

Creating Custom Reports Using the 25Live Interactive Report Designer

[+](#) Click here to view related articles.

In This Article:

- [Interactive Report Designer Introduction](#)
- [Additional Help](#)
- [Creating a New Report](#)
- [Creating a New Page Report](#)
- [Identifying Report Sections](#)
- [Inserting Data Fields Into a Report](#)
- [Moving, Resizing, and Hiding Report Objects](#)
- [Formatting and Styling Report Objects](#)
- [Filtering Report Data](#)
- [Viewing Report Output](#)
- [Saving a Report](#)
- [The Interactive Report Designer and WebServices](#)
- [The Report Run Process](#)
- [Report Objects - Data Containers](#)
- [Required - URL Source Parameter](#)
- [JReport Run](#)
- [Formulas and Summaries](#)
- [Formula Editor](#)
- [Referencing Fields in Formulas](#)
- [Formula Syntax](#)
- [Using Formulas to Control Object Properties](#)
- [Using Formulas to Combine Text with Data](#)
- [Modifying or Deleting a Formula](#)
- [Summaries](#)
- [Adding a Custom Report to 25Live](#)
- [The Report Run Process](#)

Interactive Report Designer Introduction

The 25Live Interactive Report Designer (the Designer) is a desktop application included with 25Live and developed by Jinfonet Software that is used to produce custom report templates. The Designer visual design environment uses familiar conventions such as drag-and-drop placement, property panels, and toolbars to support every aspect of the report design process.

The Designer is also used to create and manage report resources, including data sources, queries, formulas, summaries, and parameters. These report resources are stored in a central repository called a Catalog. All reports are assigned to a Catalog and share the resources stored in the Catalog.

To help you get started, CollegeNET has created a Catalog to distribute to you. This Catalog, 25LIVE_WS.cat, contains an initial set of pre-defined resources for you to use in building your custom reports. Once you begin to create your own reports, new queries, new formulas, summaries, and so on, this Catalog will become more unique to your institution.

Pre-Requisites

The most important pre-requisite before using the Designer is to have a basic understanding of the Series25 WebServices API. You may also find it useful to be familiar with:

- Data relationships, relational data in databases, relational models, and rows of data
- Viewing XML schema and knowing how to find what you are looking for
- How your 25Live instance is set up
- The verbiage used in 25Live reports, like space search, event search, and so on
- Selecting and setting up searches to use, test, and design your report(s)

You may wish to read through some of the [Web Services documentation](#) exclusively available to customers.

Report Components

Report Template - A report template comprises static text, graphics, and placeholders for data. At runtime, a report is connected to a data source, queries are executed, and the fetched data is applied to the report template to produce a result file.

Data Sources - Data is made available to a report through the Series25 WebServices API. A Data Source roughly equates to a connection to a Series25 web service. Each Series25 web service outputs XML data in a specific structure that is documented in a schema file (.xsd). The schema file is used to create the corresponding JReport Data Source. JReport transforms the hierarchical structure of the XML web service data into a relational model. Complex elements, XML elements that have child elements, are transformed into tables, and simple elements are transformed into fields/columns.

Queries - Queries are created based on the relational model of their parent Data Source. Only the tables and columns selected in a query are made available to a report. Table relationships, such as equal (=) join and left outer (!=), are defined at this query level. Filtering can be applied to a query to further refine results returned by a web service request. Multiple queries can be created for a single Data Source, which can target subsets of the data available in the parent Data Source.

Formulas - Formulas are objects that are computed at runtime which allow you to manipulate field data by performing calculations on that data. Formulas can be used for many different purposes including:

- Performing calculations between data elements to produce values not present in the data model.
- Constructing URL strings used to request Series25 WebServices data.
- Controlling report object properties, including conditional formatting, object visibility, object position, and behavior when exported to output formats such as MS Excel.

Formulas may typically be created for use by a specific report; however, they are stored in the Catalog as children of a Data Source and can, therefore, be shared by other reports using the same data source.

Summaries - Summaries are the application of standard functions, count, average, sum, and so on, to data fields or existing formulas. Summaries can be applied to grouped sections of data as well as to the whole report, such as with a grand total. Like formulas, summaries are stored in the Catalog as children of the Data Source and can be shared by other reports using the same data source.

Parameters - A Parameter is a variable whose value is input at runtime. Each Data Source is created with a required parameter that is anticipating the appropriate Series25 WebService URL to make a data request.

Additional Help

This documentation is intended as a companion guide to be used with [Insight Software's JReport website](#), their customer care portal, and their [online user's guide](#).

You will find links to detailed topics in the *JReport Designer User's Guide* throughout this document below this icon:

Creating a New Report

Catalogs

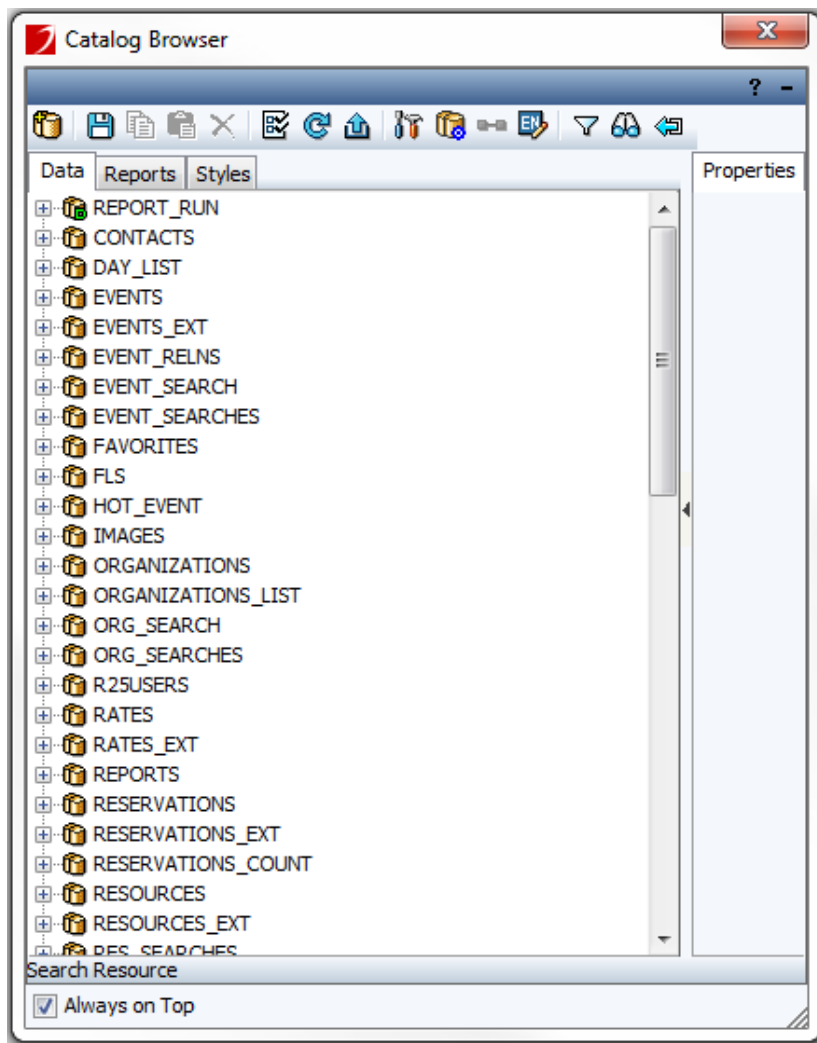
A Catalog is a file that stores all the object definitions created while developing the reports that are also stored in the Catalog. This central repository can be viewed and modified using the Catalog Browser which is a component of the Designer that assists with managing data connections and other report resources.

Report resources stored in the catalog include: Data Sources, Queries, Formulas, Summaries, Styles and Parameters. The creation of these resources can be initiated from either the report design editor or the Catalog Browser.



For information on catalogs that may not be covered in this topic, please see [The JReport Designer User's Guide](#).

- *Home > Catalogs*
- *Home > Report Development Environment > Catalog Browser panel*



Connecting to a Data Source

A Data Source represents a connection to the Series25 database via the WebServices API. CollegeNET has created a catalog file (25LIVE_WS.cat) for you that contains Data Source connection definitions to 65 of the most common WebServices used for reporting purposes. This information is intended to help you if you need to connect to another data source.

Data Source Based on XML Schema

Each Data Source is created by supplying the Designer with a schema file (.xsd) that describes the XML structure of a given WebService request result. The Designer translates the hierarchical structure of the WebServices XML into a relational model of tables and fields and injects primary and foreign key values into each table which it uses to create default table relationships derived from the schema file.

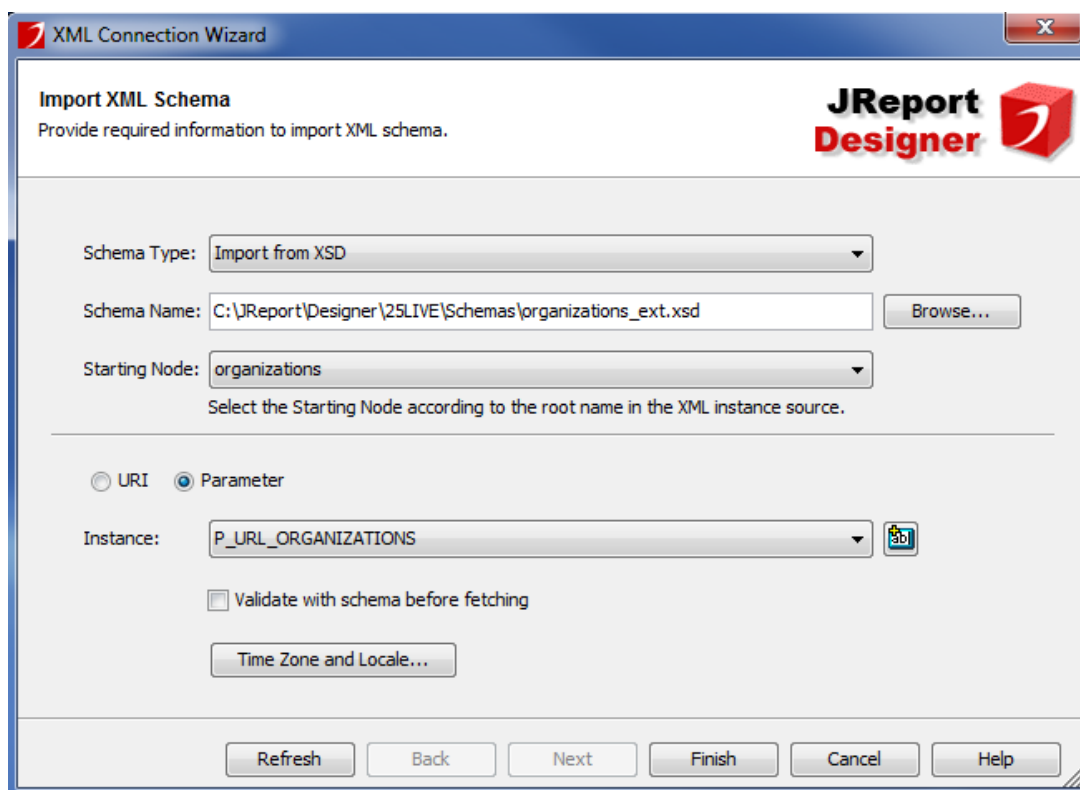
A copy of each schema file (.xsd) referenced in the 25LIVE_WS.cat is available as part of your local reporting directory at *C:\JReport\Designer\25LIVE\Schemas*.

Data Source Has Required Parameter

A required Parameter is defined for each Data Source, which is intended to receive the WebServices URL used to make

a data request. A standardized naming convention is recommended for parameters created to receive WebServices URLs. This includes a prefix of "P_URL_". The Parameter name cannot contain any literal space.

This is an example of the XML Relational Data Source Connection Wizard used to create a new Data Source.



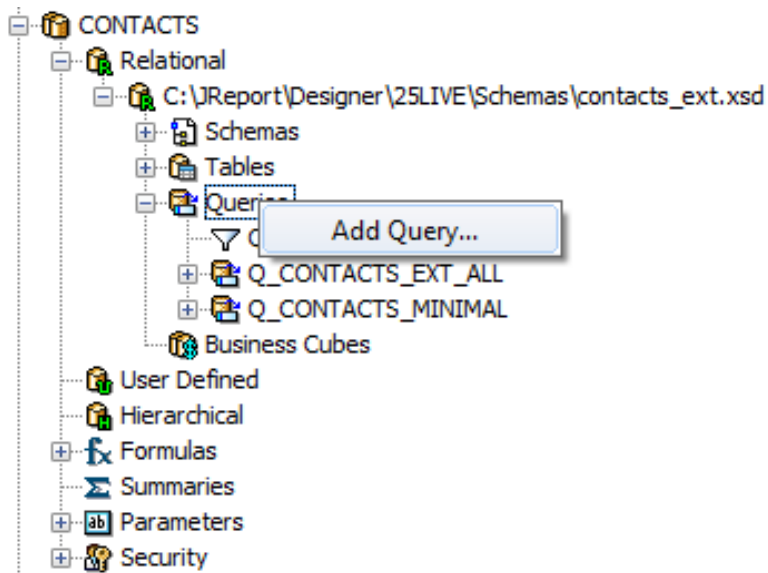
Working with Queries

Queries are created based on the relational model of their parent Data Source. A Query is used to present selected tables and fields, along with appropriate data relationships, to the report. Queries can be managed within the Catalog Browser using the Query Editor.

- *Home > Queries*

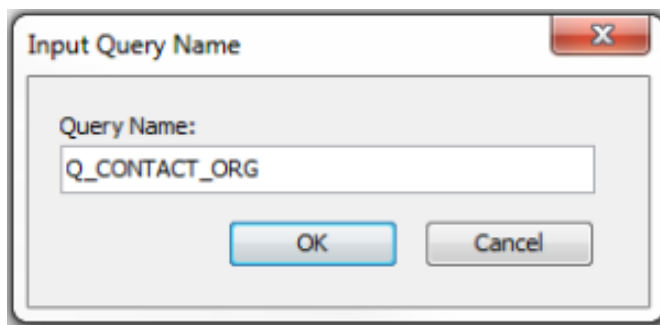
Adding a Query

Open the Catalog Browser and continue to expand the Data Source tree structure until you see the Queries node. Right-click Queries and select "Add Query...".



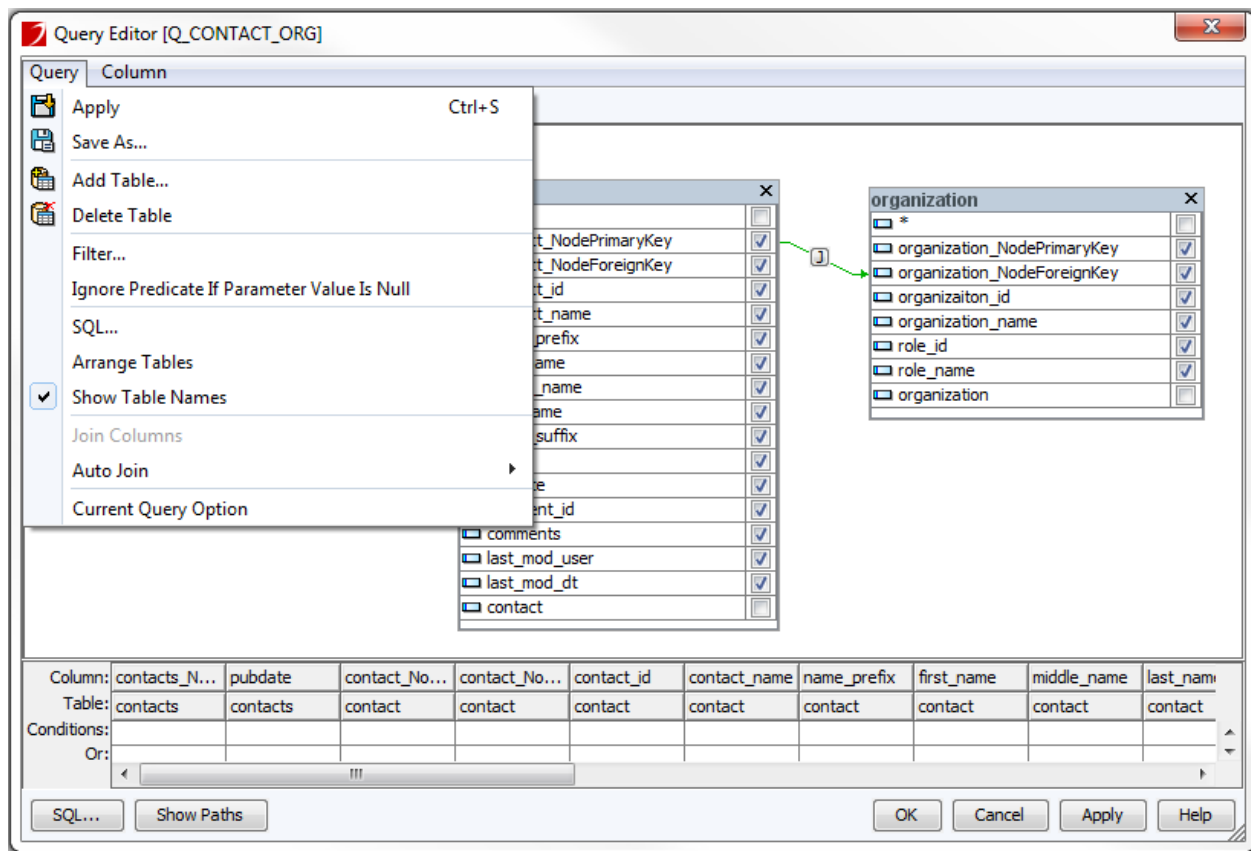
Naming a Query

A standardized naming convention (using a prefix of "Q_") is recommended for all Queries. Query names cannot contain any literal space.

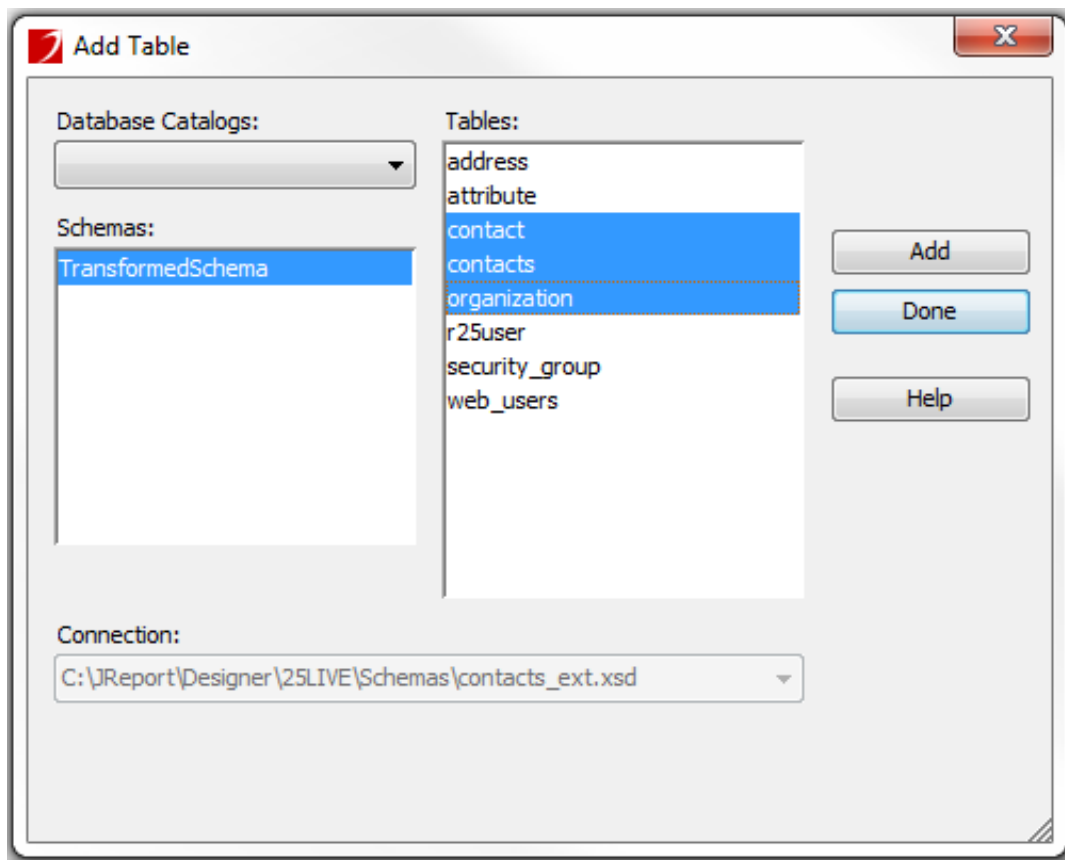


Selecting Tables for a Query

Tables are added and deleted within the Query Editor window.



From the Add Table dialogue, highlight the Tables to be added and click the Add button.



Selecting Query Fields

While in the Query Editor, put a check next to each field you may want to reference in a report.

 For information on queries that may not be covered in this topic, please see [The JReport Designer User's Guide](#).

- [Home > Dialog Reference > Query Editor window](#)



In each table, there is a field listed with the same name as the table, as shown in this example. This represents the complex element name from the WebServices XML used by JReport to create the table. It is recommended that you not check this field, as it has no associated value.

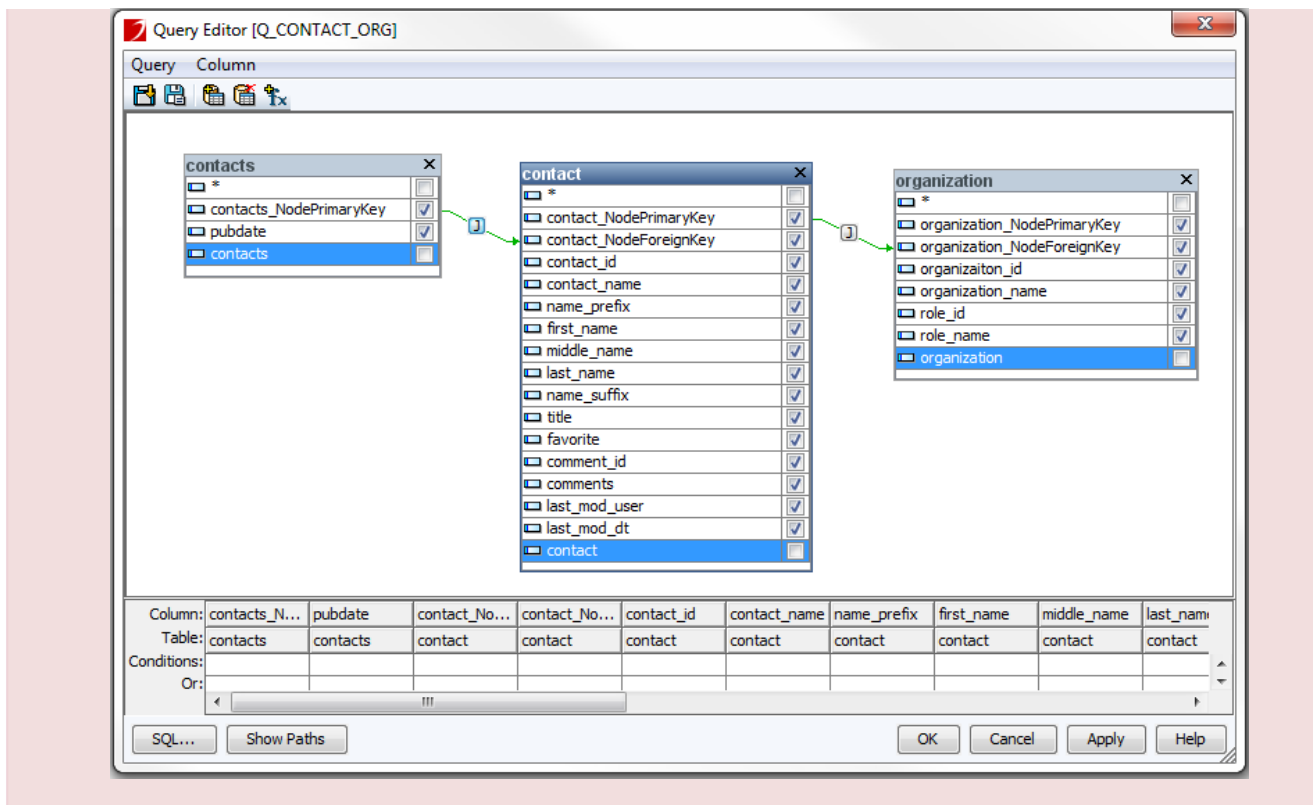


Table Joins and Relationships

When adding tables to a query, the default table joins should be established for you. Arrow lines drawn between tables provide a visual representation of their relationship. These default table joins (pre-joins) are identified when the parent Data Source is created, and are derived from the nested structure of the XML data source. By default, the Designer will link tables using an equal join [=] relationship between the NodePrimaryKey from the “link from” table and the NodeForeignKey of the “link to” table. The Designer adds these Key values to the data table when the Data Source is created.

