



# JADE Report Writer

Version 2018

JADE Software Corporation Limited cannot accept any financial or other responsibilities that may be the result of your use of this information or software material, including direct, indirect, special or consequential damages, or loss of profits. There are no warranties extended or granted by this document or software material.

You should be very careful to ensure that the use of this software material and/or information complies with the laws, rules, and regulations of the jurisdictions with respect to which it is used. No part of this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without the express written permission of JADE Software Corporation Limited.

The information contained herein is subject to change without notice. Revisions may be issued to advise of such changes and/or additions.

Copyright © 2019 JADE Software Corporation Limited.

All rights reserved.

JADE is a trademark of JADE Software Corporation Limited. All trade names referenced are the service mark, trademark, or registered trademark of the respective manufacturer.

---

# Contents

|   |          |
|---|----------|
| <b>JADE Report Writer</b>                                     | <b>4</b> |
| Introduction .....  | 4        |
| Importing the JADE Report Writer .....                        | 4        |
| Creating a Reporting View .....                               | 5        |
| Selecting Types and Features .....                            | 8        |
| Root Collections .....  | 13       |
| Exercise 1 – Creating a Bank Reporting View .....             | 17       |
| Exercise 2 – Selecting Types and Features .....               | 18       |
| Exercise 3 – Adding Root Collections.....                     | 20       |
| Creating a Report Design .....                                | 22       |
| Catalog of Available Fields .....                             | 27       |
| Viewing the JADE Report.....                                  | 27       |
| Report Scripts and Combining Data .....                       | 30       |
| Exercise 4 – Creating a Report.....                           | 32       |
| Exercise 5 – Report Scripts.....                              | 33       |
| Exercise 6 – Extracting a Report .....                        | 36       |
| Report Writer Groups.....                                     | 37       |
| The Summary Sheet .....                                       | 40       |
| Exercise 7 – Grouping Data .....                              | 42       |
| Exercise 8 – Summarizing Data.....                            | 44       |
| Root Collection Joins.....                                    | 46       |
| Setting Security Options .....                                | 49       |
| Exercise 9 – Specifying Designer Security Settings .....      | 51       |
| Exercise 10 – Specifying Configuration Security Settings..... | 52       |

---

# JADE Report Writer

---

## Introduction

The JADE Report Writer is a tool that allows end-users and application developers to develop simple reports of a JADE database from a graphical, drag-and-drop interface.

The report writer contains the following applications.

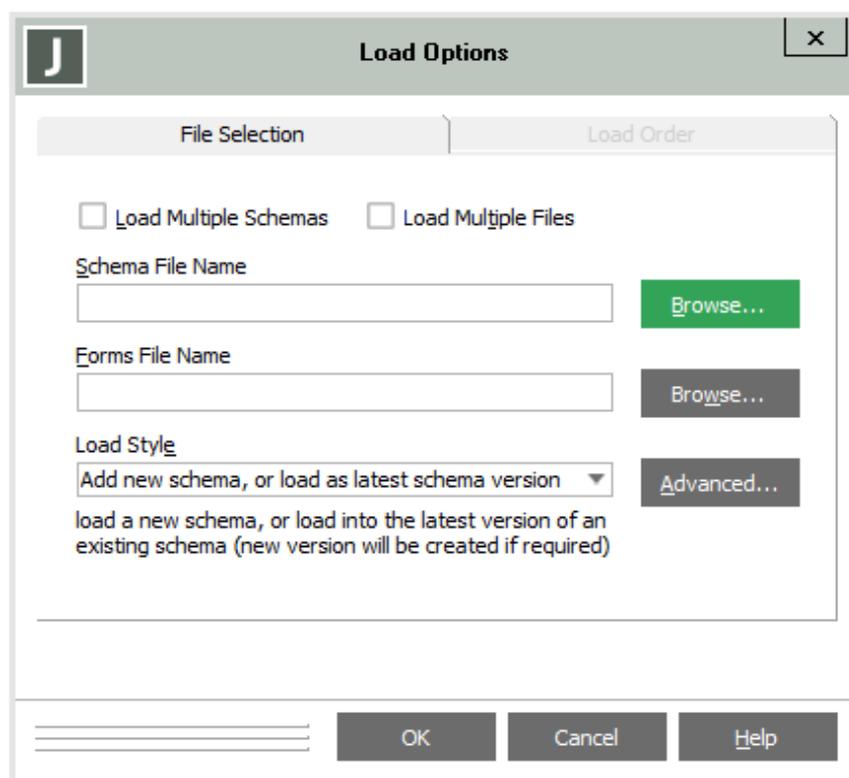
- A configuration application for creating views, which specify what is visible to the report designers
- A designer application for building reports based on the created views

## Importing the JADE Report Writer

The JADE Report Writer files are automatically included with JADE itself, so you do not need to download any additional files to use the JADE Report Writer. To use the JADE Report Writer, you need only to load the **JadeReportWriterSchema** as a subschema of the JADE RootSchema.

The **JadeReportWriterSchema** is found in the **reportwriter** folder in the directory into which you installed JADE (for example, **C:\JadeCourse\reportwriter\**).

To load the **JadeReportWriterSchema**, select **RootSchema** in the Schema Browser and then select the **Load** command from the Schema menu.



Click **Browse** at the right of the **Schema File Name** text box and then select **JadeReportWriterSchema.scn** from the **reportwriter** folder. Click **OK**, to load the JADE Report Writer as a subschema of RootSchema.

## Creating a Reporting View

Reports designed in the JADE Report Writer Designer application are based on a view of the database.

These reporting views specify which classes, and which features of those classes, are visible to the reports.

