IBM Cúram Social Program Management

Cúram Business Intelligence BIRT Developer Guide

Version 6.0.5

IBM Cúram Social Program Management

IBM

Cúram Business Intelligence BIRT Developer Guide

Version 6.0.5

Contents

Figures

Tables

Chapter 1. Introduction

1.1 About this Document

The purpose of this guide is to describe how the Cúram Business Intelligence and Reporting ToolsCúram Business Intelligence and Reporting Tools (BIRT) stack is used.

BIRT is an open source component which provides a development and execution environment for Cúram Business Intelligence and AnalyticsCúram Business Intelligence and Analytics (BIA) content. BIRT is used primarily as a tool for the development and execution of charting and tabular data with the aim of providing decision support information for front line staff (case workers), line managers and senior managers in the organization.

This document describes how BIA content is developed, integrated and executed in a runtime environment. The following are considered the key benefits for the assessment of BIRT Engine:

- Look and Feel: The visual appeal of the product release is paramount, BIRT supports the production of content that can be seamlessly integrated into the BIA Report Application.
- Licensing Terms and Cost: An important element is the ability to include basic functionality in the "Out of the Box" product using an open source version of the product; including a licensed product was not acceptable for this purpose. The second element was the licensing options for customers for more advanced features, it is critical that they are able to license tools at a cost which is proportional to the value they derive from its usage.
- Technical Merit: The technical merit of BIRT was evaluated against the three main areas of BIA functionality - Online Analytics, Reports, Dashboards and Charting.

1.2 Prerequisites

This guide is intended for any reader who will be using BIRT to create and integrate BIA content into the BIA Report Application.

1.3 Audience

Readers should have a good working knowledge of Java™ and BIRT.

Chapter 2. Business Intelligence Analytics (BIA) Overview

2.1 Business Features

The BIA component allows agencies to measure and monitor the performance of an organization, to detect gaps in processes, and to analyze the issues that are occurring. These agencies require a business intelligence solution that will provide them with the tools to support better decision making. BIA provides a solution that covers the complete spectrum of reporting tools and technologies that enable organizations to make valuable business decisions. This includes decision support information for front line staff (case workers), line managers and senior managers in the organization. The following areas are provided for:

- Embedded Analytics: Embedded Analytics are representations of aggregated data that can be interacted with by the user to construct alternative views/sub-groupings of the data which were not envisaged at design time. The infrastructure to construct embedded analytics is provided with BIA and a number of these analytics have been added out of the box to the Application. The purpose of this is to assist front line staff and managers in the decision making process. This is achieved by integrating the online application and the data warehouse which therefore provides interactive, summarized information in context.

- Interactive Dashboards and Reports: Interactive dashboards include the ability to publish graphically intuitive displays of information, including dials, gauges and traffic lights. These displays indicate the state of the performance metric, compared with a goal or target value. This data is a more in-depth view into the aggregated data in the business specific data warehouse. Reports provide the ability to create formatted and interactive reports with highly scalable distribution and scheduling capabilities. Infrastructure is provided in support of these and business specific reports will be added over time.

- Ad-hoc Reporting: The Business Intelligence infrastructure we provide, uses BIRT to render the charts as BIRT is an open sourced and low cost option for agencies that do not want an elaborate business intelligence offering. We also support more advanced offerings in the market place such as Actuate, which allows charts to be rendered in a format suitable for ad-hoc reporting. This provides easy access to the underlying reporting data, allowing a user to tailor a report to their specific needs. Please refer to our Actuate documentation for further reading.

2.2 Technical Features

BIA is comprised of the following components:

- A BIRT Engine infra-structure component Cúram BIRT ViewerCúram BIRT Viewer which executes BIRT charts and reports, this wraps the open source BIRT Engine, and is a J2EEapplication.

- The Cúram Enterprise FrameworkCúram Enterprise Framework BIA Manager component is responsible for managing the look-up of report templates when the BIRT Engine is in use. A BIRT render is also available which simplifies the process of placing BIRT charts and reports on our Report Pages.

- Our Administration application has a BIA configuration section, which contains sections to configure the viewer and sections to configure BIRT reports.

Chapter 3. BIRT System Landscape

3.1 Introduction

This section gives an overview of the BIRT architecture and system environment and describes the dependencies between the BIA third party tools that are supported.

3.2 BIRT System Environment

BIRT is an open source component which provides a development and execution environment for BIA content. It was selected for use following detailed analysis of the alternatives. Cúram V6Cúram V6 GA is the first release to incorporate BIRT technology, the list below is included to provide an overview of the key features.

- The BIRT Report Designer: The BIRT Report Designer is an Eclipse plug-in which allows a developer to write a BIRT report which can then be deployed into the BIRT Report Engine. This supports the use of a wide range of graphs (using the BIRT Charting Engine) as well as data listing. This is built on the BIRT Report Design Engine, which can be used by any Java application to create or modify report designs.
- The BIRT Charting Engine: The BIRT Charting Engine has a large number of built-in charts as well as support for user defined charts. The user defined charts are not constrained to any single technology, and so could be implemented in Adobe Flex, but all the built-in charts are implemented in Java.
- BIRT Report Engine: The BIRT Report Engine is the runtime component that renders a report design. It can produce output in a number of formats including HTML and PDF.

3.3 BIRT, Eclipse and Actuate Software Dependencies

The Actuate iServer component is an optional direct replacement for the BIRT component and offers additional support for on-line customization of pre-built content and new content creation. It is a propriety server component, which includes its own application server (non-J2EE)

The BIRT and Actuate versions are linked; Actuate 11 supports BIRT 2.6. The versions were selected based on two key factors:

- Actuate 11 was released in 2010 and hence is the most recent version available at the time Cúram v6 was released.
- Both Actuate 11 and BIRT 2.6 offer significant advances over the previous releases

Please note that Actuate 11 is an optional component and is not included in any release, hence clients must purchase Actuate separately. The version dependencies are:

- The BIRT version used is BIRT 2.6, Eclipse version 3.6 is required to support BIRT 2.6.
- BIRT 2.6 requires Eclipse 3.6, our development process supports the ability of any of our releases to move to newer versions of BIRT (as new features become available).
- Actuate 11 is based on BIRT 2.6.

Chapter 4. Build Process Reviewed

The introduction of BIRT content has resulted in a new step being added to the build process. This additional step publishes BIRT content from the BIContent folder to the BIRT Viewer.

The BIRT Viewer executes BIRT content which is then rendered on a BIA Report page. The new build processes is reviewed in the next section:

4.1 Build Process Note

Important: When viewing the Client Application using Eclipse the follow process is now required:

Buid the server: Change to the `EJBserver` folder and build the server (build server).

Buid the client: Change to the `webclient` folder and build the client (build client).

Publish BIRT content for execution: Change to the `BIContent` folder and publish your BIRT content (build client.birt).

Refresh Eclipse and Start Tomcat: Start Apache Tomcat and view the client application.

Chapter 5. Viewing and Integrating BIA Content

5.1 Introduction

This section will guide you through the processes that are required to view BIRTContent in the client application. The following sections will guide you through:

- How to import into Eclipse the projects that are required to view BIRT content.
- The post installation configuration steps required to view BIRT content (ensuring BIA content is rendered onto BIA report pages).
- How integrate BIRT content into BIA Report pages, i.e. how to use the BIRT renderer and server side classes to embed BIRT content into BIA Report pages.

5.2 Viewing BIA Reports Within Eclipse

Due to the software version requirements, Eclipse version 3.6.2 is mandated for "standard" BIA Report development. See the Third Party Tools Guide for guidance on Eclipse versions and installation. The following instructions relate to the Eclipse environment used for BIA Report development.

See the following paragraph for a description of the development environment required for BIRT development.

Please note if you are viewing existing source database content you do not need to install a BIRT development environment. If you are using H2® as your RDBMS ensure it is started in remote mode.

5.3 Eclipse - BIRT Development

BIRT is an (Eclipse) open source component which provides a development and execution environment for BIA content.

