

# BIRT Support Resource Deployment Feature Specification

Author: Rima Kanguri

## Document Revisions

Version	Date	Description of Changes
Draft 1	02/17/2006	Initial draft.
Draft 2	02/27/2006	Minor changes based on team feed back
Draft 3	03/02/2006	Updates to section 3.3
Draft4	04/04/2006	UI mockups and other updates
Draft 5	04/18/2006	Changes to section on custom resource locator and the default resource locator.
Draft 6	04/19/2006	Updated based on review comments

<b>1. Introduction .....</b>	<b>2</b>
<b>2. Use cases .....</b>	<b>3</b>
2.1 Deploy BIRT and report designs in a single war file. ....	3
2.2 Deploy BIRT and report design separately. ....	3
<b>3. BIRT Designer changes .....</b>	<b>3</b>
3.1 Windows preference.....	3
3.2 Library explorer .....	4
3.3 Publish a library file to \$BIRT_RESOURCE_PATH .....	4
3.4 Message file builder .....	5
3.5 Image builder .....	6
<b>4. BIRT Design Engine Support .....</b>	<b>9</b>
4.1 Support for custom resource locator .....	9
4.2 Registering the custom resource locator .....	9
4.3 Design Engine will add support for BIRT_RESOURCE_PATH.....	9
4.4 API Changes.....	<b>Error! Bookmark not defined.</b>
<b>5. Report Engine and Viewer.....</b>	<b>10</b>
<b>6. Future enhancements .....</b>	<b>10</b>

## 1. Introduction

BIRT report designs could use external resources like the BIRT libraries, message files, java class files. The goal of the feature is to simplify the task of deployment of resources and report designs to different environments.

This document covers the following

- Deployment of the BIRT resources in designer and runtime.
- BIRT Designer enhancements to use resources from a predefined \$BIRT\_RESOURCE\_PATH
- BIRT Design Engine enhancements to support external resources.

The following resources could be external to the report designs

- Library file (.rptlibrary)
- Message file (.properties)
- Image files (Images used by the report design)
- Report Scripting code stored in java files (.class or jar files)

BIRT Designer, BIRT Engine and BIRT Design Engine will be enhanced to support BIRT\_RESOURCE\_PATH; it supports URI syntax. Resources are stored in folders or sub folders under BIRT\_RESOURCE\_PATH

## 2. Use cases

### 2.1 Deploy BIRT and report designs in a single war file.

This is the use case where a user creates a web application, which contains the BIRT Engine, report designs. (Note: The resources may or may not be part of the web application)

- In web.xml user sets the BIRT\_RESOURCE\_PATH to the location where the resources will be deployed.  
(Note: If BIRT\_RESOURCE\_PATH is empty, then by default it points to \$BIRT\_HOME/resources folder)
- Copies all the resources into \$BIRT\_RESOURCE\_PATH
- Creates a war file containing the BIRT Report engine, viewer, report designs, report design resources like message files, libraries, images etc.
- User deploys the war file into the application server; the war files could be unzipped or not depending on the application server.

### 2.2 Deploy BIRT and report design separately.

This is the use case where BIRT application is deployed on an application server. Reports and resources are deployed separately. This use case would also cover the example web viewer.

- User deploys the BIRT application. Sets the BIRT\_RESOURCE\_PATH in web.xml file to the location where the resources will be deployed.  
Or  
Calls the Report Engine API to set the BIRT\_RESOURCE\_PATH
- Report designer deploys all the common resources, which are to be shared between reports under \$BIRT\_RESOURCE\_PATH folder.
- Deploys report designs to the target location.