If the default table joins were not established for you, click and drag your mouse cursor from the NodePrimaryKey of the “link from” table to the NodeForeignKey of the “link to” table.

- *Home > Catalogs > Managing resources in a catalog > Defining the join relationships between tables*

Equal Join

An equal join [=] relationship allows values from two or more tables to be combined when matching link values are found in each table.

Outer Join Left

In cases where data is optional, consider changing the default equal join [=] to an Outer Join Left [!=]. This outer join allows for all records in the primary table while selecting values in the lookup table when a link match is found.

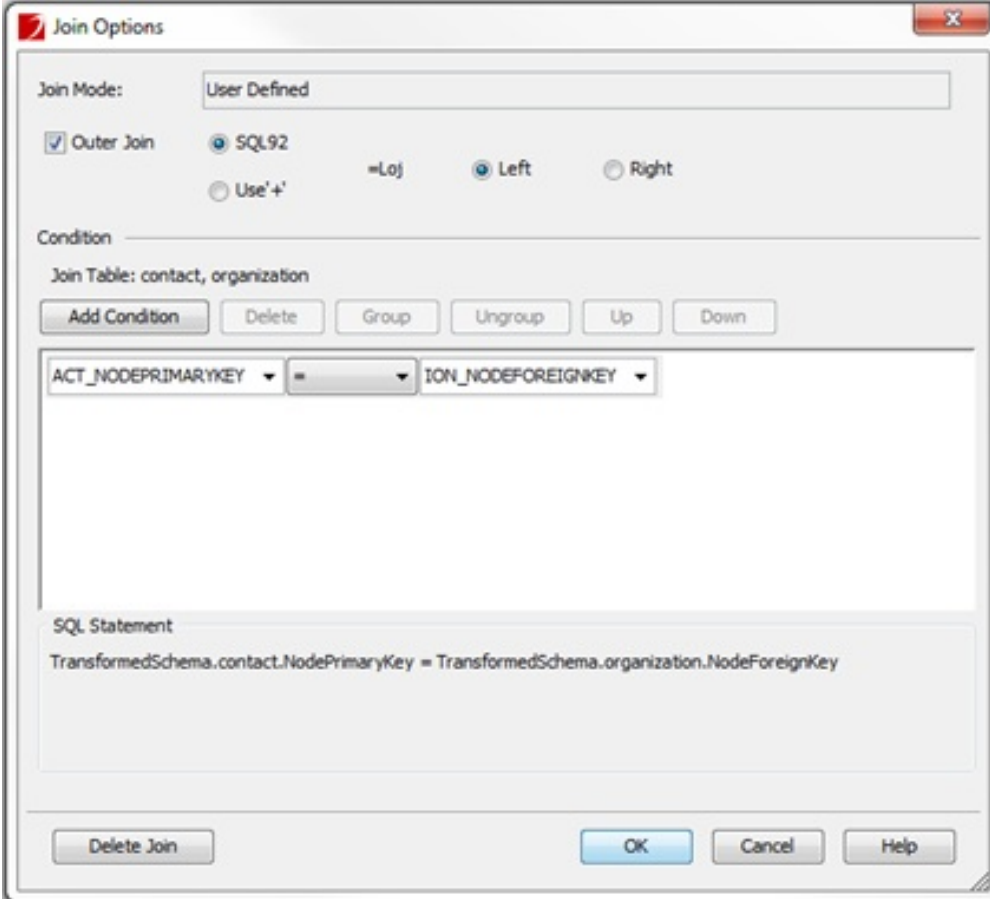
One-to-One & One-to-Many relationships

When joining tables together, it is important to make note of the type of relationship that exists between database tables. The 25Live data model contains many examples of both one-to-one and one-to-many relationships. The

number of rows generated when linking tables together is based on whether 0, 1 or many matches are found between link values. In a JReport, the details section of a report repeats once for each combination of matches found between all linked tables.

Changing to Outer Join

To change the details of an existing table join, double-click the join button on the arrow linking two tables together. This will bring you to the Join Options dialog. To change an equal join to an outer join, check the box next to Outer Join.

The image shows a 'Join Options' dialog box. At the top, 'Join Mode' is set to 'User Defined'. Below this, the 'Outer Join' checkbox is checked. There are three radio buttons for join types: 'SQL92' (selected), 'Left', and 'Right'. There is also a 'Use '+'' option. Below these, there are buttons for 'Add Condition', 'Delete', 'Group', 'Ungroup', 'Up', and 'Down'. A condition is listed: 'ACT_NODEPRIMARYKEY = ION_NODEFOREIGNKEY'. At the bottom, there is a 'Delete Join' button and 'OK', 'Cancel', and 'Help' buttons. The 'SQL Statement' field at the bottom shows: 'TransformedSchema.contact.NodePrimaryKey = TransformedSchema.organization.NodeForeignKey'.

Deleting existing links

From the Join Options dialog, click the Delete Join button. This will automatically bring you back to the Query Editor window.

Creating new links

From the Query Editor window, click and drag your mouse cursor from the NodePrimaryKey of the "link from" table to the NodeForeignKey of the "link to" table.



Table Links Can Use Fields Other Than the NodePrimaryKey and NodeForeignKey

Table links can be established using fields other than the NodePrimaryKey and NodeForeignKey. These Key values simply represent the default relationships represented by the WebServices XML structure. Other field values that are common between two tables can be used to establish a table relationship.

Saving a Query

From the Query Editor window, click the Apply button to automatically apply Query changes to the Catalog. Click the OK button when finished reviewing query details in the Query Editor to return to the Catalog Browser.

Saving a Catalog

From the Catalog Browser, click the save button to save the currently selected Catalog.

Creating a New Page Report

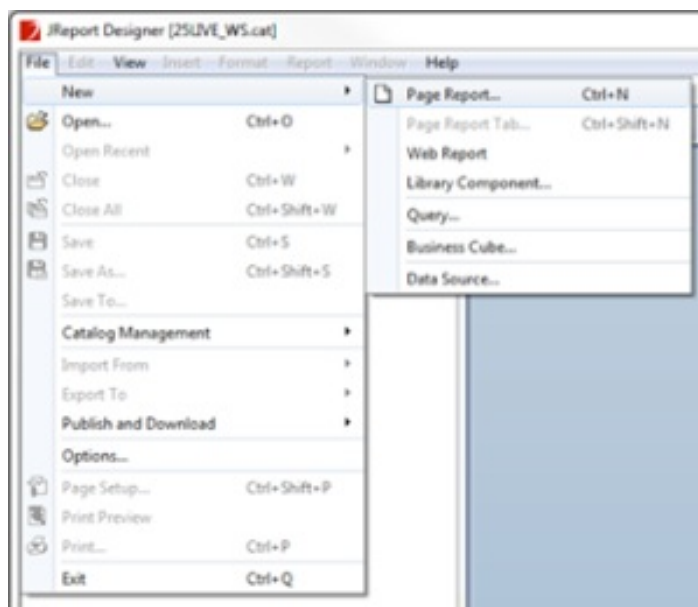
This section focuses on creating new Page Reports (.cls). Other Designer formats, such as Web Report (.wls) and Dashboards (.dsh), will be covered in later documentation editions.

 For information on reports that may not be covered in this topic, please see [The JReport Designer User's Guide](#).

- *Home > Page Reports*

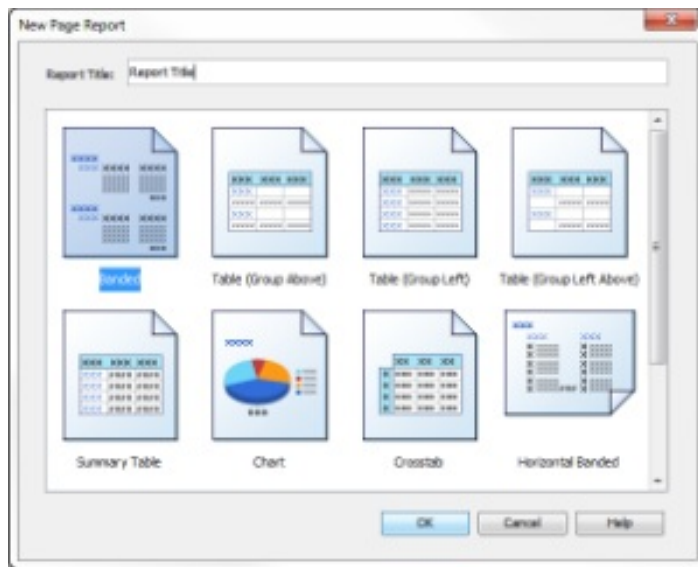
Report Design Wizard (Banded Wizard, Table Wizard)

The Designer provides a report design wizard to assist you in creating a new report. From the report design editor, choose the menu option File > New > Page Report, or press Ctrl+N.



Report Components

The report design wizard first brings you to the New Page Report dialog. Report objects representing different report types are made available.



These objects will provide a base structure to your report.

Banded object

A banded object is a kind of component that can present both grouped data and detailed data. A banded object is composed of several banded panels with which you can easily organize data fields and other elements. Multiple banded objects can be used to make a single report.

In the Designer, a banded object is a data container, which means that it displays the results of a query. Banded objects can be inserted into other data containers. When appropriate, data container links can be established between the banded object and its parent.

Banded objects are most appropriate to use when the report output result does not line up into neat rows and columns. A banded object allows for free-form placement of data fields and other report elements.

- *Home > Components > Banded objects*

Table object

The Designer assists in creating different types of table objects by anticipating where the group header should be placed. Group Above, Group Left, Group Left Above, and Summary Table are all examples of different types of table objects.

Like banded objects, tables are data containers that display the results of a query. Tables can be created to display both grouped and detailed data. Multiple table objects can be used to make a single report.

Tables provide additional structure and control over the placement of fields, grouping, and sorting data. Tables offer an efficient way to show any two-dimensional data set and are a good choice for lists or spreadsheet-type results.

- *Home > Components > Tables*

Other report components

Other report components include Chart, Crosstab, Horizontal Banded, Mailing Label, Tabular, and Blank. These

components will be discussed in later documentation editions.

Banded Wizard

This documentation focuses on the use of Banded objects for building reports. The Designer provides a Banded Wizard to assist with the initial setup of a Banded object. The Banded Wizard provides a breadcrumb display to assist with navigation that may include some or all of the following items:

Data > Display > Group > Summary > Chart > Filter > Style

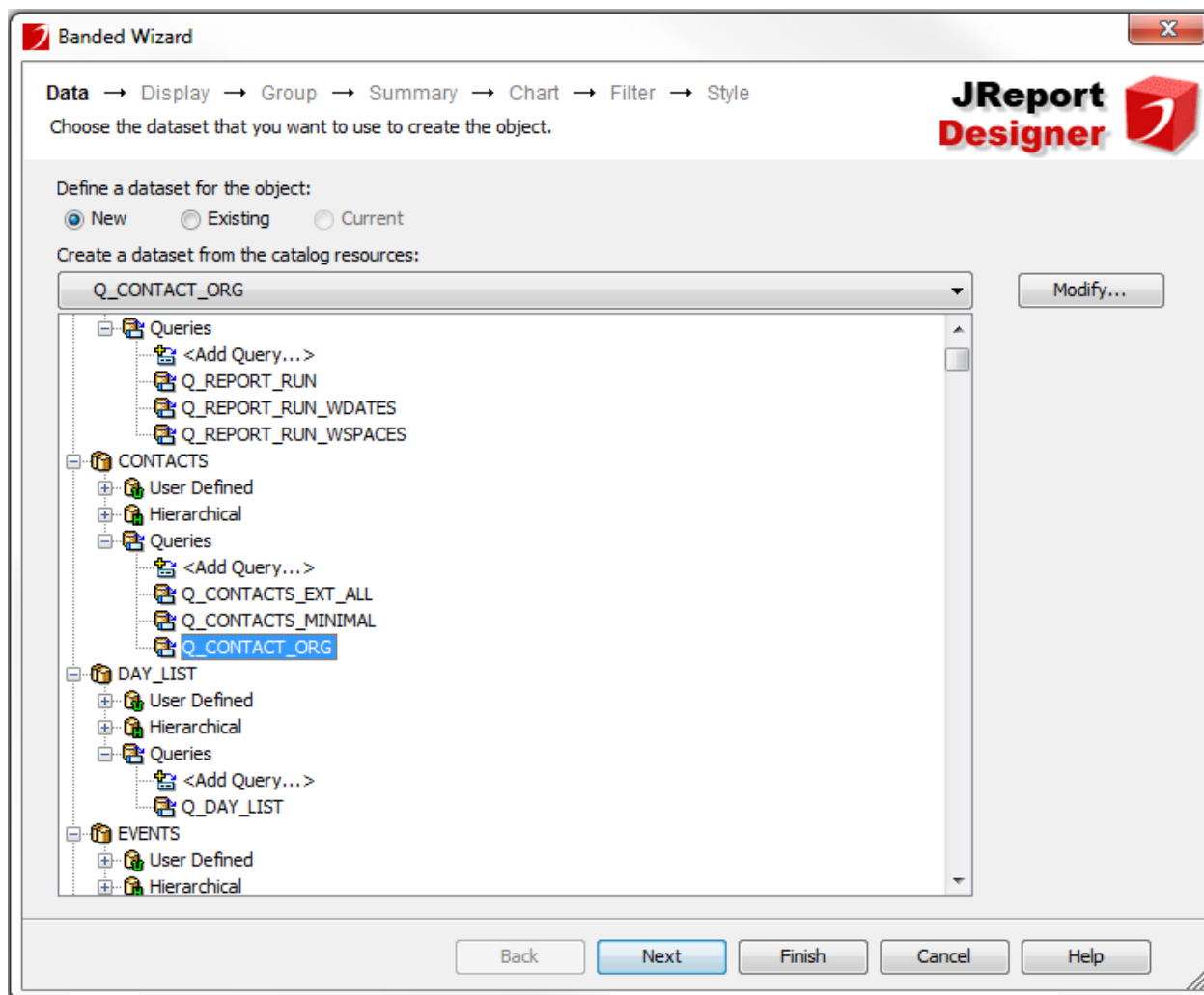
You can exit the Banded Wizard at any time by clicking the Finish button at the bottom of the Wizard dialog. To re-enter the Banded Wizard from the report design editor, right-click over the Banded object and select Banded Wizard... from the pop-up menu.

- *Home > Dialog Reference > Banded wizard*

Selecting a Query

Data >

Banded objects are data containers that require a data source. From the Banded Wizard dialog, select the Query that will serve as the data source for this Banded object or Table object.



Remember that queries belong to a parent Data Source. While a Data Source represents all the data elements available in a given WebService, a Query can present a subset of those data elements.

Multiple data container objects (Banded/Table) can be used in one report. Each data container object will be assigned one Query. A Query can be assigned to a report as New, Existing, or Current. Each Query added as New or Existing requires a unique request of WebServices data. A Query added as Current utilizes the current data set and does not require a new request of WebServices data.





Query Choice Affects Report Performance

Your choice in adding a Query as New, Existing, or Current can have a significant impact on the speed of your report. Steps should be taken to prevent multiple requests of the same WebServices data in order to optimize report performance.

Selecting fields to display

Display >

Use the arrow buttons   to move data fields back and forth from the Resources list to the Display Fields List.

Click the Sort Fields By... button to add and remove the fields you want to sort by, as well as to change sort direction

(ascending and descending). Sort Fields selected here will provide a secondary sort order to any/all Groups.

Grouping report object data

Group >


Groups help organize report data into collections having similar attributes. When a Group is selected, a Group Header panel and Group Footer panel are added to the Banded report object. By default, the Group Header panel will print once at the beginning of each data group, and the Group Footer panel will print once at the end of each data group.

Group sort order

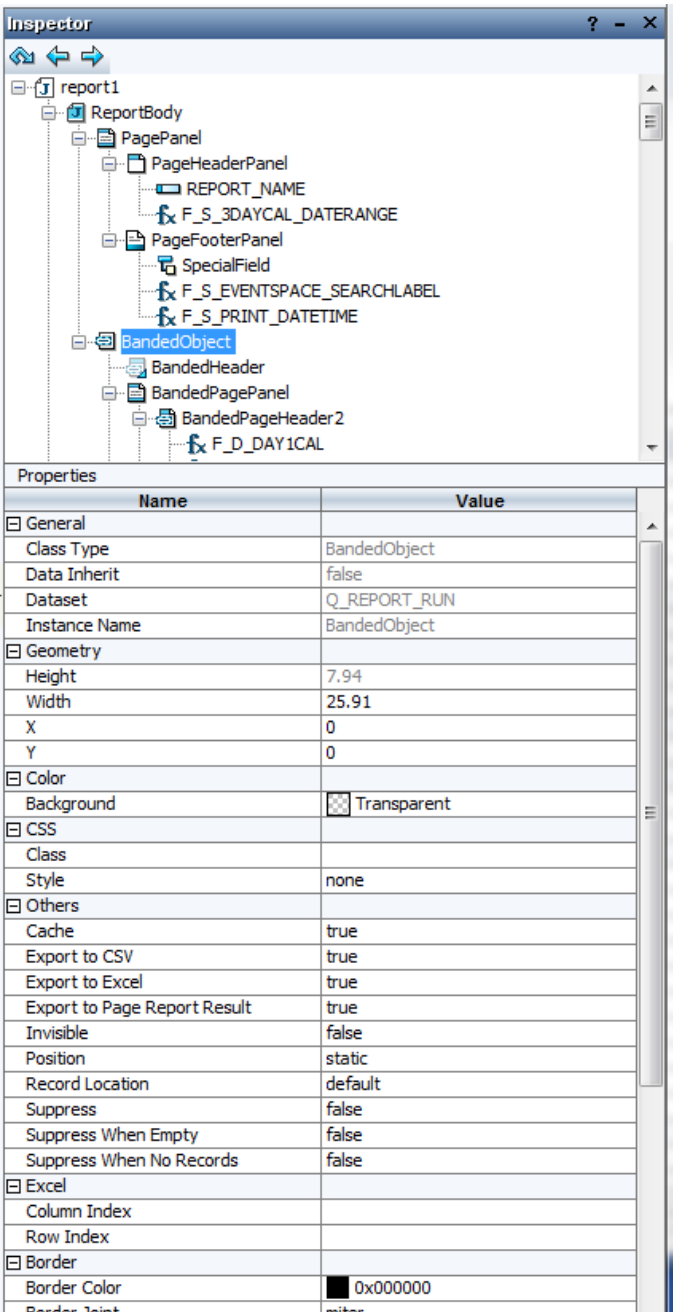
Identifying Report Sections

Report Inspector

The Designer provides a Report Inspector, which includes a Report structure panel and a Properties panel.

 For information on the report inspector that may not be covered in this topic, please see [The JReport Designer User's Guide](#).

- *Home > Report Development Environment > Inspector panel*



Report structure panel (top section)

The report structure panel can assist in identifying the different report sections. This panel displays the report content as a hierarchy or tree structure and tracks your activity as you work within the report design editor. Items selected within the structure panel are also selected in the report design editor.

Properties panel

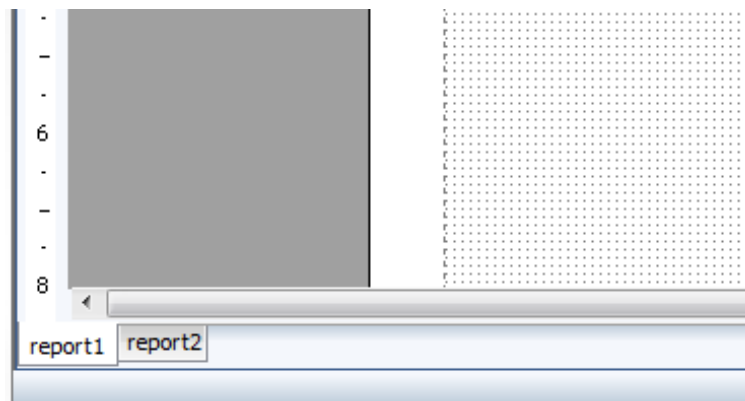
This panel lists all the properties of the object selected in the report structure panel. Each type of object has its own set of properties. Depending on the object type, the properties that a certain object holds may differ greatly from those of another. The Properties sheet has two columns: Name and Value. You can change how an object appears and behaves by changing its value in the Value column.

- *Home > Object Property Reference > Properties in the Report Inspector*

Page Report Tabs

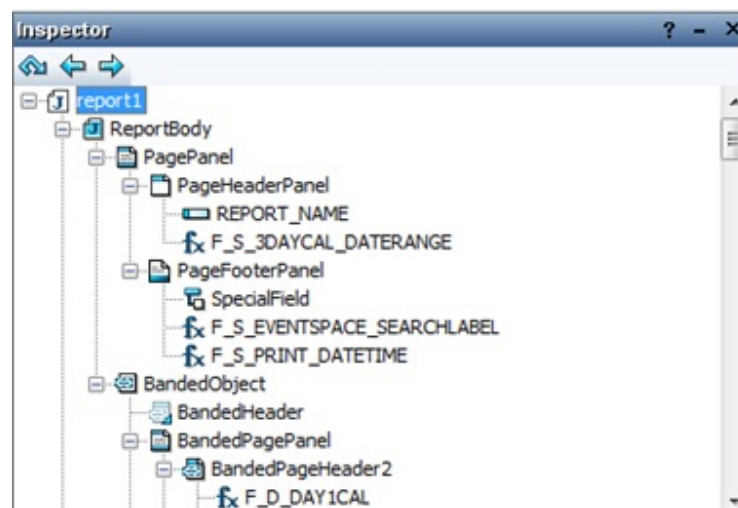
A single-page report file can have multiple report tabs. Report tabs are found in the bottom left corner of the report design editor.