The use of a second Eclipse instance for BIRT development is required as standard BIA Report development requires Eclipse 3.6.2 and BIRT development requires Eclipse 3.6.1. See the Third Party Tools Guide for a full explanation. Please note, that a BIRT development environment is not required to view BIRT content shipped with any delivery.

5.4 Viewing BIA Content Within Eclipse

This section will guide you through the process of configuring the Eclipse environment used for BIA Report development.

1. Import the CuramBIRTViewer project. To import the CuramBIRTViewer project into eclipse, go to `File->Import->General-> Existing Projects into Workspace` and click Next. Select the root directory to be `CURAM_DIR\BIApp \CuramBIRTViewer` and hit Finish.

2. Select the CuramBIRT Viewer project in eclipse, then `window->preferences->java-> compiler->compile compliance level.` and set to 1.5

3. Select the CuramBIRTViewer project in eclipse, right-click and select `Tomcat Project-Update context definition`.

4. From the Window Menu select `Preferences->Tomcat->JVM Settings` and on the classpath window select the directory button and add in the full path to the `CURAM_DIR\EJBServer \project\properties` directory from your installed location.

5. To view licensed content via the Reports Tab (ignore this section if you do not have a Reports tab), then from the Window Menu select `Preferences->Tomcat->JVM Settings` and on the classpath window select the directory button and add in the full path to the `CURAM_DIR\Reporting\project\ properties` directory from your installed location.

6. You must now publish all BIA content to the BIRT Viewer, from the directory `CURAM_DIR\BIContent` execute the command `build client.birt`.

7. Start Tomcat, to verify your BIA viewer is open for business execute the follow URL:

   a. http://localhost:9080/CuramBIRTViewer/

   b. Execute the report `/birtsamples/test.rptdesign` to verify the viewer is serving static BIRT reports.

   c. Execute the report `/components/core/birt/curamsamples/ SampleBIReportPDF.rptdesign` to verify the viewer is serving dynamic BIRT

   d. Execute the report `/components/core/birt/curamsamples/ SampleBIReportFlex.rptdesign` to verify the viewer is serving dynamic Flex based BIRT reports.

8. If you are using H2 as your RDBMS ensure it is started in remote mode.

9. If you are experiencing issues please refer to the Troubleshooting guide at the end of the document.

10. To publish content to the CuramBIRTViewer application execute the following steps.

If you are experiencing issues please refer to the Troubleshooting guide at the end of the document.

For BIA Report Dependencies, please refer to the Appendices.

5.5 Developing BIA Content/ BIRT Reports

The section entitled `Developing BIRT Reports` describes how to create or modify BIRT report design documents.

5.6 Integrating BIRT Content

The process for integrating BIRT content is described briefly in the following points and explained fully in the sections below, please note that the client must follow the `Cúram Server Developer's Guide` at all times.

1. See the section entitled `UIM Example` for an example UIM code snippet.

2. See the section entitled `Report Name` for an example report name.

3. See the section entitled `Report Configuration` for on guidance on how reports are configured and what DMX is required.

4. See the sections `POD Facade Implementation` or `General Facade Implementation` for example on how to embed BIRT in PODS and BIA Report pages respectively.

5.6.1 UIM Example

A new domain definition has been created for BIRT reports, i.e. `BIRT_REPORT`. For BIRT content to be displayed on a Report Page this domain definition must be referenced/configured for the struct field in question. When embedding a BIRT report in a POD the domain definition must be set to `BIRT_POD`. BIRT_POD is is a POD styling definition which must have been previously set.

5.6.2 Report Na

The name logical BIRT Report name is coded to a constants file, e.g. see `Const.java`. The report name is a logical name with the physical report returned by a call using the CEF utility `BIHelper` component, the report DMX (Data Mining Extensions) configuration entries must be available in the database or this method is throw an exception. See the following section on DMX configuration for an example.

```
/** Investigation Summary report. **/
  public static final String gkInvestigationsSummaryReportName =
      "InvestigationsSummary";
```

5.6.3 Reporting Configuration - DMX

Once the BIRT report is fit for purpose you must create a DMX entry for your report. The report DMX entry must be present your custom initial demo data, for examples see the DMX file `BIREPORTCONFIGURATION.dmx` Please note that the client must follow the `Cúram Server Developer's Guide` at all times, also note that the BIHelper class will throw a runtime exception if the report logical name cannot be found in the BIReportConfiguration entity. See the section below for a example DMX configuration entry.

```
<row>
        <attribute name="biReportConfigurationID">
          <value>2002</value>
        </attribute>
        <attribute name="reportName">
          <value>AuditPlanSummaryBarChart</value>
        </attribute>
        <attribute name="reportCategory">
          <value>RC2001</value>
        </attribute>
        <attribute name="reportFileName">
          <value>components\core\birt\CaseAudit\
            AuditPlanSummaryBarChart.rptdesign</value>
        </attribute>
        <attribute name="reportServlet">
          <value/>
        </attribute>
        <attribute name="width">
          <value>100%</value>
        </attribute>
        <attribute name="height">
          <value>280</value>
        </attribute>
        <attribute name="scrolling">
          <value>RS2003</value>
        </attribute>
        <attribute name="reportFrameborder">
          <value>0</value>
        </attribute>
        <attribute name="description">
          <value>Audit Plan Summary</value>
        </attribute>
        <attribute name="recordStatus">
          <value>RST1</value>
        </attribute>
        <attribute name="versionNo">
          <value>1</value>
        </attribute>
    </row>
```

5.6.4 UIM Example

The following is a UIM snippet to include a BIRT report in a BIA Report page.

```
<SERVER_INTERFACE
    CLASS="ProductDelivery"
    NAME="DISPLAY"
    OPERATION="getReassessmentResultsChart"
    PHASE="DISPLAY"
  />
  <FIELD ALIGNMENT="CENTER">
    <CONNECT>
      <SOURCE
        NAME="DISPLAY"
```

```
        PROPERTY="report"
      />
    </CONNECT>
  </FIELD>
```

5.6.5 POD Facade Implementation Example

The following is a java code snippet which includes a BIRT report in a BIA Report page.

```
protected Document getBIRTReportData()
      throws AppException, InformationalException {

  ...

  reportData = biHelper.getReportData(
        PodsConst.kCaseloadSummaryBarChartBIReportName,
        reportParameters);


  return reportData;
}
```

The following is a java code snippet which creates a pod with links to new investigation and list of investigation pages.

- workspaceDocument: Document to which the investigation summary pod is added
- contexts: A map of contexts available to the Pod
- Node: represents the pod structure to be loaded

```
public Node createPod(final Document workspaceDocument, final Map
  <String, Object> contexts)
  {

  try {

    ...

    PodBuilder pod =
    PodBuilder.newPod(workspaceDocument,
    PODTYPE.INVESTIGATIONSSUMMARYPOD);

    pod.addContent(chart, rc);

    return pod.getWidgetRootNode();

} catch (Exception e)

{ throw new AppRuntimeException(e); }
}
```

The following is java code which gets the investigation summary details

- param selection: Selected range for which investigations summary details is required
- return investigations summary document
- throws AppException: Generic Exception Signature
- throws InformationalException: Generic Exception Signature

```
protected Document
  getBIRTSummary(final String selection)
  throws AppException, InformationalException {

  ...

  //Building Report Parameters
  Date startDate = Date.getCurrentDate();
  Date endDate = Date.getCurrentDate();
```

```
Calendar calendar = startDate.getCalendar();
calendar.add(Calendar.MONTH, -CuramConst.gkOne);
startDate = new Date(calendar);

reportParameters.put(
PodsConst.knvestigationSummaryStartDateParameter,
startDate.toString());

reportParameters.put(
PodsConst.knvestigationSummaryEndDateParameter,
endDate.toString());

return biHelper.getReportData(
CuramConst.gkInvestigationsSummaryReportName,
reportParameters);

}
```

5.6.6 Non POD Facade Implementation

The following is a java code snippet which adds content to a BIA page (e.g. a content panel) with format XML data for an employer work force tab details:

- parameter caseID: Employer concern role id
- return ContentPanelBuilder
- throws InformationException: Generic Exception Signature
- throws AppException: Generic Exception Signature

```
protected ContentPanelBuilder getBIRTReportDetails(
  final long concernRoleID)

  throws AppException, InformationalException
{

  ContentPanelBuilder contentPanelBuilder =
  ContentPanelBuilder.createPanel(
  CuramConst.gkEmployerWorkforceDetail);

  contentPanelBuilder.addRoundedCorners();

  Map <String, String> reportParameters =
  new HashMap<String, String>();

  reportParameters.put(CuramConst.gkParam_ConcernRoleID,
  String.valueOf(concernRoleID));

  WidgetDocumentBuilder reportBuilder = biHelper.getDocumentBuilder(
  CuramConst.gkBIRTProspectEmployerWorkforceReport, reportParameters);

  contentPanelBuilder.addWidgetItem(reportBuilder, CuramConst.gkStyle,
  CuramConst.gkStyleBirt); return contentPanelBuilder;

}
```

The following is a java code snippet which adds content to a BIA page (e.g. a content panel) with formatted XML data for an employer work force tab details:

- parameter caseID: Employer concern role id
- return ContentPanelBuilder
- throws InformationalException: Generic Exception Signature
- throws AppException: Generic Exception Signature

```
protected ContentPanelBuilder getBIRTReportDetails(
final long concernRoleID)
   throws AppException, InformationalException
   {

   ContentPanelBuilder contentPanelBuilder =
   ContentPanelBuilder.createPanel( CuramConst.gkEmployerWorkforceDetail);

   contentPanelBuilder.addRoundedCorners();

   Map <String, String> reportParameters = new HashMap<String, String>();

   reportParameters.put(
   CuramConst.gkParam_ConcernRoleID, String.valueOf(concernRoleID));

   WidgetDocumentBuilder reportBuilder = biHelper.getDocumentBuilder(
   CuramConst.gkBIRTEmployerWorkforceReport, reportParameters);

   contentPanelBuilder.addWidgetItem(reportBuilder,
   CuramConst.gkStyle, CuramConst.gkStyleBirt);

   return contentPanelBuilder;

}
```

5.7 Writing Portable BIRT Reports across DB2 databases

The following steps need to be applied to BIRT reports if they need to be run on DB2 databases for both ZOS and non-ZOS platforms.

1. Create a new 'Viewer Configuration' parameter, via the CURAM Application, called 'db2numbertype' with a parameter value of 'bitdata'

2. Copy and paste the existing dataset for the relevant report in BIRT Eclipse and

    • Rename this new dataset to be 'zos' - all in lowercase

    • Edit the dataset and remove any parameters specified

    • Go to the query and remove any lines with question marks for parameters

3. Append the following code to the 'Before Factory' script area and enter the relevant chart name in the code below

```
if(params["db2numbertype"].value == "bitdata") {

  mytable = reportContext.getDesignHandle().
  findElement("<ENTER CHART NAME HERE>");

  mytable.setProperty( "dataSet", "zos" );

}
```

4. Click on the 'zos' data set and then select the 'script' tab Tab (located at the bottom of the 'Report Designer' window) and choose the 'beforeOpen' Script option. Edit the following code snippet to match your query needs and paste it in the report.

```
importPackage(Packages.java.math);
importPackage( Packages.biapp );

var param_<append parameter name here> = BigDecimal.valueOf
(params["<ENTER Parameter name here>"].value);

param_<append parameter name here>_long_to_string = UniqueIDUtil.longToString
(param_<append parameter name here>);

this.queryText = " <Enter the SQL select statement here>"
```

Note: The parameter name is case sensitive. You must enter it as it appears in the report.

E.g. The following is an example of a completed code section. You will need to create the relevant parameters based on the query given. In this example there is only 1 parameter called 'concernRoleID', this parameter is passed in to the report and then it is converted to be used in the SQL query as follows:

```
importPackage(Packages.java.math);
importPackage( Packages.biapp );

var param_concernroleid = BigDecimal.valueOf(params["concernRoleID"]
.value);

param_concernroleid_long_to_string = UniqueIDUtil.longToString
(param_concernroleid);

this.queryText = "SELECT  numberPermanentStaff as numberStaff"
    + ", 'PermanentStaff' as staffType"
    + " FROM Employer"
    + " WHERE concernRoleID ="  + param_concernroleid_long_to_string
    + " UNION"
    + " SELECT"
    + " numberCasualStaff as numberStaff,"
    + " 'CasualStaff' as staffType"
    + " FROM Employer"
    + " WHERE concernRoleID ="  + param_concernroleid_long_to_string
```

Chapter 6. Authentication and Entitlements

6.1 Overview

The BIRT Viewer application ensures that only authenticated users can access the BIRT Viewer application, therefore ensuring only authenticated users can execute BIRT reports.

The inclusion of an additional security measure in form of entitlement checking or authorization checking is not included in the BIRT Viewer application at present. It is critical this is reviewed if any steps are taken to include BIRT content in an externally facing (public facing) client application.

Chapter 7. Developing New Reports

7.1 Eclipse Installation

BIRT is an (Eclipse) open source component which provides a development and execution environment for BIA content. We recommend the use of a second Eclipse instance for BIRT development. This second instance ensures that newer BIRT versions can be used without changing the development process, which would otherwise be required should a single Eclipse environment be used.

An all in one Eclipse and BIRT installation is used for internal development.

See the Third Party Tools Guide and the BIRT Developer Guide which provide a full explanation. Please note the Eclipse environment for BIRT development must be installed into a folder called <eclipsebirt>, if you choose a different folder name see the trouble shooting section "Problem: build client.birt fails to execute" on how to configure the BIRT Eclipse location.

7.2 Eclipse Post Installation

It is recommended that you create a batch file to start your BIRT Eclipse development environment. e.g.
- birteclipse.bat this will be located at the root of your project.
- Please note that the BIRT eclipse environment is only used for the development of BIRT content, you must use your "normal" eclipse development for "standard" application development work.
- You will be prompted to name your work space, we recommend that you give this workspace a descriptive title, e.g. <workspaceBIRT>.

7.2.1 Configuration for BIA Charts

This section will guide you through the steps required to view BIA charts in Eclipse, BIA charts are BIRT charts which have been configured to render in Flex. Build environment support to automate the steps below may be provided in a future service pack release.

The commands below are based on BIRT 2.6.1, with the version number part of the directory name changing as the version of BIRT is upgraded, e.g. when upgrading from BIRT 2.6.0 to 2.6.1 the following directory name changed from `plugins\org.eclipse.birt.report.data.oda.jdbc_2.6.0.v20100524` to `plugins\org.eclipse.birt.report.data.oda.jdbc_2.6.1.v20100913`.

1. Copy the JDBC drivers you require from `CURAMSEDJ_DIR\drivers` e.g. the Oracle or DB2® JDBC drivers to `TOOLS_DIR\eclipsebirt\plugins\org.eclipse.birt.report.data.oda.jdbc_2.6.1.v20100909\drivers` where TOOLS_DIR is the parent directory of your BIRT Eclipse installation.

2. Copy all files in the directory (the flex extensions) from `CURAM_DIR\BIApp\CuramBIRTViewer\WebContent\webcontent` (i.e. just the files located in webcontent are required a deep copy of the whole webcontent is not required) to `TOOLS_DIR\eclipsebirt\plugins\org.eclipse.birt.report.viewer_2.6.1.v20100913\birt\webcontent`.

3. Copy 3 jar files (the flex extensions) from `CURAM_DIR\BIApp\CuramBIRTViewer\WebContent\WEB-INF\platform\plugins` to `TOOLS_DIR\eclipsebirt\plugins`.

4. Copy 1 Jar file from the directory (the flex extensions) from `CURAM_DIR\BIApp\CuramBIRTViewer\WebContent\WEB-INF\bicontent\resources\scriptlib` (e.e. just the file with a naming convention of curam-birt-jar) to `TOOLS_DIR\eclipsebirt\plugins\org.eclipse.birt.report.viewer_2.6.1.v20100913\birt\scriptlib`.