---

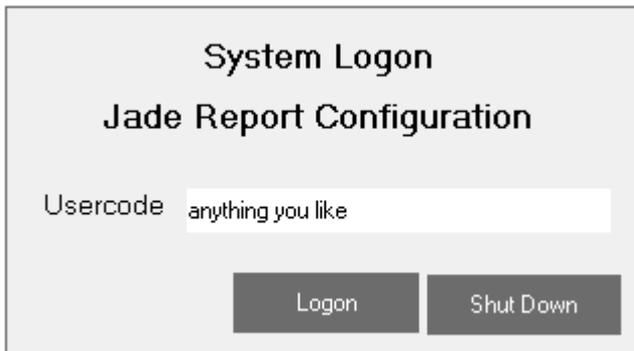
**Note** A feature can be a property or a method. However, if it is a method, the return type must be a primitive type (for example, **Integer** or **String**) rather than an object.

---

To create a reporting view, use the **ReportWriterConfiguration** application in the **JadeReportWriterSchema**. You can open this using either of the following actions.

- Select the **Run Application** toolbar button in the JADE development environment, select **ReportWriterConfiguration**, and then click **OK**.

The following dialog is displayed. You can enter any anything you like as the user code.



The dialog box is titled "System Logon" and "Jade Report Configuration". It features a text input field labeled "Usercode" containing the text "anything you like". Below the input field are two buttons: "Logon" and "Shut Down".

- Write a JadeScript method in the schema that is to be reported on, as follows.

```
startConfiguration();  
  
vars  
  rwManager : JadeReportWriterManager;  
begin  
  create rwManager transient;  
  rwManager.startReportWriterConfiguration("a user name", null);  
epilog  
  delete rwManager;  
end;
```

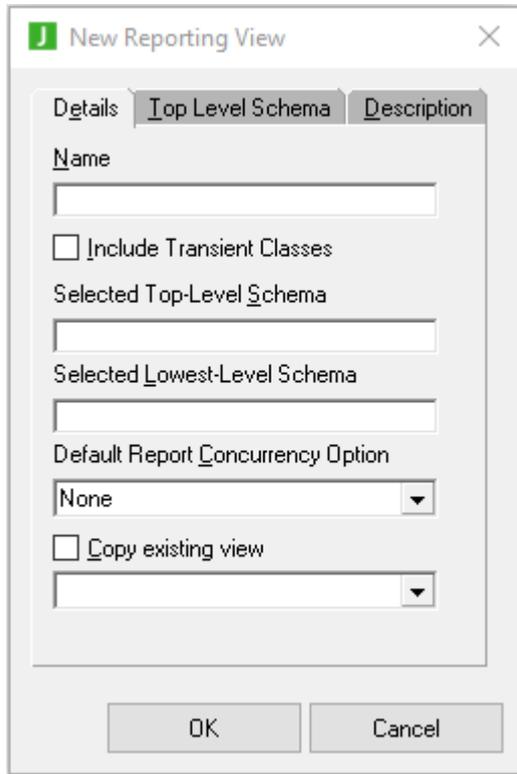
---

**Note** Later in this module, we will set up different levels of security based on the entered user name, but for now any user name is accepted.

---

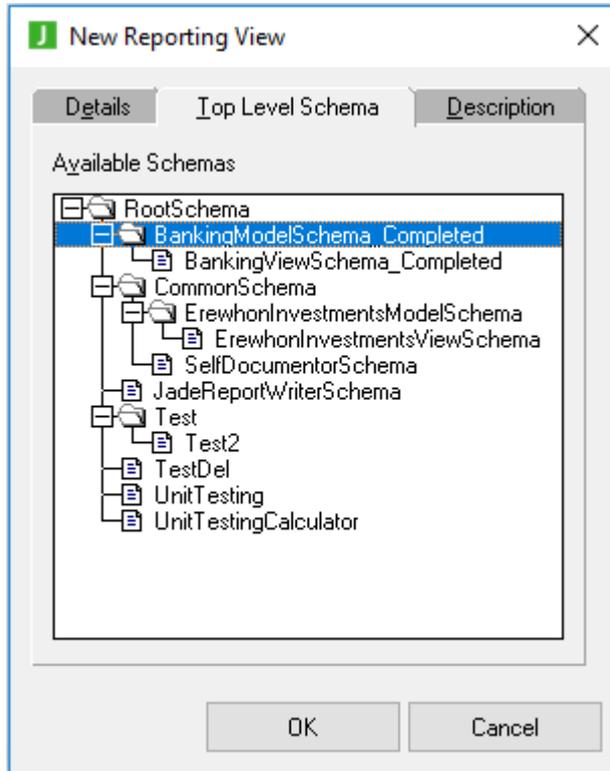
The New Reporting View dialog is then displayed in the JADE Report Configuration window.

To create a new reporting view, select the **New** command from the View menu.



You can specify a name for the view, the top-level schema, the lowest-level schema, and the default concurrency option. Alternatively, you can copy settings from an existing view, if you have one.

When you click on the **Selected Top-Level Schema** text box, a hierarchy of available schemas is displayed on the **Top Level Schema** sheet. Select the schema that you want to be the highest-level schema to include in the view, and then return to the **Details** sheet to finish filling out the form.



When you have selected the schema, click **OK** to create the view. The **Types & Features** sheet of the JADE Report Configuration window is then displayed.

## Selecting Types and Features

Use the **Types & Features** sheet of the JADE Configuration window to select which types (classes and interfaces) and features (methods and properties) will be included in your reporting view.

