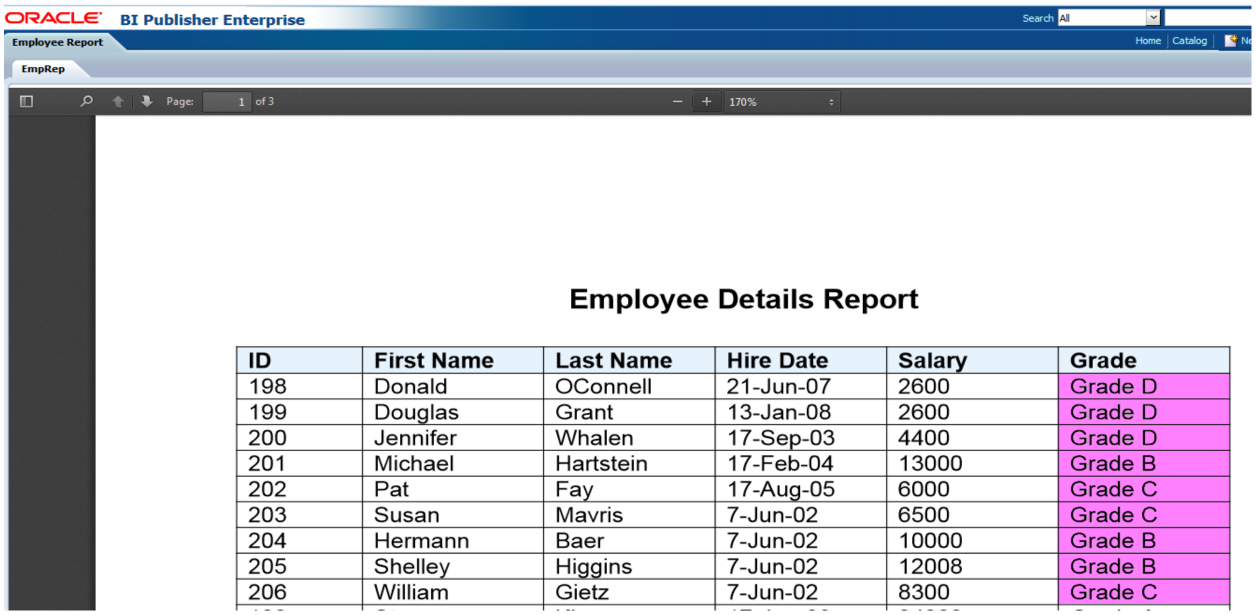


If the Report Value is not set then the Server Value will be used.

Properties	Report Value	Server Value
View Paginated		False
FO Processing		
Use BI Publisher's XSLT processor		True
Enable scalable feature of XSLT processor		False
Enable XSLT runtime optimization		True
Enable XPATH optimization		True
Pages cached during processing		50
Bidi language digit substitution type		National
Disable variable header support		False
Add prefix to IDs when merging FO		False
Enable multithreading		False
Disable external references	False	False

Figure 28. Runtime Formatting Properties

At this time the report can be executed and if everything is done correctly, the result would be something similar to the following:



Employee Details Report

ID	First Name	Last Name	Hire Date	Salary	Grade
198	Donald	OConnell	21-Jun-07	2600	Grade D
199	Douglas	Grant	13-Jan-08	2600	Grade D
200	Jennifer	Whalen	17-Sep-03	4400	Grade D
201	Michael	Hartstein	17-Feb-04	13000	Grade B
202	Pat	Fay	17-Aug-05	6000	Grade C
203	Susan	Mavris	7-Jun-02	6500	Grade C
204	Hermann	Baer	7-Jun-02	10000	Grade B
205	Shelley	Higgins	7-Jun-02	12008	Grade B
206	William	Gietz	7-Jun-02	8300	Grade C

Figure 29. Sample Report Output

Conclusion

By utilizing the powerful feature of Java extensions for BI Publisher, the power of Java programming language can be added to the BI Publisher customizations. The centralized libraries built and deployed into WebLogic, could be used across templates and reports. This would ease the maintenance of complex business logic and increase the productivity.

Oracle consulting has had many experiences in implementing custom Java extensions for BI Publisher. If you are interested in Oracle consulting to discuss more in detail about the implementation and review of your reporting or customization needs, please contact Shankar Duvvuri (shankar.duvvuri@oracle.com), Senior Principal Consultant, Oracle Business Intelligence group.

ORACLE

Oracle is committed to developing practices and products that help protect the environment

BI Publisher 11g Custom Java Extensions
May 2014
Author: Shankar Duvvuri

Oracle Corporation
World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065
U.S.A.

Worldwide Inquiries:
Phone: +1.650.506.7000
Fax: +1.650.506.7200
oracle.com

SOFTWARE. HARDWARE. COMPLETE.