## 3. BIRT Designer changes

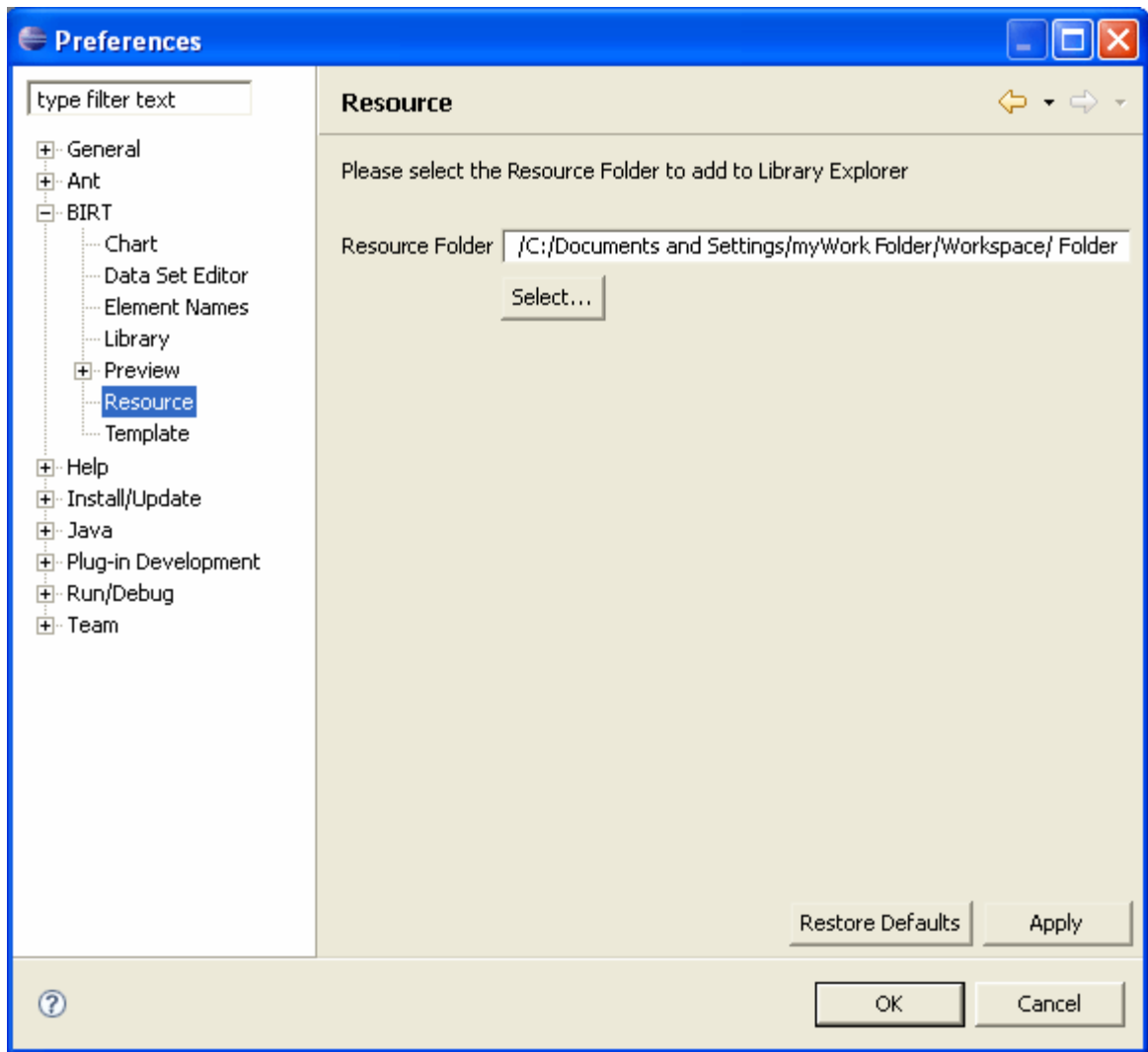
All the builders that allow user to add a resources to a report design will be enhanced to support \$BIRT\_RESOURCE\_PATH. BIRT report design created using BIRT designer will store resource paths relative to BIRT\_RESOURCE\_PATH.

Following are the UI changes

### 3.1 Windows preference

BIRT designer windows preference will be enhanced to allow the user to set the BIRT\_RESOURCE\_PATH.

