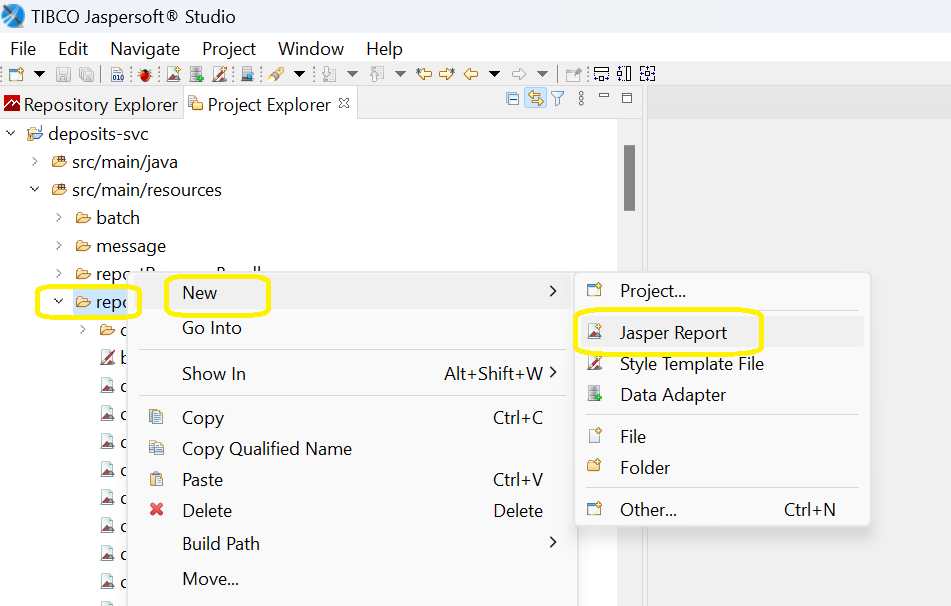
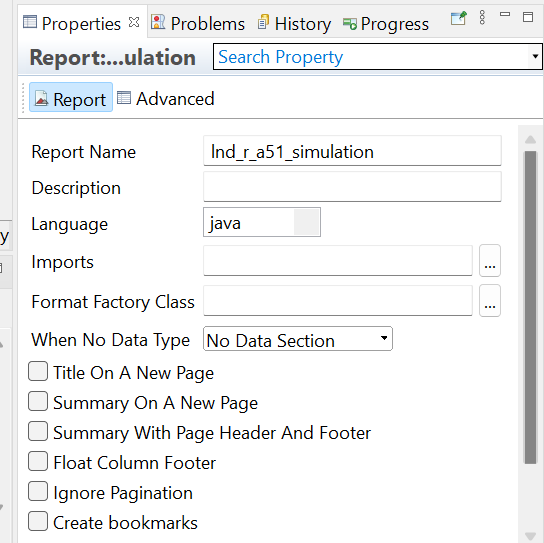
******

***Report Name***

Every report design needs a name. Its name is important because the library uses it when generating files, especially when the default behaviour is preferred for compiling, filling, or exporting the report.

The name of the report is specified using the name attribute, and its inclusion is mandatory. Spaces are not allowed in the report name - it must be a single word.

|  |
| --- |
| CBS standard naming convention |
| <module code>\_r\_<report no.>\_<report title in lower case, blank space replaced with underscore character>.  Example:  lnd\_r\_a51\_simulation  dep\_r\_921\_cash\_position  dep\_r\_922\_cash\_in\_transit |



***Language***

Report expressions are usually written using the Java language. However, one can use other languages as long as a report compiler is available to help evaluate these expressions at report-filling time.

The default value for the language property is java, meaning that the Java language is used for writing expressions, and that a report compiler capable of generating and compiling a Java class on the fly is used for producing the bytecode needed for expression evaluation at runtime.

Report compilers reference this property to see whether they can compile the supplied report template or whether a different report compiler should be used, depending on the actual scripting language.