7.2.2 Import BIRT Projects

Import the following projects into BIRT Eclipse:

- Import the <Curam BIA Content> project into eclipse, go to File->Import->General->Existing Projects into Workspace and click Next. Select the root directory to be <Curam_DIR>\BIContent and hit Finish. This is the project within which you will work.
- Import the <Curam BIRT Viewer Samples> project into eclipse from the <Curam_DIR>\BIApp\ CuramBIRTViewer\components\BIContent directory. There are a number of sample reports contained within this project which may be helpful.
- The CuramBIRTViewer project also needs to be imported. To import the <CuramBIRTViewer> project into eclipse, go to File->Import->General->Existing Projects into Workspace and click Next. Select the root directory to be Curam_DIR\BIApp and hit Finish.
- Select the CuramBIRTViewer project in eclipse, right-click ->go to properties->Java Build Path-> Libraries-> Add variable->Configure variable->New-> TOMCAT_HOME (set to DEV_ENV\tomcat).
- Select the Cúram BIRT Viewer project in eclipse, window->preferences ->java->compiler ->compile compliance level-> set to 1.5.

7.2.3 Configure BIRT Projects

- From the menu -> Preferences -> Report Design -> Resource, Select the Workspace button.
- Select the "CuramBIRTViewer Samples" project and select the resources folder, Select OK.
- The resource folder text box should now contain the value ${workspace_loc:Curam BIRT Viewer Samples(BI Stream)/resources}, Hit OK.
- Select Cúram BIRT Viewer Samples and right click and select properties.
- Select Report Design -> Resources -> Configure Workspace settings (top right hand side).
- Select OK and OK again (even though you are not changing anything this need to be completed).
- Verify your common resource are available:
  1. From the <Curam BIRT Viewer Samples> Project select the resources folder->library and double click the CEFLibrary file.
  2. The library will open in the resource explorer, (Window->show view->Resource Explorer). You can now use the standard library features.
- To verify your environment is correctly configured:
  1. Select the "CuramBIRTViewer Samples" project, within tests/curamsamples open the sample report SampleBIReportPDF.rptdesign or SampleBIReport.rptdesign.
  2. To view the report select the Preview tab or from the View Report short cut on the tool bar. You have opened and executed your first report.
- You are now ready to create your first report with data from the operational database.

7.3 Creating your first Sample Report for the BIA Reporting Application

- Select <Curam BI Content> project.
- Go to File, new, Report and if creating an overpayment sample report save the report to here <Curam_DIR>\BIContent\components\core\birt.
- Verify that the library is still available in the Resource View (from the previous steps).
- Go to the Outline View->libraries->right click and select use library, select the "CEFLibrary" from the Shared Resources\resources\library folder in the popup.
- Go to the Resource Explorer, expand "CEFLibrary" in Shared Resources\resources\library and complete the following:
  1. Data Sources ->CuramDB, right click and select add to report.
  2. Select CuramDM if creating a datamart report.
  3. Select Data Sources and if no current data source exists, right click and choose new data source. Edit the new/current data source to point to your local source database, i.e right click on datasource, click edit and set the following values:

- – Driver Class: com.ibm.dbe.jcc.DB2Driver (v3.57)
- – Driver URL: jdbc:db2://silverstreak:50000/curam
- – Username: db2admin
- – Password: ********
- The settings within this panel will be used for the Eclipse embedded preview mode only. Test your connection.
- Create a new data set using the data source configured in the previous step, paste in the SQL as an example select casetypecode as status, count (casetypecode) as casecount from caseheader group by casetypecode. If you are unable to view all entities this is due to the fact that BIRT puts a cap on the number of tables displayed when creating a dataset. Please see Troubleshooting Guide for steps to change this setting.
- Add a chart to the report we recommend that you copy and paste in a chart from one of the reports in the Cúram BIRT Viewer Samples project. Ensure the chart uses the newly created data set. Chart styling will be enforced through CSS in future versions of the Cúram BIRT Viewer.
- In order to test your report via the application pages, copy the report to be executed from the Cúram BIRT Viewer:
  1. If your Cúram BIRT Viewer is already configured (see 4.1 Configuring the Cúram BIRT Viewer), then execute the following command from the <Curam_DIR>/BIContent directory to make your reports available to the Cúram BIRT Viewer - appbuild client.birt.
  2. Run your report from the Cúram BIRT Viewers test window or view from the application page:
     - `http://localhost:9080/CuramBIRTViewer/`

7.4 Creating your first Sample Report for a BIA licensed tab

About this task

Only follow these steps if you are creating a report for a licensed BIA Reporting tab.

Procedure

1. You will need to place a BIBootstrap.properties file in the <Curam_DIR>\Reporting\project\ properties directory. Copy the sample BIBootstrap.properties.sample file and rename it to BIBootstrap.properties. Change the centraldm properties to point to to datamart you intend to work from. You only need to configure the properties for the datamart, i.e. centraldm. #

   You will also need to encrypt the passwords before entering them into the BIBootstrap.properties file. Encrypting a password is done as follows

   a. Open a command prompt from Reporting\components
   b. Run the following: 'appbuild encrypt.password -Dpassword=<p>' where <p> is the assigned password to be encrypted
   c. Enter the full encrypted password returned, for example: qqnscP4c4+s== as the password in the BIBootstrap.properties file.

   See example below.

   ```
   db.type=<databasetype> (e.g. ORA or DB2 - just put this in the property file once)
    # ORACLE connection properties for the data mart
   centraldm.db.server=<machinename>
   centraldm.db.port=1521
   centraldm.db.name=<servicename>
   centraldm.db.SID=<SID> (e.g. ORCL)
   centraldm.db.username=datamart
   centraldm.db.password=<encrypted password> (e.g. qqnscP4c4+s==)
   ```
2. Select <Curam BI Content> project.

3. Go to Window Menu -> Preferences -> Tomcat ->JVM Settings. On the classpath widow select the button directory and add in the full path to the <Curam_DIR>\Reporting \project\properties directory for your stream.

4. Create a new report e.g. if creating a report for the CEF application which is associated with a license then save the report to this location:

   <Curam_DIR>\BIContent\components\reportingForCEF.

   Platform reports associated with a license should be saved here, etc.

   <Curam_DIR>\BIContent\components\reportingForPlatform.

5. Verify that the library is still available in the Resource View (from the previous steps).

6. Go to the Outline View->libraries->right click and select use library, select the "CEFLibrary" from the popup.

7. Go to the Resource Explorer, select CEFLibrary and do the following:

   a. Data Sources->CuramDM, right click and select add to report.

   b. Select the data source and right client to edit the data source to point to your local source database by editing the values in the data source as follows:

      - Driver Class: com.ibm.db2.jcc.DB2Driver (v3.57)
      - Driver URL: jdbc:db2://silverstreak:50000/curam
      - User Name: db2admin
      - Password: ********

      The settings within this panel will used for the Eclipse embedded preview mode only. Test your connection.

8. Create a new data set using the data source configured in the previous step, paste in the SQL as an example

   select casetypecode as status, count (casetypecode) as casecount from caseheader group by casetypecode.

9. Add a chart to the report we recommend that you copy and paste in a chart from one of the reports in the Cúram BIRT Viewer Samples project. Ensure the chart uses the newly created data set. Chart styling will be enforced through CSS in future version of the Cúram BIRT Viewer.

10. In order to test your report via the application pages, copy the report to be executed from the Cúram BIRT Viewer:

    - 1. If your Cúram BIRT Viewer is already configured (see Eclipse Installation section), then execute the following command from the web client directory to make your reports available to the Cúram BIRT Viewer: appbuild client.birt.

    - 2. Run your report from the Cúram BIRT Viewer test window or view from the application page

      `http://localhost:9080/CuramBIRTViewer/`

7.5 BIA Application Content Compliance

To take-on a one of our new versions do not modify any report designs directly, copy the report and then make any changes you required.

