

White Paper

Fabasoft Integration for Eclipse BIRT

Fabasoft Folio 2026

Contents

- 1 Introduction 3
- 2 Software Requirements..... 3
- 3 Getting Started..... 3
- 4 FSCBIRT@1.1001:CreateReport 6
- 5 Generating the Source XML..... 6
- 6 Creating a Report With Eclipse BIRT Report Designer 8
 - 6.1 Create a New Project 9
 - 6.2 Get Sample XML Data 9
 - 6.3 Specify a Data Source..... 9
 - 6.4 Specify Data Sets10
 - 6.5 Arranging the Report12
- 7 Including Images15
 - 7.1 folioimages15
 - 7.2 inlineimages18
- 8 Appendix.....20
 - 8.1 Eclipse BIRT Demo Report Design.....20

1 Introduction

This document describes how to create reports from Fabasoft Folio objects utilizing Eclipse BIRT (Business Intelligence and Reporting Tools).

2 Software Requirements

System environment: All information contained in this document implicitly assumes a Microsoft Windows environment or a Linux environment.

Supported platforms: For detailed information on supported operating systems and software see the software product information on the Fabasoft distribution media.

Descriptions in this document are based on the following software:

- Eclipse 2025-09
- Eclipse BIRT 4.21

3 Getting Started

All you need to get your first generated BIRT report is contained in this chapter. Detailed information about the introduced approaches can be found in the following chapters. As prerequisite you need a Fabasoft Folio installation with Fabasoft app.ducx. More information about Fabasoft app.ducx can be found in the white paper “An Introduction to Fabasoft app.ducx”.

Create a new app.ducx project (in this case BIRT@1.506) with an object model, a use case and a user interface file. Additionally create a BIRT report design file (`report.rptdesign`) and put it in the `resources` folder of the app.ducx project.

The following example defines an object model for a simple logbook management. With a context menu command a report for the logbook management can be created. To keep this first example as simple as possible the source XML data is hard-coded and the demo BIRT report design can be found in chapter 8.1 “Eclipse BIRT Demo Report Design”. How to generate XML data from Folio properties is described in chapter 5 “Generating the Source XML”. A brief introduction to the Eclipse BIRT Report Designer is provided in chapter 6 “Creating a Report With Eclipse BIRT Report Designer”.

Example

app.ducx Object Model Language

```
objmodel BIRT@1.506
{
  import COOSYSTEM@1.1;

  // creates an object with the BIRT report design as content
  instance FSCBIRT@1.1001:RPTDesign BirtReport {
    content = file("resources/report.rptdesign");
  }
}
```

```

// a logbook management may contain several logbooks
class LogbookManagement : CompoundObject {
    compound = true;
    common = true;
    unique Logbook[] vmlogbooks {
        child = true;
    }
}

// a logbook has a description and may contain several trip logs
class Logbook : CompoundObject {
    compound = true;
    common = true;
    string[] logdescription;
    unique TripLog[] logtriplogs {
        child = true;
    }
}

// defines data of a trip
class TripLog : BasicObject {
    datetime trlfrom;
    datetime trluntil;
    unsigned float(6,1) trlmiles {
        init = 0.0;
    }
}
}

```

app.ducx Use Case Language

```

usecases BIRT@1.506
{
    import COOSYSTEM@1.1;
    import FSCVAPP@1.1001;
    import FSCVENV@1.1001;
    import FSCVIEW@1.1001;
    import FSCEXPEXT@1.1001;
    import COOMAPI@1.1;
    import FSCACROBAT@1.1;

    menu usecase CreateBirtReport on selected {
        variant Object {
            impl = expression {
                // a demo XML
                string xmlstr = "<data><objname>Logbook Management</objname><logbook
id='COO.1.506.1.1002523'><objname>First
Logbook</objname><logdescription>Description first
logbook</logdescription><logtriplog><objname>First Trip
Log</objname><trlfrom>2011-12-15T09:00:00</trlfrom><trluntil>2011-12-
15T10:10:00</trluntil><trlmiles>10</trlmiles></logtriplog><logtriplog><objname>Sec
ond Trip Log</objname><trlfrom>2011-12-16T14:00:00</trlfrom><trluntil>2011-12-
16T16:30:00</trluntil><trlmiles>20</trlmiles></logtriplog></logbook><logbook
id='COO.1.506.1.1002527'><objname>Second
Logbook</objname><logdescription>Description second
logbook</logdescription><logtriplog><objname>Trip Log 1</objname><trlfrom>2011-12-
19T23:00:00</trlfrom><trluntil>2011-12-
20T04:00:00</trluntil><trlmiles>600</trlmiles></logtriplog><logtriplog><objname>Tr
ip Log 2</objname><trlfrom>2011-12-22T15:00:00</trlfrom><trluntil>2011-12-
22T16:00:00</trluntil><trlmiles>40</trlmiles></logtriplog></logbook></data>";

                content xmlcont = coort.CreateContent();
                xmlcont.SetContent(coortx, 1, 65001, xmlstr);
                // creates the report
                content result = coouser.FSCBIRT@1.1001:CreateReport
                    (xmlcont, #BirtReport)[3];
                // PDF object containing the report
                FSCACROBAT@1.1:PDFObject pdf =
                    #FSCACROBAT@1.1:PDFObject.ObjectCreate()[2];
            }
        }
    }
}

```

```

        pdf.content = result;
        pdf.content.contextextension = "pdf";
        pdf.objname = "Report " + coort.GetCurrentDateTime(coobj);
        cooroot.ObjectLock(true, true);
        cooroot.objchildren += pdf;
    }
}
}
}

```

app.ducx User Interface Language

```

userinterface BIRT@1.506
{
    import COOSYSTEM@1.1;
    import COOATTREDIT@1.1;
    import CODESK@1.1;
    import COOSEARCH@1.1;
    import FSCVAPP@1.1001;

    // defines a context menu entry for the LogbookManagement object class to
    // be able to generate a report
    extend class LogbookManagement {
        menus {
            default = {
                expansions {
                    MenuContextExpansion {
                        entries = {
                            MenuCreateBirtReport
                        }
                    }
                }
            }
        }
    }
}
}

```

report.rptdesign

```
// see chapter 8.1 "Eclipse BIRT Demo Report Design"
```

To get your first report, perform the following steps:

1. After you have compiled and run this project create a new "LogbookManagement" object.
2. Open the context menu of the logbook management and execute "MenuCreateBirtReport".

A PDF report based on the demo XML is generated and is put on the desktop.

First Logbook

Description first logbook

2011-12-15T09:00:00	2011-12-15T10:10:00	10
2011-12-16T14:00:00	2011-12-16T16:30:00	20

Second Logbook

Description second logbook

2011-12-19T23:00:00	2011-12-20T04:00:00	600
2011-12-22T15:00:00	2011-12-22T16:00:00	40

To generate the report, the use case `FSCBIRT@1.1001:CreateReport` is used. Detailed information about this use case can be found in the next chapter.

Next steps:

- To improve this first example replace the hard-coded XML with generated XML as described in chapter 5 “Generating the Source XML”.
- To improve the layout create your own report or extend the demo report (see chapter 6 “Creating a Report With Eclipse BIRT Report Designer”).

4 FSCBIRT@1.1001:CreateReport

Fabasoftware Folio provides the use case `FSCBIRT@1.1001:CreateReport` to create BIRT reports.

```
FSCBIRT@1.1001:CreateReport (  
  content xmldata,  
  object reportdesign,  
  out content report,  
  optional string format)
```








Parameters:

- `xmldata`
A content containing the XML file that provides the data that should be displayed in the report.
- `reportdesign`
A report design file created with the Eclipse BIRT Report Designer.
- `report`
A content containing the resulting report.
- `format`
A string containing the desired format of the resulting report.
This may be “pdf” (default), “doc”, “ppt”, “xls”, “html”, “odp”, “ods” or “odt”.
- For details of the report design or format see <http://www.eclipse.org/birt>.

5 Generating the Source XML

The following example describes how to generate easily XML data within Fabasoftware app.ducx Expressions. Alternatively, Java standard functionality can be used to generate XML data in a convenient way.

First create some demo data. A logbook management is needed that contains some logbooks containing some trips. Edit the trips’ properties and insert dates and miles.

- ▼  Logbook Management
 - ▼  First Logbook
 - ▶  First Trip Log
 - ▶  Second Trip Log
 - ▼  Second Logbook
 - ▶  Trip Log 1
 - ▶  Trip Log 2

The master plan is to put the data in a dictionary and generate the XML structure with the `FSCEXPEXT@1.1001:Value2XML()` action. Replace the use case file as used in chapter 3 “Getting Started” with the following use case file.

