

# BIRT Application and BIRT Report Deployment Functional Specification

Version 1: October 6, 2005

## Abstract

*This document describes how the user will deploy a BIRT Application and BIRT reports to the Application Server.*

## Document Revisions

Version	Date	Description of Changes
Draft 1	9/6/2005	Initial Draft
Draft 2	9/26/2005	Restructured the document
Version 1	10/6/2005	Revised the document after the review meeting

## Contents

- 1. Introduction.....2**
- 2. BIRT Report Deployment.....2**
  - 2.1 Deployment Folder.....2
  - 2.2 Report Deployment.....3
  - 2.3 Resource Search Rule.....3
- 3. BIRT Application Deployment.....3**
  - 3.1 Put new Plugins in a proper place.....4
  - 3.2 Configure BIRT Application before the Deployment.....4
  - 3.3 Deploying BIRT Application.....4
- 4. Special BIRT Development under WAR Deployment.....4**
  - 4.1 BIRT Plugin Development under WAR Deployment.....4



## 1. Introduction

There are two types of deployment tasks in BIRT: BIRT Application deployment and BIRT report deployment. The deployment of BIRT Application and BIRT reports are made separate, because BIRT Applications are developed based on BIRT Report Engine and change infrequently after the development, while BIRT reports are only the input to the application and tend to change more frequently. With this separation, user is free to deploy BIRT reports without touching the deployed BIRT Application.

This document covers both types of the deployment tasks. From the deployment viewpoint, BIRT users are roughly divided into three groups:

1. Report User whose task is to design and deploy BIRT reports.
2. Administrator whose task is to deploy BIRT Application into application server and config it properly.
3. BIRT Developer whose task is to extend BIRT by developing new BIRT plugins or writing BIRT Applications based on BIRT Report Engine.

## 2. BIRT Report Deployment

In the development environment, BIRT reports files as well as other resources such as images, Jar files, Java script files, libraries and property files, are organized into a report projects. Report project is a concept that is only relevant to the Eclipse-based environment, and does not correspond to a physical entity in a deployed environment. Instead, BIRT report design and other related resource files are stored into folders on the deployment machine. BIRT does not require a specific folder structure.

BIRT report engine APIs requires that the requests pass in full path for a report design or report document file. Design engine loads resources that are referenced by the design file based on a resource search rule.

### 2.1 Deployment Folder

In general, user can follow the following guidelines to create his own deployment folder structure:

1. If a resource is only used by one report design, it is recommended that the resource is embedded into the report design. For example, if an image is only used in one report, it is recommended to put it into the report design.
2. If a resource is shared by several report designs in one deployment folder, the resource could be put outside the report designs but inside the deployment folder. It's recommended to create a separate sub-folder for each kind of resources for ease of maintenance. For example, if an image is used in several reports, user could create an "images" folder under the deployment folder and put the image there. User specifies the relative path to the image in reports that use the image.
3. If a resource is to be used by reports in several deployment folders, user can either put the resource into a BIRT library or a common place which can be accessed from reports in different deployment folders.

- If the shared resource does not change frequently, the resource should be packaged into a BIRT library. User will then include the library into different deployment folders
- If the shared resource will be changed frequently, the resource could be put into a common place which can be accessed by different deployment folders. A common way to do it is to create a “shared” deployment folder, put the shared resource in it, and reference it with relative paths in different deployment folders.

## 2.2 Report Deployment

For a typical report deployment task, user needs to conduct the following steps:

1. The report user uses the tool of his choice to copy the report designs and related files to the deployment environment. He should make sure that the relative paths in the report design can continue to work.
2. User needs to config his BIRT Application properly such that it is able to locate the new reports. For example, in BIRT Viewer, user needs to set BIRT\_VIEWER\_REPORT\_ROOT to the root location of report designs.
3. If the deployed report project doesn't contain all the resources it references, Design engine will load resources based on the resource search rule at runtime. User needs to config the deployment environment properly so that DE can find the resources at runtime.