This sheet contains four main panes, which are manipulated to describe the scope of the reporting view, which are:

- **Types (Classes and Interfaces)**, displaying all classes contained within all schemas between the top-level schema and the lowest-level schema, inclusive.

The screenshot shows a pane titled "Types (Classes and Interfaces)" with a tree view. The tree structure is as follows:

- Object
  - Bank
  - BankAccount
  - Customer
  - MoneyPit
- Interfaces
  - IWithdrawable

The "Customer" class is highlighted in blue. A green arrow icon is visible on the right side of the pane header.

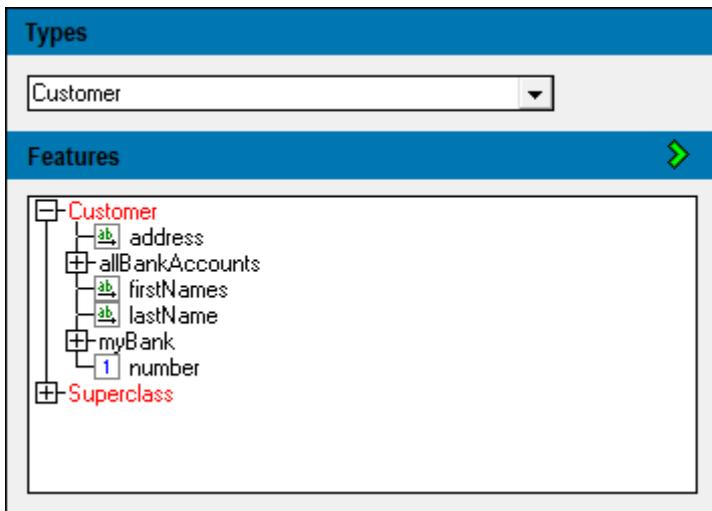
- **Selected Types**, displaying a list of all classes that have been selected to be part of the reporting view.

The screenshot shows a pane titled "Selected Types" containing a table with the following data:

| Alias       | Type        | Show Methods             |
|-------------|-------------|--------------------------|
| Customer    | Customer    | <input type="checkbox"/> |
| BankAccount | BankAccount | <input type="checkbox"/> |

The "Customer" row is highlighted in blue. A green double-checkmark icon is visible on the right side of the pane header.

- **Features**, displaying a list of all available features of the current type. To change the current type, select it in the **Selected Types** pane or select a type from the **Types** combo box above the **Features** pane.

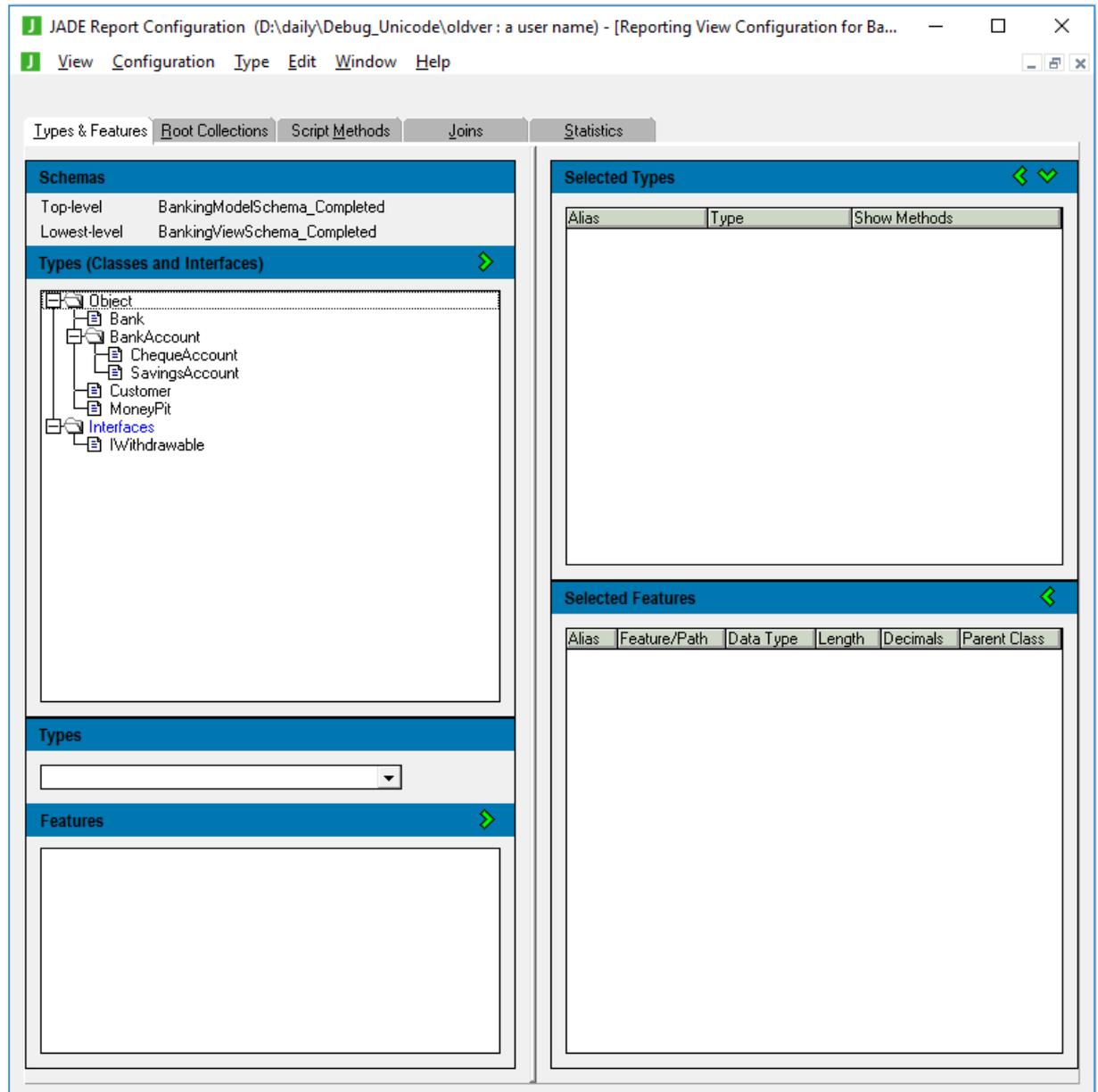


- **Selected Features**, displaying a list of features that have been selected for the current type.