Example

app.ducx Use Case Language

```

usecases BIRT@1.506
{
  import COOSYSTEM@1.1;
  import FSCVAPP@1.1001;
  import FSCVENV@1.1001;
  import FSCVIEW@1.1001;
  import FSCEXPEXT@1.1001;
  import COOMAPI@1.1;
  import FSCACROBAT@1.1;

  // gets executed when calling the context menu command
  menu usecase CreateBirtReport on selected {
    variant Object {
      impl = expression {
        // GenerateData() collects the meta data of a logbook management
        // the implementation can be found below
        dictionary data = coobj.GenerateData();
        // creates XML from the data dictionary
        string xmlstr = coobj.Value2XML(data);
        content xmlcont = coort.CreateContent();
        xmlcont.SetContent(cootx, 1, 65001, xmlstr);
        // a BIRT PDF report is generated based on the XML data and
        // the #BirtReport defined in the object model language
        content result = coouser.FSCBIRT@1.1001:CreateReport
          (xmlcont, #BirtReport)[3];

        // contains the PDF report
        FSCACROBAT@1.1:PDFObject pdf =
          #FSCACROBAT@1.1:PDFObject.ObjectCreate()[2];
        pdf.content = result;
        pdf.content.contextextension = "pdf";
        pdf.objname = "Report " + coort.GetCurrentDateTime(coobj);
        // put the report on the desk of the user
        cooroot.ObjectLock(true, true);
        cooroot.objchildren += pdf;
      }
    }
  }

  // collects the meta data of a logbook management and returns a dictionary
  usecase GenerateData(retval dictionary result) {

```

```

// use case implementation for the LogbookManagement
variant LogbookManagement {
  impl = expression {
    result = {
      // the generated XML will contain a tag "objname" with the
      // computed value of coobj.objname
      objname: coobj.objname
    };
    dictionary[] logbooks;
    dictionary[] logbookkids;
    // for all logbooks in the logbook management
    for (Logbook logbook: coobj.vmlogbooks) {
      logbooks += logbook.GenerateData();
      // IDs are necessary to be able to filter the rows with XPath
      // expressions in the report design (more information can be found
      // in chapter 6 "Creating a Report With Eclipse BIRT Report Designer"
      logbookkids += { id: logbook.objaddress };
    }
    result.SetEntry("logbook", logbooks);
    result.SetEntry("@logbook", logbookkids);
  }
}

// use case implementation for the LogbookManagement
variant Logbook {
  impl = expression {
    result = {
      objname: coobj.objname,
      logdescription: coobj.logdescription
    };
    dictionary[] logtriplogs;
    for (TripLog triplog: coobj.logtriplogs) {
      logtriplogs += triplog.GenerateData();
    }
    result.SetEntry("logtriplog", logtriplogs);
  }
}

// use case implementation for the TripLog
variant TripLog {
  impl = expression {
    result = {
      objname: coobj.objname,
      trlfrom: coobj.trlfrom,
      trluntil: coobj.trluntil,
      trlmiles: coobj.trlmiles
    };
  }
}
}
}

```

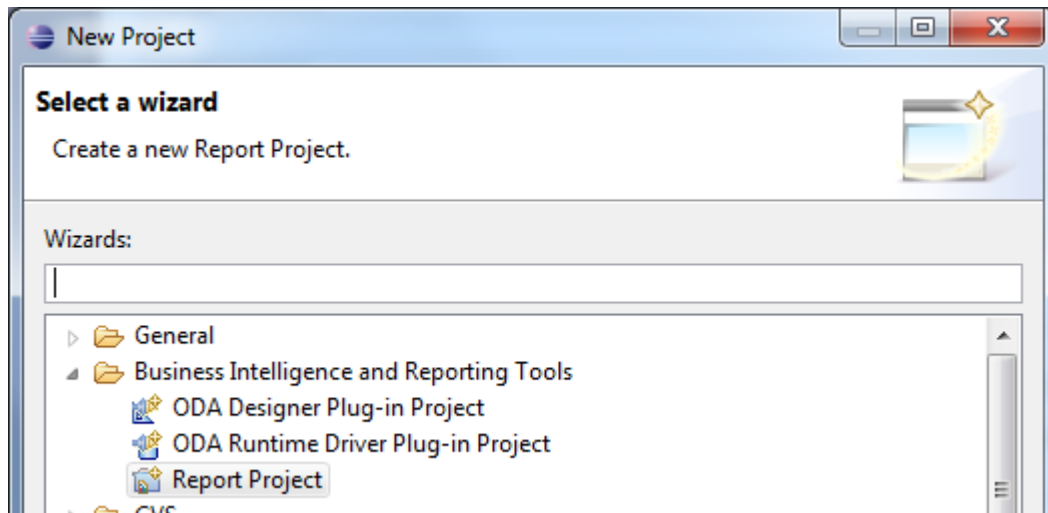
As result the report is generated based on the data provided in the Folio objects. The last step is to improve the demo report with the Eclipse BIRT Report Designer.

6 Creating a Report With Eclipse BIRT Report Designer

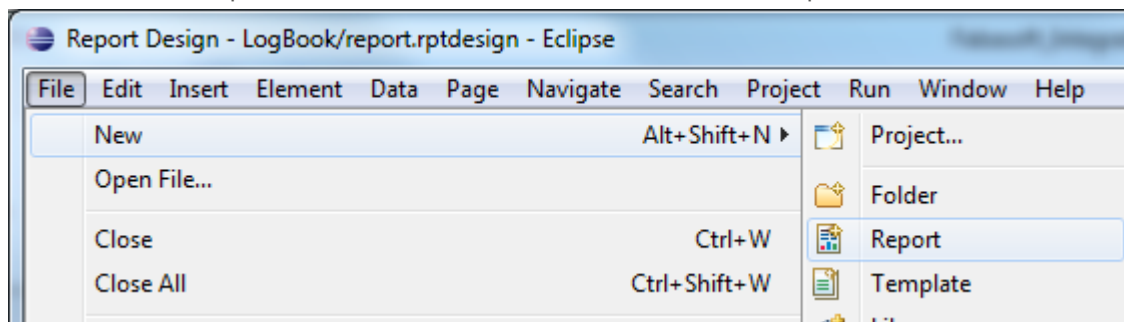
This chapter provides a brief introduction to the Eclipse BIRT Report Designer. It is highly recommended to read dedicated BIRT literature to get an impression what BIRT has to offer. In the following chapters a BIRT report that is able to display the logbook management data is created from scratch.

6.1 Create a New Project

1. Get Eclipse BIRT (<http://eclipse.org/birt>).
2. Create a new "Report Project" and follow the wizard.



3. Create a new "Report" and follow the wizard (create a blank report).



6.2 Get Sample XML Data

The design process is based on a sample data source. In this case the generated XML data should be the data source. To avoid writing the XML file manually you can add following lines to the `CreateBirtReport` use case.

Example

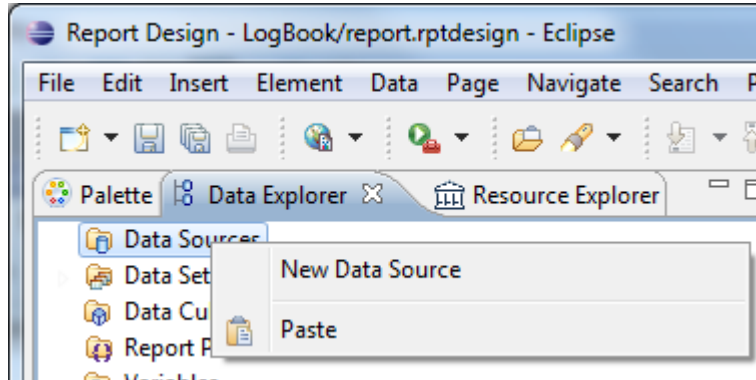
```
GENCONT@1.1:ContentObject resultxmlobj =
    #GENCONT@1.1:ContentObject.ObjectCreate() [2];
resultxmlobj.content.contcontent = xmlcont;
resultxmlobj.content.contextextension = "txt";
resultxmlobj.objname = "Result XML " + coort.GetCurrentDateTime(coobj);
```

Now you can search for the content object in Fabasoft Folio and use the XML as data source (may also be useful for debugging purposes).

6.3 Specify a Data Source

Use the previously generated sample XML as data source.