- *Home > Page Reports > Managing report tabs in a page report*



Report Container

A single report container exists for each tab in a Page Report. A report container serves as the base for all other report objects, panels, and sections. The name of the report container is displayed at the top of the hierarchy within the report structure panel of the Report Inspector.



ReportBody

Each report container has a child ReportBody with properties that allow you to control style and behavior options affecting the entire report.

PagePanel

The PagePanel contains a PageHeader and PageFooter. These panels appear at the top and bottom of each report

page. The PageHeader panel is ideal for a report title that you intend to repeat at the top of each page. The PageFooter is ideal for a page number and print date.

Banded Object

By default, a banded object consists of these panels:

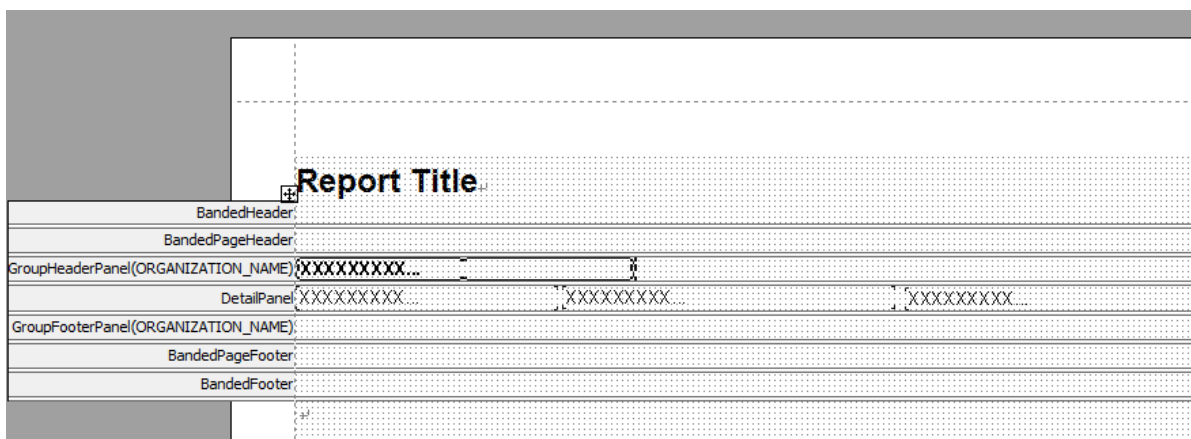


This document describes sections of a banded object. Similar sections exist for a table object.

- **BandedHeader**
 - This panel appears once at the beginning of the banded object.
- **BandedPageHeader**
 - This panel repeats at the top of each page following a page break.
- **Detail Panel**
 - This panel appears once per data record for the query assigned to the banded object.
- **BandedPageFooter**
 - This panel appears at the bottom of each page containing the banded object.
- **BandedFooter**
 - This panel appears once, at the end of the banded object.

For each Group added to the report object, the Designer will add a Group Panel:

- **GroupHeaderPanel**
 - This panel appears at the beginning of each report object group. The Repeat property (true/false) assigned to the GroupHeaderPanel can be set to true which will allow the GroupHeaderPanel to repeat at the top of each page following a page break.
- **GroupFooterPanel**
 - This panel appears at the bottom of each report object group. If the Fill Whole Page property (true/false) is set to true, it forces a page break between report object groups.



Inserting Data Fields Into a Report

Data fields can be inserted into a report by using the wizard or directly from the report design editor by dragging a field from the Resource View into the desired section of the report object.

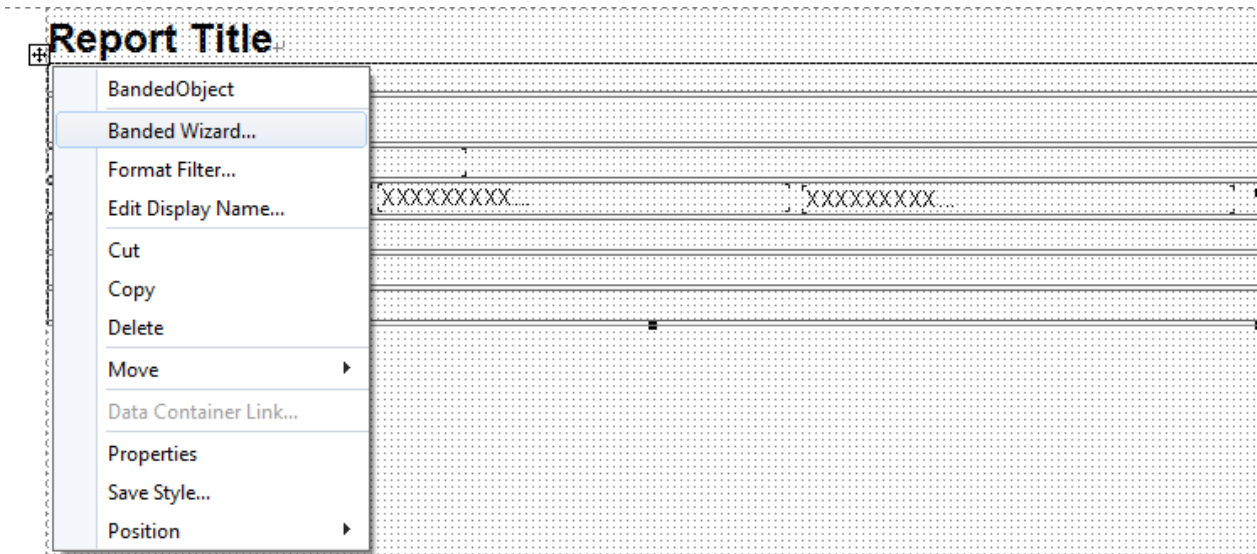


A Data Container Can Only Include Fields From the Assigned Query

A data container, banded object, or table object provides the primary context for placement of data fields. A data container can only include those data fields belonging to the assigned query.


From the Wizard

To re-enter the Banded Wizard, right-click over any unoccupied section of the Banded report object. From the menu, select the Banded Wizard item.




Within the Banded Wizard, click the Display breadcrumb.

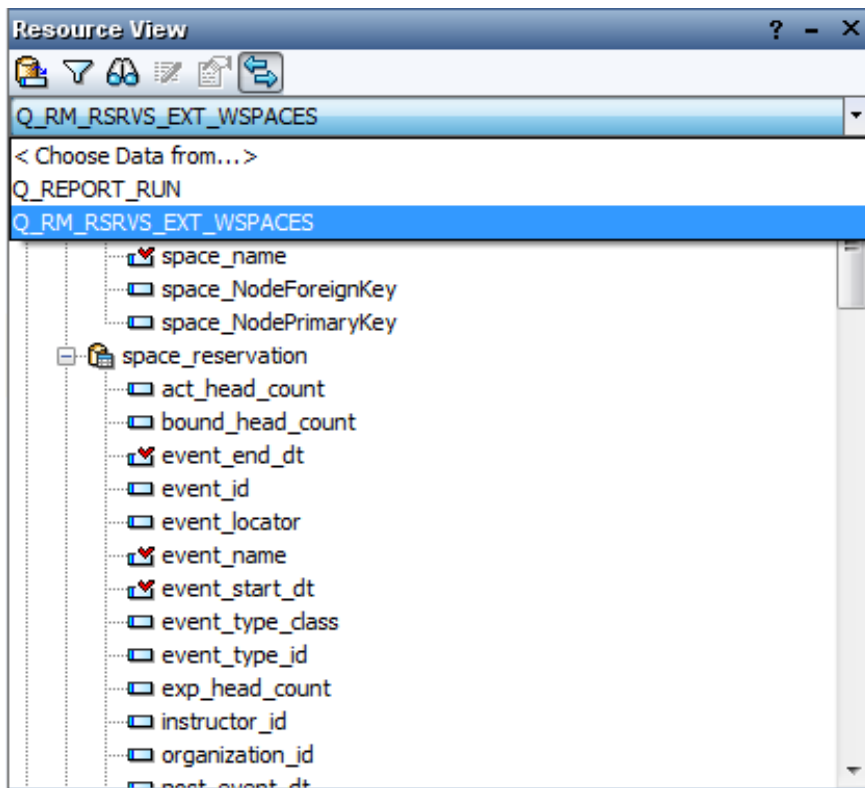
Display >

Use the arrow buttons   to move data fields back and forth from the Resources list to the Display Fields List.

From the Resource View Panel

Click and drag data elements from the Resource View panel into the Banded report object. Data elements include Query fields, Formulas, Summaries, and Parameters. Remember that multiple queries can be used in one report, but only one Query is assigned to each Banded report object.

A drop-down list within the Resource View shows all queries available in the report. On the Resource View toolbar, click the  button to link Resource View with the report design editor. When a Banded object is selected in the report design editor, the assigned Query appears in the Resource View.



Moving, Resizing, and Hiding Report Objects

A report is made up of a collection of objects, data container objects, data field objects, text label objects, and so on. There are several options available that allow you to resize and/or reposition an object:

- **Resize:**
 - Click to select an object from the report design editor or the Inspector panel. A border will highlight the selected object and allow you to stretch it vertically and horizontally. Each object has a specific height and width, which is displayed in the Properties view panel. Retype values for height and width to change the dimensions of the selected object.



Changing the Default Unit of Measure

These dimensions represent either inches or centimeters. The default unit of measure can be changed by selecting File > Options from the Designer main menu bar, choosing the Editor category, then adjusting the Display Attributes and Units.

- **Reposition:**
 - Click and drag a field object to reposition it within the same panel or to move it to another panel.

Setting Properties To Assist with Field Position

In addition to the Height and Width properties mentioned above, these other properties may assist with field placement.

- *Home > Object Property Reference*
- **X** (data type: float) / **Y** (data type: float)
 - X specifies the horizontal coordinate of the top left corner of the object, relative to its parent container.
 - Y specifies the vertical coordinate of the top left corner of the object, relative to its parent container.
- **Padding** (data type: float)
 - Bottom, Left, Right, and Top padding can be applied to create space between object content and its border.
- **Auto Fit** (data type: boolean)
 - Specifies whether to adjust the width and height of the object according to the contents.
- **Word Wrap** (data type: boolean)
 - Specifies whether to wrap the text content of the object to multiple lines.

Setting Properties to Control Object Visibility

There are several properties, each expecting a boolean value, that can be used to control the visibility of objects and sections within a report. This includes Banded objects, panels within a Banded object, and field objects. Properties that control object visibility can be viewed in the Report Inspector, Properties panel.



Show/Hide Button

A toggle button is available in the report design editor that allows you to show or hide report objects. This toggle controls the view of these items in the design tab to assist you with object placement.

- **Invisible**
 - All formulas and calculation will still be performed if the property is set to true.
- **Suppress**
 - All formulas and calculations will be skipped if the property is set to true.
- **Suppress When Empty**
 - Specifies whether to display the object in the report results when no record is returned to it.
- **Suppress When No Records**
 - Specifies whether to display the object in the report results when no record is returned to its parent data container.

Formatting and Styling Report Objects

A variety of properties are available that allow you to affect the style and format of individual report objects. The Properties panel view changes to display the properties appropriate to the selected report object. The property presents format options appropriate to the data type of a selected field object.

- *Home > Object Property Reference*

Applying Properties Conditionally Via Formula

Properties that control format and style can often be applied conditionally. A formula can be created to specify a set of conditions that return a response to be applied to the property.

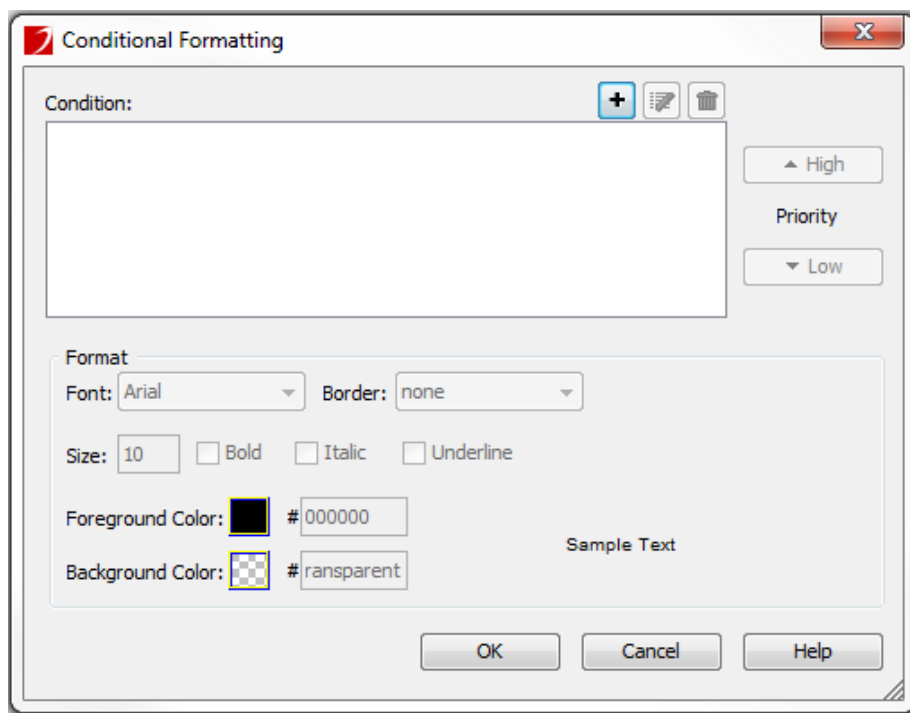
Start by creating the formula that returns the conditional response to be applied to the property value. Make sure the result of the formula matches the expected data type of the property.

In the report design editor, left-click the desired report object. In the Properties panel view, choose the property to be updated. Next to the property value, click the formula button to display a drop-down list where you can select the formula you created.

Conditional Formatting Dialog

Conditional formatting for selected format options can also be applied using the Conditional Formatting dialog. Right-click over any data field object, formula, summary, or parameter as it appears in the report. From the menu, choose the Conditional Formatting option.

- *Home > Dialog Reference > Conditional Formatting dialog*



Filtering Report Data

Within the Designer, data filters can be applied in several different areas. Choosing the appropriate option depends on the intended scope of the data filter.

Dataset Filter

A dataset is the set of data built from the result of a query. A dataset contains data fields, formulas, summaries, and parameters. A dataset filter is applied to the entire report's use of the query data.

 For information on dataset filters that may not be covered in this topic, please see [The JReport Designer User's Guide](#).

- *Home > Dialog Reference > Dataset Filter dialog*

Click the Dataset Filter button from the Resource View to access the Dataset Filter dialog.

Or

From the report design editor main menu bar:

- Select Report > Dataset Management.
- Highlight the query in which to apply a filter.
- Click the Filter tab.

- *Home > Datasets > Managing datasets*

Format Filter

A format filter is applied to a specific data container, typically a Banded or Table object. To access the Format Filter dialog, right-click over an unoccupied section of the Banded or Table object, then select Format Filter from the menu.

- *Home > Queries > Editing a query > Filtering the fields of a query > Filtering with the filter format*

Query Filter

A filter can be applied when creating or editing a query using the Query Editor. Access the Query Editor from the Catalog Browser. Right-click over a query name and select Edit Query. In the Query Editor, notice that all the selected query fields appear in columns at the bottom of the window. A query filter applies to all uses of the query by all reports.



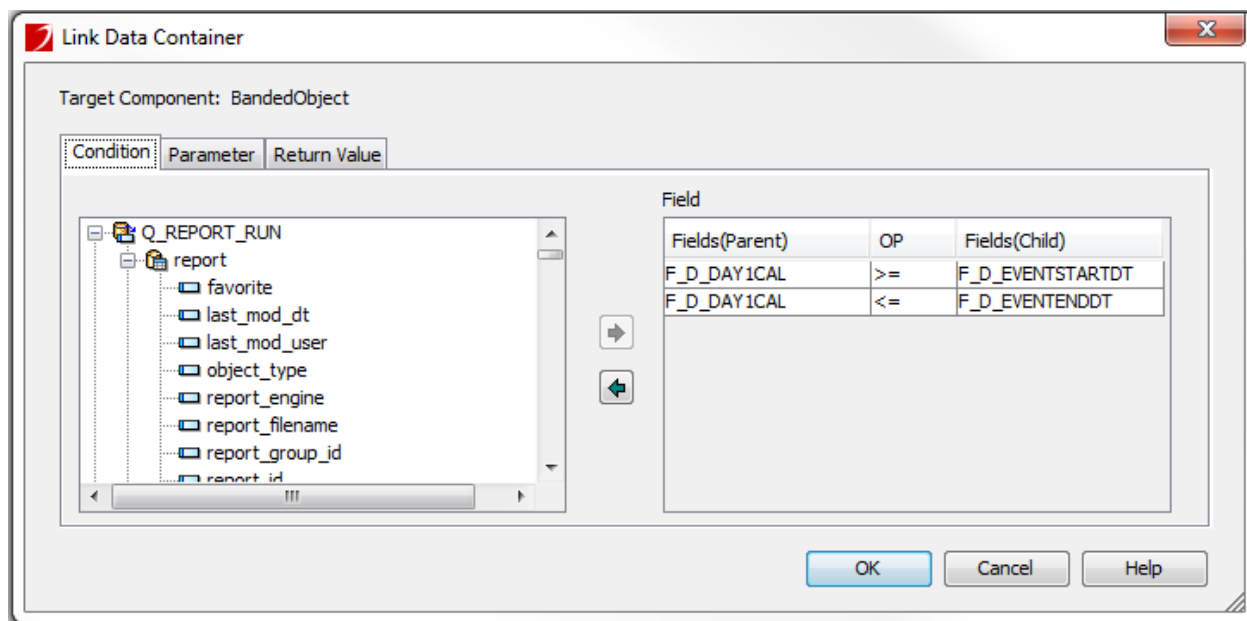
For information on query filters that may not be covered in this topic, please see [The JReport Designer User's Guide](#).

- *Home > Queries > Editing a query > Filtering the fields of a query*

Link Data Container – Condition

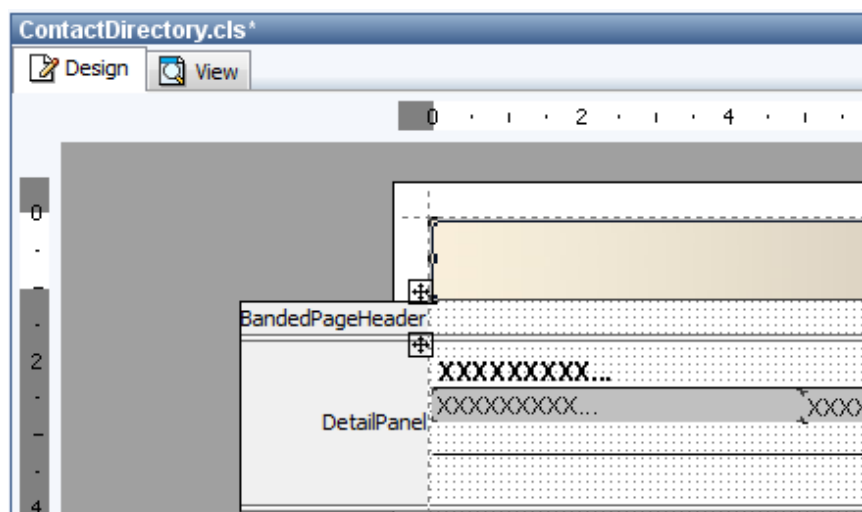
Reports are often created with multiple data containers. These are often embedded, with one or more child Banded/Table objects linked to a parent Banded object. When this link is established, Conditions can be used to dynamically filter out data of the child data container. More detail about linking data containers is provided in [JReport and WebServices](#).

- *Home > Dialog Reference > Link Data Container dialog*
- *Home > Components > Tables > Setting up data container link*



Viewing Report Output

The Designer allows you to view report results as you go. Much of your report development takes place in the Design tab of the report design editor. To view report results, click the View tab appearing near the top of the report design editor.



When viewing report results, the Designer fills the report template with actual data. Up until this point, you have been working with the relational representations of the data produced by the schema used to create a Data Source. You will likely be asked to provide a Series25 WebServices URL that the report will use to make a data request.


The WebServices URL will look something like this:

http://some_server_name/r25d/servlet/wrd/run/contacts.xml?session_id=B1FDB903EBE5A3ECA5E7183B47EE5914

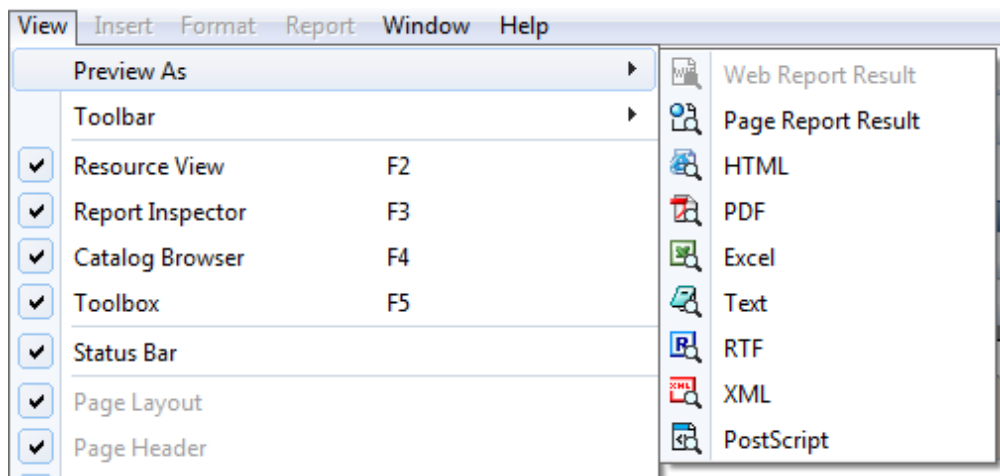
Additional Preview Formats

By default, report results are brought to a JReport preview window. You can preview report results in other formats by

selecting View > Preview As from the main menu bar. The report result will open in the appropriate program based on the settings of your computer's operating system.

 For information on previewing reports that may not be covered in this topic, please see [The JReport Designer User's Guide](#).

- *Home > Page Reports > Previewing a page report*




Page Report Result interactive viewer

Selecting Preview As > Page Report Result opens the report within an interactive browser window. This viewer capability approximates similar capabilities supported by the JReport Server. Viewing reports within this interactive viewer will not be supported in the initial release of JReport. Future editions of this documentation will discuss this interactive environment in more detail.

Saving a Report

Default Page Report Format

By default, a page report is saved as a binary file with a .cls extension, a format that provides optimal performance. To save a report, from the main menu bar of the report design editor, select File > Save.

 For information on previewing reports that may not be covered in this topic, please see [The JReport Designer User's Guide](#).

- *Home > Page Reports > Saving a page report*

Only page reports with this file extension should be published to the JReport Server.

Page reports must be saved to the same directory as the Catalog file (25LIVE_WS.cat).

Other Report File Formats

- **Report Template (.rpt)**
 - This is a file saved in text format, which can be edited in any text editor. This format can be useful for troubleshooting as the report logic is exposed in plain text.

