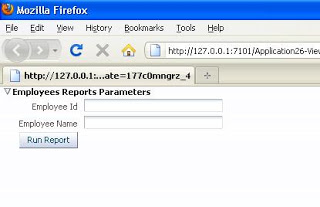
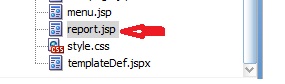
**Call Oracle Reports From Your ADF Application**

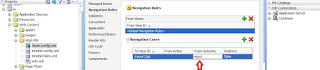
One of the main issues that can face oracle ADF Developers is calling oracle reports.  
This post will illustrate step by step how to call oracle reports.  
  
1- Create new ADF Application and make new .jspx page (this page will be a parameter page which will send the parameters to the report).  
  
2- Insert inside page inputTexts which will hold the parameters and insert one commandButton which will run the report.  
  
[](http://4.bp.blogspot.com/_gtyEuru9PjQ/TE19a5Bt3aI/AAAAAAAAAF4/g6wXfFX4JAc/s1600/report2.JPG)  
  
3- Bind the command button to the method in backbean and write this code:  
OracleReportBean reportBean =  
new OracleReportBean("oracleServerId", "Port", null);  
reportBean.setReportServerParam(OracleReportBean.RS\_PARAM\_DESTYPE,"DisplayType");// which will be one of the [cashe - file - mail - printer]  
reportBean.setReportServerParam(OracleReportBean.RS\_PARAM\_DESFORMAT,"Format"); // Which will be onr of the [HTML - HTML CSS - PDF - SPREADSHEET- RTF].  
reportBean.setReportParameter(parameterName1, parameterValue1);// as report name  
reportBean.setReportParameter(parameterName2, parameterValue2);  
reportBean.setReportParameter(parameterName3, parameterValue3);  
String url = reportBean.getReportServerURL();  
reportBean.openUrlInNewWindow(url);  
as in picture  
[](http://3.bp.blogspot.com/_gtyEuru9PjQ/TE176Fk_mpI/AAAAAAAAAFw/Tf_Fnyx2oGM/s1600/report1.JPG)  
4- Make a new java file which will be OracleReportBean (assume its name will be OracleReportBean.java) and write this code inside OracleReportBean.java:  
  
import java.util.HashMap;  
import java.util.Iterator;  
  
import javax.faces.context.FacesContext;  
  
import org.apache.myfaces.trinidad.render.ExtendedRenderKitService;  
import org.apache.myfaces.trinidad.util.Service;  
  
  
/\*\*\*  
\* This bean performs the following functions:  
\* Constructing the Report URL with various parameters passed in by client.  
\* Sending the report request.  
\* Execute printed report and retrieve return status in the indicated format.  
\* Execute a binary report (PDF, RTF, XML, with DESTYPE CACHE) and return the  
\* InputStream for processing.  
\*/  
public class OracleReportBean {  
  
/\* Report Servlet Host Settings \*/  
private String http\_host = null;  
private String http\_port = null;  
  
/\* Default path as per generic Oracle Appserver install \*/  
private String servlet\_path = "/reports/rwservlet";  
  
/\* Report Servlet URL params \*/  
public static final String RS\_PARAM\_SERVER = "server";  
public static final String RS\_PARAM\_USERID = "userid";  
public static final String RS\_PARAM\_REPORT = "report";  
public static final String RS\_PARAM\_ENVID = "envid";  
public static final String RS\_PARAM\_DESTYPE = "destype";  
public static final String RS\_PARAM\_DESFORMAT = "desformat";  
public static final String RS\_PARAM\_STATUSFORMAT = "statusformat";  
public static final String RS\_PARAM\_DESNAME = "desname";  
public static final String RS\_PARAM\_PAGESTREAM = "pagestream";  
public static final String RS\_PARAM\_DELIMITER = "delimiter";  
public static final String RS\_PARAM\_ORIENTATION = "orientation";  
public static final String RS\_PARAM\_DISTRIBUTE = "distribute";  
public static final String RS\_PARAM\_FILETYPE = "FILETYPE";  
public static final String RS\_PARAM\_REP\_VA = "REP\_VA";  
  
  
private String value\_keyMap = null;  
  