- BIRT\_RESOURCE\_PATH. This stores the path to the BIRT resources, by default it is set to \$EclipseWorkspace/BIRTResource folder



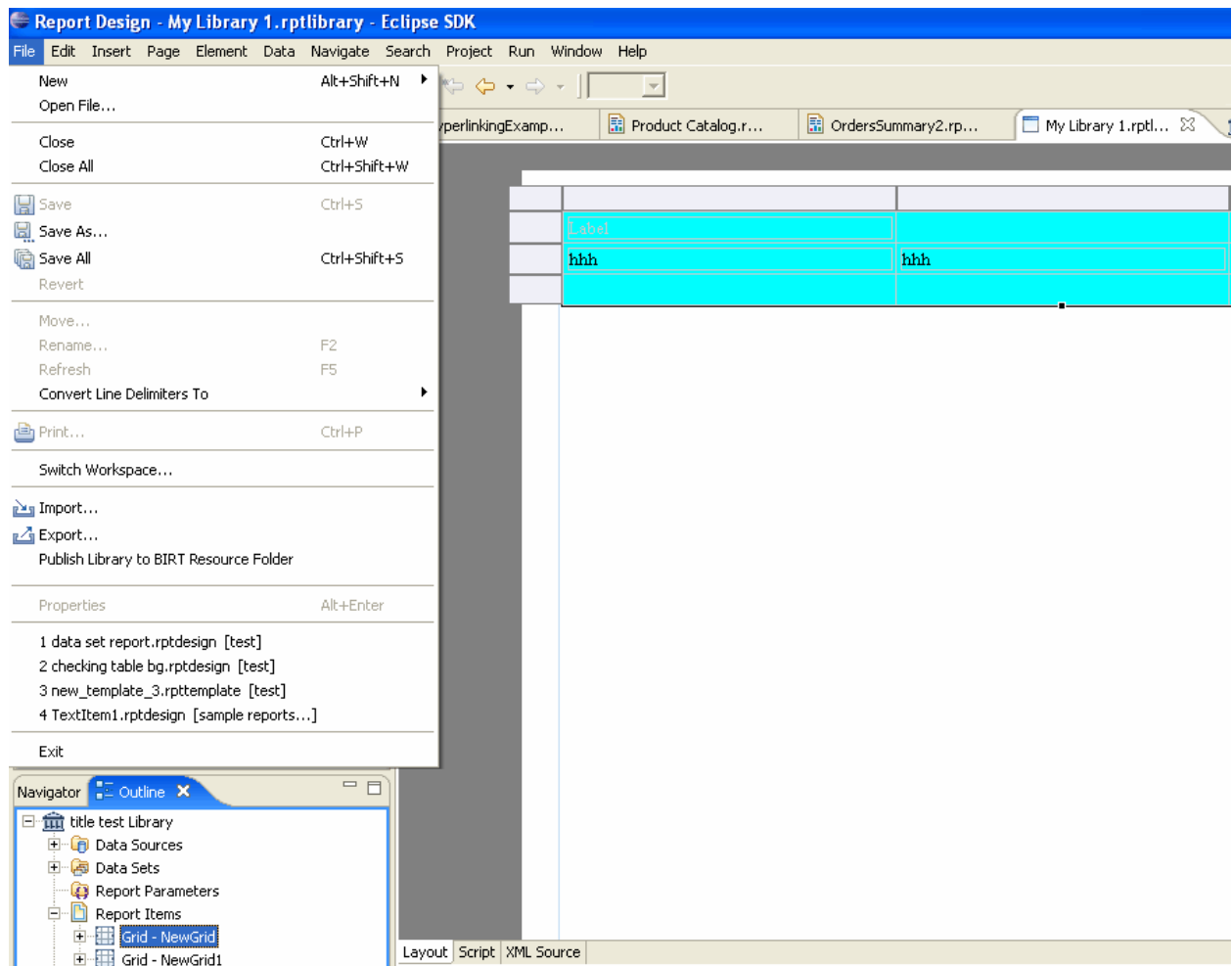
### 3.2 Library explorer

Currently the library explorer lists all the libraries in the current project and external libraries (added explicitly)