- **Self Contained Page Report (.clx)**
 - This is a binary file format which contains not only the report's layout but also the catalog with its own resources.
- **Page Report XML Format (.cls.xml)**
 - This file format follows the XML standard and can be edited with both the JReport Designer and external XML editors. This format can be useful for troubleshooting as the report logic is exposed in plain text.

Saving a Page Report to a Different Catalog

A single Catalog file (25LIVE_WS.cat) sits at the center of each report development environment. You can have multiple report development environments each with its own 25LIVE_WS.cat file. This is common when there are multiple report developers, each working with their own copy of the 25LIVE_WS.cat file. In this situation, there might typically be a central report development environment with a master 25LIVE_WS.cat file. When a report is saved to a different catalog, the catalog resources used by this report are merged with the new catalog.



For information on previewing reports that may not be covered in this topic, please see [The JReport Designer User's Guide](#).



Keep 25LIVE_WS.cat Name

This only works when the multiple catalog file names match, which is why CollegeNET recommends the catalog name remain as 25LIVE_WS.cat. Keeping this as the catalog file name allows you to share report files.

To save a page report to a different catalog, from the report design editor main menu bar:

- Click File > Save To.
- Browse to the location of the target catalog, then click Save.
- In the Warning message box, click Yes.
- This saves your report into a target catalog file, which means:
 - The report and its relevant files will be copied to the directory where the target catalog file is located.
 - The resources (query, formulas, parameters, and so on) that are referenced by this report in the current catalog will be merged to the target catalog.

- *Home > Catalogs > Merging catalogs*

The Interactive Report Designer and WebServices

Data Sources and Series25 WebServices

A Data Source represents a connection to the Series25 database via the Series25 WebServices API. CollegeNET has created a catalog file, 25LIVE_WS.cat, containing Data Source definitions for over 65 of the most common WebServices used for reporting purposes.

Each Data Source is created by supplying a schema file (.xsd) which describes the structure of a Series25 WebServices

XML data response. The Designer translates this XML structure into a relational model of tables and fields. Table relationships are interpreted based on the XML hierarchical structure.

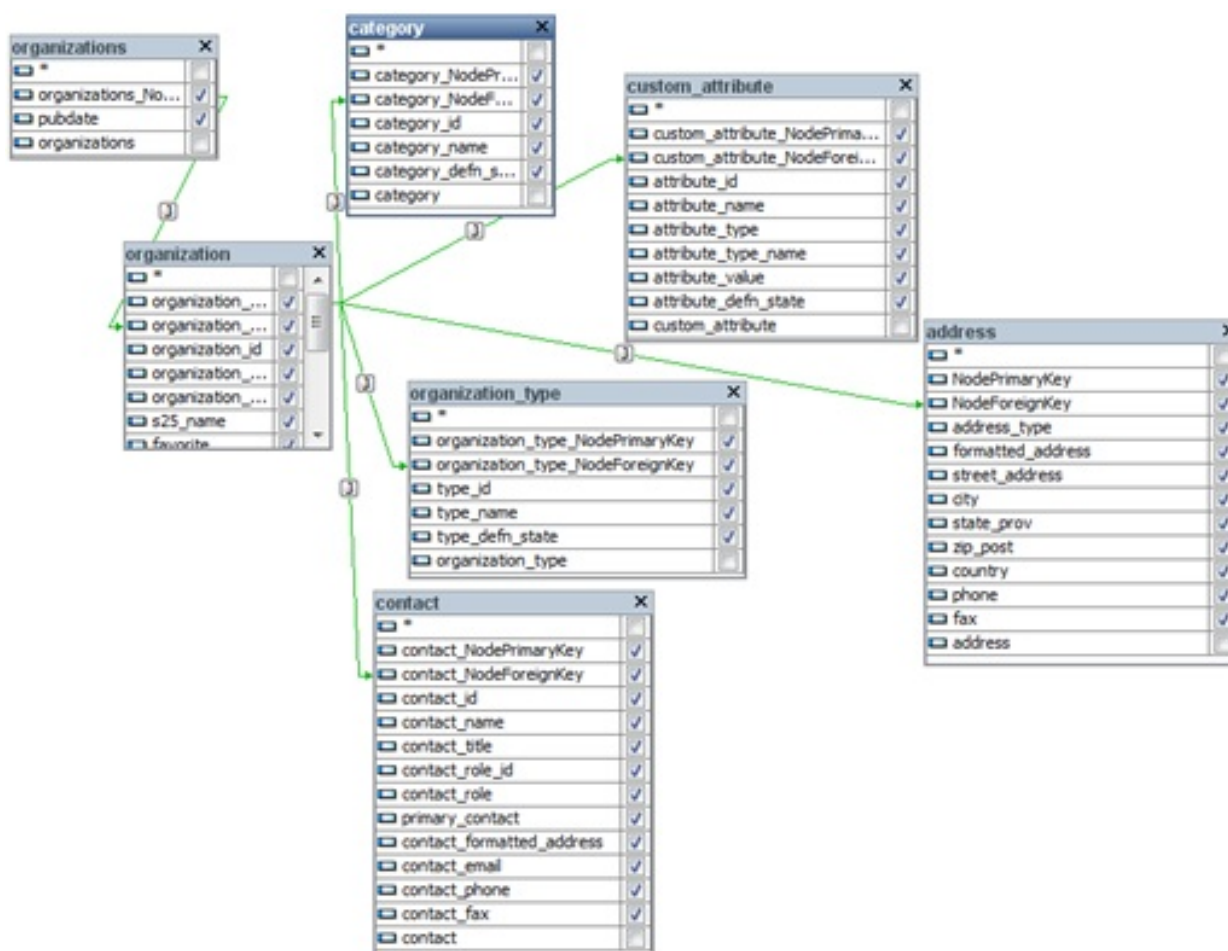
Queries are created for each Data Source and can represent more targeted subsets of the data available within the parent Data Source. Banded/Table objects are assigned a Query, which provides data for the report.

Example: Data Source – ORGANIZATIONS

The ORGANIZATIONS Data Source represents a connection to the ORGANIZATION.XML and ORGANIZATIONS.XML Series25 WebService. This WebService can present varying amounts of detail for each organization depending on the value of the scope=argument. A corresponding Query can be created that anticipates the appropriate amount of detail returned by the WebService request.

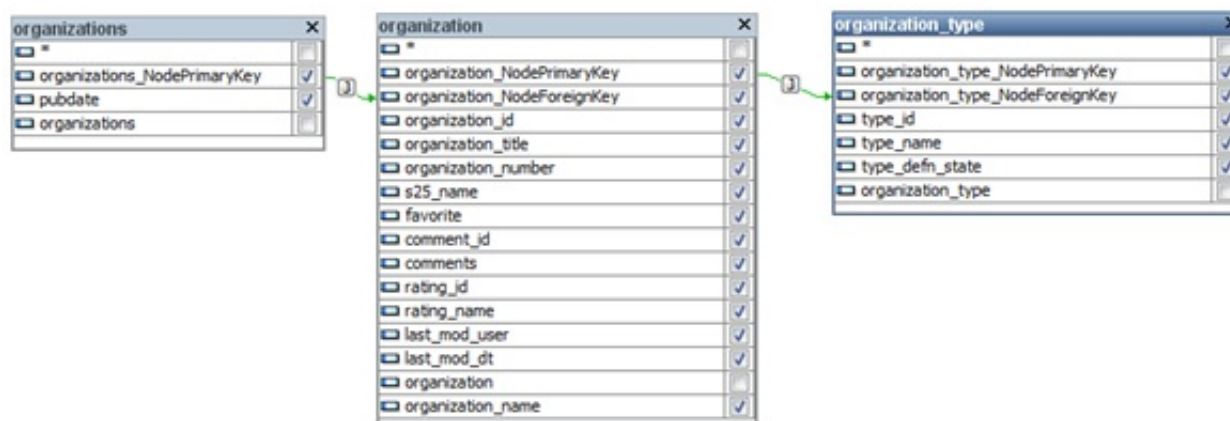
Query Name: Q_ORGANIZATIONS_EXT_ALL

This Query has been created in the 25LIVE_WS.cat file and is assigned to the ORGANIZATIONS Data Source. This Query anticipates a scope=extended request of the ORGANIZATION(S) WebService. Here is a relational view of this Query from the Query Editor:



Query Name: Q_ORGANIZATIONS_MINIMAL

This Query, assigned to the same ORGANIZATIONS Data Source, anticipates a scope=minimal request of the ORGANIZATION(S) WebService. Here is a relational view of this Query from the Query Editor:



In each example, only the field elements selected in the Query are made available to the reports Banded/Table object.

The Report Run Process

Each time a report is run from 25Live, a record is established in the database that records the report parameters supplied by the user. A 25Live user would recognize typical report parameters as Start Date, End Date, Space Search and Event Search. In the database, these values are stored in fields, date_parm1, date_parm2, rm_query_id and ev_query_id. A JReport must be allowed to retrieve this report run record using WebServices. Once retrieved, the JReport can use these parameter values to construct URLs used to make WebServices data requests.

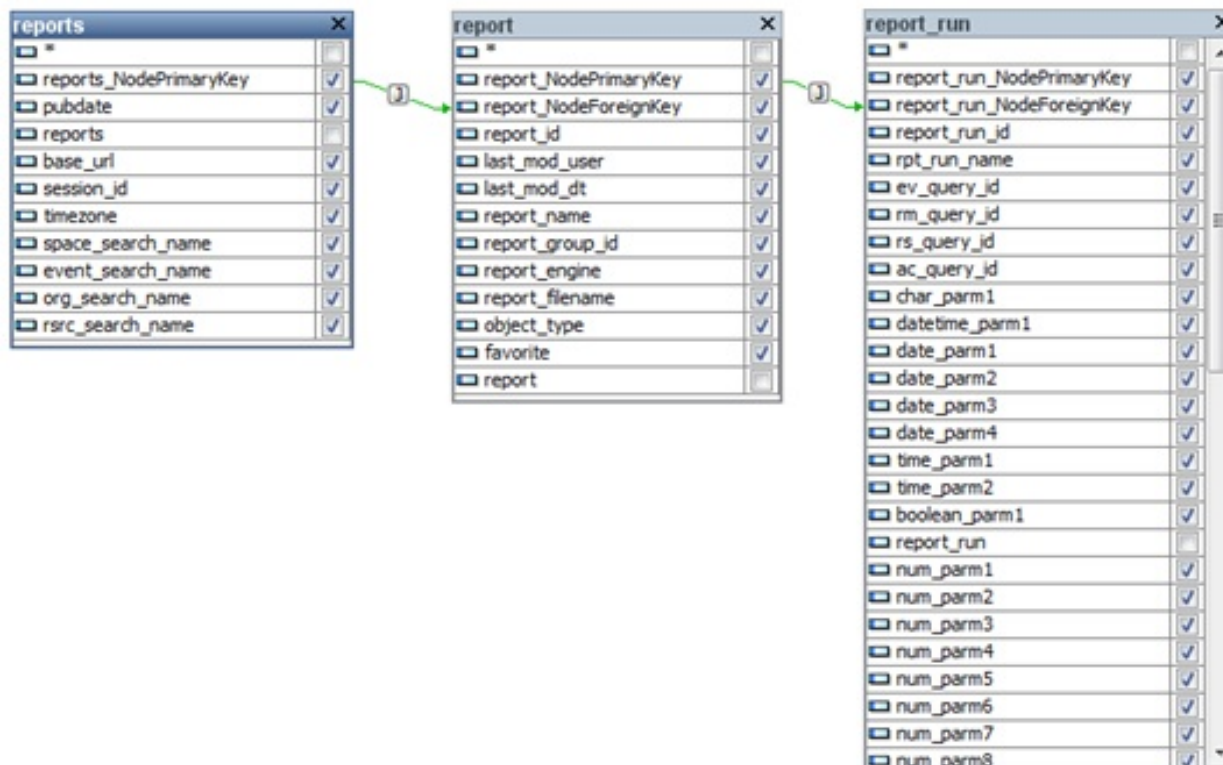
REPORT_RUN Data Source

The REPORT_RUN Data Source has been added to 25LIVE_WS.cat and contains all the necessary data elements for constructing WebServices URLs. A Query derived from the REPORT_RUN data source, assigned to a Banded report object, provides the foundation for every JReport run from 25Live.

The REPORT.XML WebService provides much of the data presented in this Data Source, which includes the report parameters supplied by the user at runtime. Additional data elements, such as base_url and session_id, have been added to facilitate the construction of a WebServices URL.

Query name: Q_REPORT_RUN

This query references three tables: reports, report, and report_run. These tables should be considered the core tables available within the REPORT_RUN data source and are typically included in any query assigned to this data source. This Query will present one row of data for the report_run session.

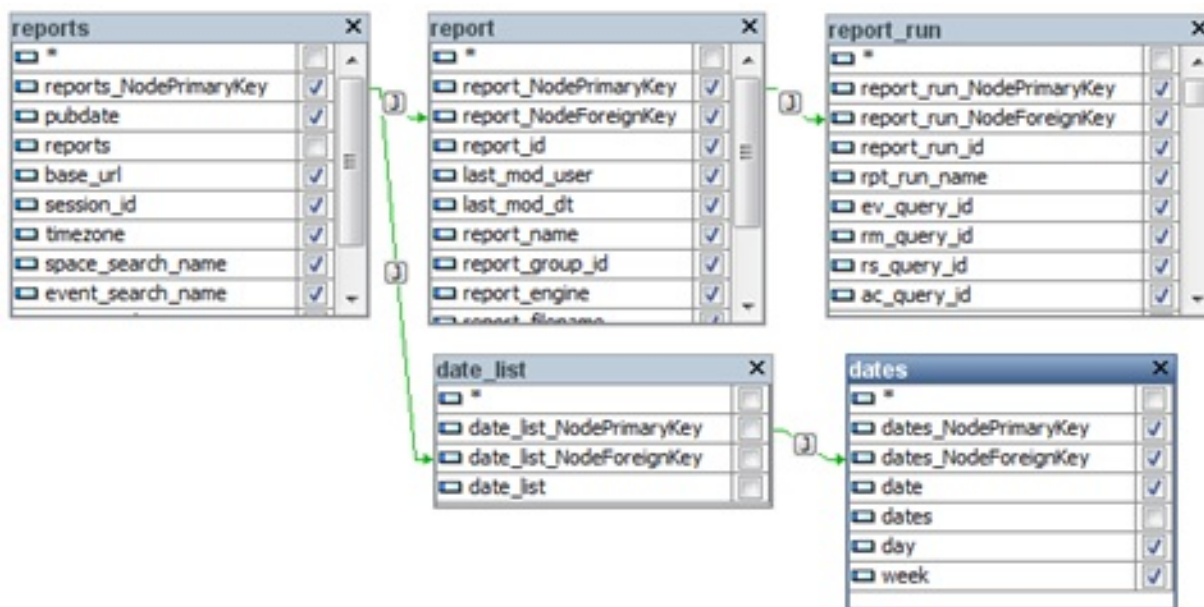


To assist with organizing a JReport, other data elements have been added to the REPORT_RUN Data Source, including:

dates

A dates table is made available that contains one row for each date appropriate to the report parameter date range supplied by the user at runtime.

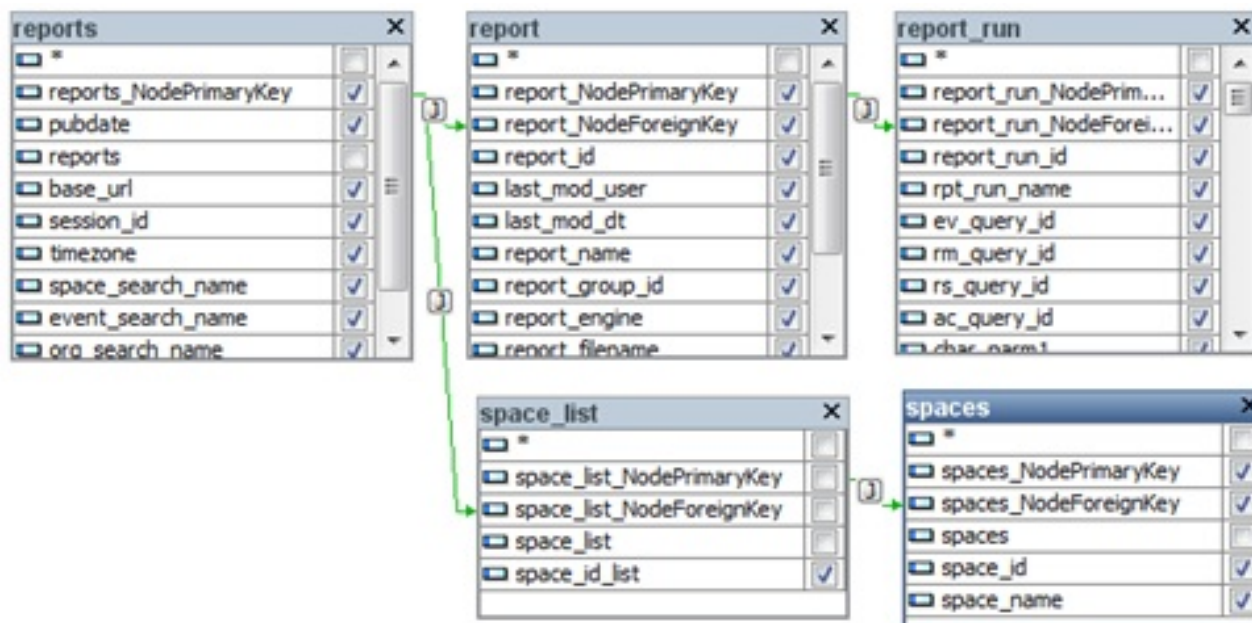
- **Query name: Q_REPORT_RUN_WDATES**
 - This Query is useful for reports that are designed to print per date.



space_list > spaces

A space_list and spaces table is made available when a space search is provided as a report parameter. The space_id_list field, in the space_list table, provides a comma-separated list of all spaces included in the supplied space search. The spaces table contains one row for each space included in the supplied space search.

- **Query name: Q_REPORT_RUN_WSPACES**
 - This Query is useful for reports that are designed to print per space.

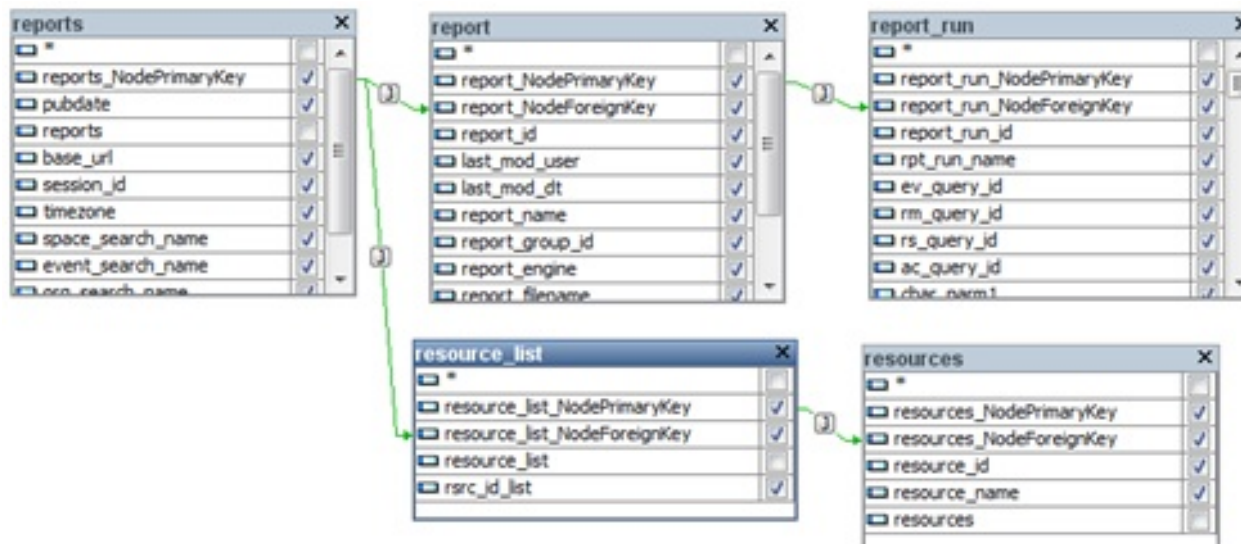


resource_list > resources

A resource_list and resources table is made available when a resource search is provided as a report parameter. The

rsrc_id_list field in the resource_list table provides a comma-separated list of all resources included in the supplied resource search. The resources table contains one row for each resource included in the supplied resource search.

