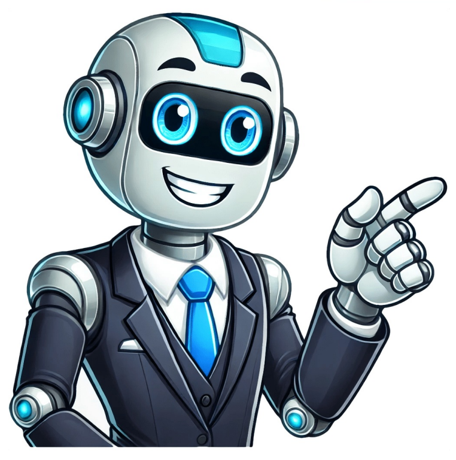

I have jrxml file, I want to compile this to get .jasper. How do I compile and get that jasper file? I have jrxml file, I want to compile this to get .jasper. How do I compile and get that jasper file? I have jrxml file, I want to compile this to get .jasper. How do I compile and get that jasper file? I have jrxml file, I want to compile this to get .jasper. How do I compile and get that jasper file? Share — copy and redistribute the material in any medium or format for any purpose, even commercially. Adapt — remix, transform, and build upon the material for any purpose, even commercially. The licensor cannot revoke these freedoms as long as you follow the license terms. Attribution — You must give appropriate credit , provide a link to the license, and indicate if changes were made . You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use. ShareAlike — If you remix, transform, or build upon the material, you must distribute your contributions under the same license as the original. No additional restrictions — You may not apply legal terms or technological measures that legally restrict others from doing anything the license permits. You do not have to comply with the license for elements of the material in the public domain or where your use is permitted by an applicable exception or limitation . No warranties are given. The license may not give you all of the permissions necessary for your intended use. For example, other rights such as publicity, privacy, or moral rights may limit how you use the material. I have jrxml file, I want to compile this to get .jasper. How do I compile and get that jasper file? data-mc-breadcrumbs-count=3 data-mc-toc=True>Viewing: This section presents an example of uploading a JRXML with resources to the server. This report has two images that need to be added. Selecting the Main JRXML File If you want to validate the JRXML before uploading it, use Jaspersoft Studio. The server doesn't validate the JRXML when you upload it. This procedure shows you how to set up a name for the report in the repository and select the main JRXML file that references all other elements. To upload the main JRXML for this example: 1. Log into the server as administrator and select View > Repository. If you log in as a user, you can upload a report unit to the server, but this example requires an administrator login to access the image resources. 2. Locate the folder where you want to add the report. For example, go to Public > Samples > Reports. 3. Right-click the Reports folder and select Add Resource > JasperReport from the context menu. The Set Up the Report page of the JasperReport wizard appears. Add Resource appears on the context menu only if you have write permission to the folder. 4. In Naming, enter the name and description of the new report and accept the generated Resource ID: • Name – Display name of the report: New Simple Report • Resource ID – Permanent designation of the report object in the repository: New_Simple_Report • Description – Optional description displayed in the repository: This is a simple example 5. Select Upload a Local File and Browse to /samples/reports/AllAccounts.jrxml. This example shows how to upload a JRXML file from the samples folder in the installation directory. You can also select a JRXML from the repository. In "Required Set Up Values" you can see the Set Up the Report page. Required Set Up Values 6. Click Controls & Resources. The Controls & Resources page appears. If the server detects that the report needs additional resources, you'll be prompted to locate those resources. Suggested Resources in the Resources List A JRXML file doesn't embed resources, such as images. When the server uploads the JRXML, it tries to detect any missing resources. For example, Suggested Resources in the Resources List shows that two image files are missing: I have jrxml file, I want to compile this to get .jasper. How do I compile and get that jasper file? • 日本語 • 5.19 • 5.2 • 5.3 • 5.8 To work with and customize report templates in .jrxml format, you must first import them to Jaspersoft Studio using the following steps. How to Import Report Templates Step 1: Create a folder to store your reports In Jaspersoft Studio, navigate to the Project tab in the left-hand part of the screen. Select MyReports. Right-click MyReports and then select New > Folder. In the dialog box that pops up, select parent folder MyReports and name the new folder reports. This is the folder that will contain your report templates. Click Finish. Step 2: Import report template files to the new reports folder Right-click the reports folder and select Import. In the dialog window that pops up, select File System. Click Next. Choose the folder where you have stored the report template files that you previously downloaded from the Contact Center Administrator application. These files will have .jrxml file extensions in their names. Select these files and click Finish. Select the files to add to the reports folder Example: How to Modify .jrxml Report Templates Once you have downloaded a report template, you can use the visual editor of the Jaspersoft Studio to import and modify .jrxml report templates (e.g., remove some of the existing metrics or add new ones). Report templates in the .jrxml format that are imported to Jaspersoft Studio will look as shown: What follows is an example of how one might modify a .jrxml report template. In this example, the column Case is removed from the out-of-the-box Agent Activity Report and the formatting/visual appearance of the report is changed. Step 1: Open file agent_activity.jrxml In Jaspersoft Studio, navigate to My Reports > reports. Find and double-click the agent_activity.jrxml file to open it. Step 2: Select the