7.6 Data Sources and JDBC Connections

The BIRT Infrastructure defines two data sources for use, namely:

- CuramDB: This is the data source name that must be used for all BIRT content when querying the application database.

- CuramDM: this is the data source name that must be used for all BIRT content when querying the application reporting data-mart.

The data-source must be manually create on the application servers, see the Third Party Tools Guides.

7.7 Internationalization and Localization

BIRT reports support localization and internationalization. Please follow the below steps to localize your BIRT report. When creating your BIRT Report also create corresponding.properties files in the same location. See naming convention section below.

- Create report with corresponding default.properties file. These are the 2 files you will deliver e.g.
  1. SampleWithTextLocalised.rptdesign
  2. SampleWithTextLocalised.properties
- Run the client.birt command to publish these properties files so they are available to BIRT Eclipse.
- In Report Designer open the report you wish to localize.
- In the layout pane of the report click anywhere on the white space. Go to the Properties section and click on Resources.
- In the Resource file field click browse select the appropriate.properties file from the Properties folder.
- For each label, go the Localization tab in the Property Editor and choose the appropriate resource from the list for the Content key property.
- Test the report in BIRT Eclipse Click on the menu->Preferences->Report Design->Preview and choose your locale. Click ok and run the report. It should display the text for the corresponding locale you have chosen.
- Test the report running in Tomcat - When running report append the following parameter to the end of the report URL &__locale=en / &__locale=en_US / &__locale=de_DE.
  – E.G. MyReport.rptdesign&__locale=en_US
- These.properties file names should follow the convention "ReportName_lang_country.properties", where lang and country are two-letter ISO codes. For example for German translation use "myreport_de_DE.properties", for US English use "myreport_en_US.properties".
- The ".properties" files must be ASCII encoded. All characters with codes above 127 ASCII must be escaped using this pattern: \uNNNN, where NNNN is the four-hex-digit Unicode representation of the original character. In theory you could escape all characters, but really it is not needed for those under 128 ASCII.
- The contents of the properties files should follow standard BIA naming conventions:
  – Help.PageDescription=This page allows you to add a text translation to a localizable text record. A localizable text record allows for application text to be localized.
  – PageTitle.StaticText1=Add Text Translation.
  – ActionControl.Label.Save=Save.
  – ActionControl.Label.Save.Help=The Save action updates the record using the information entered on the page.
  – ActionControl.Label.Cancel=Cancel.
  – ActionControl.Label.Cancel.Help=The Cancel action dismisses the page.
  – Cluster.Title.Details=Details.
  – Field.Title.Language=Language.
  – Field.Title.Language.Help=The language type for the text translation e.g. en, en_US.
  – Field.Title.Text=Text.
  – Field.Title.Text.Help=The actual translated text.
  – Report.Title = Title of Report
  – Report.chart.xaxis.title = X-axis title, e.g. time
  – Report.chart.yaxis.title = Y-axis title, e.g. no. of people participating
  – Report.chart.title = Title of Chart.
  – Report.chart.yseries1 = Series 1 of Y-values for comparing multiple rows

7.8 Chart Types/Formats Supported

An extension to the open source BIRT chart API is provided which allows BIRT charts to be rendered in Flex. Every feature and every setting that is available in a BIRT chart is not currently supported in our BIA Reporting flex equivalent. This section describes the features supported.

The BIA Reporting Charting Flex extension currently supports the following features.

7.8.1 Chart Types

The following chart types are supported:

- Pie Chart.
- Bar Chart - vertical and horizontal.
- Line Chart (vertical bar chart only).

7.8.2 Chart Legends

With each of these chart types the legend can be defined to be displayed above, below, left or right of the chart.

7.8.3 Chart Titles

The chart title is supported in the Flex chart; however, most charts will be rendered in a Pod or within a cluster. Both of these elements will already have a title, so in most cases BIRT chart titles will not be enabled.

7.8.4 Axis Labels

The label titles are now also supported in the Flex chart; however, most charts will be rendered in a Pod or within a cluster. Both of these elements will already have a label title, so in some cases BIRT chart labels may not be needed.

7.8.5 Charts with Multiple Series

When multiple series are required in a chart the Value Y Series features must be used to add series to a chart. The Optional Y Series Grouping feature is not supported in Flex. There is a known issue in the BIRT 2.6.0 release which limits the use of this feature in any event.

Chapter 8. Design Guidelines

8.1 Real Time KPI Design Guidelines

There are two classifications of Key Performance Indicators (KPIs) that use the operational database for their data:

1. Embedded Analytics charts and graphics that are embedded on standard BIA Report Pages.
2. Reporting Analytics/Dashboards charts and graphics that are associated with a Reporting license.

It is possible that either an Embedded analytic or a Reporting KPI will use the Cúram transactional system as their data source. On a case by case basis an assessment of each KPI must be undertaken to access:

1. If the KPI is performant in its own right?
2. Does the KPI adversely affect the performance of other BIA users?

If the answer to either of the previous questions is yes, it is recommended to take one of the below courses of action. In terms of finalizing the summary table design and next steps, these are the tactical options:

1. Review your BIRT content and apply all performance improvement techniques available to you and benchmark to verify your KPI is per formant.
2. If option 1 does not resolve your performance issues then design a summary table with as many de-normalized columns as needed to resolve your performance issues.

8.2 Summary Table Design Guidelines

Before starting the design of your summary table, please verify the following points still hold true:

- That the business need for this KPI to be real time is absolute.
- That the performance of the SQL for this KPI against the transaction tables is not acceptable or its impact is negative on other database usage.

Below are general design criteria for summary tables:

1. The summary table must have "de normalized" columns to help avoid table searches/scans, thus increasing the performance of record identification process, for example:
   - Adding Case Super, Case Owner and Case Type columns to a summary table might mean that the need to hit the Case header table to identify what case records are required is eliminated. The identification of the records is now carried out against the summary table.
2. The summary table should also contain other keyed columns that facilitate the retrieval of other related data items if needed, this also allows the summary table to be future proofed to some degree.
   - For example, a "CaseID" column on a summary table provides access to other case header data, but vitally the record set has already been identified in point 1 so joining to the case header table is as performant as possible.

8.3 BIRT Development Standards

Please follow the below standards when developing Reports in BIRT.

8.3.1 Naming Standard

Do not leave any spaces in the file name.

- **CCS Report Name Example:** P15_KPI1_PlacementStability.rptdesign

- **CEF Report Name Example:** OverpaymentCaseFinancialReport.rptdesign
- **Data Source Name:** curamdm
- **Data Set Name:** MonthlyPlacementStability (CCS e.g.)
- **Report Parameter Name:** P_Org
- **Data Set Parameter Name:** P_DS_Org

8.3.2 External Parameters

The following guidelines are for external parameters that are being passed into the BIRT report:

- **Operational IDs** (e.g. caseID, concernroleID etc.): Set parameter as 'Decimal' in report. Param is passed without any quotes.
- **Numbers** (excluding Operational IDs): Set parameter as Integer in report. Param is passed without any quotes.
- **Strings:** Set parameter as String in report. Param is passed without any quotes.
- **Dates:** Set parameter as Date in report. Param is passed without any quotes. Date must be in the form MM/DD/YYYY.

8.3.3 SQL

- All SQL should be written to return the minimum number of records back to BIRT.
- If possible, all calculations and aggregations should be performed at the database level.
- SQL should be portable, i.e. it should be able to run against both Oracle and DB2.
- Do not user vendor specific functions.

8.3.4 Chart Type

Please set the Output Format to be PNG. This is set in the Select Chart Type tab of the Edit Chart window.

8.3.5 Sorting Columns on X-Axis E.g. Displaying Month Name

Please note that the recommended approach for a Data-mart KPI is:

- The SQL statement should join to the DM_DIMTIMEPEROD dimension to get the month short name or month long name.
- No java script is required when displaying short or long month names.

If you need to rename the labels to something that does not exist on the database then the instructions below show how axis labels can be overridden.

If an Optional Y Series Grouping is added to a Report then whatever is displayed on the X-Axis is automatically defaulted in as the Sort Order. This means that if the X-Axis is displaying Month Names then they will be sorted alphabetically, which is incorrect.