The image shows the 'Selected Features - Customer' pane, which contains a table listing the selected features for the 'Customer' type. The table has the following data:

| Alias       | Feature/Path | Data Type | Length | Decimals | Parent Class |
|-------------|--------------|-----------|--------|----------|--------------|
| First Names | firstNames   | String    | 25     | 0        | Customer     |
| Last Name   | lastName     | String    | 15     | 0        | Customer     |

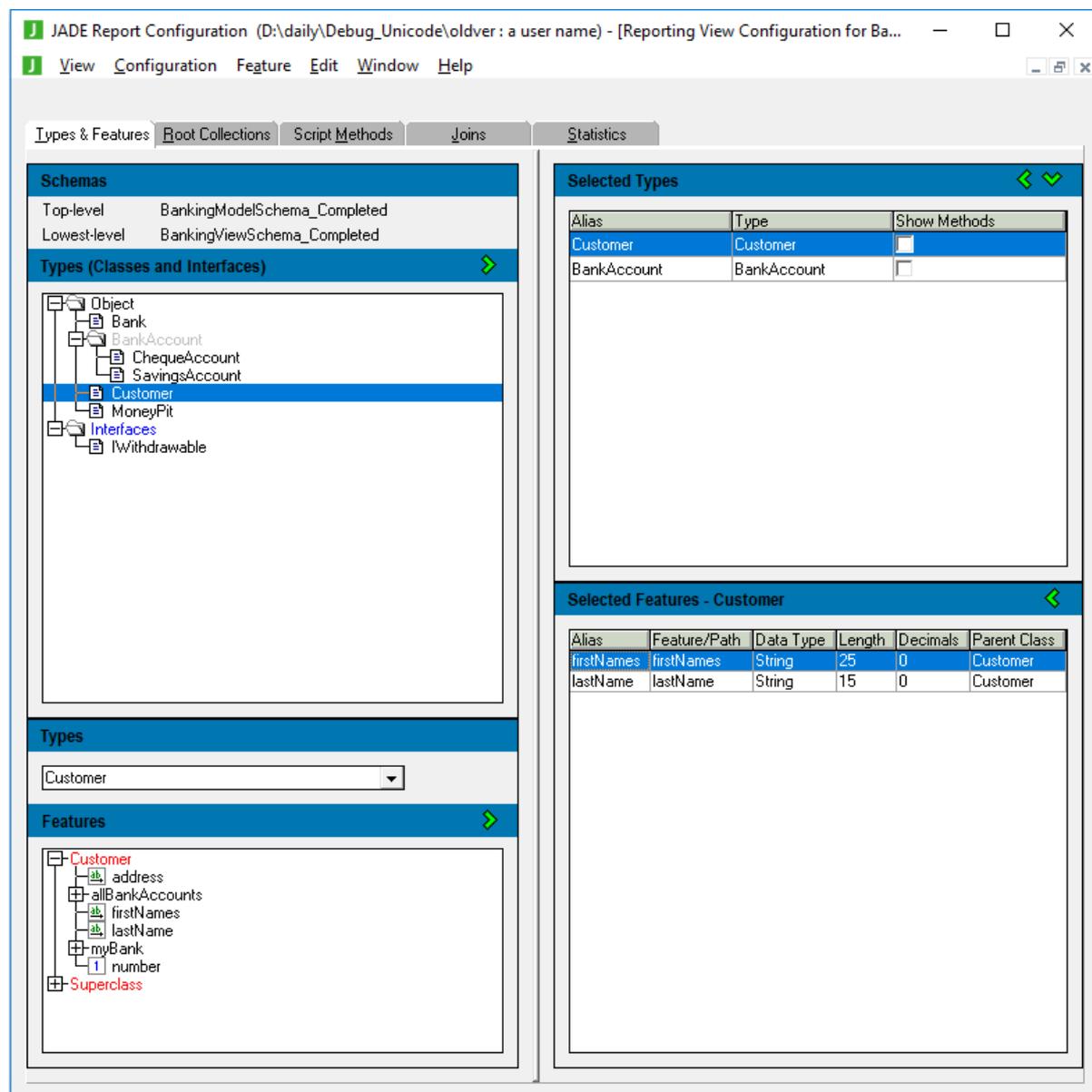
The following is an example of all four panes on the JADE Report Configuration window.



Each of the following actions adds a class to the selected types.

- Select the class and then click the green arrow in the **Types (Classes and Interfaces)** pane header.
- Click and drag the class from the **Types (Classes and Interfaces)** pane to the **Selected Types** pane.
- Double-click the class in the **Types (Classes and Interfaces)** pane.

The following is an example of classes added to selected types.



Each of the following actions adds a feature to the selected features.

- Select the feature and then click the green arrow in the **Features** pane header.
- Click and drag the feature from the **Features** pane to the **Selected Features** pane.
- Double-click the feature in the **Features** pane.