elements you wish you change Select elements $R{col_case_id} in the group header frame, and $F{case_number} in the detail frame of the last column. You can select both elements at the same time. Right-click the selected elements and select Delete to remove these elements. Selecting columns to delete Step 3: Adjust the presentation of report elements as desired Select the report frames and reduce their widths by dragging the elements to the right position. You can use the grid lines to help you find the edges of each element. To reduce the width of the report, click into the design view (the gray part of the screen, below the report) of the report in focus. That will bring up the Properties pane on the right-hand side of the screen. Report template properties At the bottom of Properties, select Edit Page Format. In the window that pops up, you can reduce the width of the report, as well as make other adjustments as needed. If a report fails to build In some versions of Jaspersoft Studio, Groovy scripting language jar file needs to be updated, as the one supplied with Jaspersoft is not always compatible with the BP formatter scriptlet. If during report compilation an error appears complaining of JRScriptletException class missing, a Groovy library matching the version used in Jaspersoft studio plugins folder must be downloaded from the official groovy distribution (use groovy-all-X.X.X.jar from "embedded" subfolder). The official groovy jar has to be renamed to have exact same name as the supplied groovy jar, as it uses "_" instead of "-" in the name.< Previous | Next > I have jrxml file, I want to compile this to get .jasper. How do I compile and get that jasper file? This section presents two examples, one simple and one complex, that illustrate many aspects of uploading JRXML files to create a report unit. Each example is split into several sub-sections, one for each different kind of element in the report unit. Each sub-section can be referenced individually when adding your own reports with different combinations of elements. To create the simple report in this example, you need access to the following sample data installed with the server: • /samples/reports/AllAccounts.jrxml • /samples/image/logo.jpg files The simple report has file resources but no input controls. The sample report resources consist of two images. Later in this chapter, a complex report example covers other types of resources: • Fonts • Resource bundles that adapt a report to other locales • Scriptlets The report includes a custom query that demonstrates how a report unit can modify the JRXML report it is based upon. To upload a report unit to the server, first set up a name in the repository. Next, upload the main JRXML file that references all other elements. The following procedure describes how to upload a main JRXML. To upload the main JRXML for the simple report example: 1. Log in to the server as administrator and select View > Repository: If you log in as a user, you can upload a report unit to the server, but the example requires an administrator login to access the image resources. 2. Locate the folder where you want to add the report. In this example, go to Organization > Reports > Samples. 3. Right-click the Samples folder and select Add Resource > JasperReport from the context menu. Add Resource appears on the context menu only if your user account has write permission to the folder. The Set Up the Report page of the JasperReport wizard appears: Set Up the Report Page 4. In Naming, enter the name and description of the new report, and accept the Resource ID generated as you type the name: • Name – Display name of the report: New Simple Report • Resource ID– Permanent designation of the report object in the repository: New_Simple_Report • Description – Optional description displayed in the repository: This is a simple example 5. Select Upload a Local File and Browse to locate samples/reports/AllAccounts.jrxml. The JRXML file is a report created in another editor such as iReport. You can either upload a new JRXML file or select a JRXML file from the repository. Set Up the Report Page 6. Click Controls & Resources. The server uploads the main JRXML file and displays the list of suggested resources: Suggested Resources in the Resources List Review the hyperlinked names of images: • Logolink • AllAccounts_Res2 You must upload the suggested file resources to the report. The report cannot be validated by the server if the resources are not added. When the server uploads the main JRXML file, it creates a list of the resources that are referenced in the JRXML. On the Controls & Resources page of the JasperReport wizard, you can add resources and input controls to the report unit. If the report you upload does not reference any file resources, you can skip to section Defining the Data Source. To upload the suggested file resources for the simple report example: 1. Click Add Now in the same row as LogoLink. The Locate File Resource page appears. 2. Choose Select a resource from the Repository, and Browse to locate Images/JR Logo. 3. The first image in this example is in the sample data in the repository. A Resource Selected from the Repository on the Locate File Resource Page To locate file resources referenced in the main JRXML, you can either upload an image file or choose an existing image in the repository. Any image file you choose can be associated with the resource ID in the main JRXML. The Add a Report Resource page appears: Properties of a Resource Being Added to a Report The file resource within the report unit has name, resource ID, and description properties. These are visible only within the report unit, they do not redefine the properties on the repository object that was referenced. The Resource ID field matches the resource name referenced in the main JRXML. 