- The library explorer will be enhanced to show all the libraries in the BIRT\_RESOURCE\_PATH folder and subfolder only.
- The behavior will be same in non RCP and RCP BIRT

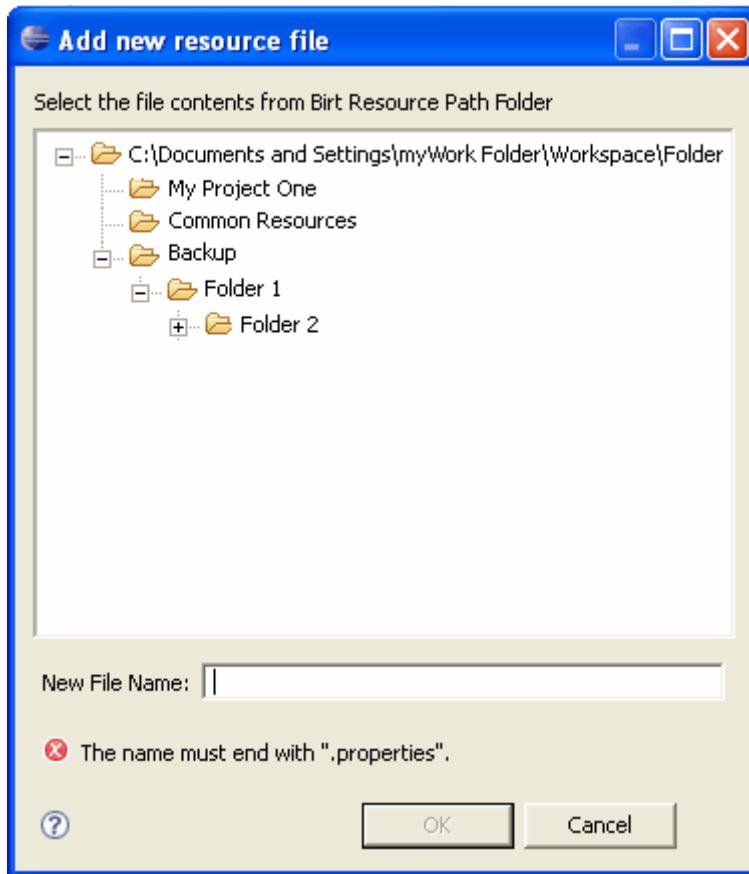
### 3.3 Publish a library file to \$BIRT\_RESOURCE\_PATH

- For .rptLibrary file, a new File menu to “Publish Library to BIRT Resource Folder” will be supported. This will allow a user to copy the library to \$BIRT\_RESOURCE\_PATH.