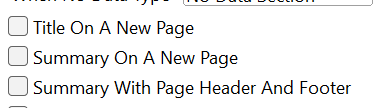
***Empty Data Source Behaviour***

The data source for a report might not contain any records. In this case, it is not clear what the output should be. Some people may expect to see a blank document and others may want some of the report sections to be displayed anyway.

The whenNoDataType attribute lets users decide how the generated document should look when there is no data in the data source supplied to it. Possible values of this attribute are as follows:

* NoPages - This is the default setting. The generated document will have no pages in it. Viewers might throw an error when trying to load such documents.
* BlankPage - the generated document will contain a single blank page
* AllSectionsNoDetail - All the report sections except the detail section will appear in the generated document.
* NoData - the generated document will contain only the NoData section

***Title and Summary Section Placement***



To display the title or/and summary section on a separate page, set one or both of the following attributes to true:

* isTitleNewPage
* isSummaryNewPage

Both of these Boolean attributes are set to false by default.

When the summary section stretches to a new page, or starts on a new page altogether, it is not accompanied by the page header and page footer. In order to force the page header and page footer to reappear on summary trailing pages, set the isSummaryWithPageHeaderAndFooter to true; it is set to false by default.

***Column Footer Placement***

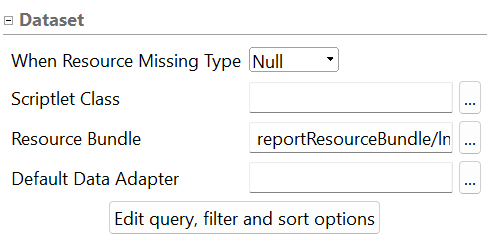
The isFloatColumnFooter Boolean property lets users customize the behavior of the column footer section. By default, this section is rendered at the bottom of the page, just above the page footer. In certain cases, it is useful to render it higher on the page, just below the last detail or group footer on that particular column. To do this, set the isFloatColumnFooter property to true.

***Scriptlet Class***

There can be multiple scriptlet instances associated with a report. The scriptletClass attribute is just a convenient way to specify the scriptlet class in case there is only one scriptlet associated with the report.

This attribute is used by the engine only if no value is supplied for the built-in [REPORT\_SCRIPTLET](https://jasperreports.sourceforge.net/api/net/sf/jasperreports/engine/JRParameter.html#REPORT_SCRIPTLET) parameter. If neither the attribute nor the parameter is used, and no other scriptlet is specified for the report using named scriptlet tags the reporting engine uses a single [JRDefaultScriptlet](https://jasperreports.sourceforge.net/api/net/sf/jasperreports/engine/JRDefaultScriptlet.html" \o "class in net.sf.jasperreports.engine) instance as the report scriptlet.