1. Create a new data source.



2. Select "XML Data Source" and specify the path to the sample XML file.

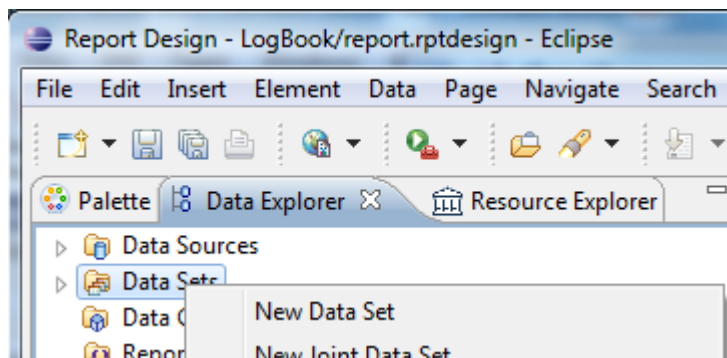
6.4 Specify Data Sets

Based on the data source, data sets can be created. For the logbook management example two data sets are required:

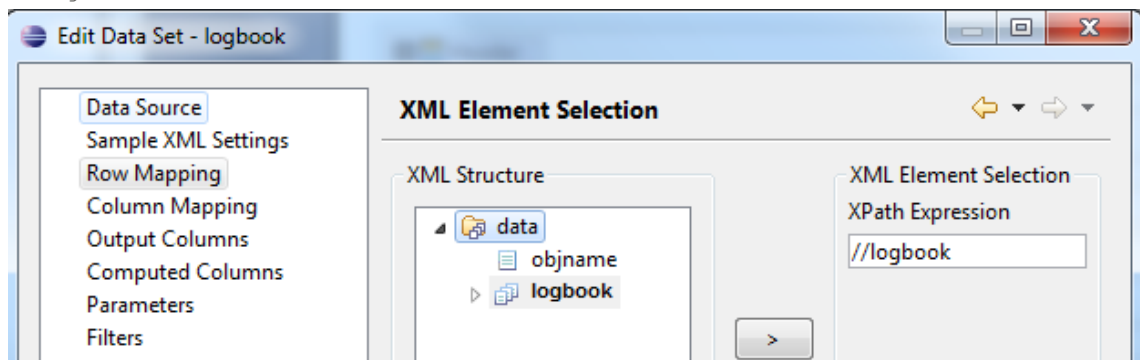
- A logbook management can contain several logbooks.
- A logbook can contain several trips.

To put this hierarchy into relation the two data sets are used. The first data set selects the logbooks and the second data set selects the trips. The ID attribute of the logbook is used to match the trips with the corresponding logbook. This is a generic pattern when working with such hierarchies.

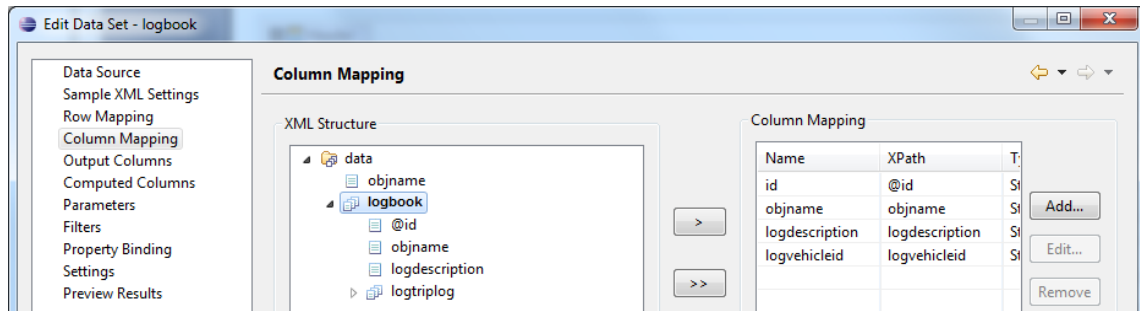
1. Create a new data set.



2. Name the data set "logbook".
3. To select all logbook XML elements, define as "Row Mapping" the XPath expression `//logbook`.



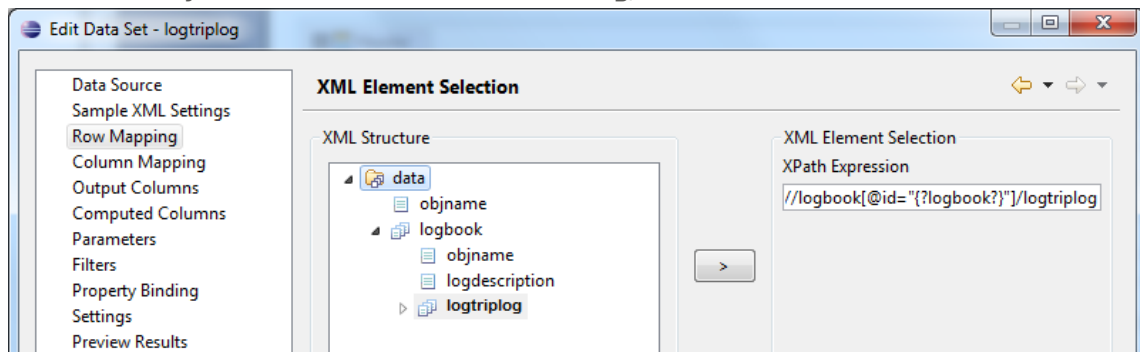
- In the "Column Mapping" section add the values that should be display in the report for logbooks or are used for computation (ID attribute).



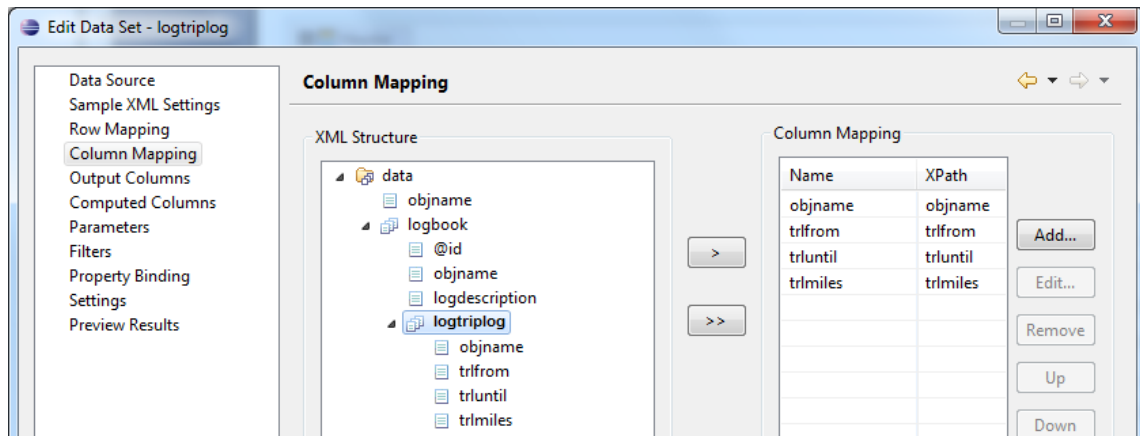
To create the second data set, perform the following steps:

- Create a second data set named "logtriplog".
- As "Row Mapping" use the XPath expression `//logbook[@id="{?logbook?}"]/logtriplog`. The XML attribute "id" is used to match the trip with the corresponding logbook.

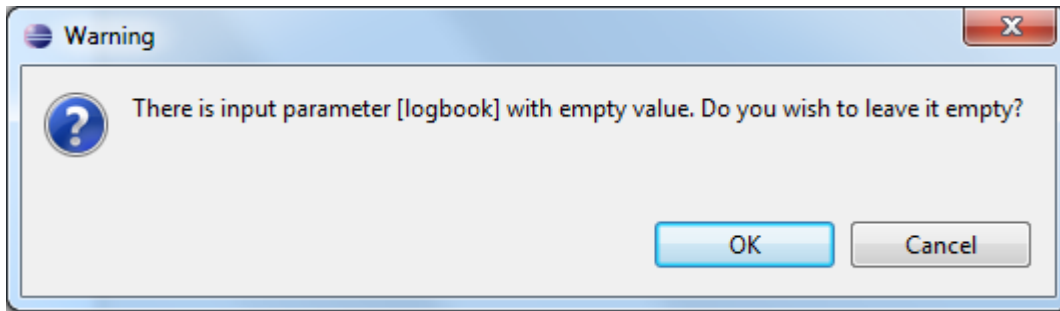
Note: BIRT only allows XML attributes for filtering, not XML elements.



- In the "Column Mapping" section add the values that should be display in the report for trips.