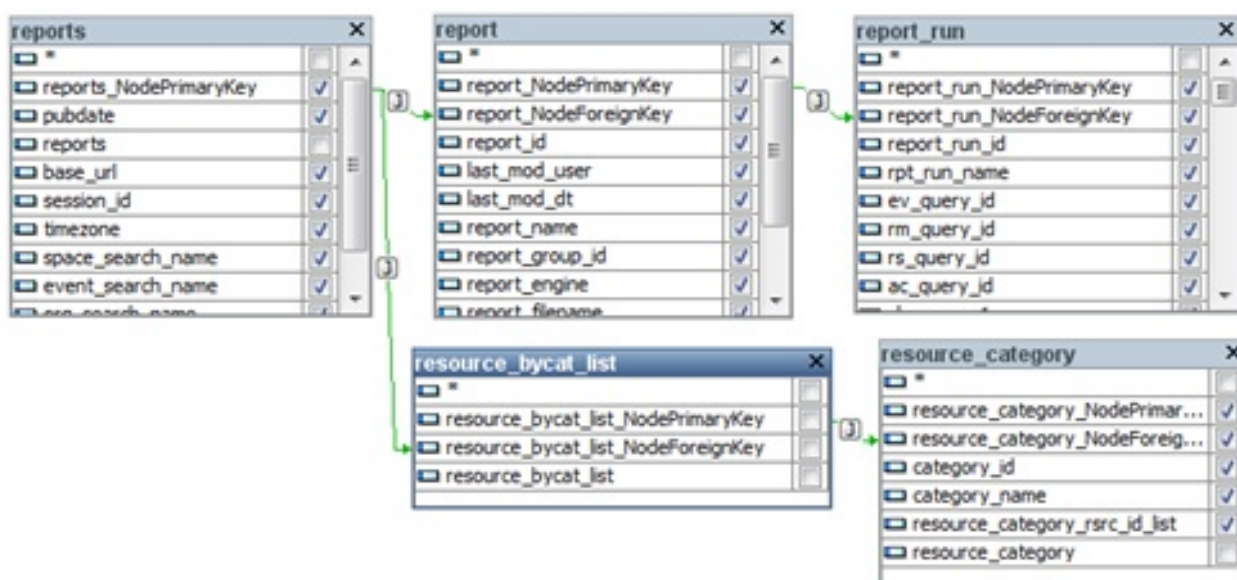
- **Query name: Q_REPORT_RUN_WRESOURCES**
 - This Query is useful for reports that are designed to print per resource.



resource_bycat_list > resource_category

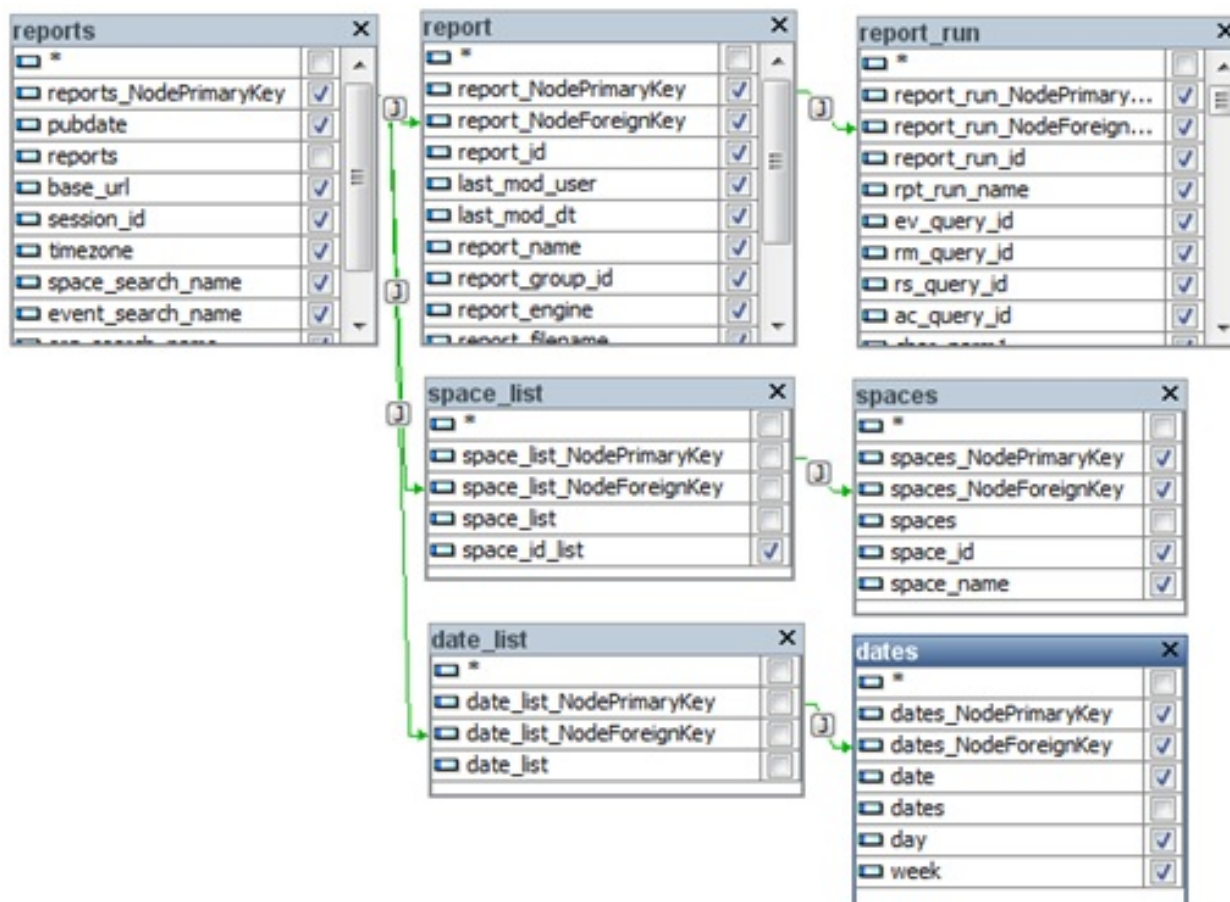
The resource_category table contains one row for each resource category represented by a population of resources. The resource_category_rsrc_id_list field in the resource_category table provides a comma-separated list of all resources assigned to a given category. A resource search supplied as a report parameter will set the population of resources. If a resource search is not supplied, all resources are represented.

- **Query name: Q_REPORT_RUN_WRsrcListByCat**
 - This Query is particularly useful to help identify the appropriate category for a resource.



A Query assigned to the REPORT_RUN Data Source can be created that will include two or more of these additional data elements.

- **Query name: Q_REPORT_RUN_WDATES_SPACES**
 - This Query is especially useful for a report designed to print per date and per space. The Query will return one row for each combination of date and space.



Report Objects – Data Containers

The structure of a page report is provided by Banded and/or Table objects. Each of these report objects is assigned a Query. When a Banded or Table object is assigned a Query, it is referred to as a data container and can contain other embedded report objects, as well as data elements, from its assigned Query.

Typically, a single report has multiple data containers, each assigned a Query belonging to a different Data Source.

For example: Event Listing Report

Report Name							
BandedHeader							
BandedPageHeader							
DetailPanel							
BandedPageFooter							
BandedFooter							

Name	Type	Status	Reference	Organization	Title	Start Date	End Date
XXXXXXXXXX	XXXXXXXXXX	XXXXXX	XXXXXXXXXX	XXXXXXXXXX	XXXXXXXXXX	MMM dd yyyy	MMM dd yyyy

The red border above highlights the parent Banded object for the Event Listing Report. This data container is assigned a query called Q_REPORT_RUN, which is derived from the Data Source REPORT_RUN. The Banded object has panels; BandedHeader, BandedPageHeader, DetailPanel, BandedPageFooter, and BandedFooter that help organize report content. The DetailPanel contains an embedded Banded object that acts as a sub-report that is linked to the parent.

The embedded Banded object is assigned a query called Q_EVENTS_MINIMAL, which is derived from the data source EVENTS.

Report Name							
BandedHeader							
BandedPageHeader							
DetailPanel							
BandedPageFooter							
BandedFooter							

Name	Type	Status	Reference	Organization	Title	Start Date	End Date
XXXXXXXXXX	XXXXXXXXXX	XXXXXX	XXXXXXXXXX	XXXXXXXXXX	XXXXXXXXXX	MMM dd yyyy	MMM dd yyyy

The red border above highlights the embedded Banded object that has been placed in the DetailPanel of the parent. The labels for each column are located in the BandedPageHeader of the embedded Banded object. The DetailPanel of the embedded Banded object contains the primary report details and prints once for each row in the dataset supplied by the query Q_EVENTS_MINIMAL.

Required – URL Source Parameter

Every Data Source requires a source parameter value formatted as a URL. This URL is used to request WebServices data at runtime. The REPORT_RUN data source has a required parameter, P_REPORT_RUN, which is passed automatically when a report is run from 25Live. When a report is run from the JReport Designer, this URL parameter value must be supplied manually.

URL Formulas

URL parameter values for data sources assigned to embedded report objects are supplied dynamically by linking the parent report to the embedded report.

URLs are constructed using a JReport formula, which generates a string value by combining data elements with text.

For example: JReport formula name, F_URL_EVENTS_MINIMAL

```

1  → @base_url +
2  'events.xml?query_id='+
3  @ev_query_id +
4  '&session_id=' +
5  @session_id

```

- **Line 1: @base_url**
 - @base_url is a data element available within the REPORT_RUN data source. The base_url is unique to each institution and is the base to all WebServices URLs.
- **Line 2: 'events.xml?query_id='**
 - This is plain text that identifies the name of the WebService, events.xml, along with the filter argument query_id=.
- **Line 3: @ev_query_id**
 - @ev_query_id is a data element available within the REPORT_RUN data source. The query_id of an event search, provided as a report parameter at runtime, is mapped to this data element.
- **Line 4: 'session_id='**
 - This is plain text that identifies the session_id argument added to the WebService request.
- **Line 5: @session_id**
 - @session_id is a data element available within the REPORT_RUN data source. The session_id argument must be appended to every WebServices URL to provide authentication to the data request.

At runtime, the data elements @base_url, @ev_query_id, and @session_id are populated with values.

For example:

Line 1: http://25live.collegenet.com/25live/data/school/run/

Line 2: events.xml?query_id=

Line 3: 1234

Line 4: &session_id=

Line 5: 58BE0815149559B41287050CA8F854AA

Combined values in the URL:

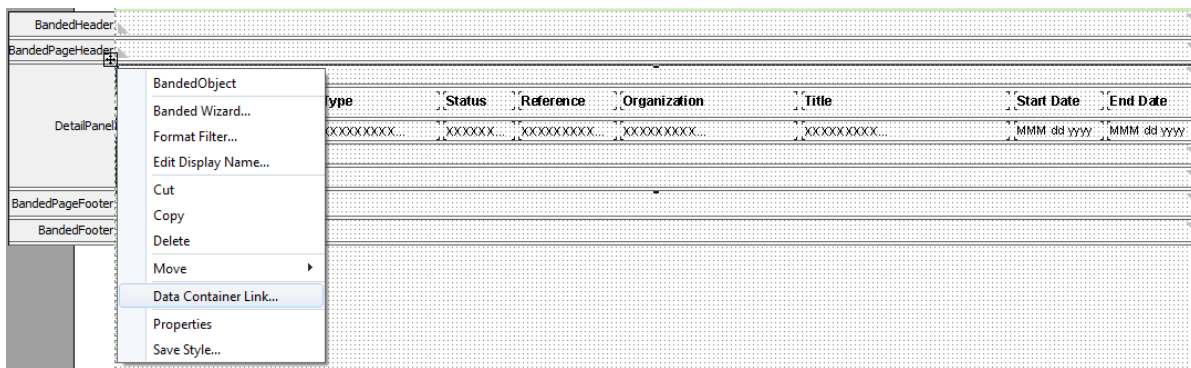
http://25live.collegenet.com/25live/data/school/run/events.xml?
query_id=1234&session_id=58BE0815149559B41287050CA8F854AA

Linking Embedded Reports

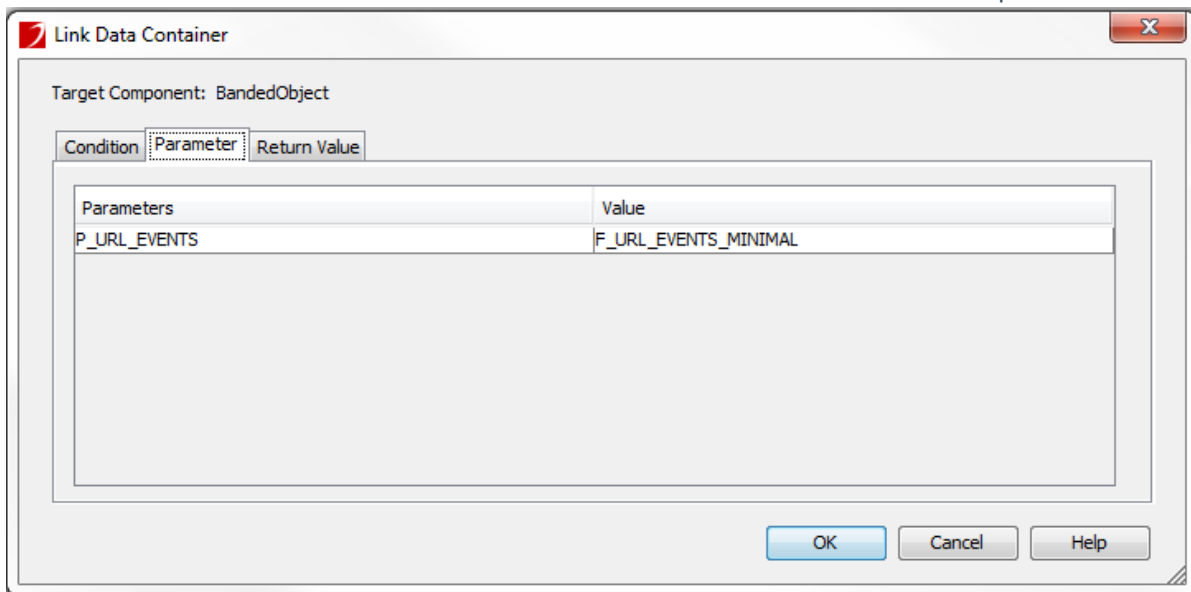
In the Event Listing example, the embedded report object is assigned to the query Q_EVENTS_MINIMAL. This query is derived from the EVENTS data source, which requires a URL source parameter called P_URL_EVENTS. The parent data container is linked to the embedded data container by passing the formula, F_URL_EventsExt_Customers, to the parameter P_URL_EVENTS_EXT.

Follow these steps to link an embedded report object:

1. Left-click the embedded report object to be linked. A small box should appear in the top left corner. Right-click the small box to get a menu. Select Data Container Link...From the Link Data Container dialog, choose the Parameter tab. The URL source parameter should be listed in the Parameters column. For the Value column, select the URL formula that was created in the parent data container.



- From the Link Data Container dialog, choose the Parameter tab. The URL source parameter should be listed in the Parameters column. For the Value column, select the URL formula that was created in the parent data container.



JReport Run

JReport Run is a new standard report transform that has been added to 25Live. This report was created to assist with running reports from the Designer by exposing the URL source parameter value required by the REPORT_RUN Data Source. This URL source parameter is used by the Designer to retrieve a representative parameter set adequate for running any report.

While developing and testing custom reports using the Designer, you will be prompted for this URL source parameter value.

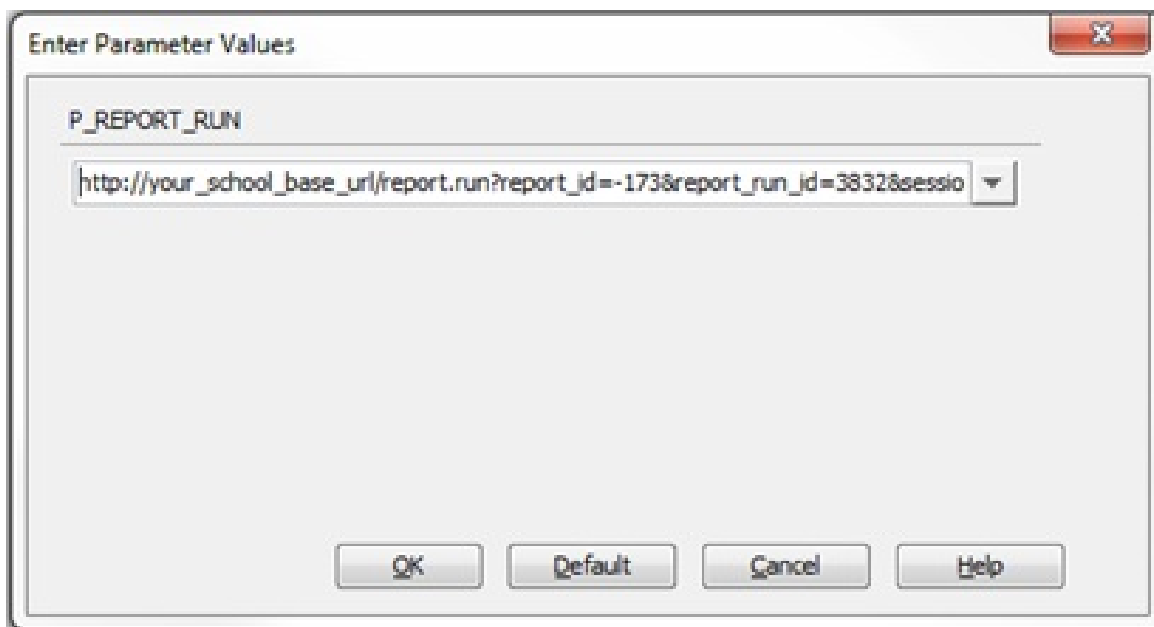
This value is returned as part of the XML data in an element called report_run_url. Copy and paste this URL value, when prompted, which allows the Designer to request this parameter set data from WebServices.

For example:

Copy from: JReport Run result XML document

```
<r25:report_run>
<r25:report_run_id>3832</r25:report_run_id>
<r25:rpt_run_name>JReport Run</r25:rpt_run_name>
<r25:ev_query_id>25639</r25:ev_query_id>
<r25:rm_query_id>25315</r25:rm_query_id>
<r25:rs_query_id>25156</r25:rs_query_id>
<r25:ac_query_id>25499</r25:ac_query_id>
<r25:num_parm1>22153</r25:num_parm1>
<r25:char_parm1>motuwethfr</r25:char_parm1>
<r25:datetime_parm1>2010-10-01T12:00:00-07:00</r25:datetime_parm1>
<r25:date_parm1>2010-10-11</r25:date_parm1>
<r25:date_parm2>2010-10-24</r25:date_parm2>
<r25:date_parm3>2009-05-01</r25:date_parm3>
<r25:date_parm4>2009-05-13</r25:date_parm4>
<r25:time_parm1>08:00:00</r25:time_parm1>
<r25:time_parm2>17:00:00</r25:time_parm2>
<r25:boolean_parm1>F</r25:boolean_parm1>
<r25:true_value>T</r25:true_value>
<r25:false_value>F</r25:false_value>
<r25:report_run_url>
http://your_school_base_url/report.run?report_id=-173&report_run_id=3832&session_id=220F784179705B2C2845649CB6
DDAEE1
</r25:report_run>
</r25:report_run>
```

Paste to: JReport Designer, Enter Parameter Values dialog



Target a Data Sample

When a JReport Run report is selected from 25Live, the user is given the opportunity to identify a representative sample of report parameters that can be applied to running just about any report. At runtime, this parameter set is saved in the database with a corresponding ID number stored in the report_run_id field of the report_run table. The report result can be viewed in a browser, text editor, or XML editor.

Follow these instructions to run JReport Run from 25Live:

1. Sign in to 25Live and open the **More** menu. Under the Go to Tool section, tap the **25Live Reports** link. Locate the report titled "JReport Run" under **Other Reports**.
2. Before running the report, identify the report parameters to be used as you develop a JReport using the JReport Designer. When selecting these parameters, it is recommended that you identify sample/representative data, selecting, for example, a date range of 1 week instead of 1 month, an event search that includes 100 events instead of 1000, and so on. Identifying more restrictive parameters will speed your effort in developing a report.

Choose a report from the list, then complete the parameters to run the report.

JReport Run

Select Report Parameters

Start Date:	<input type="text" value="Sun Jun/1/2014"/>	
End Date:	<input type="text" value="Sun Jun/15/2014"/>	
Comparison Start Date:	<input type="text" value="Tue Jul/1/2014"/>	
Comparison End Date:	<input type="text" value="Mon Jul/14/2014"/>	
Event Search:	<input type="text" value="Special Events 2014"/>	
Location Search:	<input type="text" value="ARTS Rooms"/>	
Resource Search:	<input type="text" value="Personnel"/>	
Organization Search:	<input type="text" value="All Orgs (130)"/>	
Start Time:	<input type="text" value="08:00 am"/>	
End Time:	<input type="text" value="05:00 pm"/>	
Date/Time Parm 1:	<input type="text" value="Mon Jun/30/2014"/>	<input type="text" value="08:00 am"/>
Boolean Parm 1:	<input type="radio"/> Yes <input checked="" type="radio"/> No	
Numeric Parm 1:	<input type="text" value="21573"/>	
Day(s) of Week:	<div style="display: flex; flex-wrap: wrap; justify-content: space-around;"> <div style="text-align: center;"><input checked="" type="checkbox"/> Sunday</div> <div style="text-align: center;"><input checked="" type="checkbox"/> Monday</div> <div style="text-align: center;"><input checked="" type="checkbox"/> Tuesday</div> <div style="text-align: center;"><input checked="" type="checkbox"/> Wednesday</div> <div style="text-align: center;"><input checked="" type="checkbox"/> Thursday</div> <div style="text-align: center;"><input checked="" type="checkbox"/> Friday</div> <div style="text-align: center;"><input checked="" type="checkbox"/> Saturday</div> </div>	

3. Run the report. The report is designed to output an XML file to be viewed in a browser. Select a browser to open the report result XML file.

4. Find the element `<r25:report_run_url>` contained within the `<r25:report_run>` node of the report result XML output. It will look something like this:

```
<r25:report_run_url>
http://your_school_base_url/report.run?report_id=-173&report_run_id=3832&session_id=220F784179705B2C284564
9CB6DDAEE1
</r25:report_run_url>
```

5. Copy and paste this URL into the Designer parameter dialog when you are prompted to supply a value for `P_REPORT_RUN`. Copy only the URL. Do not copy the `<r25:report_run_url>` element tags.

Formulas and Summaries

Formulas

Formulas allow you to create new values by manipulating and evaluating existing field data. Formulas can reference data fields, other formulas, summaries, and parameters.



For information on formulas that may not be covered in this topic, please see [The JReport Designer User's Guide](#).

The result of a formula can be referenced in other formulas, inserted into a report, and used to control object properties.

Here are some things to remember when working with formulas:

- Formulas are stored in the Catalog and may be shared by multiple reports.
- Changes made to formulas will be reflected in all reports sharing the formula.
- The formula name must be unique across the entire Catalog.
- Formulas are assigned within the Catalog to a Data Source and can be referenced by any report using the same Data Source.
- The result of a formula must be of a single data type.

- *Home > Formulas*
- *Home > Formula Reference*

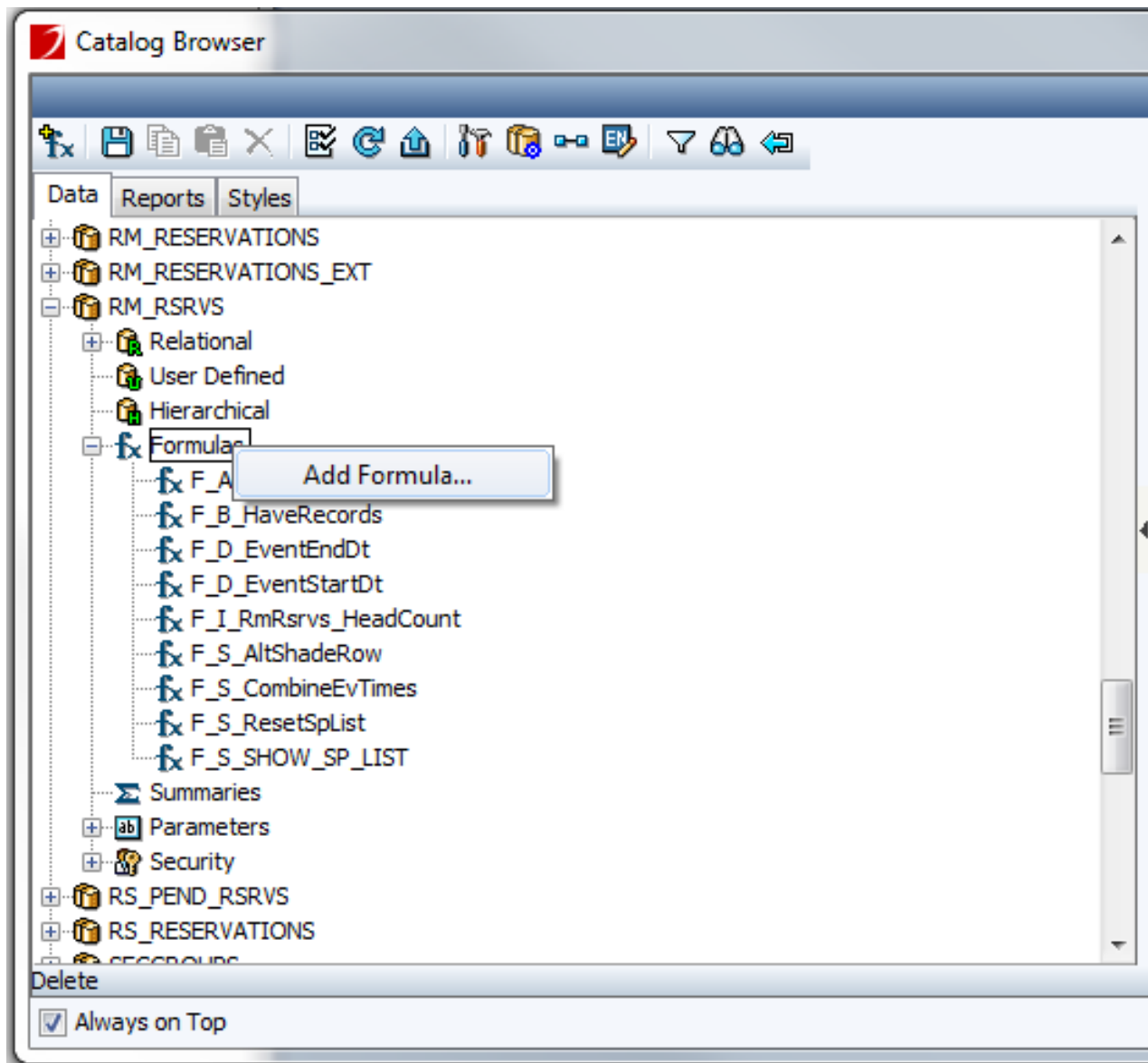
Creating a New Formula

Formula creation is performed within the Formula Editor dialog and can be initiated from within Catalog Browser or report design editor.