***Resource Bundle***

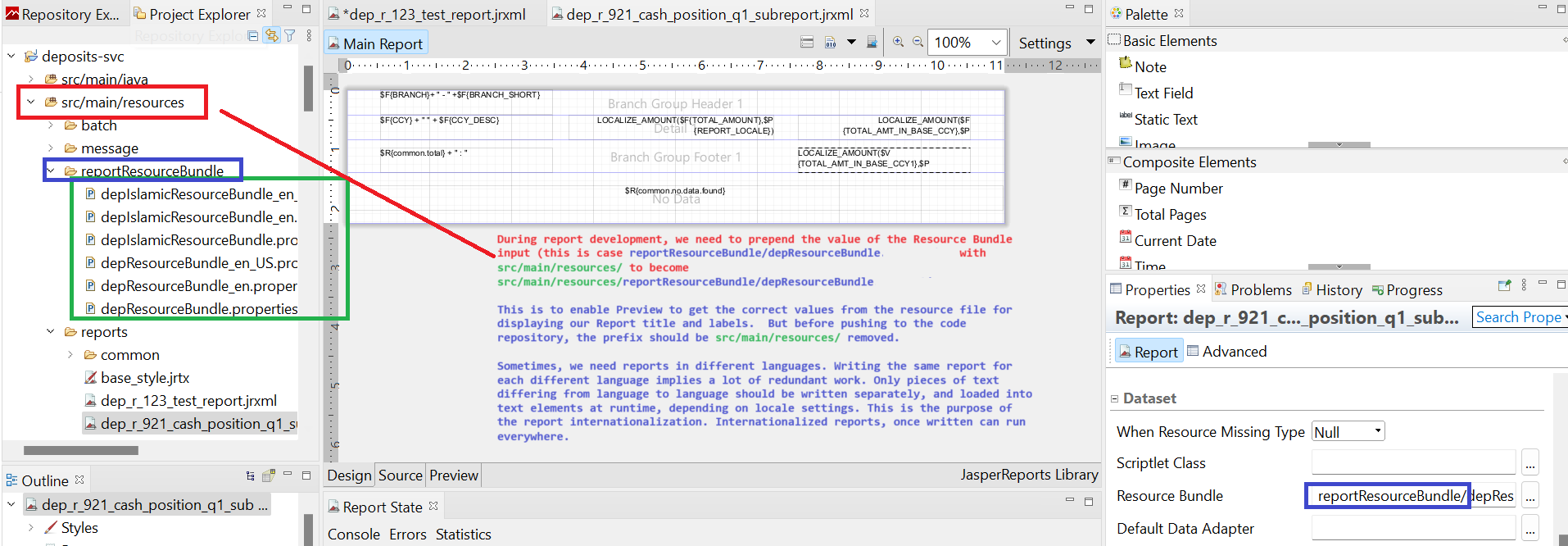
******

To generate reports in different languages from the same report template, associate a resource bundle with the template and make sure that the locale-specific resources inside report expressions are retrieved based on the $R{} syntax.

There are two ways to associate the java.util.ResourceBundle object with the report template.

The first is a static association made by setting the resourceBundle property of the report template object to the base name of the target resource bundle.

A dynamic association can be made by supplying a java.util.ResourceBundle object as the value for the [REPORT\_RESOURCE\_BUNDLE](https://jasperreports.sourceforge.net/api/net/sf/jasperreports/engine/JRParameter.html#REPORT_RESOURCE_BUNDLE) parameter at report-filling time.



***Missing Resources Behavior***

The whenResourceMissingType property allows users to choose the desired behavior of the engine when it deals with missing locale-specific resources in the supplied resource bundle. There are four different values that can be used to deal with missing resources:

* Null - The null value is used
* Empty - An empty string is used
* Key - The key is used
* Error - An exception is raised in case a locale-specific resource is not found in the supplied resource bundle for the given key and locale

***Pagination***

When the isIgnorePagination property is set to true, the report-filling engine will completely ignore page break-related settings inside the report template and generate the document on a single, very long page. The value of this property can be overridden at runtime using the optional, built-in [IS\_IGNORE\_PAGINATION](https://jasperreports.sourceforge.net/api/net/sf/jasperreports/engine/JRParameter.html#IS_IGNORE_PAGINATION) parameter.

***Formatting Numbers, Dates and Times***

The formatFactoryClass attribute lets users specify the name of the factory class implementing the [FormatFactory](https://jasperreports.sourceforge.net/api/net/sf/jasperreports/engine/util/FormatFactory.html" \o "interface in net.sf.jasperreports.engine.util) interface, which should be instantiated by the engine in order to produce java.text.DateFormat and java.text.NumberFormat objects to use for date and number formatting in the current report.

This attribute specifying the factory class name is used only if no value is supplied for the built-in [REPORT\_FORMAT\_FACTORY](https://jasperreports.sourceforge.net/api/net/sf/jasperreports/engine/JRParameter.html#REPORT_FORMAT_FACTORY) parameter.