5. Accept the default property values for the image resource by clicking Next. The Controls & Resources page appears again, showing that the LogoLink resource was added. 6. On Controls and Resources, click Add Now in the AllAccounts_Res2 row. The Locate File Resource page appears again. 7. Select Upload a Local File, click Browse to locate /samples/images/logo.jpg, then click Next: Uploading a Resource from the File System on the Locate File Resource Page The server uploads the file. The Add a Report Resource page appears, showing the properties for the file resource: Properties of an Uploaded Resource on the Controls & Resources Page 8. Accept the default naming of the image by clicking Next. The JasperReport wizard returns to the Resources List page that shows the addition of all resources referenced in the main JRXML: Suggested Resources Added to the Simple Report Data sources belong to the report engine, in this case JasperReports Server, and cannot be defined in the main JRXML. Therefore, in the process of creating a report unit, you must define a data source in the server that the report can use. From the Data Source page, you select a data source in the repository or create a new data source on-the-fly. To define the data source for the simple report example: 1. Proceed to the Data Source page. The three choices for the data source are: • Do not link a data source – Select or define the data source at a later time. You see an error if you run the report in this state. • Click here to create a new data source – Define a new data source that is available only within the report unit. • Select data source from repository – Select an existing data source from the repository. 2. In this example, select Select data source from the Repository and click Browse to locate /Data Sources/JServer JNDI Data Source. Data Source Page 3. Click Query to proceed to the Query page. The query in the report unit determines what data is retrieved from a data source. You can use an existing query or define a new one. You can create multiple reports that look the same but contain different data by defining different queries for the same JRXML file. The simple report example defines a custom query that displays accounts from a single country. To define a custom query for the simple report example: 1. Proceed to the Query page. The three choices for the query are: • Do not link a Query – Uses the query, if there is one, defined within the main JRXML. If there is no query in the main JRXML, you see an error when trying to run the report. • Click here to create a new Query – Lets you define a new query that is available only within the report unit. • Select a Query from the Repository – Select an existing query object previously defined and stored in the repository. The AllAccounts.jrxml file uploaded in section Uploading the Main JRXML already contains a query. This example overrides the existing query by defining a new query in the report unit. 2. On the Query page, select the Click here to create a new Query option: Query Page 3. Click the link, Click here to create a new Query. The Name the Query page appears where you can enter the properties for the local query. The locally defined query and its properties are visible only within the report unit. 4. In this example, the query must retrieve only Canadian accounts. Enter the following values: • Name – CanadaAccounts • Resource ID – CanadaAccounts • Description – Query for New Simple Report in User Guide Name the Query Page The Link a Data Source to the Query page appears. When specifying a local query for the report unit, you can select a data source that overrides the one associated with the report unit. You can select a query from the repository or define a new one. Selecting None uses the default data source that you selected in section Defining the Data Source. 6. Select Do not link a data source to use the report unit's existing data source, then click Next. The Define the Query page appears. 7. Select SQL in the Query Language drop-down and enter the following Query String to retrieve only Canadian accounts: SELECT * FROM accounts WHERE billing_address_country = "Canada" ORDER BY billing_address_city Entering a Locally Defined Query on the Edit Query Page 8. Click Save to save the query. The Customization page appears. No customization is required. 9. Click Submit to submit the new report unit to the repository. To submit a new report unit to the repository, click Submit on any page, from the Set Up page to the Customization page, skipping pages you do not need. When you click Submit, the server validates all the elements of the report unit that you entered, and saves the new report unit in the repository with the type Report. It's always a good idea to test the report and review the output. After clicking Submit, a message appears at the top of the repository page, indicating that the report was successfully saved. The report appears in the repository with the properties you specified on the Set Up page: New Simple Report Added to the Repository If the report was not validated and saved, you may be prompted to review your entries. To run the report and view the output, click its name, New Simple Report. As shown in the figure below, all the accounts listed are in Canada. Output of the New Simple Report On the last page of the report are the two logo images that you added as file resources: Output of the New Simple Report Showing the Image Resources For more information about running reports, see section Running a Simple Report. This section describes the creation of a more complex report than the one described in the previous section. The complex report includes a full set of input controls, a scriptlet class JAR