Please follow these steps to resolve the problem:

- Put the month number on the x-axis.
- Sort it in Ascending order.
- On the Layout tab highlight the chart and click the Script tab
- Select the beforeDrawAxisLabel function
- See this URL for sample code:

```
http://www.birt-exchange.org/org/forum/index.php/topic/
        10988-changing-x-axis-labels-only/
```

8.4 Data-mart KPI Development - Querying a variable depth Org Unit structure

Most reports will have an Org Unit Parameter, which may or may not be displayed to the User to enter.

The Org Unit Dim (DM_DIMORGUNIT) stores the Org Hierarchy of all Org Units that have been extracted into the Data Warehouse.

The following logic will need to be added to the Data Set Query to retrieve the correct records from the Data Mart:

```
where org_dim.orgunitkey in (select Org.orgunitkey
    from DM_DIMORGUNIT Org where Org.orgunitparentname is null and
    Org.orgunitid in (select T4.orgunitid from DM_DIMORGUNIT T4 where
    T4.orgunitparentname is not null and T4.orgunitparentid = ? ))
```

This logic returns all of the Child Org IDs that belong to the specified Parent ID. The logic may also be implemented in the From clause of the Data Set.

8.5 Drilling Down From a BIRT Report to a Reporting Application Page

Follow the below development steps when drilling down to a Reporting Application page from a BIRT report.

The development process is:
- Pass in the page id as a parameter to the report.
- Add the script below to the onclick event.
- BIRT stores quotes as " in the xml file. Change the string back to quotes in the xml source tab in BIRT.

```
window.parent.dojo.require("curam.util.Navigation");
    var emptyString = "";

    if (categoryData == "Completed")
     {
    window.parent.curam.util.Navigation.goToUrl(
    "CCSInvestigation_initialContactsCompletedPage.do?numberOfDays=" +
    emptyString + "complianceType" + categoryData);
     }

    else if (categoryData == "Pending")
     {
    window.parent.curam.util.Navigation.goToUrl(
    "CCSInvestigation_initialContactsPendingPage.do?numberOfDays="  +
    emptyString + "complianceType" + categoryData);
     }

    else if (categoryData == "Overdue")
     {
    window.parent.curam.util.Navigation.goToUrl(
    "CCSInvestigation_initialContactsOverduePage.do?numberOfDays=" +
    emptyString +"complianceType" + categoryData);
     }
```

8.6 Formatting of Axis labels

The following paragraphs outline the features supported when displaying axis labels on Report Charts rendered in Flex.

There are four build-in formats supported by Report Charts rendered in Flex, the formats are based on the data types used in the chart:

- Integers will be displayed without and formatting (1,2,3,100,101,1000,10001, 100001).
- Decimal values will be displayed to two decimal places using the locale in use (i.e. 1.00 or 1,00).
- Date/Time values will use the Java Medium format, again based on the locale in use (e.g. Jan 12, 1952 or 3:30pm).
- Anything else will be treated as a string and the value rendered will be displayed without change.

8.7 Accessibility and BIRT Charts

The following paragraphs outline what Chart properties must be set to meet accessibility standards.

The chart title must be set to ensure a chart is accessible, the charts accessibility is not related to the visibility of the chart title, the title can be set to visible or not-visible:

- Within Eclipse, double click on the chart to view the chart wizard.
- Navigate to the Format Chart tab and select the title property in the left navigation pane.
- Enter the chart title, depending on your user interface requirements make the title visible or not visible.

Chapter 9. BIRT Compliance

A full compliance statement on all BIA artifacts will be available in Future Service Pack Releases

9.1 BIRT Reports

A compliant process for changing BIA report design document is to copy the report and to change the copy. Changing existing BIRT report design documents will result in a more difficult upgrade process.

Chapter 10. Creating a Datamart Data Source on WebLogic and WebSphere

10.1 Introduction

BIA Reporting is supported on both Oracle V6 and DB2.

10.2 WebSphere for OWB

The BIRT Viewer application has been tested using a non-XA data source and a XA data source. The recommendation for creating a Datamart Data Source on WebSphere® is to use the non-XA data source.

10.2.1 Creating the Data Source Login Alias

IBM® DB2, IBM DB2 for z/OS® and Oracle Database are the databases supported. The Administrative Console can be used to configure a login alias for both the DB2 and Oracle data sources as follows:

- Navigate to Security -> Global Security.
- Expand the Java Authentication and Authorization Service option in the Authentication section and select the J2C authentication data option.
- Click New to open the Configuration screen.
- Set the following fields:

  Alias = CuramReporting User ID = <database username>

  Password = <database password> Description = The database security alias where <database username> and <database password> are set to the username and password used to login to the database.
- Click OK to confirm the changes.

10.2.2 Setup the Oracle Non-XA Database Driver

- Navigate to Resources->JDBC->JDBC providers.
- Choose the scope appropriate for your environment.
- Click New to add a new driver. This will open a configuration screen.
- Select Oracle from the the list in the Database type drop down supplied.
- Select Oracle JDBC Driver from the list in the Provider type drop down supplied.
- Select the Connection pool data source from the list in the Implementation type drop down supplied.
- Set the Name field to be Oracle JDBC Driver for BIA Reporting, if not filled in automatically set the description.
- Click Next to continue.
- Review the Class path and set the environmental variable to the location of the ojdbc6.jar file, e.g. C:\CC\DevEnv\6.0\CuramSDEJ\drivers. Click next to continue.
- Review the properties on the configuration screen that opens. No changes should be required.

10.2.3 Setup the Oracle XA Database Driver

- Navigate to Resources->JDBC->JDBC providers.
- Choose the scope appropriate for your environment.
- Click New to add a new driver. This will open a configuration screen.
- Select Oracle from the the list in the Database type drop down supplied.
- Select Oracle JDBC Driver from the list in the Provider type drop down supplied.

- Select the XA data source from the list in the Implementation type drop down supplied.
- Set the Name field to be Oracle XA JDBC Driver For BIA Reporting, if not filled in automatically set the description.
- Click Next to continue.
- Review the Class path and reset the environment variable to the location of the ojdbc6.jar file, e.g. C:\CC\DevEnv\6.0\CuramSDEJ\drivers. Click next to continue.
- Review the properties on the configuration screen that opens. There should be no need to change any of them.
- Click Finish to confirm the changes.

10.2.4 Set up the Oracle Non-XA Database Driver Data Source

- Navigate to Resources->JDBC->Data Source.
- Choose the scope appropriate for your environment
- Start the process for creating a new data source.
- Click New to add a new data source.
- Set the fields as follows:
  - Data source name: curamdm
  - JNDI Name: jdbc/curamdm
  - Click Next.
  - Select an existing JDBC driver, the driver created in Section "Setup the XA Database Driver" and hit next.
- Set the URL Value field to the value below where serverName is the name of the server hosting the database. Port is the port number the database is listening on and databaseName is the SID of the database. This is the URL to connect using Oracle SID name:

  jdbc:oracle:thin:@serverName:port:databaseName
- Set the Data store helper class name to be Oracle,e.g for Oracle 10g data store helper.
- Leave all other fields untouched unless a specific change is required and click Next:
  - Set the Component-managed authentication alias drop down value to <none>.
  - Set the Mapping-configuration alias drop down value to <none>.
  - Set the Container managed-configuration alias drop down value to <Alias created in Section "Setup the non-XA Database Driver" >
- Leave all other fields untouched unless a specific change is required and click Next.
- Click Finish to confirm the changes and continue.

10.2.5 Setup the DB2 Environment Variable

- Navigate to Environment-WebSphere variables.
- Select the DB2UNIVERSAL_JDBC_DRIVER_PATH link from the list of environment variables. This will open the configuration screen for this variable.
- Set the Value field to point to the directory containing the Type 4/Type 2 drivers. This is normally the drivers directory under the SDEJ installation, e.g. D:\Curam\CuramSDEJ\drivers.
- Click OK to confirm the changes.