If neither of the attribute nor the parameter is used, the engine will eventually instantiate the [DefaultFormatFactory](https://jasperreports.sourceforge.net/api/net/sf/jasperreports/engine/util/DefaultFormatFactory.html" \o "class in net.sf.jasperreports.engine.util) implementation of the factory interface, which produces java.text.SimpleDateFormat and java.text.DecimalFormat objects for date and number formatting. This attribute or the built-in [REPORT\_FORMAT\_FACTORY](https://jasperreports.sourceforge.net/api/net/sf/jasperreports/engine/JRParameter.html#REPORT_FORMAT_FACTORY) parameter should be used only if the report relies on custom date and number formatters.

***Report Styles***

A report style is a collection of style settings declared at the report level. These settings can be reused throughout the entire report template when setting the style properties of report elements.

The name attribute of a <style> element is mandatory. It must be unique because it references the corresponding report style throughout the report.

One can use isDefault="true" for one of your report style declarations to mark the default for elements that do not or cannot have another style specified.

Each report style definition can reference another style definition from which it will inherit some or all of its properties. The style attribute inside a <style> element specifies the name of the parent report style.

All report elements can reference a report style to inherit all or part of the style properties. A report style declaration groups all the style-related properties supported throughout the library, but an individual element inherits only those style properties that apply to it. The others will be ignored.

***Page Format***

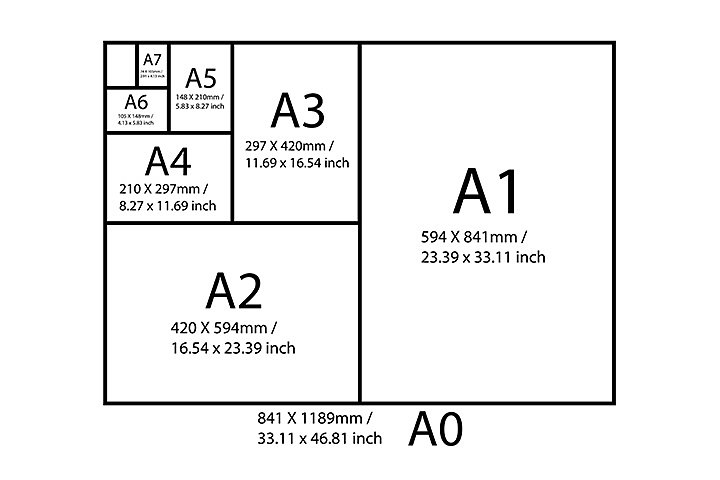
***Page Size***

There are two attributes at this level to specify the page size of the document that will be generated: pageWidth and pageHeight. Like all the other JasperReports attributes that represent element dimensions and position, these are specified in pixels. JasperReports uses the default Java resolution of 72 dots per inch (DPI). This means that pageWidth="595" will be about 8.26 inches, which is roughly the width of an A4 sheet of paper.

The default page size corresponds to an A4 sheet of paper:

pageWith="595" pageHeight="842"

|  |  |  |
| --- | --- | --- |
|  | Portrait | |
|  | Page Width | Page Height |
| A4 | 595 | 842 |
| **A3** | **842** | **1190** |
| A2 | 1190 | 1684 |
| A1 | 1684 | 2384 |
| A0 | 2384 | 3370 |
|  | Landscape | |
|  | Page Width | Page Height |
| A4 | 842 | 595 |
| A3 | 1190 | 842 |
| A2 | 1684 | 1190 |
| A1 | 2384 | 1684 |
| A0 | 3370 | 2834 |



|  |  |
| --- | --- |
|  |  |

***Column Count***

JasperReports lets users create reports with more than one column on each page.

Multicolumn report templates also have an associated column-filling order specified by the printOrder attribute.

By default, the reporting engine creates reports with one column on each page.

***Print Order***

For reports having more than one column, it is important to specify the order in which the columns will be filled. One can do this using the printOrder attribute with 2 possible values:

* Vertical - columns are filled from top to bottom and then left to right. This is the default print order.
* Horizontal - columns are filled from left to right and then top to bottom

***Page Orientation***

The orientation attribute determines whether the documents use the Portrait or the Landscape format.

This orientation attribute is useful only at report-printing time to inform the printer about the page orientation, and in some special exporters. The default page orientation is Portrait.