To set an alias for a feature, double-click the feature name in **Alias** column of the table in the **Selected Features** pane and then specify the required name.

| Alias       | Feature/Path | Data Type | Length | Decimals | Parent Class |
|-------------|--------------|-----------|--------|----------|--------------|
| First Names | firstNames   | String    | 25     | 0        | Customer     |
| Last Name   | lastName     | String    | 15     | 0        | Customer     |

---

**Note** There is no need to save or confirm your actions. The reporting view is automatically updated as you make changes.

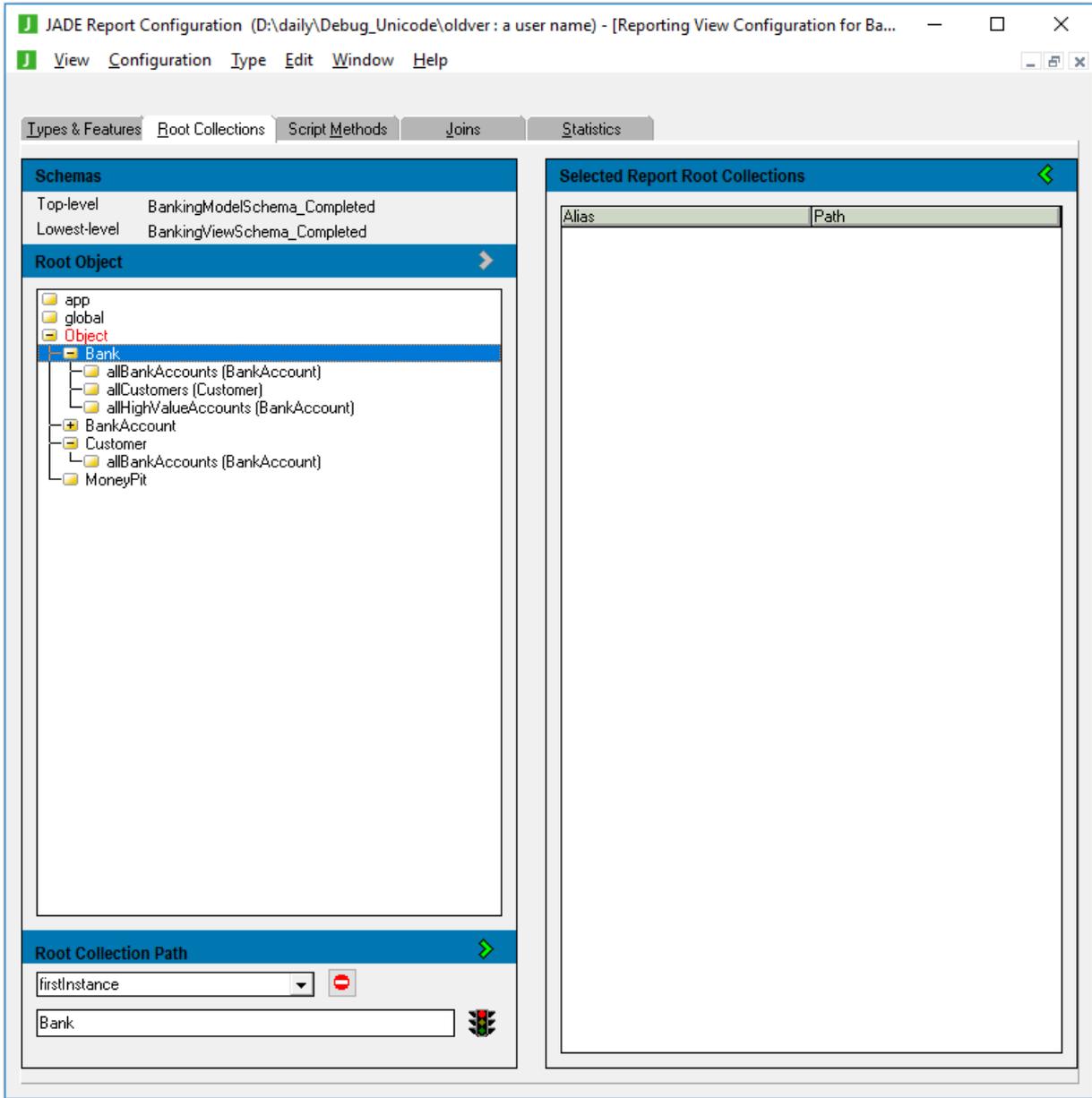
---

## Root Collections

Reports are based on root collections, which provide the primary source of data for your reports. These root collections must be existing collections containing objects that have been selected in the **Types & Features** pane of the JADE Report Configuration window.

To set the root collections for your reporting view, use the **Root Collections** sheet of the JADE Report Configuration window.

The **Root Collections** sheet contains a representation of all collections in the schemas between the top-level schema and lowest-level schema, inclusive.



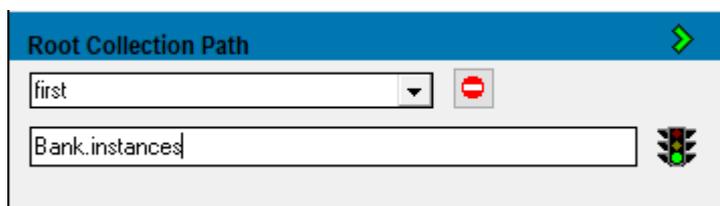
**Note** Classes themselves are represented by *class-name.instances* (for example, **Client.instances**, and therefore can be selected as a root collection. However, it is more usual to use a collection that has been defined explicitly in the schema.