- *Home > Formulas > Creating a formula*

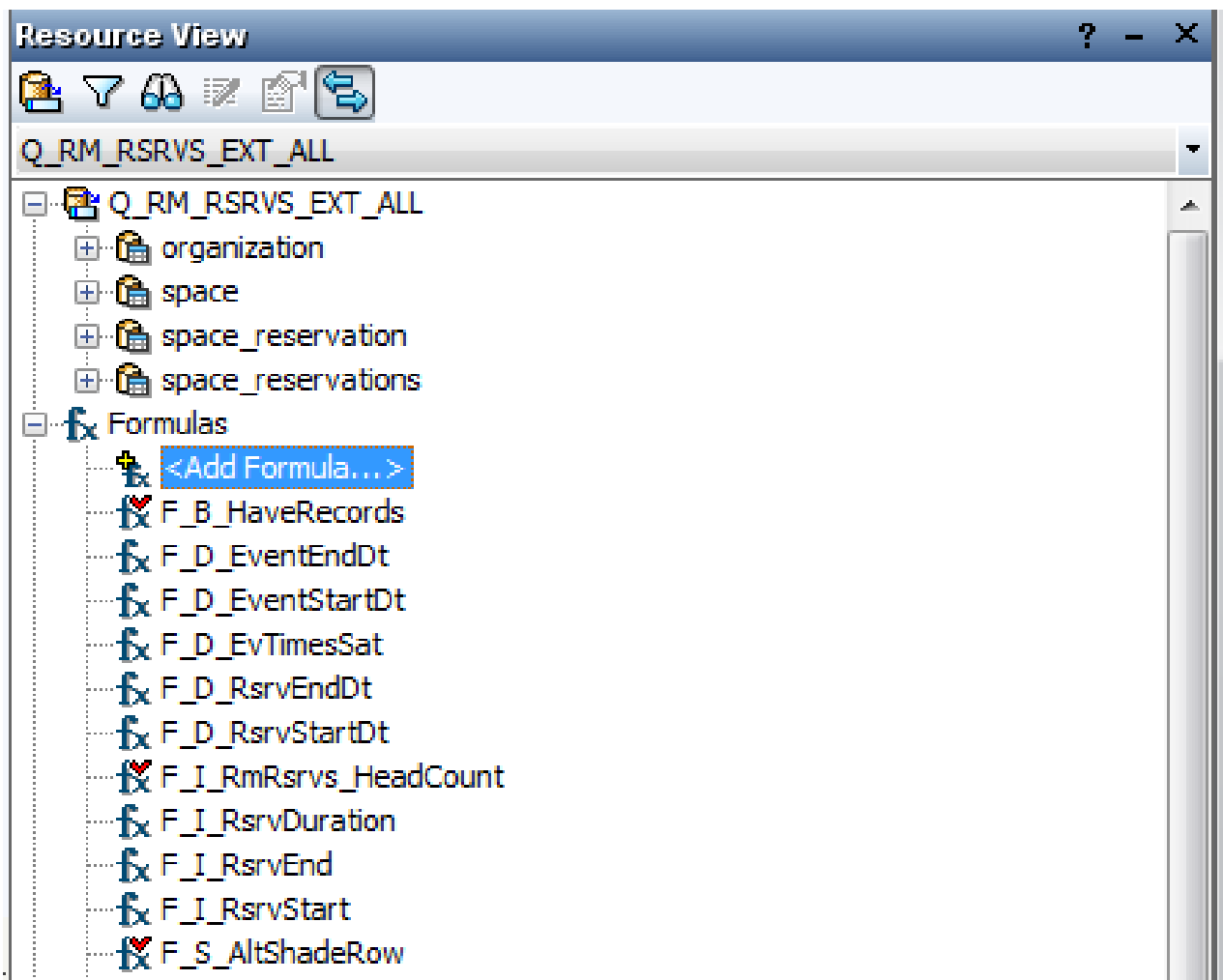
From within the Catalog Browser

From the Catalog Browser, expand the Data Source in which to associate the formula. Right-click over the Formulas node and select Add Formula....



From within the report design editor

To add a new formula from the Resource View panel of the report design editor, start by left-clicking on the Query belonging to the Data Source in which the formula will be associated. Click the <Add Formula...> option.



Formula Naming Convention

A standard naming convention is recommended that includes a prefix, "F_", followed by an abbreviation for the data type of the formula result. This naming convention helps organize formulas in lists. The formula name should not contain any blank spaces or special characters.


Data Type	Recommended Prefix
String	F_S_
String [Array]	F_SA_
Integer	F_I_
Integer [Array]	F_IA_

Number	F_N_ (where result might include a decimal value)
Number [Array]	F_NA_
Boolean	F_B_
Date	F_D_
Time	F_T_
dateTime	F_DT_
*URL	F_URL_

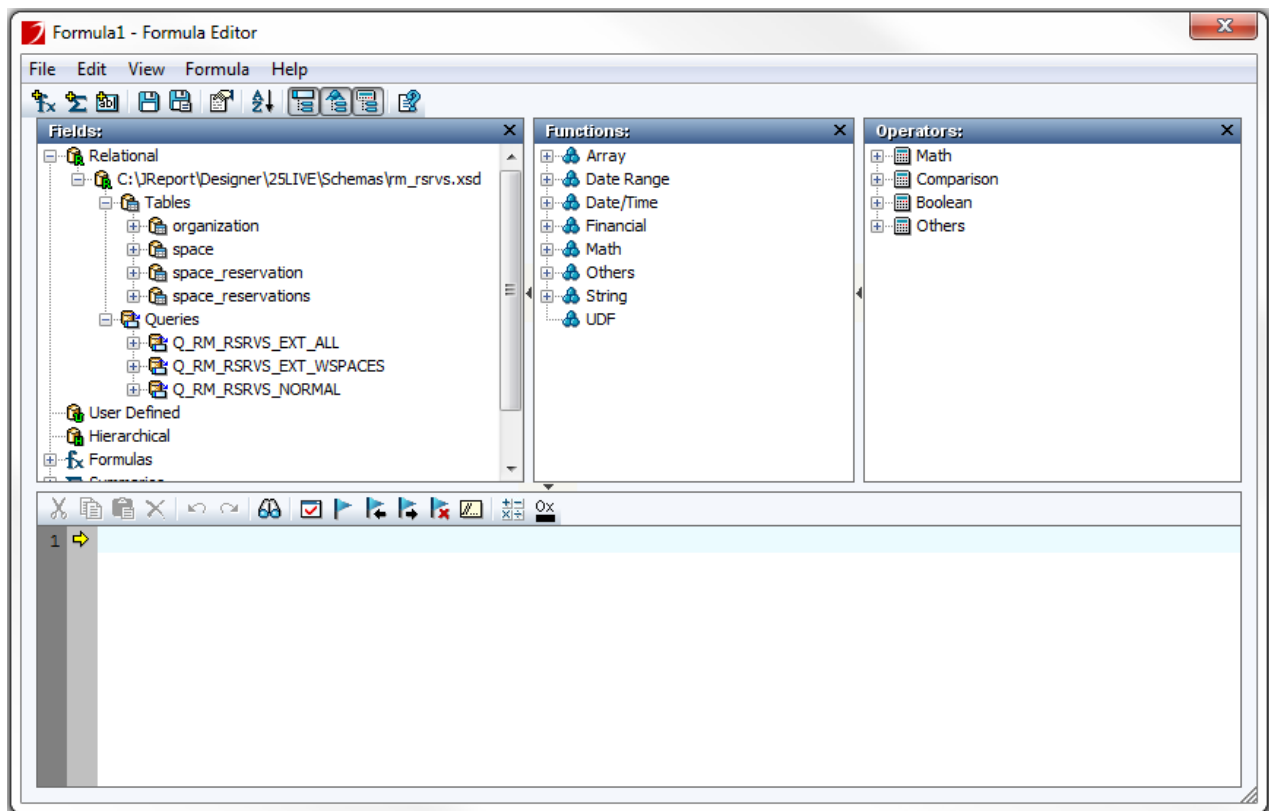
**A formula used to construct a URL will return a result of data type string. The naming convention "F_URL_" is recommended because of its intended use in requesting Series25 WebServices data.*

Formula Editor

The Formula Editor is used to create and edit formulas. Panels across the top of the Editor window allow you to select report Fields, Functions, and Operators to be used in the formula.

 For information on creating formulas that may not be covered in this topic, please see [The JReport Designer User's Guide](#).

- [Home](#) > [Dialog Reference](#) > [Formula Editor window](#)



Fields Panel

Fields are presented in a hierarchical structure with either the parent Data Source or Query at the top. The Fields panel contains a list of data fields, other Formulas, Summaries, Parameters, and Special Fields.



Data fields, Formulas, Summaries, and Parameters appear with the "@" symbol prefix when referenced within a formula.

Functions Panel

A function is a built-in procedure or subroutine used to evaluate, make calculations on, or transform data. JReport supplies many different standard functions that may assist with building formulas. Functions typically require one or more values as input and return a result that can be further evaluated or output as the formula result.

- *Home > Formula Reference > Built-in functions*

Operators Panel

An operator is a special symbol that describes an operation or an action to take place between two or more values.

- *Home > Formula Reference > Operators*



Use the Double Equal "==" Symbol to Compare Values

Use the double equal "==" symbol as the operator when comparing values. The single equal "=" symbol is

used as the operator in setting the value of a local variable or return value.

Referencing Fields in Formulas

Data Fields

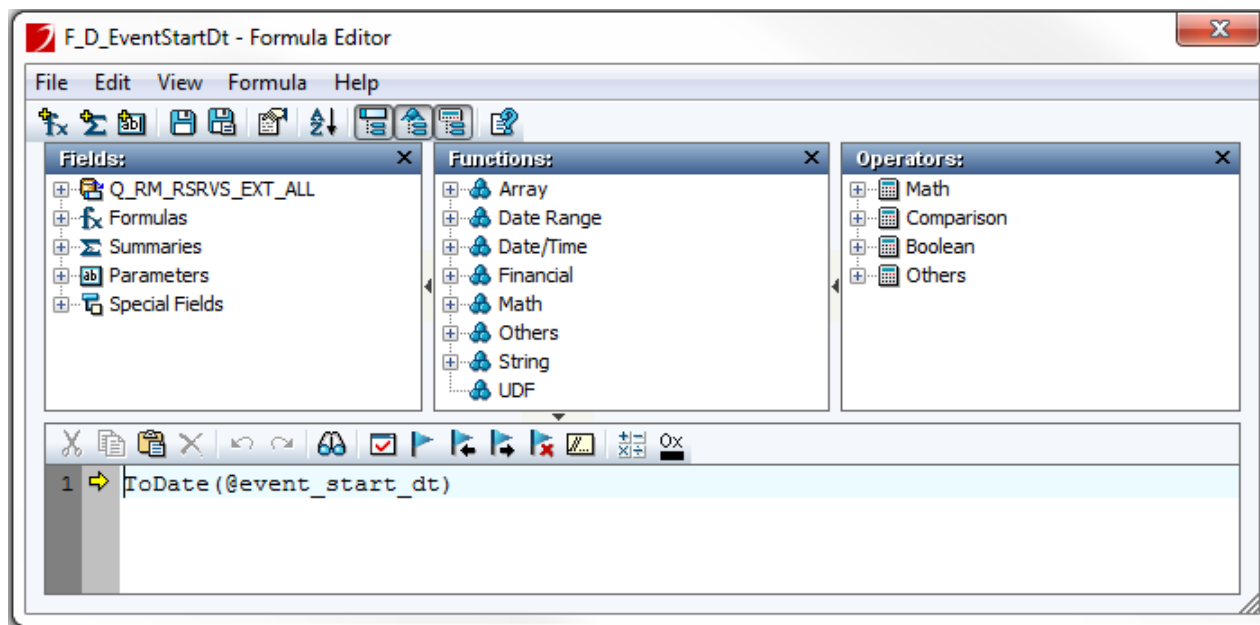
Data fields can be selected from the Fields panel of the Formula Editor. Data fields are listed in alphabetical order within their assigned Table. Tables included in a parent Data Source or a more targeted Query appear at the top of the hierarchical Fields display.



Formulas are often created for use in a specific report. Reports are assigned Queries which establish the data fields appropriate for use in the report. Formulas should only reference data fields belonging to the Queries assigned to the report.

In the example below, the formula uses a function, ToDate(), to extract the date portion from the event_start_dt dateTime data field.

Formula name: F_D_EventStartDt

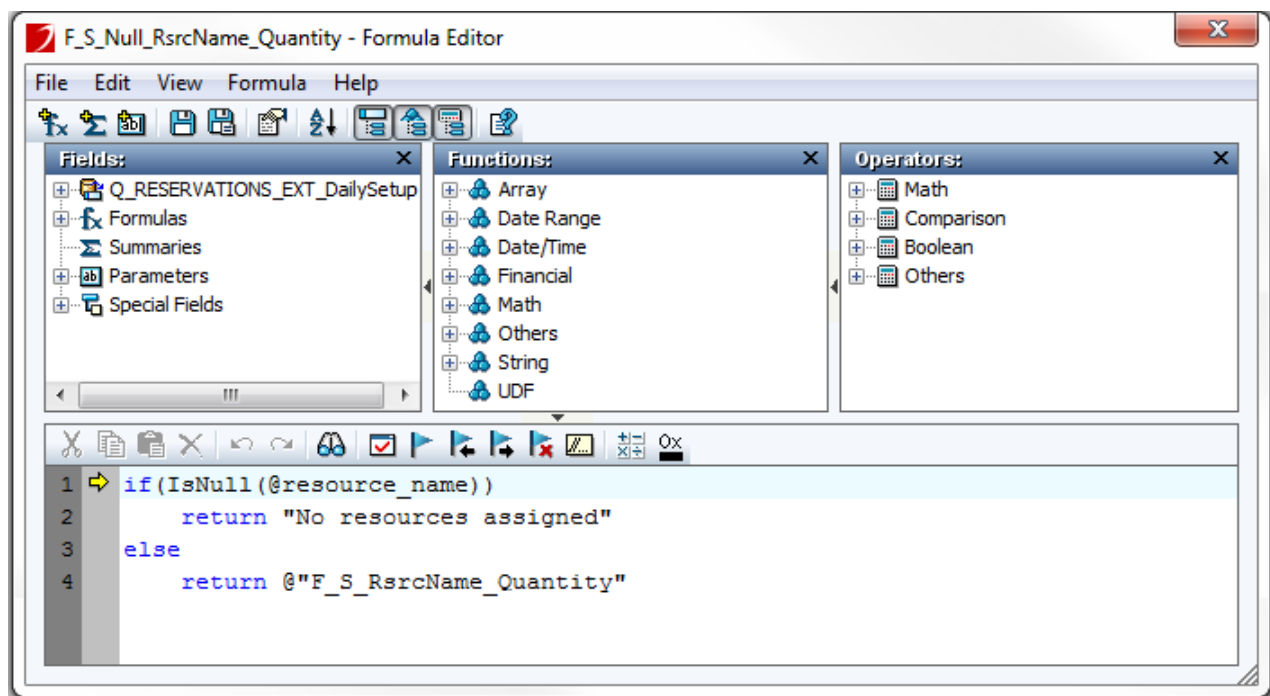


Formula Fields

Previously created formulas can be referenced in a newly created formula. All previously created formulas associated with the selected Data Source appear in this list. Care should be taken to reference formulas containing data fields included in the selected report Query.

This formula references the data field, resource_name, and a previously created formula, F_S_RsrcName_Quantity. If no resource is assigned to a given reservation, the formula will return a message, "No resources assigned." Else, the result of the previously created formula, which concatenates the resource_name and resource_count data fields, will be returned.

Formula name: F_S_Null_RsrcName_Quantity

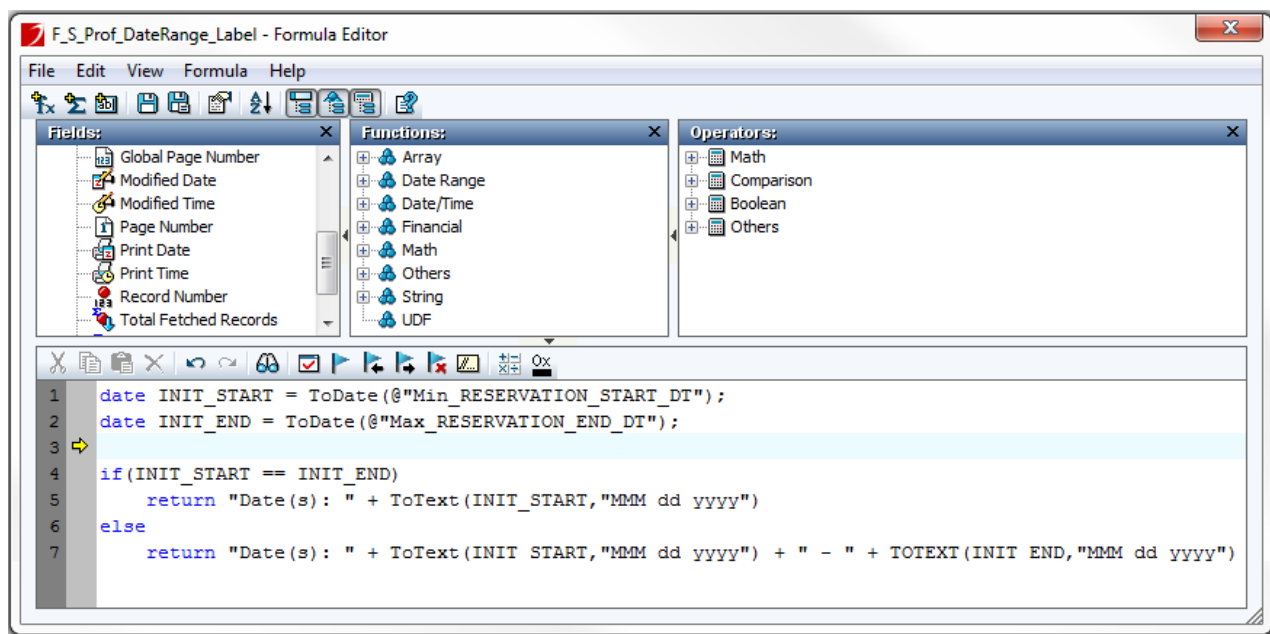


Summary Fields

A summary is a special kind of formula and can be referenced by other formulas. Summaries can represent grand totals or subtotals for data groups.

This formula references two summaries, Min_RESERVATION_START_DT and Max_RESERVATION_END_DT. These summary fields calculate the dateTime of the first and last reservation occurrence for a given reservation profile. Local variables, INIT_START and INIT_END, are used to capture the date portion of these dateTime summary values. If these dates are equal, a Dates(s) label is printed with a single date, else the Dates(s) label will show a date range.

Formula name: F_S_Prof_DateRange_Label

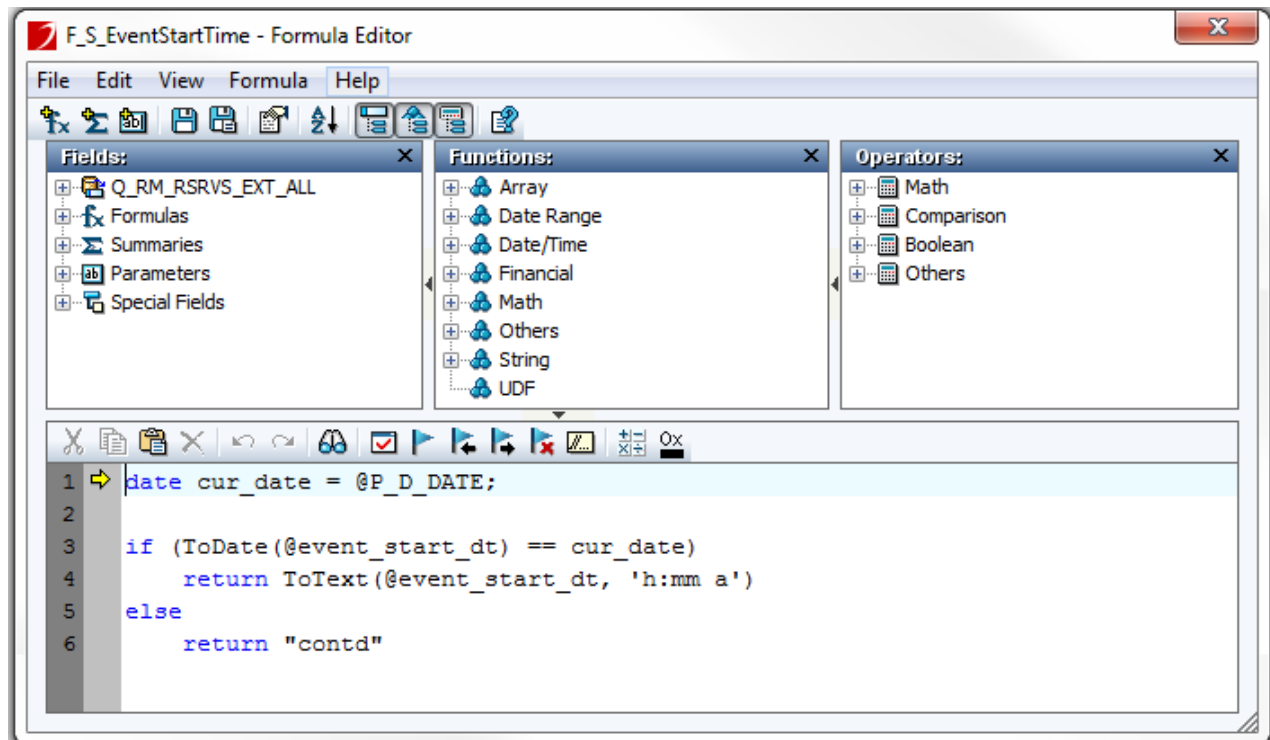


Parameter Fields

A parameter is a variable whose value is determined at runtime. Parameters values are established prior to the evaluation of report formulas and can be passed through different report objects.

This formula references a parameter field called P_D_DATE. A parent Banded object can be designed to pass a value for each date, in the report date range, to this parameter field in an embedded Banded object. This formula uses the value of this parameter to provide a current date context when compared to the event_start_date.