## 2.3 Resource Search Rule

BIRT uses a resource search algorithm to locate any external resource referenced by the report design. Different kinds of resources can have different resource search rules. The detailed resource search rules are defined in the corresponding BIRT specifications respectively. See the following table for details.

Resource Type	BIRT Specification
JAR File	Scripting Specification
JavaScript File	Scripting Specification
Library	Library Specification
Image	Design Engine Search Path
Property File	Design Engine Search Path

## 3. BIRT Application Deployment

This section is mainly for BIRT Administrators. When deploying a BIRT application, a BIRT Administrator usually needs to do the followings:

- Put new plugins in a proper place
- Configure the BIRT Application properly
- Deploy the BIRT Application to the Application Server

### 3.1 Put new Plugins in a proper place

Extensibility is one of the great features of BIRT. User is able to develop his own plugins to extend the functionalities provided by BIRT Engine, just as writing a new plugin in Eclipse. For example, user could develop a new BIRT emitter which converts BIRT report into Excel format.

Similar to Eclipse, in order for BIRT platform to recognize the new plugins provided by user, BIRT requires the user to put his new plugins into a special “plugins” folder from which BIRT Engine will recognize the new plugins at startup and load them when needed at runtime.

So, in the deployment of the BIRT Application, BIRT Administrator is required to put all the newly developed plugins into the BIRT Application’s plugins folder on the deployment machine. For example, BIRT viewer requires BIRT Administrator to put the new plugins under folder WEB-INF\plugins.

### 3.2 Configure BIRT Application before the Deployment

BIRT Application usually specifies some configuration variables to allow BIRT Administrator to customize its behaviors. Although most of the configuration variables have default value, it is highly recommended that the BIRT Administrator reads the configuration document of the BIRT Application he’s working with, and sets those configuration variables properly. For most of the BIRT Applications, the configuration variables are in the file web.xml under folder WEB-INF.

### 3.3 Deploying BIRT Application

Deploying a BIRT Application is not much different from deploying other web applications. Eclipse has a Web Tools Project which provides a framework for developing, deploying, testing and debugging J2EE applications on J2EE Server. Eclipse WTP supports most of the popular Application Servers such as Tomcat, WebLogic and Websphere, and can handle the deployment of a web application automatically. For detailed information on Eclipse WTP, please refer to <http://www.eclipse.org/webtools/>.

If user doesn’t use Eclipse WTP, he can use the deployment tools provided by the application server or even copy files directly to the application server.

## 4. Special BIRT Development under WAR Deployment

User can choose to deploy a BIRT Application by using either file deployment or WAR deployment approach. When deploying BIRT Application as a WAR file, there are some special requirements for BIRT plugin development. This section talks about those special requirements.

The target audience for this part is BIRT Developers. Hence, it contains technical details at the implementation level. You may skip this section if you are a BIRT Administrator or a BIRT Report User.

### 4.1 BIRT Plugin Development under WAR Deployment

If the plugin is supposed to run as a standalone application or only will be deployed to application server as a file deployment, user is free to use any file APIs in the development.

If the plugin is supposed to run in WAR deployment environment, File APIs to access the resources inside the WAR should not be used. Instead, resource based operations should be used. Please note only file operations that access resources inside the WAR file need to make this change. This also implies that the BIRT should not write anything into WAR. Temporary files folder or logging folder should be set to outside of WAR file. The caller application (e.g. BIRT Viewer) will set those temp file folders and logging folders when the application starts.

Avoiding using File APIs is not something special to Eclipse developers. Eclipse platform provides APIs for locating plugin resources (files, libraries etc.) in an abstract way, regardless of where the plugin is deployed (whether it's in a directory, in a ZIP file, or remotely in a web server). A properly written Eclipse plugin can be deployed as a ZIP file in the "Eclipse/plugins" directory and it should still run fine. The key to make this work is that the plugin must always use the following methods (instead of using File API) to access resources located within the plugin.

- `IPlatform.getBundle()`
- `IBundle.getEntryPath()`
- `IBundle.getEntry()`

BIRT Platform and Bundle provide these functions. In a server environment, they can help BIRT developers avoid using file APIs when accessing resources inside the WAR.