10.2.6 Set up the DB2 Database Driver Provider

- Navigate to Resources-JDBC-JDBC providers.
- Note: The appropriate scope where the data source is to be defined should be selected at this point.
- Click New to add a new driver. This will open a configuration screen.
- Select DB2 from the list in the database type drop down supplied.

- Select the DB2 Universal JDBC Driver Provider from the list in the Provider type drop down supplied.
- Select the XA data source from the list in the Implementation type drop down supplied.
- Click Next to continue.
- Review the properties on the configuration screen that opens. There should be no need to change any of them unless you are planning to connect to a zOS database. If so, verify that ${DB2UNIVERSAL_JDBC_DRIVER_PATH} field is pointing at the correct directory for your system. For example, it should point at the directory containing the DB2 Connect license jar, db2jcc_license_cisuz.jar provided by IBM for zOS connectivity.
- Click Next and then Finish to confirm the changes.

10.2.7 Set up the DB2 Database Driver Data Source

- Select the DB2 Universal JDBC Driver Provider (XA) now displayed on the list of JDBC Providers. This will open the configuration screen for the provider.
- Select the Data sources link under Additional Properties.
- Click New to add a new data source.
- Set the fields as follows:
  - Data source name: curamdm
  - JNDI Name: jdbc/curamdm
- Set the Component-managed authentication alias drop down value to: <valid for database>; Set the Mapping-configuration alias drop down value to: DefaultPrinicipalMapping. Set the Container-managed authentication alias drop down value to: <valid for database>; where the <valid for database> alias used is the one set up in the Creating the Data Source Login Alias section above.
- Leave all other fields untouched unless a specific change is required and click Next to continue.
- Click Finish to confirm the changes and continue.
- Select the newly created DatasourceName data source from the displayed list.
- Select the Custom Properties link under Additional Properties.
- Select the fullyMaterializeLobData entry.
- Set the value to be false.
- Click OK to confirm the change.

10.3 WebLogic

The BIRT Viewer application has been tested using a non-XA data source and a XA data source. The recommendation for creating a Datamart Data Source on WebLogic is to use the non-XA data source. data source.

Create a non-XA data source called <curamdm>, as below with the following settings, create a connection pool data source with no global transactions, add the Cúram node as a target.

Open the Administration Console as detailed in the previous section.
- Navigate to <DomainName>+Services+JDBC+Data Sources.
- Click the New button; Enter the following fields:
  - Name: curamdm
  - JNDI Name: jdbc/curamdm
- Change the Database Type to be Oracle and set the Database Driver to be Oracles Driver (Thin XA) for Instance connections. Versions:9.0.1,9.2.0,10,11.
- Click the Next button.
- Leave the default for Transaction Options and click the Next button.

- Set the URL Value field to the value below where serverName is the name of the server hosting the database. Port is the port number the database is listening on and databaseName is the SID of the database. This is the URL to connect using Oracle SID name:

  jdbc:oracle:thin:@serverName:port:databaseName
- Leave all other fields untouched unless a specific change is required.
- Click the Next button.
- Review the settings and click the Next button.
- Select Targets, select the Cúram node.
- Click the Finish button.

It is advised to restart the AdminServer at this point, to ensure the changes are correct. To do this:
- Navigate to <DomainName>+Environment+Servers.
- Select the Control tab, then select AdminServer in the Server's list and click Shutdown+When work completes.
- Click the Yes button to shutdown the AdminServer.

Appendix A. TroubleShooting

This section details possible troubleshooting tips and fixes that the user may find helpful.

A.1 Problem: cannot view BIRT content through Eclipse with H2

Solution:

Please ensure H2 is started in remote mode.

A.2 Problem: build client.birt fails to execute.

Solution:

A build command exists which publishes BIRT content to the Cúram BIRT Viewer. This build command will only execute cleanly if an environment variable 'DEVENV' is created. This variable 'DEVENV' must point to the parent directory of the BIRT Eclipse folder.

If the eclipse install folder is named "eclipsebirt" no further action is required. Otherwise, change the property <version-eclipsebirt=eclipsebirt>, e.g. if birt eclipse is installed into directory named <BIEclipse> then update the property to <version-eclipsebirt= BIEclipse>.This property is located in the file <BIRTthird_party_version.properties> within the BIApp directory.

A.3 Problem: The library with the namespace CEFLibrary is not found.

Solution:

In Eclipse using the view outline model, right click on the report title and select refresh library. Please ensure your workspace is correctly configured, ensure all the steps in section Develop New Reports have been completed.

A.4 Problem: Error when executing a report.

If you get a message (the message text may be vague) that states you cannot connect to the database or connection is null or oracle message file is missing, the first task is to verify that the JDBC drivers are present in the ODA drivers directory.

Solution:

Verify that the JDBC jar files are present in the ODA drivers directory, if this occurs within Eclipse copy the drivers to the ODA plugins directory within the Eclipse install path. If you find this error when viewing the application through tomcat then verify that the drivers have been copied by the client.birt build target which is executed from the BIContent directory.

A.5 Problem: Eclipse dies when previewing a BIRT report.

Solution:

This was found to be an issue of the JVM used by Eclipse, we change the birteclipse.bat to use the following setting -vm %DEVENV%\eclipsejre\bin\javaw.exe.

A.6 Problem: Report Page only displays one report when more that one report exists on the operational workspace.

Solution:

There may be a race condition for loading the JDBC drivers resulting in one report failing. A fix will be provided in due course, the current solution is to initialize your BIRT environment before your first Report Page loads. Do this by going to your CuramBIRTViewer\List page and execute any report (this will bootstrap BIRT).

A.7 Problem: Tomcat closes on report preview or ODA SQL preview.

Solution:

The addition of BIRT may result in Tomcat having memory issues for the permanent generation heap, to resolve:

- Add the following parameter to Tomcat "-XX:MaxPermSize=256m"
- Go to: Menu ->Preferences->Tomcat->JVM Settings->Append to JVM Parameters->Add -> "-XX:MaxPermSize=256m"

A.8 Problem: BIRT Viewer does not run / List of reports not present.

Solution:

To ensure everything was pulled down correctly follow these steps:

- Delete the BIAppBIRTInfastructureVersion.txt file from <CURAM_DIR>/ ActualVersions.
- Run 'appbuild client.birt' from <CURAM_DIR>/ BiContent.
- Verify the following folder exists. (<CURAM_DIR>\BIApp\CuramBIRTViewer\WebContent\WEB-INF\classes\biapp).
- Start tname server, start operational database server, start tomcat.
- View BIRT Viewer from:
  1. `http://localhost:9080/CuramBIRTViewer/`

A.9 Problem: Exception thrown when running report

The below error is returned when trying to run a report:

```
SEVERE: An error happened while running the report. Cause:
java.lang.NullPointerException
at java.util.Hashtable.put(Hashtable.java:396)
at java.util.Hashtable.put(Hashtable.java:396)
at java.util.Hashtable.putAll(Hashtable.java:470)
at org.eclipse.birt.data.engine.executor.DataSource.
<init>(DataSource.java:76)at org.eclipse.birt.data.engine.
executor.DataSourceFactory.
getDataSource(DataSourceFactory.java:75)
at org.eclipse.birt.data.engine.impl.
PreparedOdaDSQueryOdaDSQueryExecutor.getDataSource
(PreparedOdaDSQuery.java:241)
```

Solution:

Ensure that you have completed this below step:

- Go to Menu -> Preferences -> Tomcat ->JVM Settings. On the classpath widow select the button directory and add in the full path to the EJBServer\project\properties directory for your stream.

A.10 Problem: Unable to view all entities after creating dataset.

There is a cap on the number of tables that BIRT displays when creating a Data Set.

Solution:

You can change the number of tables and schemas that BIRT pulls back from the Database. Go to Menu -> Preferences -> Report Design -> Data Set Editor -> JDBC Data Set and increase the number of tables.

A.11 When importing OWB meta data the OWB validation process results in a build failure due to 'warnings' being present ETL processes.