/\* Static values for destination formats \*/  
public static final String DESFORMAT\_PDF = "PDF";  
public static final String DESFORMAT\_HTML = "HTML";  
public static final String DESFORMAT\_POSTSCRIPT = "POSTSCRIPT";  
public static final String DESFORMAT\_DELIMITED = "DELIMITED";  
public static final String DESFORMAT\_XML = "XML";  
public static final String DESFORMAT\_RTF = "RTF";  
  
/\* Static values for destination types\*/  
public static final String DESTYPE\_MAIL = "mail";  
public static final String DESTYPE\_PRINTER = "printer";  
public static final String DESTYPE\_FILE = "file";  
public static final String DESTYPE\_LOCAL\_FILE = "localFile";  
public static final String DESTYPE\_CACHE = "cache";  
  
/\* Static values for distribute \*/  
public static final String DISTRIBUTE\_YES = "YES";  
public static final String DISTRIBUTE\_NO = "NO";  
  
/\*Static values for status format \*/  
public static final String STATUSFORMAT\_XML = "XML";  
public static final String STATUSFORMAT\_HTML = "HTML";  
  
/\* Static values for report orientation \*/  
public static final String ORIENTATION\_PORTRAIT = "PORTRAIT";  
public static final String ORIENTATION\_LANDSCAPE = "LANDSCAPE";  
public static final String ORIENTATION\_DEFAULT = "DEFAULT";  
  
/\* HashMap to hold individual report parameters\*/  
private HashMap reportParams = new HashMap();  
  
/\* HashMap to hold report server params \*/  
private HashMap reportServerParams = new HashMap();  
  
/\* Report Servlet \*/  
private StringBuffer reportURL = new StringBuffer();  
// private String XMLReturnStatus = null;  
  
  
/\*\*\*  
\* Constructor  
\*/  
public OracleReportBean(String p\_http\_host, String p\_http\_port,  
String p\_servlet\_path) {  
http\_host = p\_http\_host;  
http\_port = p\_http\_port;  
  
/\* If the servlet path is null, we assign the default path. \*/  
if (p\_servlet\_path != null) {  
servlet\_path = p\_servlet\_path;  
}  
  
/\* Default the status format to XML  
setReportServerParam(RS\_PARAM\_STATUSFORMAT,STATUSFORMAT\_XML);\*/  
  
}  
  
  
/\*\*\*\*\*  
\* Private utility methods ...  
\*  
\*/  
private String buildKeyValueString(HashMap p\_map) {  
  
String map\_key = null;  
String param\_sep = "&";  
String param\_equal = "=";  
StringBuffer keyValueBuffer = new StringBuffer();  
  
if (!p\_map.isEmpty()) {  
  
Iterator mapKeys = p\_map.keySet().iterator();  
  
while (mapKeys.hasNext()) {  
map\_key = (String)mapKeys.next();  
keyValueBuffer.append(map\_key).append(param\_equal).append(p\_map.get(map\_key));  
  
if (mapKeys.hasNext()) {  
keyValueBuffer.append(param\_sep);  
}  
}  
}  
return keyValueBuffer.toString();  
}  
  
/\* Construct the URL for accessing the Oracle Reports Servlet \*/  
  
private void constructURL() {  
String param\_sep = "&";  
  
/\* Clearout current URL \*/  
reportURL = new StringBuffer();  
  
/\* HOST Section \*/  
  
reportURL.append("http://");  
reportURL.append(http\_host);  
  
if (http\_port != null) {  
reportURL.append(":").append(http\_port);  
}  
  
/\* Add "/" separator if necessary. \*/  
if (servlet\_path.indexOf("/") > 0) {  
reportURL.append("/");  
}  
  
reportURL.append(servlet\_path);  
reportURL.append("?");  
  
if (value\_keyMap != null) {  
reportURL.append(value\_keyMap).append(param\_sep);  
}  
  