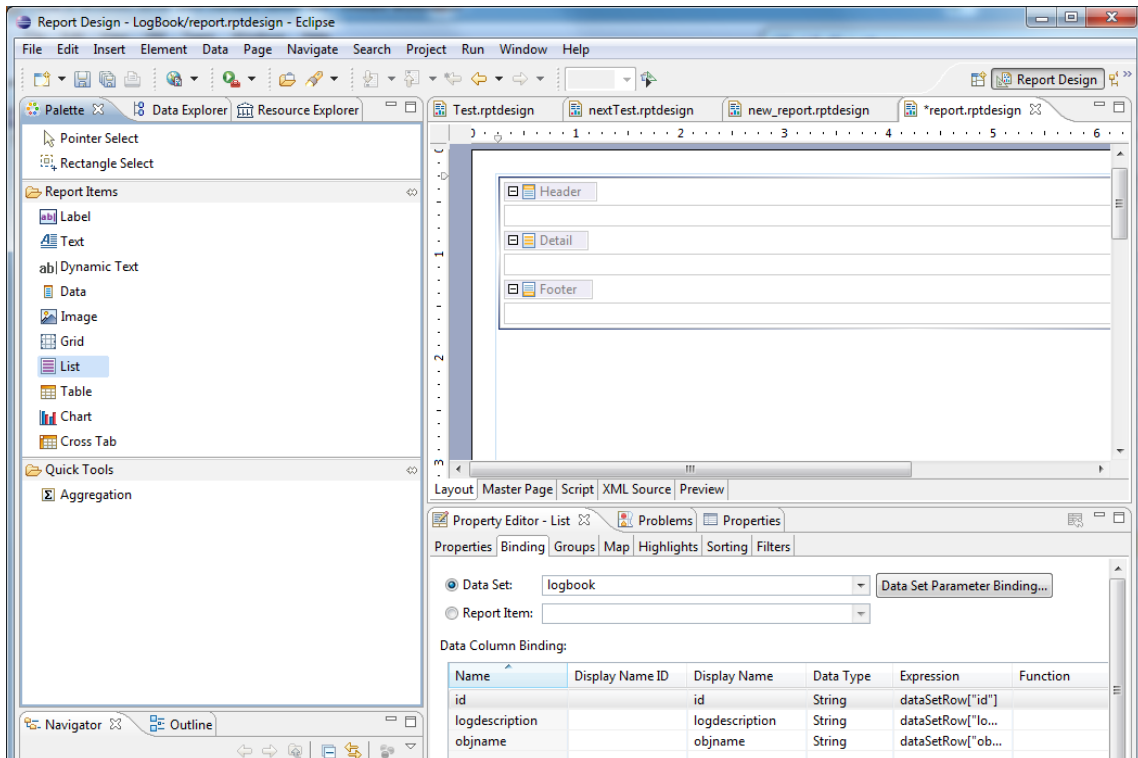
- When finishing the creation of the data set click "OK" to let the "logbook" parameter empty.



6.5 Arranging the Report

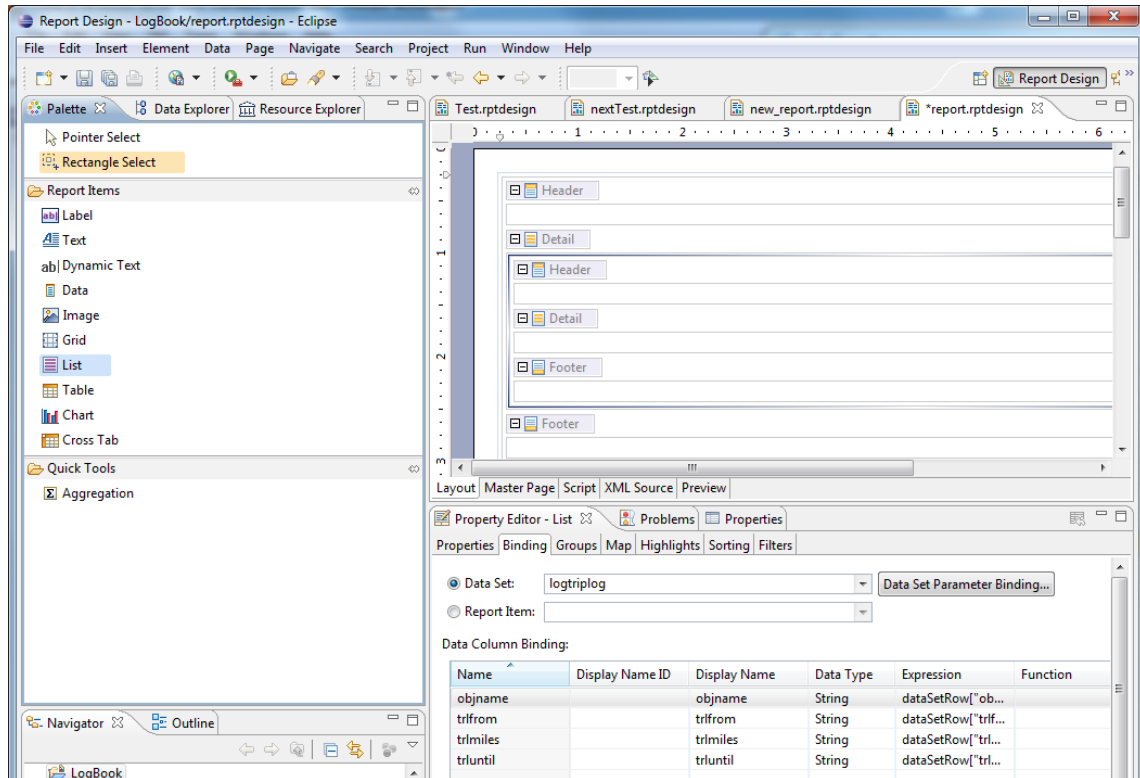
You can arrange the parts of the report mainly with drag and drop. In this example two lists are used that have a binding to the two data sets. The second list containing the trips uses the "logbook" parameter to display the trips that belong to the corresponding logbook.

- Switch to the "Palette" tab and drag a "List" to the report.

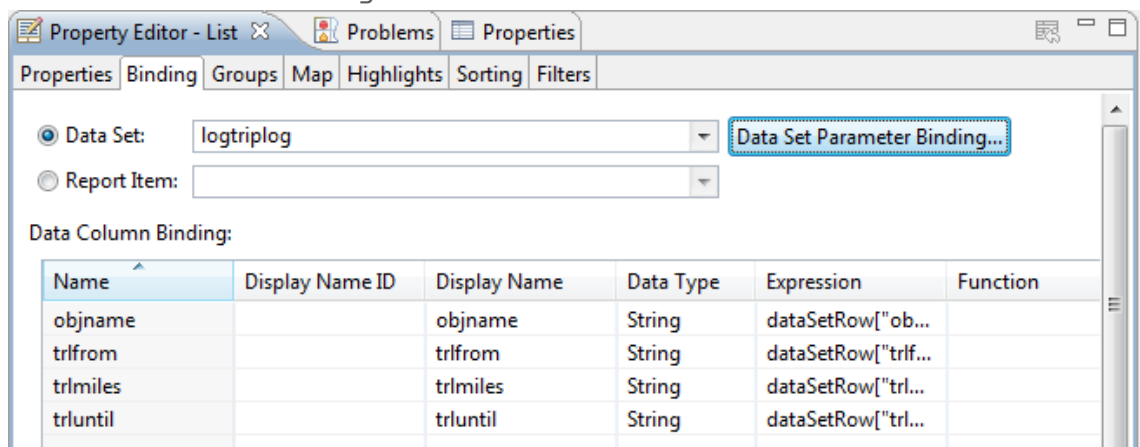


- Select "logbook" as binding in the "Property Editor" for this list.
- Drag a second list in the "Detail" section of the first list.

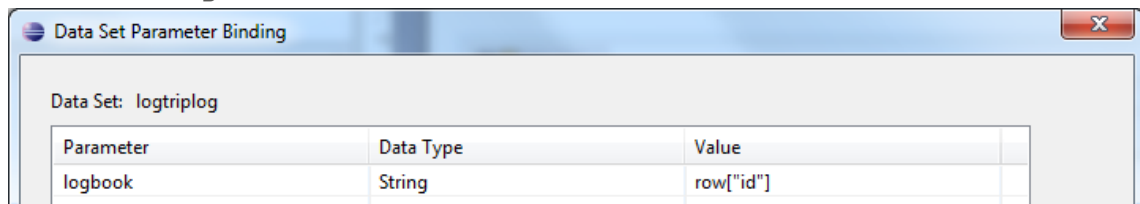
- Select "logtriplog" as binding in the "Property Editor" for this list.