Formula name: F_S_EventStartTime

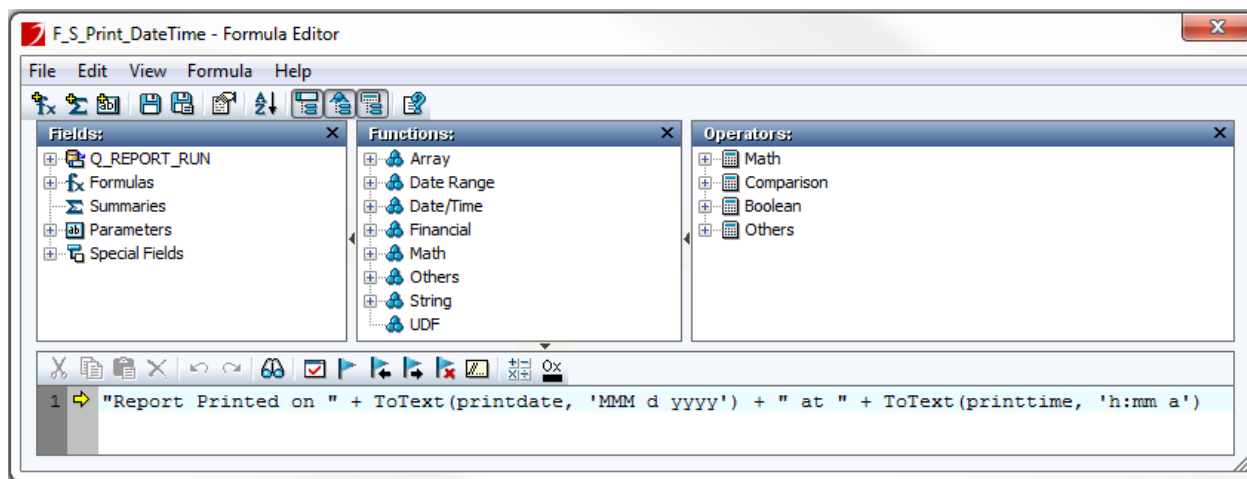


Special Fields

Special fields reference system information, such as Print Date, Print Time, Total Records, etc., that can be used in building a formula.

In this formula example, two special fields, printdate and printtime, are referenced. The formula returns a string value that combines text with these special fields and is typically placed in the report footer.

Formula name: F_S_Print_DateTime



Formula Syntax

Syntax rules

A JReport formula is largely written in plain text and includes a statement, or sequence of statements, often used to evaluate, manipulate, and compare report fields. Formula syntax is unique to JReport, however, it may share similarities to JavaScript and SQL.

 For information on formula syntax that may not be covered in this topic, please see [The JReport Designer User's Guide](#).

- *Home > Formula Reference > Formula syntax*

Here are some basic rules that will assist you in successfully validating a formula:

- Text strings referenced within a formula must be enclosed in quotation marks.
- Data fields, formulas, summaries, and parameters, referenced within a formula must be prefixed with the @ sign.
- Text used in identifying functions or operators is not case-sensitive.
- Text intended to target data field values is case-sensitive.
- The data type of a formula result must be constant.
- When comparing values, use the double equals operator "==".
- When setting the value of a local variable, use the single equal operator "=".
- A semicolon ";" should be used to separate multiple statements used in a formula.

Formula Evaluation Levels

Formulas can be categorized into 6 levels. These levels determine when the actual statements in your formula will be executed. JReport Engine will make several passes over the data to determine when formulas are executed. Even though an individual formula appears to be at a certain level, if it references anything in the higher level then it also becomes the higher level. Formula levels include Constant Record pass one, Record pass two, Group, Global, and Page.



For information on formula syntax that may not be covered in this topic, please see [The JReport Designer User's Guide](#).

- *Home > Formulas > Formula levels*

Comments

Formula comments are notes included with a formula to explain its design and operation. Comments do not print and do not affect the formula. However, comments do appear in the Formula Editor. You can use comments to explain the purpose of a formula or the steps involved in writing it. This can also be a good place to leave notes to yourself, especially if the formula you created is complicated. Comments must begin with two forward slashes (//) and are followed by the text of the comment. Everything that follows the slashes on the same line is treated as part of the comment.

For example:

```
//This formula returns the string "Hello"  
"Hello" //Comments can be added at the end of a line  
//Comments can occur after the formula text
```

Using Formulas to Control Object Properties

A formula result can be used to control various object properties at runtime. Properties are used to control an object's format, position, style, and behavior. The data type of a formula result must match the data type required by the object property.

- *Home > Formulas > Using formulas to control object properties*

Properties Panel

Object properties are assigned using the Properties panel of the Report Inspector. Hover your mouse over, or left-click to select, the property row to see a property description including the required data type.

Inspector

- BandedHeader
 - BandedPagePanel
 - BandedPageHeader
 - DetailPanel
 - RM_RESERVATIONS_RSRV_ID
 - RSRV_ID
 - RM_RESERVATIONS_RSRV_START_DT**
 - RSRV_START_DT
 - EVENT_NAME
 - BandedPageFooter
 - BandedFooter
 - TOC
 - DataSet

Name	Value
Right Line	none
Shadow	false
Shadow Color	0x000000
Top Line	none
Pattern	
Pattern Color	0x000000
Pattern Style	none
Text Format	
Auto Fit	false
Bold	false
Font Face	Arial
Font Size	9
Format	M/d/yyyy h:mm a
Horizontal Alignment	left
Ignore	
Italic	
Maxim	
Striket	
Underl	
Vertical Alignment	center
Word Wrap	false
Others	
Cache Value	true
Column Name	rm_reservations_rsrv_start_dt
Data Mapping File	
Detail Report	
Display Null	

Format

Specifies the display format of the column's value. The format varies with the column's data type and can be defined manually.

Data type: String

Depending on which property is selected, a drop-down list of suggested values may be available. If you are allowed to apply a formula result to the property control, a button appears next to the selected property value.

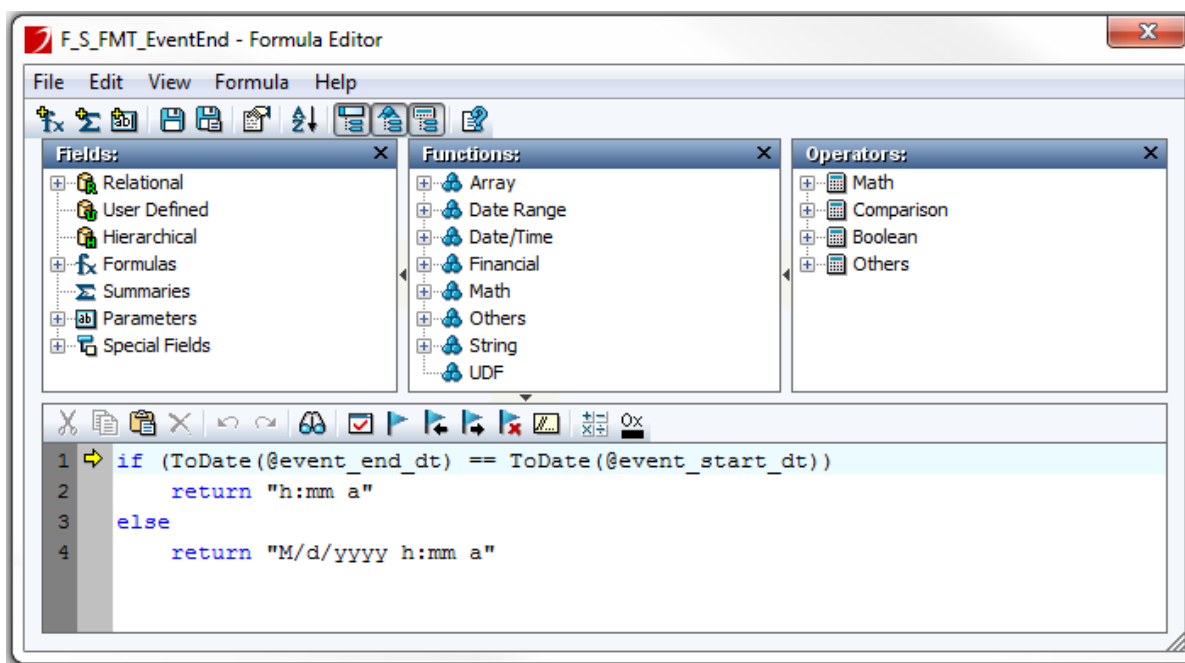
Assigning a Formula to a Property


Follow these steps to assign a formula result to an object property control.

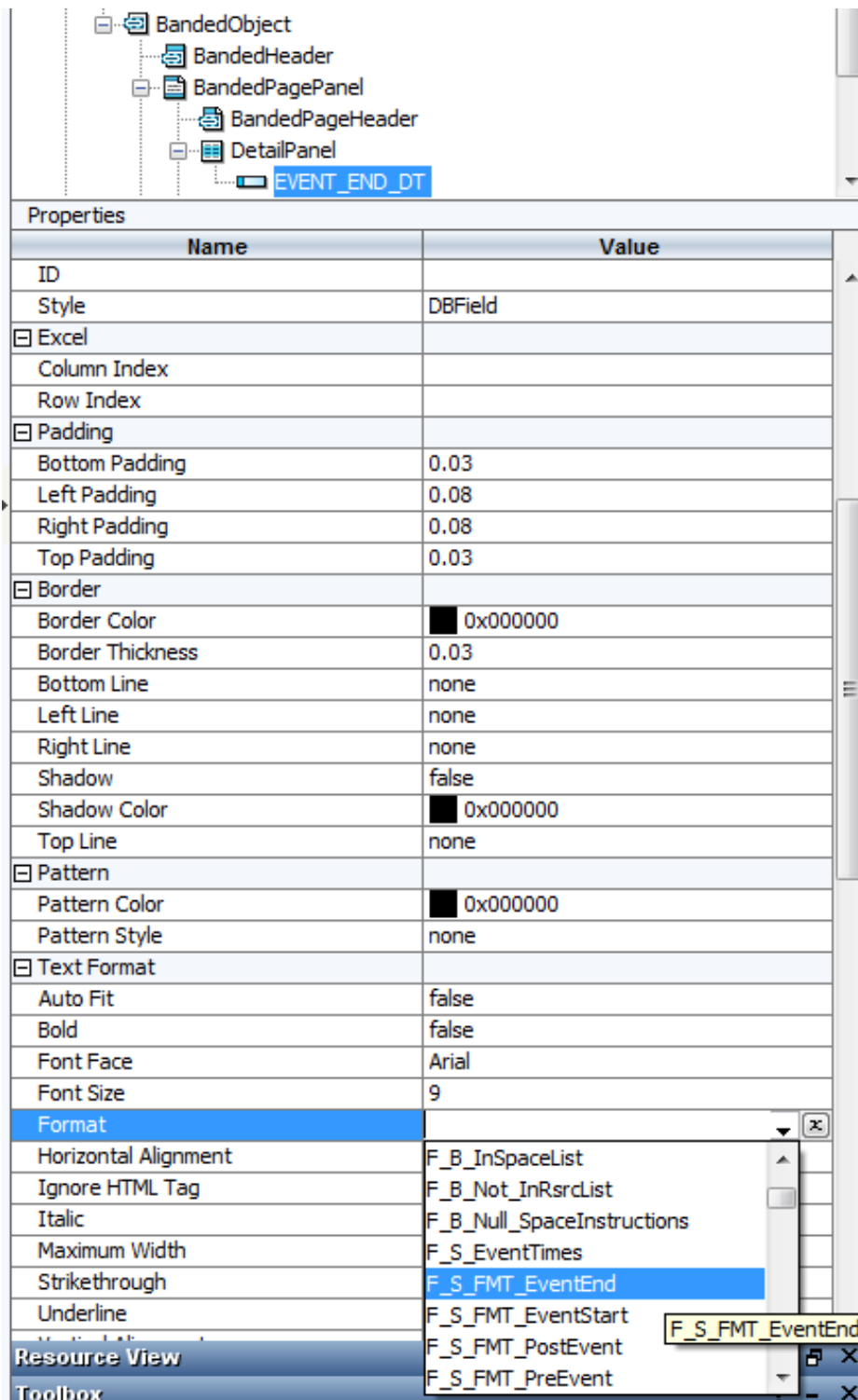
1. Create the formula.

This formula returns a data type string that can be applied to the Format property of a dateTime field. If the date of event_end_dt and event_start_dt are the equal, a time only format is returned, else a dateTime format is returned.

Formula name: F_S_FMT_EventEnd



2. In the body of the report, or in the report structure panel of the Report Inspector, click to select the report object. The Properties panel will display those properties appropriate for the selected object.
3. Select the currently assigned property value in the Properties panel.
4. Click the  button, which should re-populate the items in the drop-down menu to include those formulas where the data type of the formula result matches the expected data type of the property control.
5. Select the desired formula from the drop-down menu.



6. To change the drop-down list items back to the static choices, click the  button.

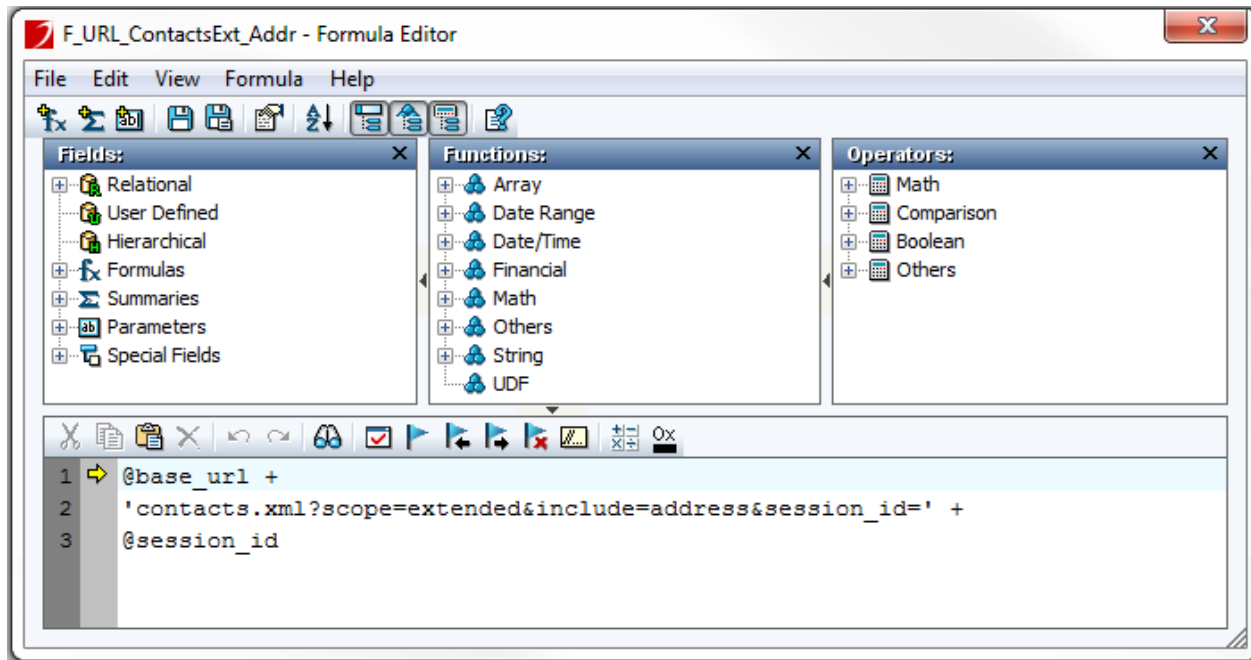
Using Formulas to Combine Text with Data

Each Data Source requires a parameter value that anticipates a Series25 WebServices URL used to request data. A formula is used to construct the URL string, which combines text with data fields. At runtime, the formula value is

passed to the required parameter, and a WebServices data request is made.

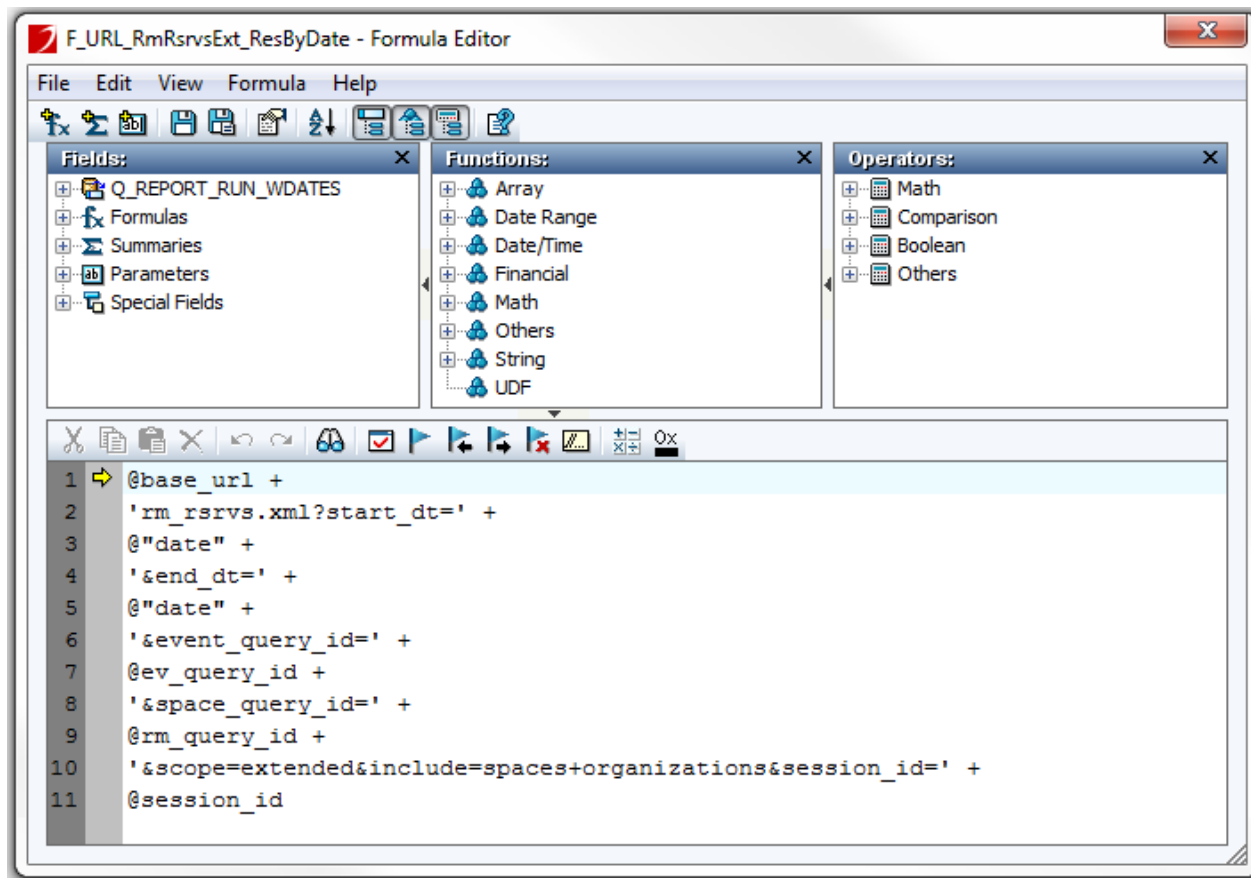
This formula returns a result string that can be linked to a Data Source parameter used to retrieve Series25 WebServices data. This string uses the + operator to combine data fields, @base_url and @session_id, with text. The @base_url and @session_id data fields are referenced within the query Q_REPORT_RUN.

Formula name: F_URL_ContactsExt_Addr



This URL formula example is used in the Reservations By Date standard report. This report has a parent Banded object that is assigned the Q_REPORT_RUN_WDATES query. The DetailPanel of the parent Banded object is evaluated for each date in the report date range as dictated by the result of the assigned Query. By referencing the data field @date, this formula logic will return a unique URL that can be used to request rm_rsrvs.xml data for each date. An embedded Table object, responsible for printing the reports reservation details, is placed in the DetailPanel of the parent Banded object. This formula is linked to the URL source parameter, P_URL_RM_RSRVS, of the embedded Table object.

Formula name: F_URL_RmRsrvsExt_ResByDate



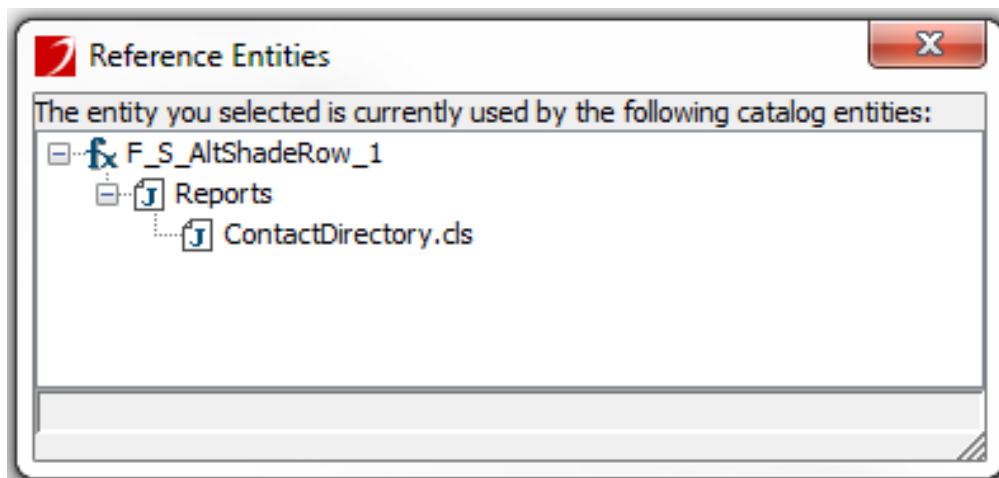
Modifying or Deleting a Formula

To edit a formula from the Resource View of the report design editor or from the Catalog Browser under the Data Source with which the formula is assigned, right-click and select Edit Formula... from the menu.

Formulas can be deleted only from within the Catalog Browser. Formulas are removed from a report by removing all report references to the formula.

Remember, multiple reports can share the same formula. Changes made to a formula will appear in each report sharing the formula.

A reference table feature within the Catalog Browser can assist you with identifying which Catalog resources are referenced by which reports.



To access this reference table information, right-click over a formula, as displayed within the Catalog Browser, and select Reference Entities from the menu.



For JReport Designer version 13.1, a bug was discovered in the reference table feature that may prevent some report resource relationships from displaying. This issue is scheduled to be fixed in the next release.

Summaries

A summary allows you to apply an aggregate function, count, average, sum, etc., to a group of data or to a whole dataset. Different aggregate functions are made available dependent upon the data type of the field being summarized. Group summaries can be assigned to static groups designated by a Group By value or dynamically based on their placement within a report object.



For information on summaries that may not be covered in this topic, please see [The JReport Designer User's Guide](#).

- [Home > Summaries](#)

Summaries are often created for specific report use, however, it is important to remember that they are stored in the Catalog as a child of a parent Data Source. Summaries can be shared by multiple reports. If a summary has been created for a static group, based on a designated Group By value, it can only be referenced within a report object having the same Group By.