/\*Construct Report Server Parameter URL component\*/  
reportURL.append(buildKeyValueString(reportServerParams));  
  
if (!reportServerParams.isEmpty()) {  
reportURL.append(param\_sep);  
}  
  
/\*Construct Report Parameters URL Component\*/  
reportURL.append(buildKeyValueString(reportParams));  
}  
  
/\*\*\*  
\* Getters and Setters for the Reports Servlet  
\* URL parameter values.  
\*/  
  
public void setReportServerParam(String p\_param, String p\_value) {  
reportServerParams.put(p\_param, p\_value);  
}  
  
public String getReportServerParam(String p\_param) {  
if (reportServerParams.containsKey(p\_param)) {  
return (String)reportServerParams.get(p\_param);  
} else {  
return null;  
}  
}  
  
/\* Set/Get the value of a Reports KeyMap file \*/  
  
public void setKeyMap(String p\_keyMap) {  
value\_keyMap = p\_keyMap;  
}  
  
public String getKeyMap() {  
return value\_keyMap;  
}  
  
/\* Add/Update and retrieve individual report parameters \*/  
  
public void setReportParameter(String paramName, String paramValue) {  
  
reportParams.put(paramName, paramValue);  
}  
  
public String getReportParameter(String paramName) {  
  
if (reportParams.containsKey(paramName)) {  
return (String)reportParams.get(paramName);  
} else {  
return null;  
}  
  
}  
  
/\*\*\*\*  
\* Construct and return a URL that can be used to execute the report.  
\*/  
public String getReportServerURL() {  
constructURL();  
return reportURL.toString();  
}  
  
public void openUrlInNewWindow(String url) {  
ExtendedRenderKitService erks =  
Service.getRenderKitService(FacesContext.getCurrentInstance(),  
ExtendedRenderKitService.class);  
StringBuilder strb = new StringBuilder("window.open('" + url + "');");  
erks.addScript(FacesContext.getCurrentInstance(), strb.toString());  
}  
}  
  
  
Oracle Report URL should be at this format  
http://myServer:myport/reports/rwservlet?destype=cache&desformat=PDF&report=myreport.jsp&myparameter=myvalue  
  
5- Save and Run the application.  
  
You can see other way to call oracle report from this link   
<http://sameh-nassar.blogspot.com/2012/05/call-oracle-reports-from-your-adf.html>

**Call Oracle Reports From Your ADF Application (Hide Report Parameters From URL)**

Hello ADF Developers,  
  
In my previous post <http://sameh-nassar.blogspot.com/2010/07/call-oracle-reports-from-your-adf.html> I explain a way to call oracle report but this way has a problem. The problem is the report parameters appear in the URL. In this post I will explain another way to call oracle report without displaying report parameters in URL.  
to call oracle report follow this steps:  
1- Make a new .jsp page (for example the page name is **report.jsp**).

[](http://2.bp.blogspot.com/-959ic_3Iksc/T7WDP-7baSI/AAAAAAAAAKg/tWwtQIEpS88/s1600/1.jpg)

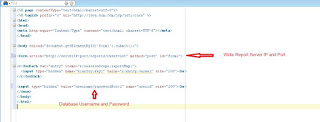
2- In faces-config.xml make a navigation rule to report.jsp with outcome "report".