The screenshot displays the JADE Report Writer interface. At the top, the 'Schemas' section shows 'Top-level' as 'BankingModelSchema\_Completed' and 'Lowest-level' as 'BankingViewSchema\_Completed'. Below this is the 'Root Object' section, which contains a tree view of the schema structure. The tree is expanded to show the 'Bank' class, which includes sub-objects like 'allBankAccounts (BankAccount)', 'allCustomers (Customer)', 'allHighValueAccounts (BankAccount)', 'BankAccount', 'Customer', 'allBankAccounts (BankAccount)', and 'MoneyPit'. The 'Bank' class is highlighted in blue. At the bottom, the 'Root Collection Path' section shows a dropdown menu with 'firstInstance' selected, a red prohibition sign, and a text box containing 'Bank'. To the right of the text box is a traffic light icon with the red light illuminated, indicating that the path is invalid.

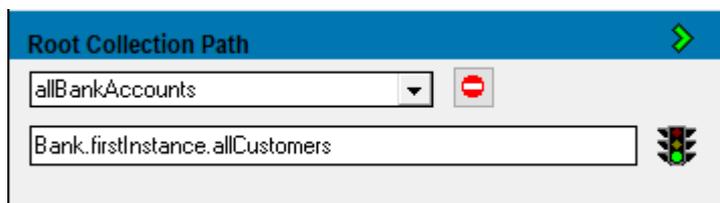
When selecting a class or collection, the **Root Collection Path** pane will show which reference is currently highlighted, with a traffic light symbol showing whether it is valid as a root collection.

As the **firstInstance** property of the Bank class is a singular **Bank** object rather than a collection, this path is invalid, and the traffic light shows **red**.

As **Bank.instances** is a collection of all the **Bank** objects, the path is valid, and the traffic light shows **green**. However, this is likely not a useful collection to report on.



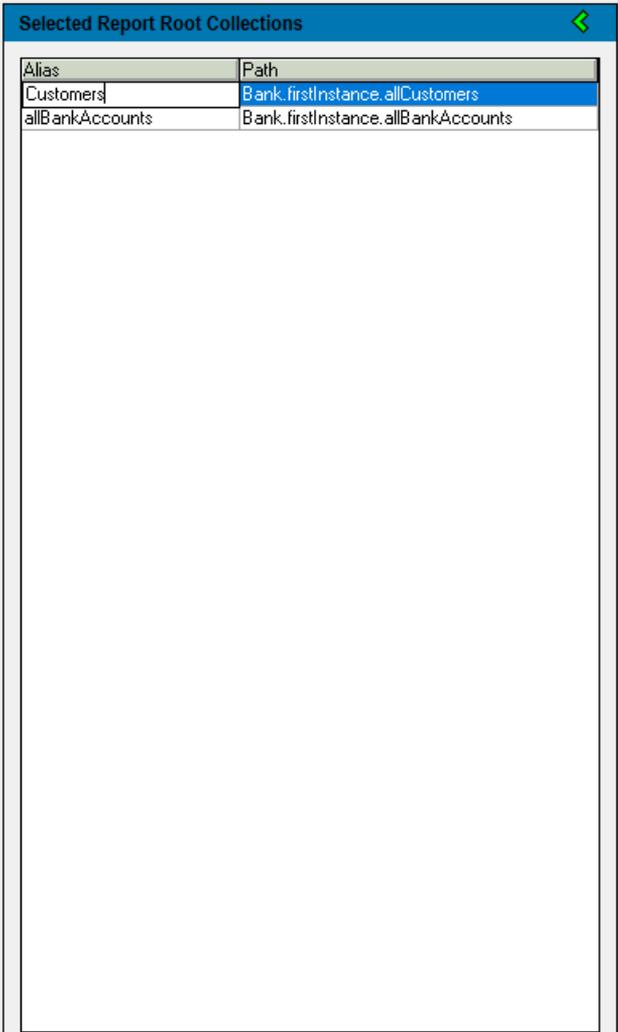
As **Bank.firstInstance.allCustomers** is a collection containing **Customer** objects, the path is valid, and the traffic light shows **green**.



When you have selected a valid collection to add as a Root Collection, click the green arrow button at the right of the header, to add it to the **Selected Report Root Collections** pane.

Alternatively, you can add it by double-clicking it in the **Root Object** pane.

As with types and features, you can set an alias if you do not want to use the variable name of the collection.



| Alias           | Path                               |
|-----------------|------------------------------------|
| Customers       | Bank.firstInstance.allCustomers    |
| allBankAccounts | Bank.firstInstance.allBankAccounts |

## Exercise 1 – Creating a Bank Reporting View

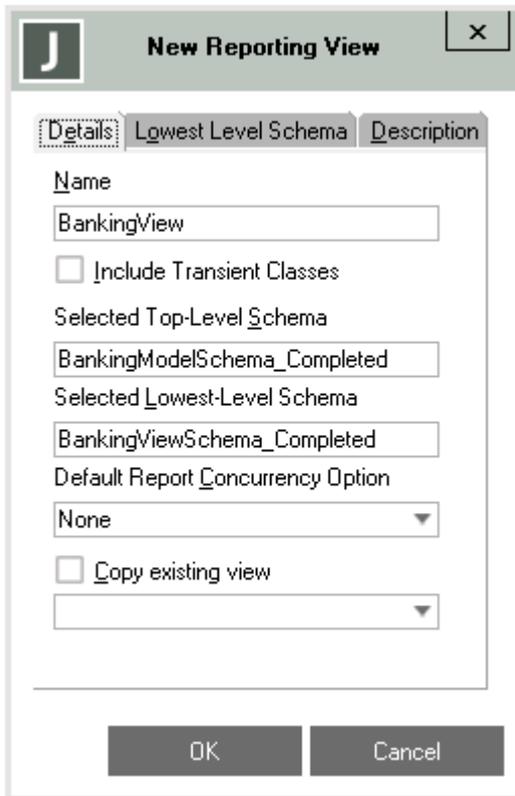
In this exercise, you will add a reporting view to the Banking system created in the JADE Developer's course.