file, and resource bundles. To create the report, you must have access to all its resources, including the SalesByMonth.jrxml file and the SalesByMonthDetail.jrxml file (the sub-report). The complex report also uses a datatype and an image in the repository that are part of the sample data installed with the server. To simplify learning how to upload a complex report, familiarize yourself with the SalesByMonth.jrxml file and the resources it requires. The report file and its resources are found in the /samples directory. The example uses these resources to create a report that behaves just like the SalesByMonth report in the Reports > Samples folder of the repository. The beginning of the complex example resembles the beginning of the simple example. For detailed instructions, see sections Uploading the Main JRXML and Uploading Suggested File Resources. The following procedure assumes you are familiar with how to add a resource as described in the previous section. To upload the main JRXML and suggested resource files for the complex report example: 1. Log in to JasperReports Server as administrator and select View > Repository. If you log in as a user, you can upload a report unit to the server, but the example requires an administrator login to access the image resources. 2. Locate the folder where you want to add the report. For example, go to Organization > Reports > Samples. 3. Right-click the Samples folder and select Add Resource > JasperReport from the context menu. Add Resource appears on the context menu only if your user account has write privilege to the folder. 4. Enter the following properties for this example on the Set Up the Report page of the JasperReport wizard: • Name – New Complex Report • Resource ID– New_Complex_Report • Description – This is a complex example 5. On the Set Up the Report page, select Upload a Local File. 6. Click Browse to locate the file /samples/reports/SalesByMonth.jrxml. 7. Click Controls & Resources. On the Controls & Resources page, the list of suggested resources indicates that you need a sub-report (the SalesByMonthDetail.jrxml file) and a logo image. 8. On the Controls & Resources page, upload the sub-report: Click Add Now in the same row as SalesByMonthDetail. a. Select Upload a Local File. b. Click Browse to locate the file /samples/reports/SalesByMonthDetail.jrxml. c. On the Locate File Resource page. e. On the Add a Report Resource page. f. Click Next to accept the default report resource name and resource ID. 9. On the Controls & Resources page, upload the logo image resource: Click Add Now in the same row as Logo. g. On the Locate File Resource page, click Select a resource from the Repository. h. Click Browse to locate the file /Images/JR Logo. The Add a Report Resource page is displayed. Suggested Resources Added to the Report on Add a Report Resource Page j. On the Add a Report Resource page, click Next to accept the default name, resource ID, and description. The second suggested resource is added. The JasperReport wizard cannot detect every type of resource referenced in the main JRXML. You must be aware of other file resources needed in the report and upload them from the Controls & Resources page. In addition, you must know how they are referenced and give the file resources exactly the same name so they can be associated. When you upload a TrueType font as a resource, the file must have the .TTF file extension. The complex report example needs a scriptlet JAR and two language bundles (Java properties files). To upload the undetected file resources for the complex report example: 1. Add and upload the scriptlet JAR file: On the Controls & Resources page, click Add Resource. k. On the Locate File Resource page, select Upload a Local File, and click Browse to locate the /samples/jars/scriptlet.jar file. m. The Add a Report Resource page indicates that the file was successfully loaded and automatically detected as a JAR. Enter the following information; the Resource ID is referenced in the main JRXML file, so do not change it: Name – Scriptlet Resource ID– Scriptlet Description – Scriptlet JAR for complex example Adding the Scriptlet JAR as a Report Resource shows these values entered on the Add a Report Resource page: Adding the Scriptlet JAR as a Report Resource 3. Add and upload the English resource bundle: On the Controls & Resources page, click Add Resource again. n. Select Upload a Local File, click Browse to locate the file /samples/resource_bundles/sales.properties. o. In Locate File Resource, click Next. The Add a Report Resource page indicates that the file was successfully loaded and automatically detected as a resource bundle (a Java properties file): Adding the English Resource Bundle on the File Resource Pages p. Enter the following information: Name – sales.properties Resource ID – sales.properties Description – Default English resource bundle Enter the resource ID field exactly as it is shown. 