- To define the parameter that is used for filtering the trips that belong to the logbook, click "Data Set Parameter Binding".

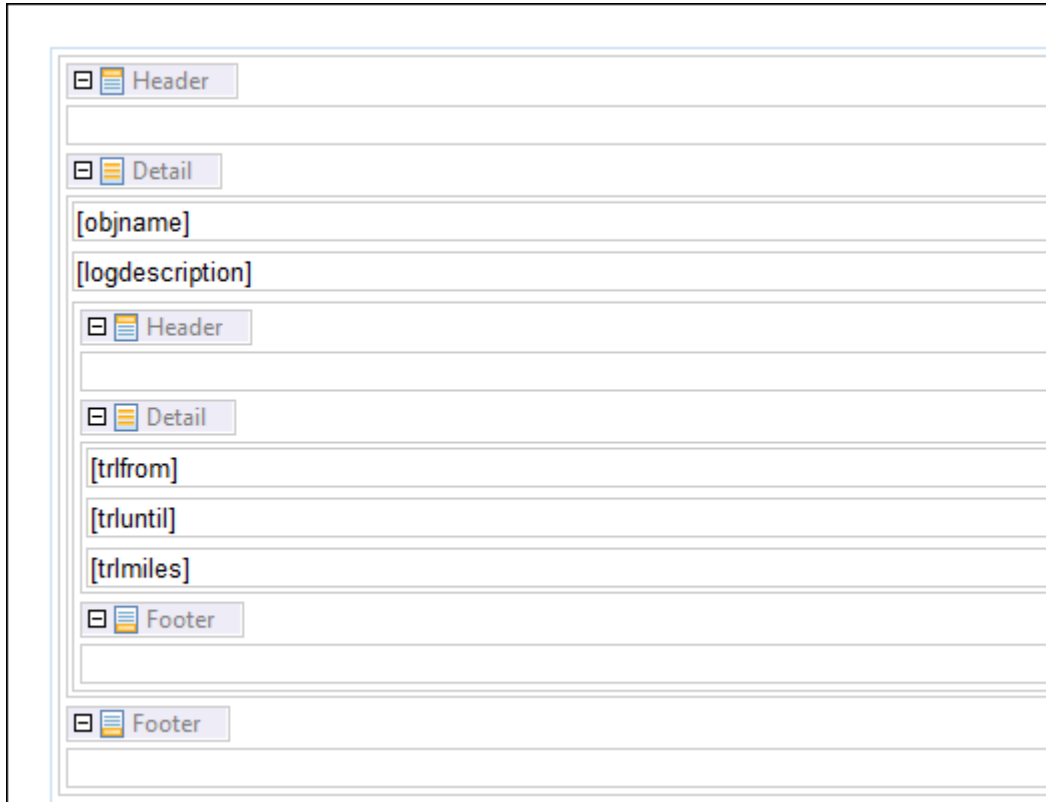


- Specify `row["id"]` as value for the "logbook" parameter. This way the XML attribute "id" is used for filtering.

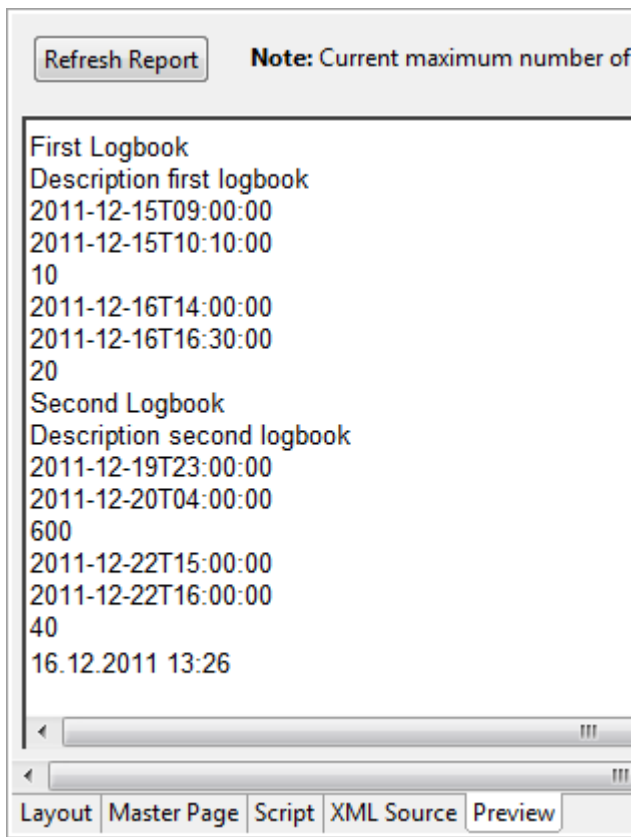


- Switch to the "Data Explorer" tab and drag "objname" and "logdescription" in the detail section of the first list right before the second list.

8. Drag "trlfrom", "trluntil" and "trlmiles" in the detail section of the second list.



9. Switch to the "Preview" tab.



Now the desired report functionality is given. BIRT offers a lot of possibilities to improve the layout and format the values of a report. When you are glad with your report put it in the

resources folder of your app.ducx project, run the project again and generate a PDF report with the new design.

A report of a fully featured logbook management project may look like this:

Logbook Details

L 881 FS

Total Mileage 3704
Vehicle ID L 881 FS

Trip Log for "L 881 FS" (Open, Nov 3, 2011 - Nov 29, 2011)

Trips

Starting Mileage	120520	Ending Mileage	120724
Departure on/at	03.11.2011 08:00	Arrival on/at	03.11.2011 10:10
Place of Departure	Linz	Destination	Wien
Driver	Stephan Leiner	Trip Mileage	204
Duration	2h 10m		
Starting Mileage	120724	Ending Mileage	120910
Departure on/at	03.11.2011 16:30	Arrival on/at	03.11.2011 18:20
Place of Departure	Wien	Destination	Linz
Driver	Stephan Leiner	Trip Mileage	186
Duration	1h 50m		
Starting Mileage	120910	Ending Mileage	120945
Departure on/at	04.11.2011 07:30	Arrival on/at	04.11.2011 08:10
Place of Departure	Linz	Destination	Perg
Driver	Stephan Leiner	Trip Mileage	35
Duration	40m		
Starting Mileage	120945	Ending Mileage	120980
Departure on/at	04.11.2011 17:00	Arrival on/at	04.11.2011 17:40

7 Including Images

Beside textual metadata also images may be included in a report. Static images can be included directly in the report design whereas dynamic images are defined in the source XML data. There are two ways to include images:

- folioimages
- inlineimages

When using the `folioimages` tag only the object addresses of the images are included in the XML data. When using the `inlineimages` tag an identifier and the image data itself is included in the XML data.

7.1 folioimages

In order that the images are available when generating the report the XML must contain a `folioimages` tag, followed by `image` tags containing the object address of the image object.

Example

```

<data>
...
<folioimages>
  <image>COO.1.506.1.1002712</image>

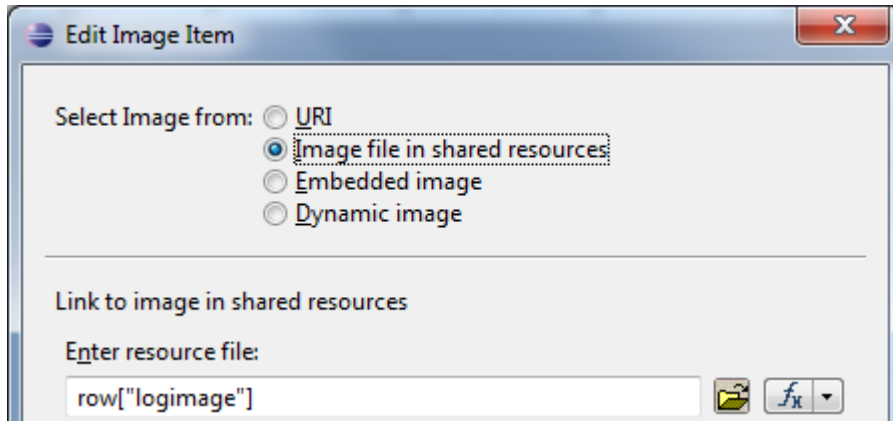
```

```

    <image>COO.1.506.1.1002713</image>
  </folioimages>
</data>

```

When generating the report the defined images are available in the shared resource folder. Thus, in the BIRT Report Designer you can define an "Image file in shared resources".



The following example extends the previous one including an image for each logbook. The example is not self-containing, it just depicts the differences.