1. Open the JADE Report Configuration window by coding and running the following JadeScript method.

```
startConfiguration();

vars
    rw : JadeReportWriterManager;
begin
    create rw transient;
    rw.startReportWriterConfiguration("User", null);
epilog
    delete rw;
end;
```

2. Select the **New** command from the View menu to create a new reporting view.
3. Enter **BankingView** as the name, **BankingModelSchema** as the top-level schema, and **BankingViewSchema** as the lowest-level schema. Leave the default report concurrency option as **None**.

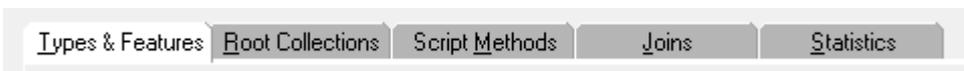


4. Click **OK**.

## Exercise 2 – Selecting Types and Features

In this exercise, you will set the classes, methods, and properties to be included in the newly created reporting view.

1. Ensure that the **Types & Features** sheet is selected. (It should be, by default.)



2. Select the **BankAccount** and **Customer** classes and add them to the **Selected Types** pane.
3. Change the alias for **BankAccount** to **Account**.

- Your **Selected Types** pane should show the following.

| Selected Types |             |                          |
|----------------|-------------|--------------------------|
| Alias          | Type        | Show Methods             |
| Account        | BankAccount | <input type="checkbox"/> |
| Customer       | Customer    | <input type="checkbox"/> |

- For the **Account (BankAccount)** type, add the **balance** feature.

**Tip** To show the features of **BankAccount**, make sure it is selected in the **Selected Types** pane, or select it from the list in the **Types** pane.

| Selected Features - Account |              |           |        |          |              |
|-----------------------------|--------------|-----------|--------|----------|--------------|
| Alias                       | Feature/Path | Data Type | Length | Decimals | Parent Class |
| balance                     | balance      | Decimal   | 12     | 2        | BankAccount  |

- For the **Customer** type, add the **firstNames** and **lastName** features.

- Change the alias of **firstNames** to **First Name** and the alias of **lastName** to **Last Name**.

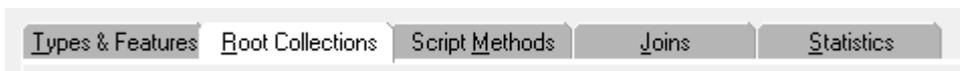
**Selected Features - Customer** 

| Alias      | Feature/Path | Data Type | Length | Decimals | Parent Class |
|------------|--------------|-----------|--------|----------|--------------|
| First Name | firstNames   | String    | 25     | 0        | Customer     |
| Last Name  | lastName     | String    | 15     | 0        | Customer     |

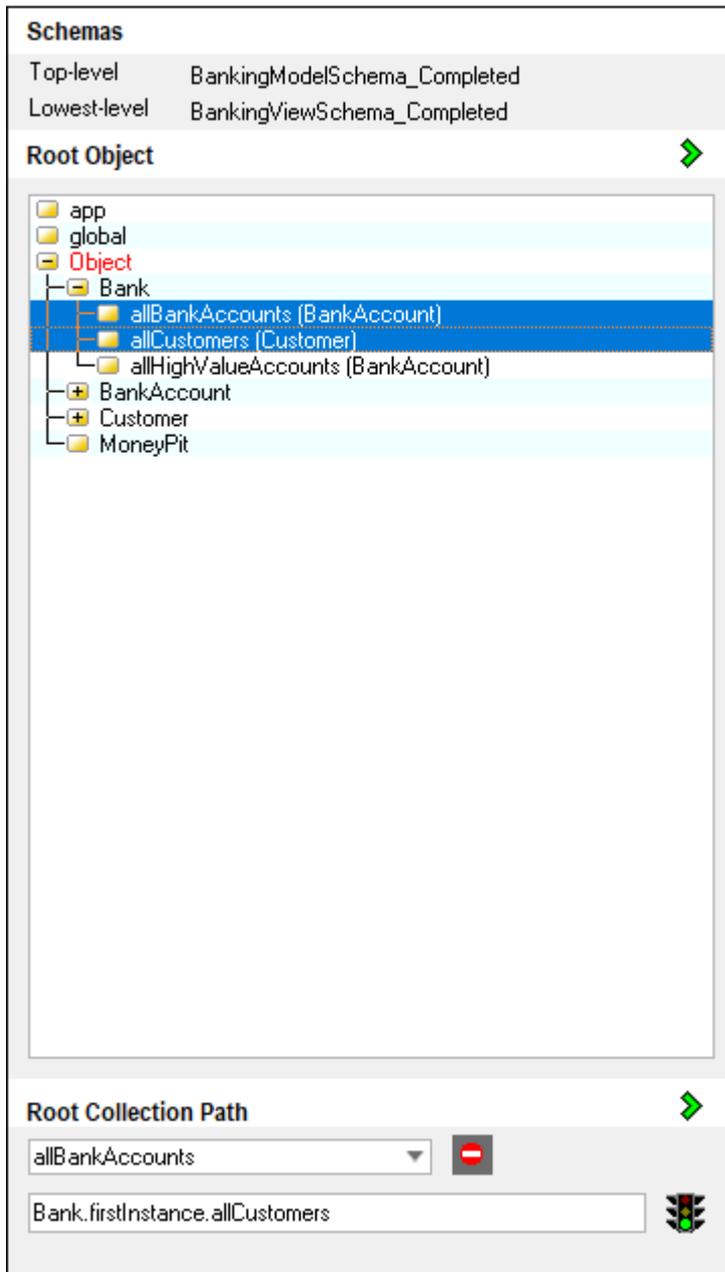
### Exercise 3 – Adding Root Collections

In this exercise, you will add a pair of root collections to the reporting view, to specify the scope of the report.