***Page Margins***

Once the page size is decided, one can specify what margins the reporting engine should preserve when generating the reports. Four attributes control this: topMargin, leftMargin, bottomMargin, and rightMargin.

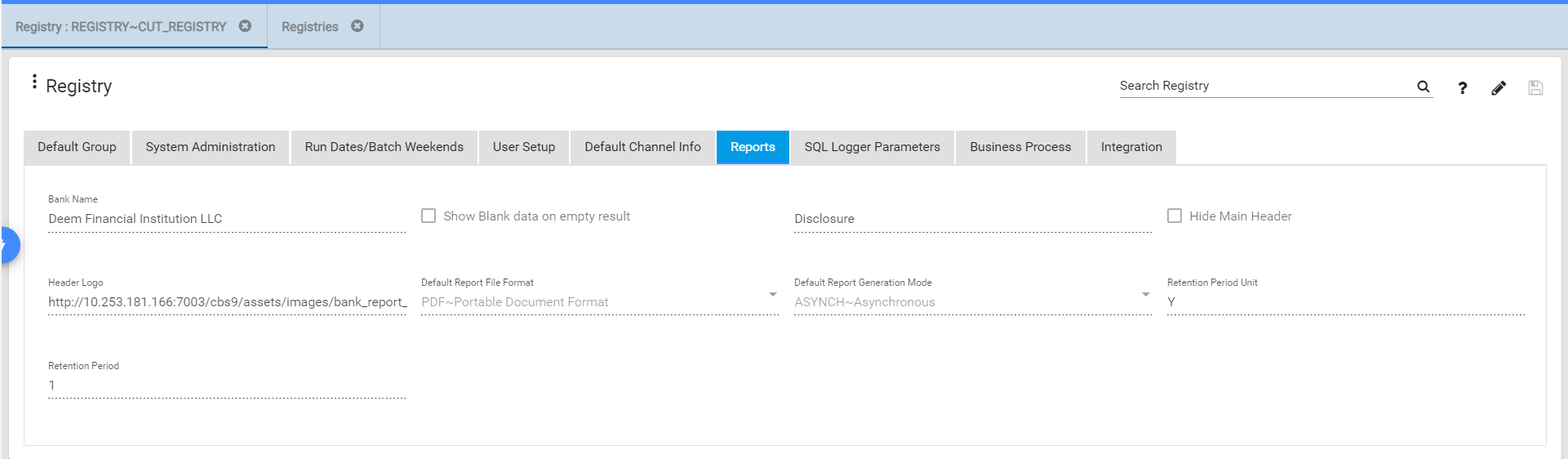
The default margin for the top and bottom of the page is 20 pixels. The default margin for the right and left margins is 30 pixels.

***Column Size and Spacing***

Reports may have more than one column, as specified in the columnCount attribute. However, the reporting engine has to know how large a column can be and how much space should be allowed between columns. Two attributes control this: columnWidth and columnSpacing.

Also, when a report is compiled JasperReports checks whether the width of the overall columns and the space between them exceed the specified page width and page margins.

Since there is only one column by default, the default column spacing is 0 pixels and the default column width is 555 pixels (the default page width minus the default left and right margins).



<template><![CDATA[$P{BASE\_LOCATION} + "base\_style.jrtx"]]></template>

<parameter name="REPORT\_NAME" class="java.lang.String"/>

<parameter name="BASE\_LOCATION" class="java.lang.String"/>

<parameter name="RUN\_DATE" class="java.util.Date"/>

<parameter name="BANK\_NAME" class="java.lang.String"/>

<parameter name="SUPPRESS\_NO\_DATA" class="java.lang.Boolean">

<defaultValueExpression><![CDATA[false]]></defaultValueExpression>

</parameter>

<parameter name="DISCLOSURE" class="java.lang.String"/>

<parameter name="HIDE\_MAIN\_HEADER" class="java.lang.Boolean"/>

<parameter name="LOGO" class="java.lang.String"/>