Example

app.ducx Object Model Language

```

objmodel BIRT@1.506
{
  import COOSYSTEM@1.1;
  import FSCWEBCONT@1.1001;

  class Logbook : CompoundObject {
    compound = true;
    common = true;
    string[] logdescription;
    // defines an image for the logbook
    ImageObject logimage;
    unique TripLog[] logtriplogs {
      child = true;
    }
  }
}

```

app.ducx Use Case Language

```

usecases BIRT@1.506
{
  import COOSYSTEM@1.1;
  import FSCVAPP@1.1001;
  import FSCVENV@1.1001;
  import FSCVIEW@1.1001;
  import FSCEXPEXT@1.1001;
  import COOMAPI@1.1;
  import FSCACROBAT@1.1;

  usecase GenerateData(retval dictionary result) {
    variant LogbookManagement {
      impl = expression {
        result = {
          objname: coobj.objname

```

```

};
dictionary[] logbooks;
dictionary[] logbookkids;
// images from all logbooks
string[] logimages;
// provides the necessary folioimages parent
dictionary folioimages;
for (Logbook logbook: coobj.vmlogbooks) {
    logbooks += logbook.GenerateData();
    logbookkids += { id: logbook.objaddress };
    // add all logbook images
    logimages += logbook.logimage.objaddress;
}
result.SetEntry("logbook", logbooks);
result.SetEntry("@logbook", logbookkids);
// this dictionary will contain all logbook images
folioimages = coort.CreateDictionary();
// add the logbook images
folioimages.SetEntry("image", logimages);
// add the folioimages dictionary to the result dictionary
result.SetEntry("folioimages", folioimages);
}
}
variant Logbook {
    impl = expression {
        result = {
            objname: coobj.objname,
            logdescription: coobj.logdescription,
            // in the report the image corresponding to the logbook can be
            // accessed using the object address and file extension
            logimage: coobj.logimage.objaddress + "." +
                coobj.logimage.content.contextension
        };
        dictionary[] logtriplogs;
        for (TripLog triplog: coobj.logtriplogs) {
            logtriplogs += triplog.GenerateData();
        }
        result.SetEntry("logtriplog", logtriplogs);
    }
}
variant TripLog {
    impl = expression {
        result = {
            objname: coobj.objname,
            trlfrom: coobj.trlfrom,
            trluntil: coobj.trluntil,
            trlmiles: coobj.trlmiles
        };
    }
}
}
}
}

```

The resulting XML looks like this:

Result

```

<data>
  <objname>Logbook Management</objname>
  <logbook id="COO.1.506.1.1002523">
    <objname>First Logbook</objname>
    <logdescription>Description first logbook</logdescription>
    <!-- generated with logimage: coobj.logimage.objaddress + "." +
      coobj.logimage.content.contextension
    -->
  </logbook>
</data>

```

```

        <logimage>COO.1.506.1.1002712.jpg</logimage>
        ...
    </logbook>
    ...
    <!-- generated with result.SetEntry("folioimages", folioimages); -->
    <folioimages>
        <image>COO.1.506.1.1002712</image>
        ...
    </folioimages>
</data>

```

Note: In the Eclipse BIRT Report Designer the images are displayed broken. For testing purposes you can put some image files corresponding to the sample XML data in the shared resources folder of the Eclipse BIRT Report Designer.

7.2 inlineimages

In order that the images are available when generating the report the XML must contain a `inlineimages` tag, followed by `image` tags containing the object address of the image object.

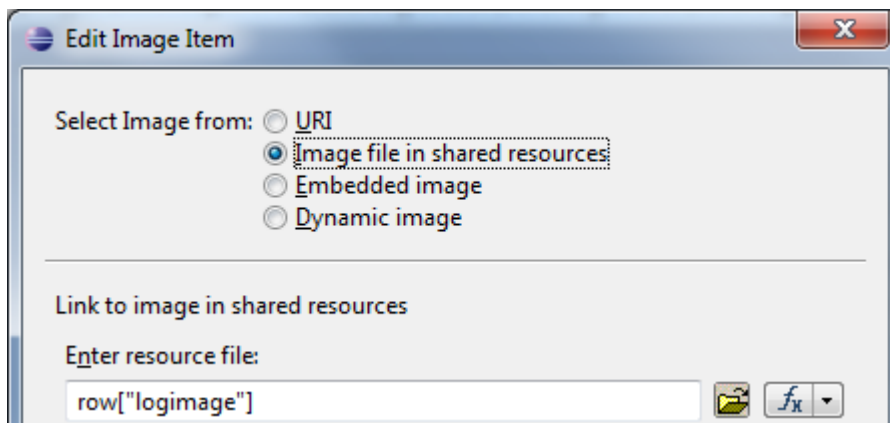
Example

```

<data>
    ...
    <inlineimages>
        <image name="COO.1.506.1.1002712.jpg"/>/9j/4AAQSk...</image>
        <image name="COO.1.506.1.1002713.jpg"/>/9j/4AAQSk...</image>
    </inlineimages>
</data>

```

When generating the report the defined images are available in the shared resource folder. Thus, in the BIRT Report Designer you can define an "Image file in shared resources".



The following example modifies the previous one including an inline image for each logbook. The example is not self-containing, it just depicts the differences.

Example

app.ducx Object Model Language

```

objmodel BIRT@1.506
{

```

```

import COOSYSTEM@1.1;
import FSCWEBCONT@1.1001;

class Logbook : CompoundObject {
    compound = true;
    common = true;
    string[] logdescription;
    // defines an image for the logbook
    ImageObject logimage;
    unique TripLog[] logtriplogs {
        child = true;
    }
}

```

app.ducx Use Case Language

```

usecases BIRT@1.506
{
    import COOSYSTEM@1.1;
    import FSCVAPP@1.1001;
    import FSCVENV@1.1001;
    import FSCVIEW@1.1001;
    import FSCEXPEXT@1.1001;
    import COOMAPI@1.1;
    import FSCACROBAT@1.1;

    usecase GenerateData(retval dictionary result) {
        variant LogbookManagement {
            impl = expression {
                result = {
                    objname: coobj.objname
                };
                dictionary[] logbooks;
                dictionary[] logbookkids;
                // images from all logbooks
                string[] logimages;
                // provides the necessary inlineimages parent
                dictionary inlineimages;
                dictionary inlineimagesids;
                string[] inlineimagesbytes;
                for (Logbook logbook: coobj.vmlogbooks) {
                    logbooks += logbook.GenerateData();
                    logbookkids += { id: logbook.objaddress };
                    // add all logbook images
                    logimages += logbook.logimage.objaddress + "." +
                        logbook.logimage.content.contextextension;
                    // the data of the images Base64-encoded
                    inlineimagesbytes +=
                        coouser.Base64Encode(logbook.logimage.content.contcontent);
                }
                result.SetEntry("logbook", logbooks);
                result.SetEntry("@logbook", logbookkids);
                // provides the "name" XML attribute for the images
                for (string logimage: logimages) {
                    inlineimagesids += { name: logimage };
                }
                // this dictionary will contain all logbook images
                inlineimages = coort.CreateDictionary();
                // add the logbook images
                inlineimages.SetEntry("image", inlineimagesbytes);
                inlineimages.SetEntry("@image", inlineimagesids);
                // add the inlineimages dictionary to the result dictionary
                result.SetEntry("inlineimages", inlineimages);
            }
        }
    }
}

```

```

}
variant Logbook {
  impl = expression {
    result = {
      objname: coobj.objname,
      logdescription: coobj.logdescription,
      // in the report the image corresponding to the logbook can be
      // accessed using the object address and file extension
      logimage: coobj.logimage.objaddress + "." +
                coobj.logimage.content.contextextension
    };
    dictionary[] logtriplogs;
    for (TripLog triplog: coobj.logtriplogs) {
      logtriplogs += triplog.GenerateData();
    }
    result.SetEntry("logtriplog", logtriplogs);
  }
}
variant TripLog {
  impl = expression {
    result = {
      objname: coobj.objname,
      trlfrom: coobj.trlfrom,
      trluntil: coobj.trluntil,
      trlmiles: coobj.trlmiles
    };
  }
}
}
}
}
}

```

The resulting XML looks like this:

Result

```

<data>
  <objname>Logbook Management</objname>
  <logbook id="COO.1.506.1.1002523">
    <objname>First Logbook</objname>
    <logdescription>Description first logbook</logdescription>
    <!-- generated with logimage: coobj.logimage.objaddress + "." +
      coobj.logimage.content.contextextension
    -->
    <logimage>COO.1.506.1.1002712.jpg</logimage>
    ...
  </logbook>
  ...
  <!-- generated with result.SetEntry("inlineimages", inlineimages); -->
  <inlineimages>
    <image name="COO.1.506.1.1002712.jpg">/9j/4AAQSk...</image>
    ...
  </inlineimages>
</data>

```

8 Appendix

8.1 Eclipse BIRT Demo Report Design

The following XML data describes a report that can be used with the examples throughout this white paper.