5. Add and upload the Romanian Resource bundle as described in the following steps: On the Controls & Resources page, click Add Resource again. q. Select Upload a Local File and click Browse to locate the file /samples/resource_bundles/sales_ro.properties. r. In Locate File Resource, click Next. The Add a Report Resource page indicates that the file was successfully loaded and automatically detected as a resource bundle (a Java properties file). s. Enter the following information: Name – sales_ro.properties Resource ID – sales_ro.properties Enter the resource ID field exactly as it is shown. Description – Romanian resource bundle Adding the Romanian Resource Bundle Controls & Resources lists all the files: Undetected File Resources Added on the Controls & Resources Page If you want to upload a different file for a named resource, click its resource ID in the Resources list and locate the new file or repository object. You can also change the name and description of the resource, but not its resource ID. If there is a mistake in a resource ID, remove the resource and re-add it. To remove the file resource, locate the ID in the list of resources on Controls & Resources, and click Remove at the end of the row. Input controls are standard graphical widgets that take user input in a pre-determined format and pass it into the report. When a report is designed and configured for input controls, a user can modify run time selections, such as the WHERE filter clauses in SQL parametrized queries. The server displays the input control widgets with the report and regenerates the report when the user changes the input values. Input controls correspond to the parameters defined in JRXML reports, such as $P{name}. You must define an input control for every parameter in the report. The server maps the value that the user enters for the input control to the parameter of the same name. The server returns an error if there is no parameter name that matches the name of an input control. You must define an input control even when parameters have default values. To prevent users from changing the default, you can make the input control read-only or invisible. Input controls are complex objects that can be configured in many ways. However, you must create the input controls so that they return the exact type, range, and number of values expected by the report. In general, you must define a datatype for user entries, such as number or text, that the input control accepts. You also must specify that the input control use one of the following widget types: • Boolean – A yes/no value determined by a check box. • Single value – A text, number, date, or date/time, optionally constrained by a minimum value, maximum value, or both. Text input can also be constrained by a matching pattern. Single values provide a text box for the user to enter a value, but date and date/time also provide a calendar icon. • Choice of multiple values – The choices offered to the user can either be a static list of values that you specify or a dynamic list of values returned by a separate query. After determining the list of values, choose the widget type: • Drop-down list to select a single value. • Radio buttons to select a single value. • Multi-select list to select multiple values. • Check boxes to select multiple values. All choices are presented and selected by their text representation, but the report may process the input value, for example, to interpret a string of digits as a number. The query in the SalesByMonth.jrxml file of the complex example has several parameters that require input controls. Each input control is a different type to demonstrate how each can be added to the report unit. Input controls and their datatypes can be locally defined or reused from the repository. In the example, you define most components locally to understand how to use the JasperReport wizard. The simplest input control is a text box into which the user types a value. In this example, the datatype for the value is a number, meaning that the server verifies that the user input is numeric. To add a text input control to the complex report example: 2. On the Controls & Resources page, click Add Input Control. The Locate Input Control page appears: Locate the Input Control page 3. Select Define an Input Control in the next step. 5. On the Create Input Control page, accept the default type of input from the Type drop-down: Single Value. 6. Enter the other properties for the input control: The name is referenced in the main JRXML file, so you must enter it exactly as shown. • Prompt Text – The label that the user sees next to the widget for this input: Text Input Control • Parameter Name – The name of the parameter in the report that receives the user value: TextInput • Description – An optional description that appears only within the report wizard: leave blank in this example • Mandatory, Read-only, Visible – A setting that affects how the input control is displayed: check only Visible. Properties of the Text Input Control You can reuse input controls that you created for similar reports if the parameter name is consistent. The server does not run a report if it has input controls that do not match any parameter name. Also, you must create the input control in the repository, you cannot reuse input controls that you defined locally in another report. 8. In Locate Datatypes, select Define a DataType in the next step, and click Next: Specifying a Locally Defined Datatype You can locate a datatype already stored in the repository. Datatypes can be simple definitions of the expected input (numbers, text, date, or date/time). Datatypes can also include range restrictions that the server enforces. To reuse a datatype saved in the repository, make sure its type and range are compatible with the input control you want. 9. In Set the Datatype Kind and Properties, enter the properties for the locally defined datatype associated with the input control: Select Number from the drop-down. u. Enter a name – Integer Type v. Enter a resource ID – Integer_Type The name and resource ID are required, but they are visible only when defining the input control: • Description – An optional description that appears only within the report wizard: leave blank in this example • Type – Determines the format of the data that the user may enter: Number The number format allows users to enter both integers and decimals. The report must handle both kinds of values when processing the parameter that receives the number. • Minimum value – The lower bound of the value the user may enter: leave blank in this example • Maximum value – The upper bound of the value the user may enter: leave blank in this example • Minimum is strict – Means the minimum value itself is not allowed: leave unchecked in this example • Maximum is strict – Means the maximum value itself is not allowed: leave unchecked in this example Defining the Properties of the Integer Datatype The Controls & Resources page appears with Text Input Control in the list of resources: Added Controls in the Resources List of the Report Wizard A simple check box provides a true/false (boolean) input that can be used to conditionalize some part of the report. To add a simple check box input control to the complex report example: 1. Continuing with the previous example, on the Controls & Resources page, click Add Input Control. 