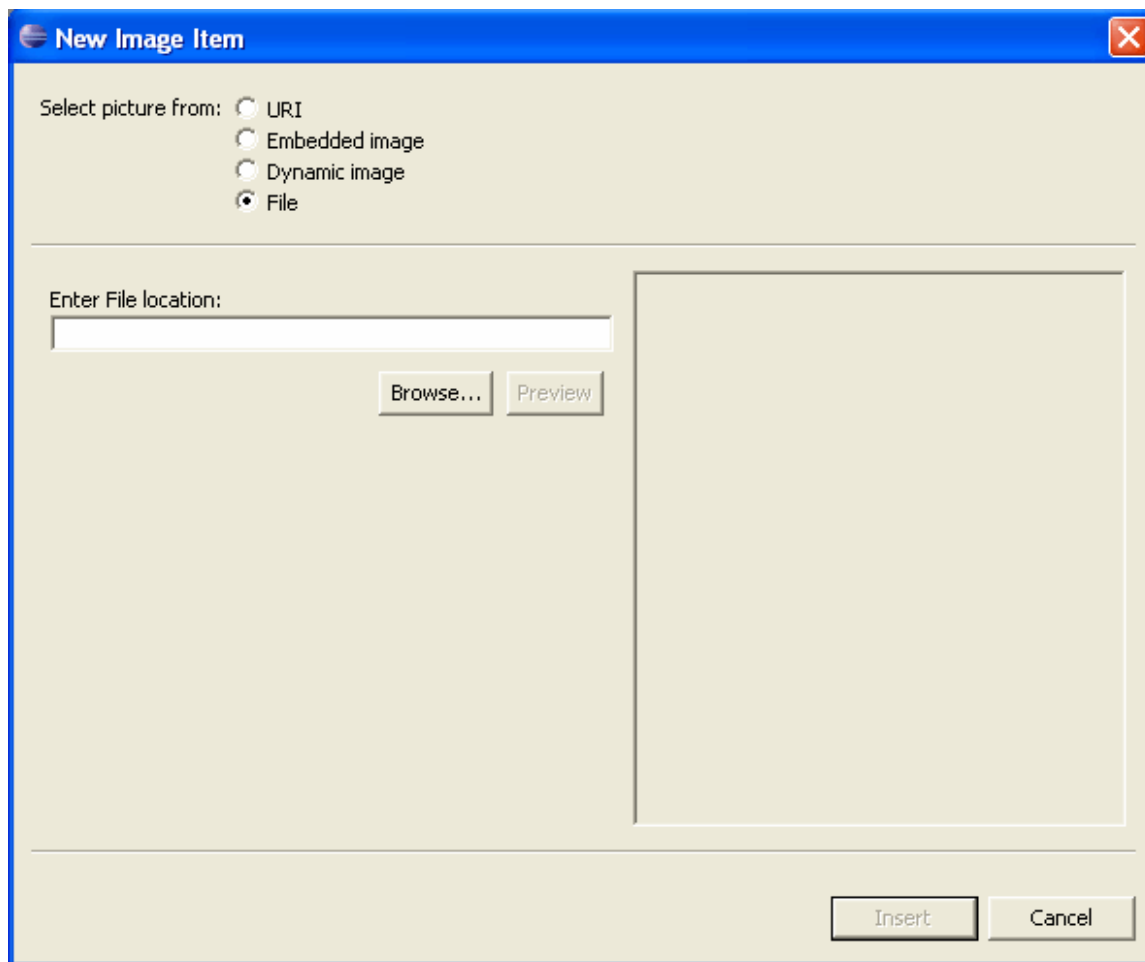
### 3.4 Message file builder

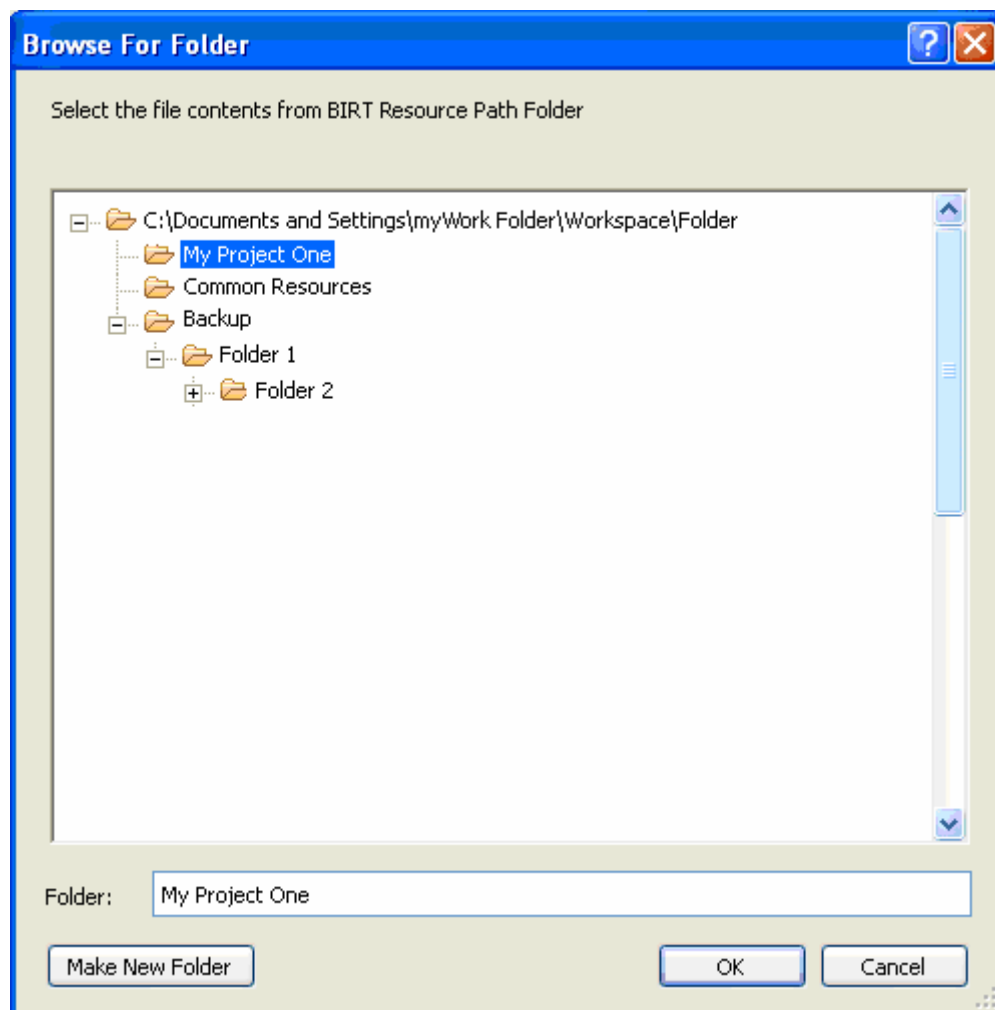
- The builder will be enhanced to list the properties files in the \$BIRT\_RESOURCE\_PATH folder.
- New message files can be added by using the resource builder.



### 3.5 Image builder

Image builder will be enhanced to support the “Browse” button. User is allowed to browse the file system under \$BIRT\_RESOURCE\_PATH folder and select an image file.







## 4. BIRT Design Engine Support

The following section describes the changes in the Design Engine to simplify the deployment of BIRT resources.

### 4.1 BIRT Default Resource locator

By default Model uses the following resource location algorithm

- Look for the BIRT resources relative to BIRT\_RESOURCE\_PATH (The absolute location of the resource is computed as \$BIRT\_RESOURCE\_PATH + relative resource path)
- Next look for the resources under all the model plug-in fragments (User could have created several fragments to the model plug-in)

### 4.2 Support for custom resource locator


The resource locator can be customized. DE already supports the IResourceLocator interface, to write a custom resource locator user needs to implement the IResourceLocator.

- Below is the IResourceLocator interface already supported by DE.

Public interface IResourceLocator

```
{  
Public URL findResource( ModuleHandle moduleHandle, String String  
Resourcefilename, int Type);  
}
```

DE provides the implementation for the default resource locator. This resource loader is based on the file system and uses the algorithm described in section 4.3

 *Note: If a user writes a custom resource locator, then the BIRT\_RESOURCE\_PATH will not be used, as the custom loader does the task of generating the absolute resource path.*

### 4.3 Registering the custom resource locator

Report Engine and Model API will support a method to register the custom resource locator class name.

### 4.4 Design Engine will add support for BIRT\_RESOURCE\_PATH.

- Enhancements to DE API SessionHandle class to support static setBIRTResourcePath() method.

## 5. Report Engine and Viewer

- BIRT ReportEngine API will provide a method to allow user set the BIRT\_RESOURCE\_PATH. By default it is set to \$BIRT\_HOME/resource folder
- BIRT Report Engine API will provide a method to allow user to set the class name of the custom resource locator
- Viewer will support a new configurable variable BIRT\_RESOURCE\_PATH in the web.xml file. By default it is set to empty

## 6. Future enhancements

- Library explorer to show file system view.
- Enhancements to BIRT designer to support renaming of folder in the above view; renaming a folder should modify all the report designs to the new location of the resource file.