Note: Use UTF-8 as encoding for this file.

```
<?xml version="1.0" encoding="UTF-8"?>
<report xmlns="http://www.eclipse.org/birt/2005/design" version="3.2.23" id="1">
  <property name="createdBy">Eclipse BIRT Designer Version 3.7.2.v20120213 Build
  &lt;3.7.2.v20120214-1408</property>
  <property name="units">in</property>
  <property name="iconFile">/templates/blank_report.gif</property>
  <property name="bidiLayoutOrientation">ltr</property>
  <property name="imageDPI">96</property>
  <data-sources>
    <oda-data-source extensionID="org.eclipse.datatools.enablement.oda.xml"
name="Data Source" id="7">
      <property name="FILELIST">test.xml</property>
    </oda-data-source>
  </data-sources>
  <data-sets>
    <oda-data-set
extensionID="org.eclipse.datatools.enablement.oda.xml.dataSet" name="logbook"
id="8">
      <list-property name="columnHints">
        <structure>
          <property name="columnName">id</property>
          <property name="analysis">dimension</property>
          <text-property name="heading">id</text-property>
        </structure>
        <structure>
          <property name="columnName">objname</property>
          <property name="analysis">attribute</property>
          <text-property name="heading">objname</text-property>
        </structure>
        <structure>
          <property name="columnName">logdescription</property>
          <property name="analysis">attribute</property>
          <text-property name="heading">logdescription</text-property>
        </structure>
        <structure>
          <property name="columnName">logvehicleid</property>
          <property name="analysis">dimension</property>
          <text-property name="heading">logvehicleid</text-property>
        </structure>
      </list-property>
      <list-property name="parameters"/>
      <structure name="cachedMetaData">
        <list-property name="resultSet">
          <structure>
            <property name="position">1</property>
            <property name="name">id</property>
            <property name="dataType">string</property>
          </structure>
          <structure>
            <property name="position">2</property>
            <property name="name">objname</property>
            <property name="dataType">string</property>
          </structure>
        </list-property>
      </structure>
    </oda-data-set>
  </data-sets>
</report>
```

```

    </structure>
    <structure>
        <property name="position">3</property>
        <property name="name">logdescription</property>
        <property name="dataType">string</property>
    </structure>
    <structure>
        <property name="position">4</property>
        <property name="name">logvehicleid</property>
        <property name="dataType">string</property>
    </structure>
</list-property>
</structure>
<property name="dataSource">Data Source</property>
<list-property name="resultSet">
    <structure>
        <property name="position">1</property>
        <property name="name">id</property>
        <property name="nativeName">id</property>
        <property name="dataType">string</property>
        <property name="nativeDataType">12</property>
    </structure>
    <structure>
        <property name="position">2</property>
        <property name="name">objname</property>
        <property name="nativeName">objname</property>
        <property name="dataType">string</property>
        <property name="nativeDataType">12</property>
    </structure>
    <structure>
        <property name="position">3</property>
        <property name="name">logdescription</property>
        <property name="nativeName">logdescription</property>
        <property name="dataType">string</property>
        <property name="nativeDataType">12</property>
    </structure>
    <structure>
        <property name="position">4</property>
        <property name="name">logvehicleid</property>
        <property name="nativeName">logvehicleid</property>
        <property name="dataType">string</property>
        <property name="nativeDataType">12</property>
    </structure>
</list-property>
<xml-property name="queryText"><![CDATA[table0#-TNAME-
#table0#:#[//logbook]#:#{id;STRING;@id},{objname;STRING;objname},{logdescription;ST
RING;logdescription},{logvehicleid;STRING;logvehicleid}]]></xml-property>
    <xml-property name="designerValues"><![CDATA[<?xml version="1.0"
encoding="UTF-8"?>
<model:DesignValues
xmlns:design="http://www.eclipse.org/datatools/connectivity/oda/design"
xmlns:model="http://www.eclipse.org/birt/report/model/adaptor/odaModel">
    <Version>2.0</Version>
    <design:ResultSets derivedMetaData="true">

```

```

<design:resultSetDefinitions>
  <design:resultSetColumns>
    <design:resultColumnDefinitions>
      <design:attributes>
        <design:identifier>
          <design:name>id</design:name>
          <design:position>1</design:position>
        </design:identifier>
        <design:nativeDataTypeCode>12</design:nativeDataTypeCode>
        <design:precision>-1</design:precision>
        <design:scale>-1</design:scale>
        <design:nullability>Unknown</design:nullability>
      </design:attributes>
      <design:usageHints>
        <design:label>id</design:label>
        <design:formattingHints/>
      </design:usageHints>
    </design:resultColumnDefinitions>
  </design:resultSetColumns>
  <design:criteria/>
</design:resultSetDefinitions>
</design:ResultSets>
</model:DesignValues>]]</xml-property>
  <list-property name="privateDriverProperties">
    <ex-property>
      <name>MAX_ROW</name>
      <value>-1</value>
    </ex-property>
    <ex-property>
      <name>XML_FILE</name>
    </ex-property>
  </list-property>
</oda-data-set>
<oda-data-set
extensionID="org.eclipse.datatools.enablement.oda.xml.dataSet" name="logtriplog"
id="9">
  <list-property name="columnHints">
    <structure>
      <property name="columnName">objname</property>
      <text-property name="heading">objname</text-property>
    </structure>
    <structure>
      <property name="columnName">trlfrom</property>
      <text-property name="heading">trlfrom</text-property>
    </structure>
    <structure>
      <property name="columnName">trluntil</property>
      <text-property name="heading">trluntil</text-property>
    </structure>
    <structure>
      <property name="columnName">trlmiles</property>
      <text-property name="heading">trlmiles</text-property>
    </structure>
  </list-property>

```

```

</list-property>
<list-property name="parameters">
  <structure>
    <property name="name">logbook</property>
    <property name="nativeName">logbook</property>
    <property name="dataType">string</property>
    <property name="nativeDataType">12</property>
    <property name="position">1</property>
    <property name="isInput">>true</property>
  </structure>
</list-property>
<structure name="cachedMetaData">
  <list-property name="resultSet">
    <structure>
      <property name="position">1</property>
      <property name="name">objname</property>
      <property name="dataType">string</property>
    </structure>
    <structure>
      <property name="position">2</property>
      <property name="name">trlfrom</property>
      <property name="dataType">string</property>
    </structure>
    <structure>
      <property name="position">3</property>
      <property name="name">trluntil</property>
      <property name="dataType">string</property>
    </structure>
    <structure>
      <property name="position">4</property>
      <property name="name">trlmiles</property>
      <property name="dataType">string</property>
    </structure>
  </list-property>
</structure>
<property name="dataSource">Data Source</property>
<list-property name="resultSet">
  <structure>
    <property name="position">1</property>
    <property name="name">objname</property>
    <property name="nativeName">objname</property>
    <property name="dataType">string</property>
    <property name="nativeDataType">12</property>
  </structure>
  <structure>
    <property name="position">2</property>
    <property name="name">trlfrom</property>
    <property name="nativeName">trlfrom</property>
    <property name="dataType">string</property>
    <property name="nativeDataType">12</property>
  </structure>
  <structure>
    <property name="position">3</property>