2. On the Locate Input Control page, click Define an Input Control in the next step. 4. On the Create Input Control page, select the type of input from the Type drop-down: Boolean. 5. Enter the other properties for the input control. The name is referenced in the main JRXML file, so you must enter it exactly as shown: • Prompt Text – Check Box Input Control • Parameter Name – CheckboxInput • Description – Leave blank in this example. • Mandatory, Read-only, Visible – Check only visible. See "Entering the Properties of the Check Box Input Control" shows the Create Input Control page after you enter these properties: Entering the Properties of the Check Box Input Control The Controls & Resources page appears with the new check box input control. The drop-down input control gives the user a pre-determined list of choices to select from. The design of the report determines whether you must create a single-select or multi-select list. When creating the input controls you can select whether to present a single choice as a drop down list or a set of radio buttons. Multi-select controls can be either a multi-select list or a set of check boxes. Radio buttons and check boxes are usually appropriate if there are few choices, usually five or less. This example has only three pre-determined choices but uses a drop-down list. The predetermined choices are defined in a list of values object. You can either create one locally for the input control or reuse one that you created beforehand and stored in the repository. To add a drop-down input control to the complex report example: 1. Continuing with the previous example, on the Controls & Resources page, click Add Input Control. 2. On the Locate Input Control page, click Define an Input Control in the next step. 4. On the Create Input Control page, select the type of input from the Type drop-down: Single-select List of Values. 5. Enter the other properties for the input control. The name is referenced in the main JRXML file, so you must enter it exactly as shown: • Prompt Text – List Input Control • Parameter Name – ListInput • Description – Leave blank in this example. • Mandatory, Read-only, Visible – Check only visible. See "Entering the Properties of the List Input Control" shows the Create Input Control page after you enter these properties: Entering the Properties of the List Input Control 7. On the Locate List of Values page, you can locate a list of values already stored in the repository. If you have a list of values saved in the repository that you want to reuse, make sure its values are compatible with the parameter defined in the JRXML report that receives this input. For this example, select Define a list of values in the next step: Locating the List Type for the List Input Control 9. On the Add List of Values page, enter a name, resource ID, and optional description for the locally defined list of values. These properties are not visible outside of the input control. Enter the following values: • Name – list_type • Resource ID – list_type • Description – Leave blank in this example 10. In the Name Value panel, enter values to present as choices to the user. Each value has a name that is displayed instead of the actual value. The actual value is passed to the report, so you must ensure that its type and value match the parameter definition in the JRXML report. For each value that you want to add to the list, enter its name and value, then click Add. If you make a mistake, click Remove. The name for every item must be unique. For this example, enter: • Name First Item with value 1. • Name Second Item with value 2. • Name Third Item with value 3. Defining the List of Values The Controls & Resources page appears with the new List Input control. This example of adding a date input control uses a datatype that already exists in the sample data in the repository: To add a date input control to the complex report example: 1. On the Controls & Resources page, click Add Input Control. 2. On the Locate Input Control page, select Define an Input Control in the next step. 4. On the Create Input Control page, select the type of input from the Type drop-down: Single-Value. 5. Enter the other properties for the input control. The name is referenced in the main JRXML file, so you must enter it exactly as shown: • Prompt Text – Date Input Control • Parameter Name – DateInput • Description – Leave blank in this example. • Mandatory, Read-only, Visible – Check only the visible setting. See "Properties of the Date Input Control" shows the Create Input Control page after you enter these properties: Properties of the Date Input Control 7. On the Locate Datatypes page, select Select a Datatype from the Repository. Click Browse. 