Solution:

The default best practice for metadata is that it contains no errors or warnings. The build script enforces this by failing when these warnings are present. However there is a known issue with the metadata in Cúram V6 in that it does contain some warnings. To get around these build failures change the property that controls this process. In the Application.Properties file set the property 'environment.owbconfig.validate.failonwarnings=false', this allows builds to continue without failing. This issue will be resolved in a future service pack.

Appendix B. Advanced Business Intelligence and Analytics

B.1 What is Advanced BIA

Advanced BIA provides support for the integration of BIA components into the Operational Database Application via embedded analytics, Interactive Dashboards and Reports. Advanced Analytics On-Line Analytical Processing (OLAP) cubes and Data Mining are outside the scope of BIA for Cúram V6.

Advanced BIA:

- OLAP cubes enables end users to analyze data with extremely fast query and calculation performance, enabling a style of analysis known as "slicing and dicing." This capability could span a variety of storage architectures such as relational, multidimensional and in-memory.
- Data Mining is the capability to enable organizations to classify categorical variables and estimate continuous variables using advanced mathematical techniques.

B.1.1 Incremental upgrades

Incremental upgrades (up sell) between licensed options must be possible It is possible for a customer to purchase additional licensed options, at any stage in their project lifecycle and integrate these without excessive additional work. In particular any features that we are currently using should continue to work when the additional options/components are installed.

B.1.2 Large Scale aggregations

Large Scale aggregations will be supported by the Data Warehouse. It is assumed that any large scale aggregations of data will be performed in the Data Warehouse or with dedicated tables within the transactional database. As a result no consideration has been given to supporting large scale aggregations directly in the transactional database, nor to the potential issues any such aggregations might pose.

Appendix C. Deployment Architecture

C.1 Introduction

There are two key deployment options for this project:

1. Using BIRT: The BIRT Engine is deployed in a separate Application Server instance.

2. Using Actuate: When using Actuate, it is deployed as a separate instance (the product is called iServer). Multiple instances of each of these can be used, without changing the architecture, to meet customer requirements for scalability and/or reliability as needed.

This view is open to further evolution to support Advanced Analytics, for example using the Cognos BIA Engine; This may be integrated into BIA as part of future Service Packs.

C.2 Installation and Migration

The installation of a standard deployment where BIRT is co-located on a single application server instance, is the standard installation for the BIA Platform. Other deployment configurations will require manual steps, but these will not be supported by the installer.

Appendix D. Runtime Architecture

D.1 Default Runtime Architecture

The runtime architecture would structured in the following order:

1.
    - Database
    - Reporting Database
2. Combined with the following:
    - Server Application
    - BIRT Server Application
    - Actuate Engine
3. Into:
    - Client Application

    which is selected from based on configuration

D.2 BIRT Viewer Application

This is a new component which will wrap the open source BIRT Engine.

D.2.1 BIA Engine Abstraction Layer

This component is responsible for managing the look-up of report templates when the BIRT Engine is in use.

D.2.2 BIRT Viewer

This component represents the BIRT Report Engine, which will run and render the selected template. This will serve reports which in turn access the Source Database as well as the Reporting Database.

D.3 BIA Reporting Database

This component represents the Data Marts currently branded as BIA Reporting Enterprise Module, which can be queried by BIA content.

D.3.1 Rendering BIA content using BIRT

When a user opens a page containing BIA content, on a system where BIRT is the deployed BIA Engine, the functional components consisting of the following:

1. **Client Browser**
2. **Curam Client Application**
3. **Curam Server Application**
4. **Curam Database**
5. **Curam BIRT Server App**

interact with each other in the following order:

- 2 Requests a page from 1
- 3 Calls a Facade from 2
- 4 Requests Data from 3
- 4 Returns Data to 3

- 3 Returns Data to 2
- 5 Requests Data from 2
- 4 Requests Data from 5
- 4 Returns Data to 5
- 5 Returns Report to 2
- 2 Renders Page to 1

D.3.2 Rendering BIA content using Actuate

When a user opens a page containing BIA content, on a system where Actuate is the deployed BIA Engine, the functional components are similar to the BIRT in previous section and interact in the same order.

D.3.3 BIRT Performance and Scalability

The key elements are:

- Well designed SQL queries influencing performance.
- The option of using summary tables within the transactional database, where needed to ensure acceptable performance.
- The option of using data from the data warehouse where no acceptable performant alternative can be found.

D.3.4 Security

There are potentially additional issues here, where the security of data within aggregated views will need to be considered when developing BIA Content, as this is a key concern. As a basic principle the assumption is that where data has been aggregated; so that individual details can no longer be discerned, then it is acceptable to ignore security based on the underlying data. In switching to views where the underlying data is accessible, then the security must be enforced.

To this end we are recommending two approaches for constructing reports:

- Where the data will only be presented in the aggregate and anywhere that underlying data is accessible only through the BIA Application, the report should query the databases (both on-line and the data warehouse) directly.
- Where the data will be presented in a way that exposes the underlying data, or where very specific data security requirements exist, the report should make EJB calls to facades in the BIA Application to retrieve the data.

D.3.5 Availability and Resilience

Because this project will depend on Actuate components which will be deployed without using the BIA infrastructures, standard BIA practices cannot be relied on to meet availability and resilience requirements. The architecture of the system attempts to mitigate this in so far as it is possible and Actuate do have very large deployments of this solution in place.

D.3.6 Accessibility

Because we are relying on the rendering of BIRT and Actuate, some usage of non-accessible client technologies, for example JavaScript, Flex and SVG, may occur. It is intended that no accessible alternative versions of these will be provided. The option to have accessible and non-accessible versions of a report template remains an option, but based on the current level of experience with the tools, it is felt that this should not be required.

Appendix E. Viewing BIA Reports - Dependencies

E.1 Viewing BIA Reports

If any of the reports that query the Datamart are failing, please ensure the JNDI data source has been created on the Application Server. Please ensure that the data source points to a populated Datamart. See instructions within the BIRT Developer guide for further details.

In terms of features provided by the BIRT infrastructure when viewing BIA reports, it is important to note that there are 3 distinct environments, each of which has it own has it's own dependencies.

1. Application Server
2. Tomcat - BIRT Viewer
3. BIRT Eclipse

E.2 Application Server

When testing the reports using an Application server, the BIRT infrastructure automatically sets the JNDI data source property name for all reports. The JNDI data source must already have been created on the Application Server. This will override any settings within the reports.

E.3 Tomcat - BIRT Viewer

When testing the reports using Tomcat, the developer must first, manually enter data source properties within the SERVER_DIR\project\properties\Bootstrap.properties file. This will override any settings within the reports. The reports can then be tested successfully

E.4 BIRT Eclipse

When testing the reports locally within BIRT Eclipse, the developer must manually enter data source properties for each report as explained in the 'Developing New Reports' section. The developer will then use the 'Preview' tool within the report itself.

Notices

This information was developed for products and services offered in the U.S.A. IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service. IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing

IBM Corporation

North Castle Drive

Armonk, NY 10504-1785

U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing

Legal and Intellectual Property Law.

IBM Japan Ltd.

19-21, Nihonbashi-Hakozakicho, Chuo-ku

Tokyo 103-8510, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you. Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation

Dept F6, Bldg 1

294 Route 100

Somers NY 10589-3216

U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources.

IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only

All IBM prices shown are IBM's suggested retail prices, are current and are subject to change without notice. Dealer prices may vary.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing

application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

© (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs.

© Copyright IBM Corp. _enter the year or years_. All rights reserved.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at http://www.ibm.com/legal/us/en/copytrade.shtml.

Actuate is a registered trademark of Actuate Corporation.

Adobe, the Adobe logo, and Portable Document Format (PDF), are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, other countries, or both.

Apache is a trademark of Apache Software Foundation.

BIRT is a registered trademark of Eclipse Foundation.

Oracle, WebLogic Server, Java and all Java-based trademarks and logos are registered trademarks of Oracle and/or its affiliates.

Other names may be trademarks of their respective owners. Other company, product, and service names may be trademarks or service marks of others.

IBM ®

Printed in USA