```

```

        <property name="name">trluntil</property>
        <property name="nativeName">trluntil</property>
        <property name="dataType">string</property>
        <property name="nativeDataType">12</property>
    </structure>
    <structure>
        <property name="position">4</property>
        <property name="name">trlmiles</property>
        <property name="nativeName">trlmiles</property>
        <property name="dataType">string</property>
        <property name="nativeDataType">12</property>
    </structure>
</list-property>
    <xml-property name="queryText"><![CDATA[table0#-TNAME-
#table0#:#[//logbook[@id="{?logbook?}"]/logtriplog]#:#{objname;STRING;objname},{trl
from;STRING;trlfrom},{trluntil;STRING;trluntil},{trlmiles;STRING;trlmiles}]]></xml-
property>
    <xml-property name="designerValues"><![CDATA[<?xml version="1.0"
encoding="UTF-8"?>
<model:DesignValues
xmlns:design="http://www.eclipse.org/datatools/connectivity/oda/design"
xmlns:model="http://www.eclipse.org/birt/report/model/adaptor/odaModel">
    <Version>2.0</Version>
    <DataSetParameters>
        <parameter>
            <design:ParameterDefinition>
                <design:inOutMode>In</design:inOutMode>
                <design:attributes>
                    <design:identifier>
                        <design:name>logbook</design:name>
                        <design:position>1</design:position>
                    </design:identifier>
                    <design:nativeDataTypeCode>12</design:nativeDataTypeCode>
                    <design:precision>-1</design:precision>
                    <design:scale>-1</design:scale>
                    <design:nullability>Unknown</design:nullability>
                </design:attributes>
            </design:ParameterDefinition>
        </parameter>
    </DataSetParameters>
    <design:ResultSets derivedMetaData="true">
        <design:resultSetDefinitions>
            <design:resultSetColumns>
                <design:resultColumnDefinitions>
                    <design:attributes>
                        <design:identifier>
                            <design:name>objname</design:name>
                            <design:position>1</design:position>
                        </design:identifier>
                        <design:nativeDataTypeCode>12</design:nativeDataTypeCode>
                        <design:precision>-1</design:precision>
                        <design:scale>-1</design:scale>
                        <design:nullability>Unknown</design:nullability>
                    </design:attributes>

```

```

    <design:usageHints>
      <design:label>objname</design:label>
      <design:formattingHints/>
    </design:usageHints>
  </design:resultColumnDefinitions>
<design:resultColumnDefinitions>
  <design:attributes>
    <design:identifier>
      <design:name>trlfrom</design:name>
      <design:position>2</design:position>
    </design:identifier>
    <design:nativeDataTypeCode>12</design:nativeDataTypeCode>
    <design:precision>-1</design:precision>
    <design:scale>-1</design:scale>
    <design:nullability>Unknown</design:nullability>
  </design:attributes>
  <design:usageHints>
    <design:label>trlfrom</design:label>
    <design:formattingHints/>
  </design:usageHints>
</design:resultColumnDefinitions>
<design:resultColumnDefinitions>
  <design:attributes>
    <design:identifier>
      <design:name>trluntil</design:name>
      <design:position>3</design:position>
    </design:identifier>
    <design:nativeDataTypeCode>12</design:nativeDataTypeCode>
    <design:precision>-1</design:precision>
    <design:scale>-1</design:scale>
    <design:nullability>Unknown</design:nullability>
  </design:attributes>
  <design:usageHints>
    <design:label>trluntil</design:label>
    <design:formattingHints/>
  </design:usageHints>
</design:resultColumnDefinitions>
<design:resultColumnDefinitions>
  <design:attributes>
    <design:identifier>
      <design:name>trlmiles</design:name>
      <design:position>4</design:position>
    </design:identifier>
    <design:nativeDataTypeCode>12</design:nativeDataTypeCode>
    <design:precision>-1</design:precision>
    <design:scale>-1</design:scale>
    <design:nullability>Unknown</design:nullability>
  </design:attributes>
  <design:usageHints>
    <design:label>trlmiles</design:label>
    <design:formattingHints/>
  </design:usageHints>
</design:resultColumnDefinitions>

```

```

        </design:resultSetColumns>
        <design:criteria/>
    </design:resultSetDefinitions>
</design:ResultSets>
</model:DesignValues>]]></xml-property>
    <list-property name="privateDriverProperties">
        <ex-property>
            <name>MAX_ROW</name>
            <value>-1</value>
        </ex-property>
        <ex-property>
            <name>XML_FILE</name>
        </ex-property>
    </list-property>
</oda-data-set>
</data-sets>
<styles>
    <style name="report" id="4">
        <property name="fontFamily">sans-serif</property>
        <property name="fontSize">10pt</property>
    </style>
    <style name="crosstab-cell" id="5">
        <property name="borderBottomColor">#CCCCCC</property>
        <property name="borderBottomStyle">solid</property>
        <property name="borderBottomWidth">1pt</property>
        <property name="borderLeftColor">#CCCCCC</property>
        <property name="borderLeftStyle">solid</property>
        <property name="borderLeftWidth">1pt</property>
        <property name="borderRightColor">#CCCCCC</property>
        <property name="borderRightStyle">solid</property>
        <property name="borderRightWidth">1pt</property>
        <property name="borderTopColor">#CCCCCC</property>
        <property name="borderTopStyle">solid</property>
        <property name="borderTopWidth">1pt</property>
    </style>
    <style name="crosstab" id="6">
        <property name="borderBottomColor">#CCCCCC</property>
        <property name="borderBottomStyle">solid</property>
        <property name="borderBottomWidth">1pt</property>
        <property name="borderLeftColor">#CCCCCC</property>
        <property name="borderLeftStyle">solid</property>
        <property name="borderLeftWidth">1pt</property>
        <property name="borderRightColor">#CCCCCC</property>
        <property name="borderRightStyle">solid</property>
        <property name="borderRightWidth">1pt</property>
        <property name="borderTopColor">#CCCCCC</property>
        <property name="borderTopStyle">solid</property>
        <property name="borderTopWidth">1pt</property>
    </style>
</styles>
<page-setup>
    <simple-master-page name="Simple MasterPage" id="2">
        <page-footer>

```

```

        <text id="3">
            <property name="contentType">html</property>
            <text-property name="content"><![CDATA[<value-of>new
Date ()</value-of>]]></text-property>
        </text>
    </page-footer>
</simple-master-page>
</page-setup>
<body>
    <list id="10">
        <property name="dataSet">logbook</property>
        <list-property name="boundDataColumns">
            <structure>
                <property name="name">id</property>
                <text-property name="displayName">id</text-property>
                <expression name="expression"
type="javascript">dataSetRow["id"]</expression>
                <property name="dataType">string</property>
            </structure>
            <structure>
                <property name="name">objname</property>
                <text-property name="displayName">objname</text-property>
                <expression name="expression"
type="javascript">dataSetRow["objname"]</expression>
                <property name="dataType">string</property>
            </structure>
            <structure>
                <property name="name">logdescription</property>
                <text-property name="displayName">logdescription</text-
property>
                <expression name="expression"
type="javascript">dataSetRow["logdescription"]</expression>
                <property name="dataType">string</property>
            </structure>
        </list-property>
    <detail>
        <data id="11">
            <property name="marginTop">10pt</property>
            <property name="resultSetColumn">objname</property>
        </data>
        <data id="12">
            <property name="resultSetColumn">logdescription</property>
        </data>
    </list id="30">
        <property name="dataSet">logtriplog</property>
        <list-property name="paramBindings">
            <structure>
                <property name="paramName">logbook</property>
                <simple-property-list name="expression">
                    <value type="javascript">row["id"]</value>
                </simple-property-list>
            </structure>
        </list-property>
    </list-property name="boundDataColumns">

```

```

        <structure>
            <property name="name">objname</property>
            <text-property name="displayName">objname</text-
property>
                <expression name="expression"
type="javascript">dataSetRow["objname"]</expression>
            <property name="dataType">string</property>
        </structure>
        <structure>
            <property name="name">trlfrom</property>
            <text-property name="displayName">trlfrom</text-
property>
                <expression name="expression"
type="javascript">dataSetRow["trlfrom"]</expression>
            <property name="dataType">string</property>
        </structure>
        <structure>
            <property name="name">trluntil</property>
            <text-property name="displayName">trluntil</text-
property>
                <expression name="expression"
type="javascript">dataSetRow["trluntil"]</expression>
            <property name="dataType">string</property>
        </structure>
        <structure>
            <property name="name">trlmiles</property>
            <text-property name="displayName">trlmiles</text-
property>
                <expression name="expression"
type="javascript">dataSetRow["trlmiles"]</expression>
            <property name="dataType">string</property>
        </structure>
    </list-property>
    <detail>
        <grid id="31">
            <property name="marginBottom">0pt</property>
            <column id="32"/>
            <column id="33"/>
            <column id="34"/>
            <row id="35">
                <cell id="36">
                    <data id="39">
                        <property
name="resultSetColumn">trlfrom</property>
                            </data>
                        </cell>
                    <cell id="37">
                        <data id="40">
                            <property
name="resultSetColumn">trluntil</property>
                                </data>
                            </cell>
                        <cell id="38">
                            <data id="41">

```

```
                                <property  
name="resultSetColumn">trlmiles</property>  
                                </data>  
                                </cell>  
                                </row>  
                                </grid>  
                                </detail>  
                                </list>  
                                </detail>  
                                </list>  
                                </body>  
</report>
```