8. In Select Resource from Repository, expand Input Data Types, and select the Date Datatype. Selection of the Datatype from the Repository for the Date Input Control The Locate DataTypes page shows the location of this datatype in the repository, /datatypes/DateDatatype: Selection of Datatype for the Date Input Control The Controls & Resources page appears with the new Date Input Control. The last example of an input control shows how the choices available to the user can be dynamically based on the data. A query-based input control presents a dynamically-created list of choices to the user. The server performs a query whose results are used to create the list of choices. You must perform the following tasks: • Configure the query. • Designate how to display the results in the input control. • Specify the value to pass as the corresponding parameter. To add a query-based input control to the complex report example: 1. On the Controls & Resources page, click Add Input Control. 2. On the Locate Input Control page, select Define an Input Control in the next step. 4. On the Create Input Control page, select the type of input from the Type drop-down: Single-select Query. 5. Enter the naming properties for the input control: • Prompt Text – Query Input Control • Parameter Name – QueryInput • Description – Leave blank in this example. • Mandatory, Read-only, Visible – Use the default settings in this example. The parameter name is referenced in the main JRXML file, so you must enter it exactly as shown. See "Entering the Properties of the Query Input Control" shows the Create Input Control page after you enter these properties: Entering the Properties of the Query Input Control For a query-based input control, you can use a query stored in the repository. This example defines a query that is local to the input control with its own properties and query string. 7. On the Locate Datatypes page, select Define a Query in the next step. 9. On the Name the Query page, enter naming properties for the local query. For this example, enter testQuery in both the Name and Resource ID fields: Defining the Properties of the Query for the Query Input Control 11. On the Link a Data Source to the Report page, select Do not link a data source to use the same data source you defined for the report. In your own reports, you can select a different data source from the repository or define one that is local to the input control: Locating the Data Source for the Query Input Control 13. On the Define the Query page, select SQL from the Query Language drop-down. 14. Enter the following Query String to retrieve both the labels and the values to be displayed by this input control: SELECT user_name, first_name, last_name FROM users Editing the Query String for the Query Input Control 16. For each row in the results of the query, the server displays one choice in the widget you chose as the input type (drop-down, radio buttons, mutli-choice, check boxes). On the Query Information page, name the database column to be used as the input value and to be presented to the user. The column names must match those in the SELECT clause of the query string exactly. In the Value Column, enter the user_name. w. In the Visible Column, enter first_name. y. In the Visible Column, enter last_name. For each visible column that you want to display as a choice, enter the name, then click Add. If you make a mistake, click Remove. Selecting Columns for the Query Input Control The Controls & Resources page displays all the resources, including the new input controls. Input Controls and Resources 18. Next, set the input control options, described in the next section. Set the display mode in Input Control Options to complete the configuration of input controls. The options appear at the bottom of the Controls & Resources page only after you define one or more input controls. Leave Optional JSP location blank for this example. You can use the JSP Location to specify the path to a JSP file that changes the appearance of the input controls. To configure the display mode of input controls in the complex report example: 1. Specify how the input controls appear when the report runs. Select Pop-up window. You can also select Separate page to make the input controls appear in a separate browser window or In page to make input controls appear at the top the report. 2. Check Always prompt to force the server to prompt the user when the report runs. The definition of input controls in this example specified Visible and not Mandatory. When input controls are not mandatory and Always prompt is not checked on the Controls & Resources page, the user must click the Options button in the Report Viewer to change input controls; otherwise, the report runs with default input controls. Proceed to section Completing the Complex Report Example to set the data source and finish the complex report example. JRXML-based reports can include input controls whose values are determined by a user's selection in other input controls. For example, consider a report that has input controls for country, state, and city. By leveraging the cascading functionality, the options in the State input control are determined by the value selected in the Country input control. When the user selects a state, the list of values includes only those in the selected state. These cascading input controls use queries to determine the values to display in each input control field. The goal is to use input controls as parameters for a query that populates another input control. To do so, the input control's query must use a special syntax to reference a parameter name. The syntax is identical to the $P{parameterName} syntax used in queries for JasperReports. Consider an example report that returns data identified by country and city. It includes input controls called COUNTRY and CITY. The query used by the CITY input control retrieves city names from a table called accounts based on the value selected by the COUNTRY input control, for example: select city from accounts where country = $P{COUNTRY} When the user selects a country from the COUNTRY input control, the value selected is used by the query of the CITY input control; the CITY input control is refreshed to show the result. Note that there are other ways to use a parameter in a query. For an example of running a report with cascading input controls, see section Cascading Input Controls. To finish the complex report example and validate it: 1. On the Controls & Resources page of the JasperReport wizard, select Data Source. 