[](http://3.bp.blogspot.com/-rZkugGWgOYg/T7WG1TWktlI/AAAAAAAAAKw/noAlnBhEaAI/s1600/3.jpg)

3- Write this code inside the report.jsp page

|  |  |
| --- | --- |
|  | <%@ page contentType="text/html;charset=utf-8"%> |
|  | <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %> |
|  | <html> |
|  | <head> |
|  | <meta http-equiv="Content-Type" content="text/html; charset=UTF-8"></meta> |
|  | </head> |
|  |  |
|  | <body onload="document.getElementById('form1').submit();"> |
|  |  |
|  | <form action="http://serverIp:port/reports/rwservlet" method="post" id="form1"> |
|  |  |
|  | <c:forEach var="entry" items="${sessionScope.reportMap}"> |
|  | <input type="hidden" name="${entry.key}" value="${entry.value}" size="100"><br> |
|  | </c:forEach> |
|  |  |
|  | <input type="hidden" value="username/password@orcl" name="userid" size="100"><br> |
|  | </form> |
|  | </body> |
|  | </html> |

[**view raw**](https://gist.github.com/anonymous/3b1801c09d765a3f646e/raw/dd233381981fa1b346abaebcde58e8205ead4e5a/report.jsp)[**report.jsp**](https://gist.github.com/anonymous/3b1801c09d765a3f646e#file-report-jsp) hosted with ❤ by **[GitHub](https://github.com/)**

[](http://4.bp.blogspot.com/-at4Jryht-iM/T7WE4P7l9kI/AAAAAAAAAKo/binKd426IZg/s1600/2.jpg)

4- Create ADF page (.jspx or .jsf page) then add button (this button will call the report) and in the button action write this code:

|  |  |
| --- | --- |
|  | Map map = new HashMap(); |
|  | map.put("parameter1", "value"); |
|  | map.put("parameter2", "value"); |
|  | map.put("report", "reportName.rdf");// name of the report |
|  | FacesContext.getCurrentInstance().getExternalContext().getSessionMap().put("reportMap", map); |
|  | return "report";// report outcome[http://4.bp.blogspot.com/-depvQUL1Y10/T7WI1PSl0PI/AAAAAAAAAK4/Mj2axbv6c5Q/s320/4.jpg](http://4.bp.blogspot.com/-depvQUL1Y10/T7WI1PSl0PI/AAAAAAAAAK4/Mj2axbv6c5Q/s1600/4.jpg) |

Prerequisite 1. Oracle Application server 11g (forms, report) Steps 1. Create rdf, jsp file 2. configure cgicmd.dat file location C:\Oracle\Middleware\user\_projects\domains\ClassicDomain\ config\fmwconfig\servers\WLS\_REPORTS\applications\ reports\_11.1.2\configuration\cgicmd.dat add line at last keemp: userid=scott/tiger@xe server=RS\_MIS\_HVY3 desformat=pdf destype=cache paramform=no %\* restart report server 3. Create url for Calling report [http://ip:8888/reports/rwservlet?keem...](https://www.youtube.com/redirect?v=7boFh6A8lEA&event=video_description&redir_token=X-TRjLdnnkOD15iA871Zsi1Qpsx8MTU0MjA4NjUzOEAxNTQyMDAwMTM4&q=http%3A%2F%2Fip%3A8888%2Freports%2Frwservlet%3Fkeemp%2Breport%3DD%3A)\emp.rdf+P\_PID=179290 [http://ip:8888/reports/rwservlet?](https://www.youtube.com/redirect?v=7boFh6A8lEA&event=video_description&redir_token=X-TRjLdnnkOD15iA871Zsi1Qpsx8MTU0MjA4NjUzOEAxNTQyMDAwMTM4&q=http%3A%2F%2Fip%3A8888%2Freports%2Frwservlet%3F) keemp+report=file location .rdf or .jsp+parameter [http://192.168.10.222:8888/reports/rw...](https://www.youtube.com/redirect?v=7boFh6A8lEA&event=video_description&redir_token=X-TRjLdnnkOD15iA871Zsi1Qpsx8MTU0MjA4NjUzOEAxNTQyMDAwMTM4&q=http%3A%2F%2F192.168.10.222%3A8888%2Freports%2Frwservlet%3Fkeemp%2Breport%3DD%3A)\emp.rdf 4. Call Report from apex button