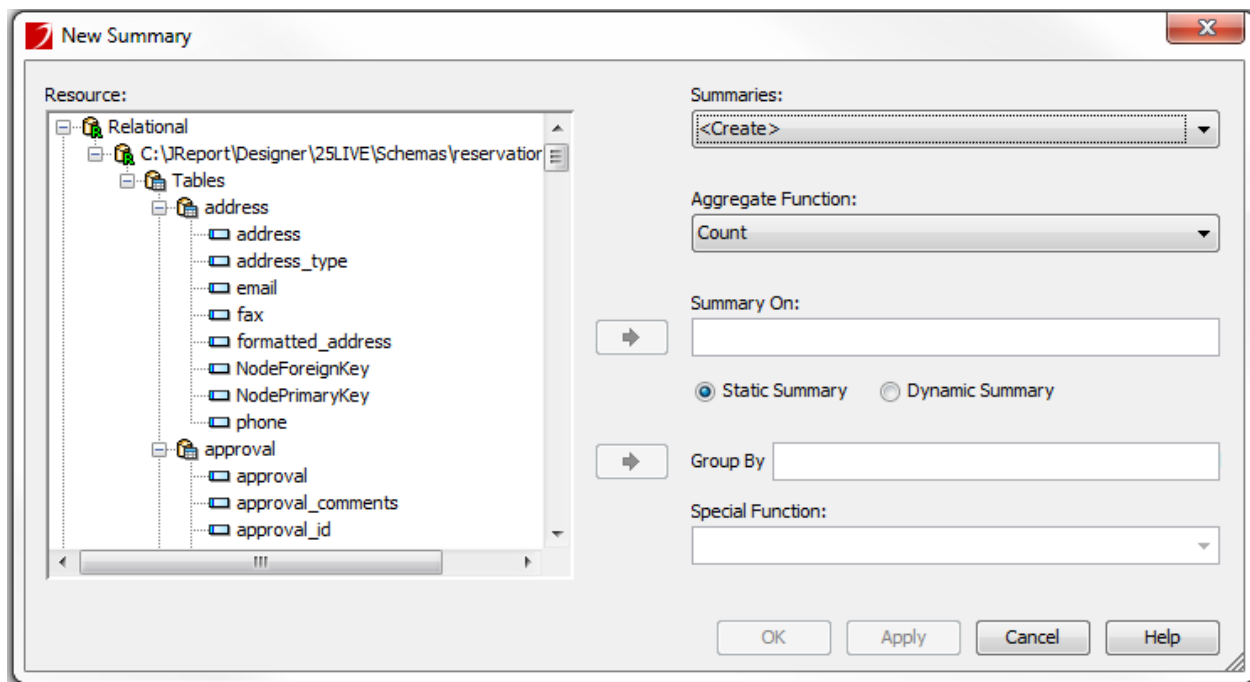
Creating a New Summary

New Summary fields can be created from the Resources View panel within the report design editor or from within the Catalog.

- [Home > Summaries > Creating a summary](#)

The New Summary dialog provides a Resource tree from which to select data fields available within the selected Data Source.

- [Home > Dialog References > New Summary dialog](#)



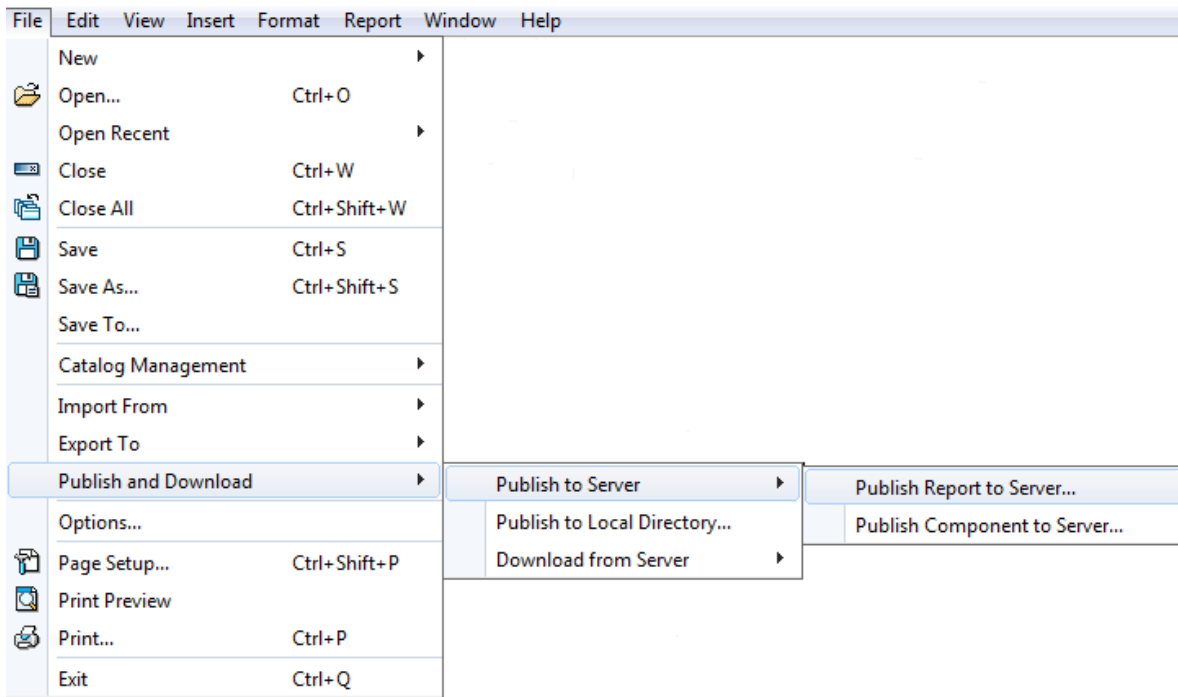
Adding a Custom Report to 25Live

Publishing Files to the JReport Server

A custom report created in the Designer is run by the end-user from 25Live using the 25Live Interactive Reporting System. At the center of this system is JReport Server Live, which is hosted by CollegeNET. When a report is ready to be integrated into the 25Live user environment, the Page Report file (.cls) and corresponding Catalog file (25LIVE_WS.cat) must be published to the CollegeNET hosted JReport Server. Your institution has been allocated a test location and a live location on the JReport Server in which to store custom report files. At runtime, 25Live is programmed to look in these locations to find all custom JReports.

The Designer has a feature to help publish report files to the JReport Server. To publish:

1. From the Designer main menu, select File > Publish and Download > Publish to Server > Publish Report to Server.



2. From the **Connect to JReport Server** dialog, input the Host, Port, and Servlet Path. These values identify the location of the JReport Server.

Host:

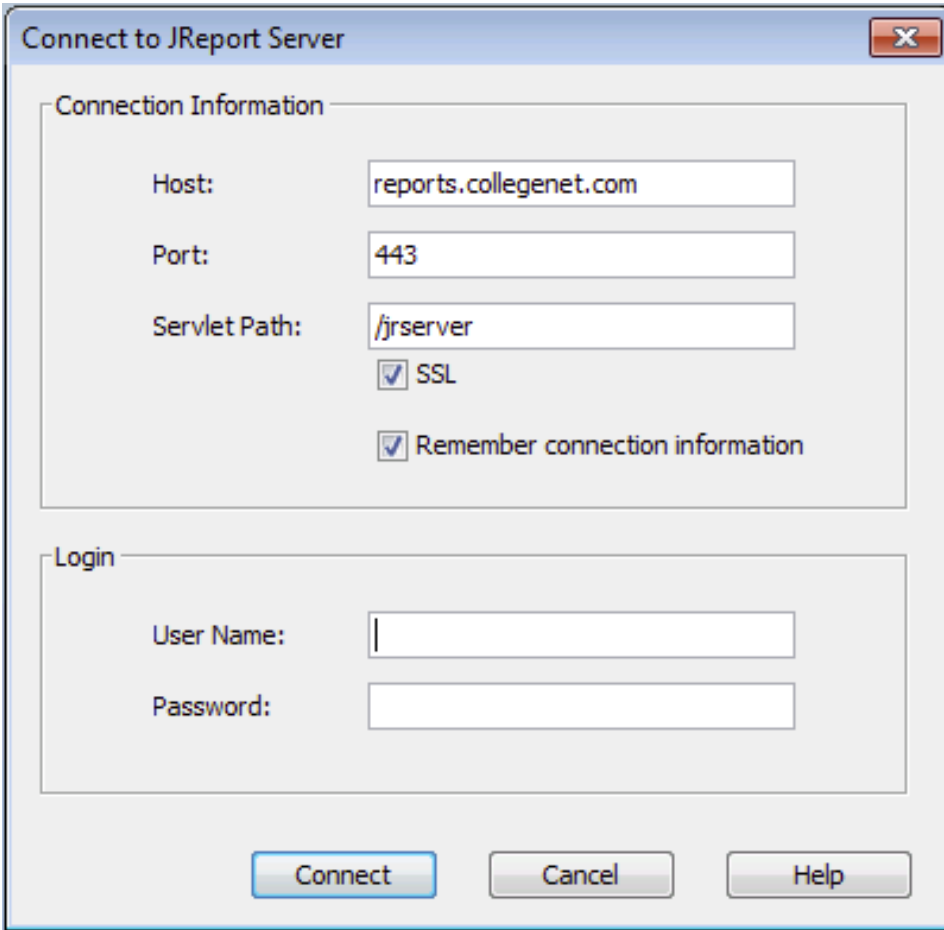
collegenet.com

Port:

443

Servlet Path:

/jrserver



The image shows a Windows-style dialog box titled "Connect to JReport Server". It has a standard close button (X) in the top right corner. The dialog is divided into two main sections: "Connection Information" and "Login".

Connection Information:

- Host:** A text field containing "reports.collegenet.com".
- Port:** A text field containing "443".
- Servlet Path:** A text field containing "/jrserver".
- SSL:** A checkbox that is checked.
- Remember connection information:** A checkbox that is checked.

Login:

- User Name:** An empty text field.
- Password:** An empty text field.

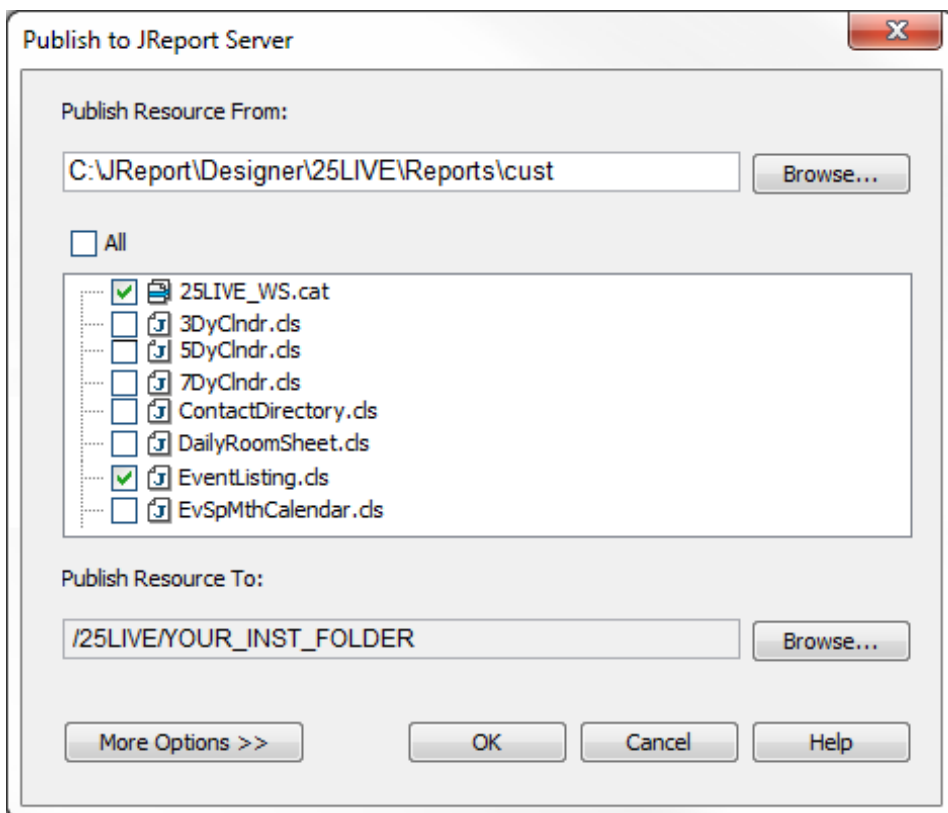
At the bottom of the dialog, there are three buttons: "Connect" (highlighted with a blue border), "Cancel", and "Help".

The User Name and Password field values will be sent to you by CollegeNET.

3. CollegeNET will also provide you with the values for the User Name and Password fields. These credentials are granted specific rights to allow for publishing report resources to your institution's location on the JReport Server.

After filling out the Connection Information and Login sections, click the Connect button, which brings you to the **Publish to JReport Server** dialog.

4. The **Publish to JReport Server Dialog** provides a list of all JReport file resources stored in your local report directory.



Publish Resource From:

This field should show the path to your local report working directory. If you need to change the directory, click the Browse... button and navigate to your local report directory. Page reports with the .cls extension, as well as the catalog file 25LIVE_WS.cat, reside in your local report working directory.

Check the box next to each report resource file you intend to publish to the JReport Server. Typically this will include a page report (.cls) and the catalog file (25LIVE_WS.cat).

Remember that what might be considered as “report logic” is actually stored in the catalog file. It is a good practice to always publish the catalog file (25LIVE_WS.cat) along with any new or updated page report (.cls).

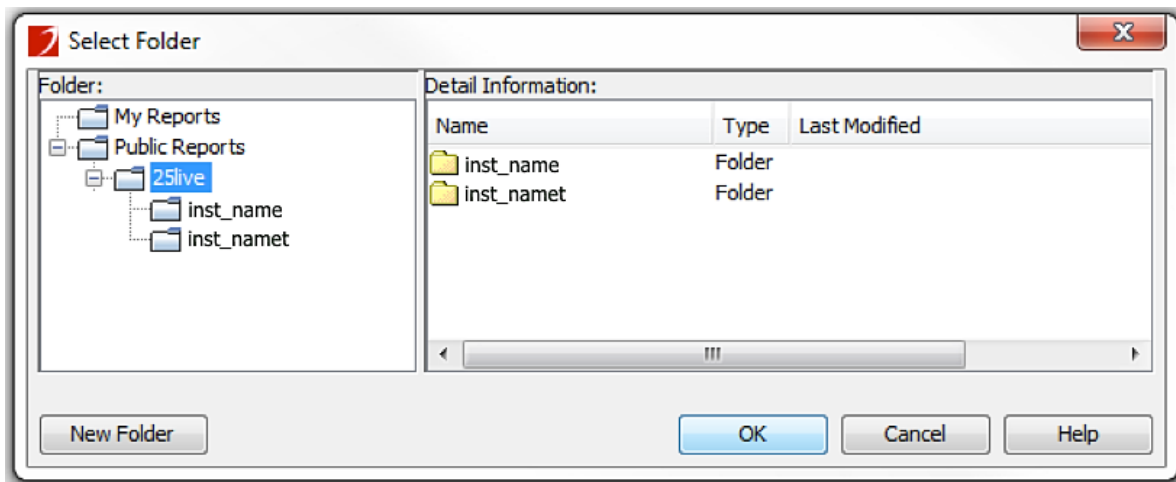
Publish Resource To:

By default, the Publish Resource To input may contain the path to your user folder path on the JReport Server. 25Live does not support access to reports in this location.



Do not publish reports to the default /USERFOLDERPATH/username directory.

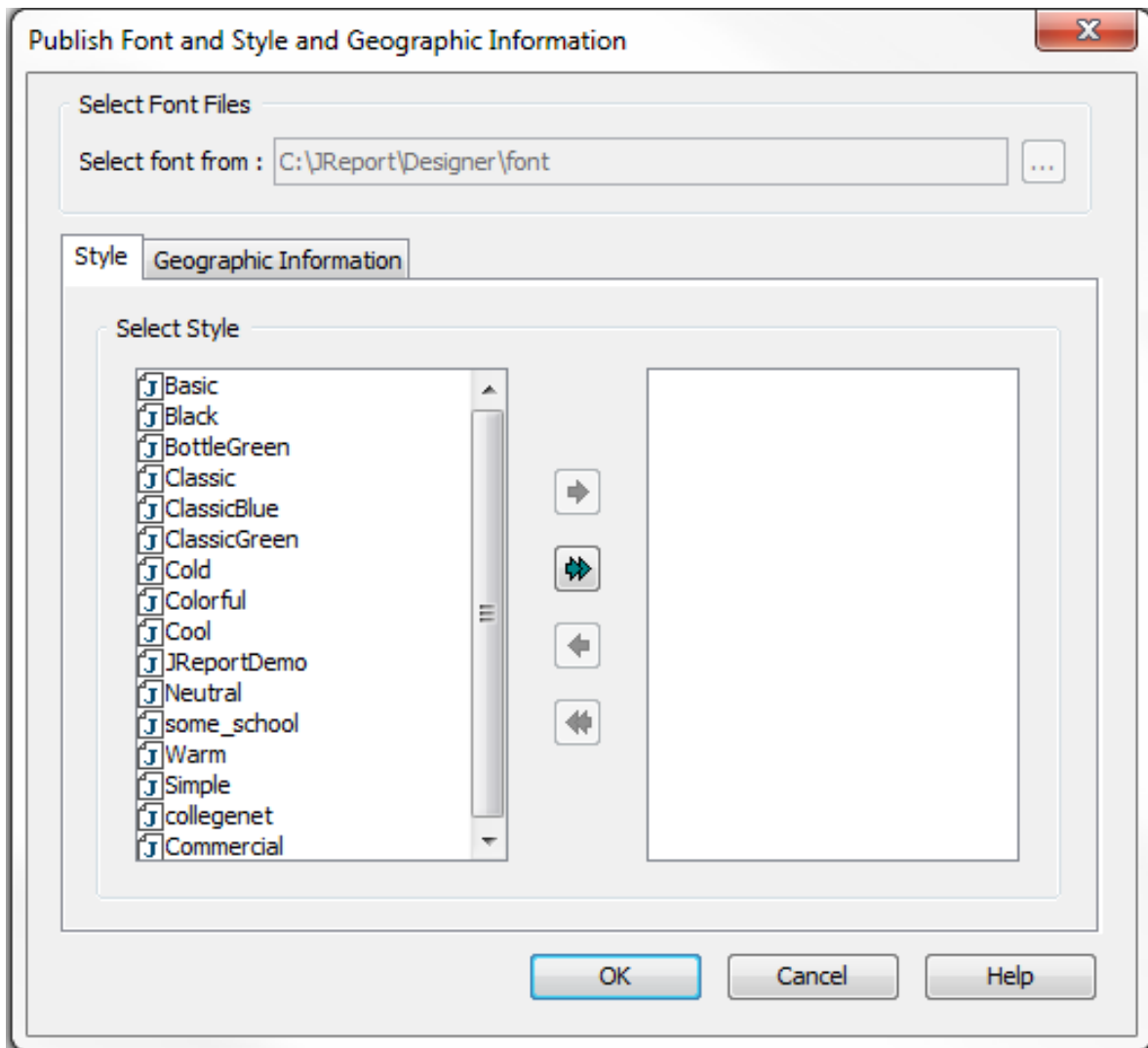
To change the Publish Resource To location, click the Browse... button. Expand the folder for Public Reports and 25live. Under the 25live folder, you should see two locations, one for test and one for production.



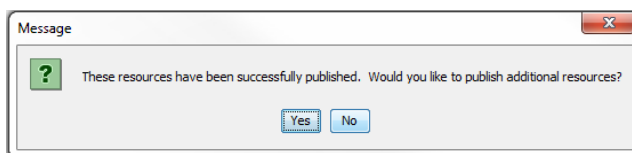
Select the desired destination folder and click the OK button.

5. Back in the **Publish to JReport Server Dialog** click the OK button to publish the selected resources to the selected destination on the JReport Server.

When presented with the **Publish Font and Style and Geographic Information** dialog, click the OK button to receive a confirmation message.



The message asks, "Would you like to publish additional resources?" Click the No button.



Defining a New Custom Report in 25Live Reports

See [Reports Admin - Managing Reports](#).

The Report Run Process

Series25 WebServices makes certain XML data documents available to assist with the report run process. One such WebService is called REPORT.XML. Much of the data available in this WebService has been provided by the database table, RPT_PARM_VALUE. When a user runs a report from 25Live, a row is stored in this table that records the report parameter values supplied by the user. These parameter values are used to construct additional WebServices URL data requests based on the requirements of the report.

While the REPORT.XML WebService contains many of the data elements used to construct a WebService URL, it does not have everything needed. To support JReport's ability to make a WebService request, additional data elements were added:

- **base_url**
 - The base_url is unique to each institution and is the base to all WebServices URLs.
- **session_id**
 - The session_id argument must be appended to every WebServices URL to provide user authentication to the data request.

The base_url and session_id were critical data elements combined with the standard output of REPORT.XML. Additional data elements were added to make the JReport development and design process more efficient. Those elements include:

- **space_search_name**
 - This data element contains the name of a space search if supplied as a parameter at runtime.
- **event_search_name**
 - This data element contains the name of an event search if supplied as a parameter at runtime.
- **org_search_name**
 - This data element contains the name of an organization search if supplied as a parameter at runtime.
- **rsrc_search_name**
 - This data element contains the name of a resource search if supplied as a parameter at runtime.
- **date_list**
 - The date_list node contains child elements <dates> for each date in the report date range. For each <dates> node, child elements for the actual date <date>, day of week <day>, and week of year <week> are available.
- **space_list**
 - If a location search is provided as a report parameter, a space_list node is available. A single element <space_id_list> contains a list of all the space_id values included in the space search. A <spaces> node containing child elements for <space_id> and <space_name> is repeated for each space.
- **resource_list**
 - If a resource search is provided as a report parameter, a resource_list node is available. A single element <rsrc_id_list> contains a list of all the resource_id values included in the resource search. A <resources> node containing child elements for <resource_id> and <resource_name> is repeated for each resource.
- **resource_bycat_list**
 - The resource_bycat_list node is useful for those reports that require placement of resources by their resource category. If a resource search is provided as a report parameter, a <resource_category> element will be created for each distinct resource category represented in the population of resource data returned by the resource search. If no resource search is provided, all resource categories will be represented. For each resource category, a comma-separated list of resource_id values is provided in the element rsrc_id_list

These data elements are added by applying an XSL transform to the standard output of REPORT.XML. This XSL transform file, jreport_run.xsl, has been added to the standard reports folder, reports/std/ws, of your WebServices installation directory. The result of this XSL transform is an XML file that has been translated into a JReport Data

Source called REPORT_RUN.



A new standard report definition has also been added to the list of reports within 25Live. This report called JReport Run can be used to assist with developing reports within the Designer.

For a Page Report to be run from 25Live, it must have a Banded object data container with a query associated with the REPORT_RUN data source. This serves as the foundation of each JReport. With this as a foundation, all other WebServices URL data requests can be constructed.

At runtime, when a JReport is run from the reports menu, 25Live will pass the URL source parameter called P_REPORT_RUN, required by the REPORT_RUN data source. This URL value is used by the JReport to retrieve the parameter values supplied by the user during this session.

[PREVIOUS: 25Live Interactive Report Designer Quick Start Guide](#)

[RETURN TO: Interactive Reports Overview](#)