2. On the Locate Data Source page, select Select data source from repository. 3. Click Browse, and select Organization > Data Sources > JServerJNDI Data Source. 5. On the Locate Query page, click Submit to save and finish the complex report example. You can skip pages of the JasperReports wizard, such as Query, to use the default settings on those pages. The server validates the report and a message appears indicating that the report was added to the repository. The report appears in the repository: New Complex Report Added to the Repository 6. Click the display name of the new report to run and view the report. Input controls appear as you defined them: Entering Input Values to Run the New Complex Report 8. Click OK or Apply to run the report with the selected input. The server enforces the proper format defined for each input control. The Text Input Control is defined as a numeric type and accepts only valid numbers, as indicated by the message in See "Output of the New Complex Report Showing Values from the Input Controls". 9. In Text Input Control, enter 3 and click OK or Apply again. 10. The sample report includes a header that displays the value of each parameter received from the input controls: Output of the New Complex Report Showing Values from the Input Controls In the Report Viewer, you can open the Input Controls window at any time by clicking the Options button. Clicking OK closes the Input Controls window; clicking Apply keeps it open to easily change values and run the report again. If running the new report results in an error, edit the report to review your settings. If you cannot identify the problem, edit the SalesByMonth sample report (in the repository at /reports/samples) and compare its settings to those you configured for the new report. A datatype in the server is more than just a classification of the data. It also includes validation criteria. The datatype description page contains these fields: Name The name of the datatype. Label The label of the datatype. Description Any additional information you want to provide about the datatype. Type The actual datatype. Possible values are Text, Number, Date, or Date-Time. Max length The maximum number of characters allowed in the field. Decimals The number of decimal places allowed in the field. Pattern A regular expression that restricts the possible values of the field. Minimum value The lowest permitted value for the field. Is strict minimum If checked, the minimum value itself is not permitted; only values greater than the minimum value are permitted. Maximum value The highest permitted value for the field. Is strict maximum If checked, the maximum value itself is not permitted; only values less than the maximum value are permitted. JRXML reports (including Topics) support the $R expressions for field names; such expressions are used to localize reports into various languages. Each $R expression refers to an entry in a resource bundle properties file that you must create. For example, if you localize a field called StoreState, you might use this JRXML: The server looks for a label with the ID of StoreState. In order for this localization to work properly, you must create the resource bundle with the correct labels, and declare the resource bundle in the header of the JRXML file. For example: In this header, simpleTable is the name of the resource bundle file that contains the StoreState property. iReport can help you localize the report by creating the resource bundle. When you upload the report, you must upload the resource bundle as well, If you edit a resource bundle in iReport, you can right-click it in repository (using the JasperReports Server Plug-in) and select Replace with Current File. By default, the server uses three fonts you can use in reports: • DejaVu Sans • DejaVu Serif • DejaVu Sans Mono Using the DejaVu fonts shipped with the server ensures that the fonts are always available in all environments, which in turn ensures the PDF is pixel perfect every time. The DejaVu fonts replace the Java logical fonts used in previous versions of the server: • SansSerif • Serif • Monospaced SansSerif, Serif, Monospaced can still be used, but should be considered deprecated. This is because these Java logical fonts map to different TTF files in different environments, and run the risk of text being cut when exported to PDF, due to font metric mismatches. Also, the java logical fonts listed above aren't recognized by browsers and some browsers substitute other font families for the Java logical fonts. For example, Firefox in a Windows environment renders the SansSerif logical font as Serif. When using the DejaVu fonts coming from JasperReports font extensions, you don't need to set any other font attributes (such as the pdfXXX attributes) in the JRXML, nor do you need to specify font mapping. The font extension file that makes these fonts available takes care of setting font attributes and mapping. For more information about DejaVu, refer to its SourceForge project at: . When you upload a TrueType font to the repository, the filename must include the correct extension (.TTF). I have jrxml file, I want to compile this to get .jasper. How do I compile and get that jasper file? I have jrxml file, I want to compile this to get .jasper. How do I compile and get that jasper file?