- Ensure the **Root Collections** sheet is selected.

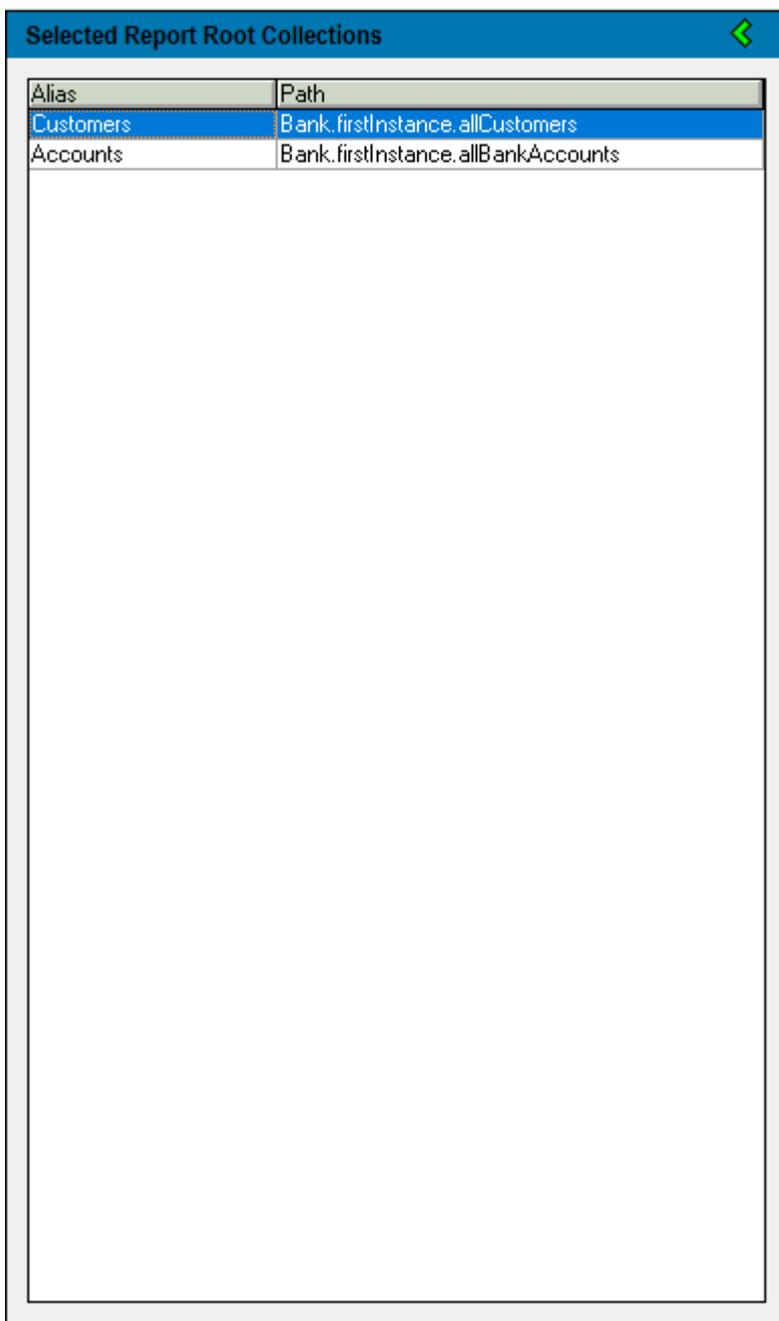


2. Select the **allBankAccounts** and **allCustomers** collections in the **Bank** class.



3. Click the green arrow button at the right of the header, to add them to the **Selected Report Root Collections** pane.
4. Change the alias of **allBankAccounts** to **Account**.
5. Change the alias of **allCustomers** to **Customers**.

6. The **Selected Report Root Collections** pane should appear as follows. (The order of **Customers** and **Accounts** in the table does not matter.)



| Alias     | Path                               |
|-----------|------------------------------------|
| Customers | Bank.firstInstance.allCustomers    |
| Accounts  | Bank.firstInstance.allBankAccounts |

7. Select the **Close** command from the View menu, and then close the JADE Report Configuration window.

## Creating a Report Design

The JADE Report Writer Designer application enables you to define the layout and content of reports based on the data provided by a reporting view of a JADE database. It provides a graphical interface, allowing you to drag and drop text, data from the database, and graphical elements directly onto the report.

While both the JADE Report Configuration and the JADE Report Designer applications can be run directly from the JADE development environment, the JADE Report Designer application has limited functionality when run directly from the JADE development environment, that is, you will be able to create reports but not print, extract, or preview them.

To have full functionality, the JADE Report Designer application should be integrated into the user system from which it reports. The simplest way to do this is from a JadeScript method, as follows.

```
startDesigner();

vars
    rwManager : JadeReportWriterManager;
begin
    create rwManager transient;
    rwManager.startReportWriterDesigner("a user name", null);
epilog
    delete rwManager;
end;
```

You can start a new report from the Welcome dialog or from the **New Report** command on the File menu of the JADE Report Designer window.

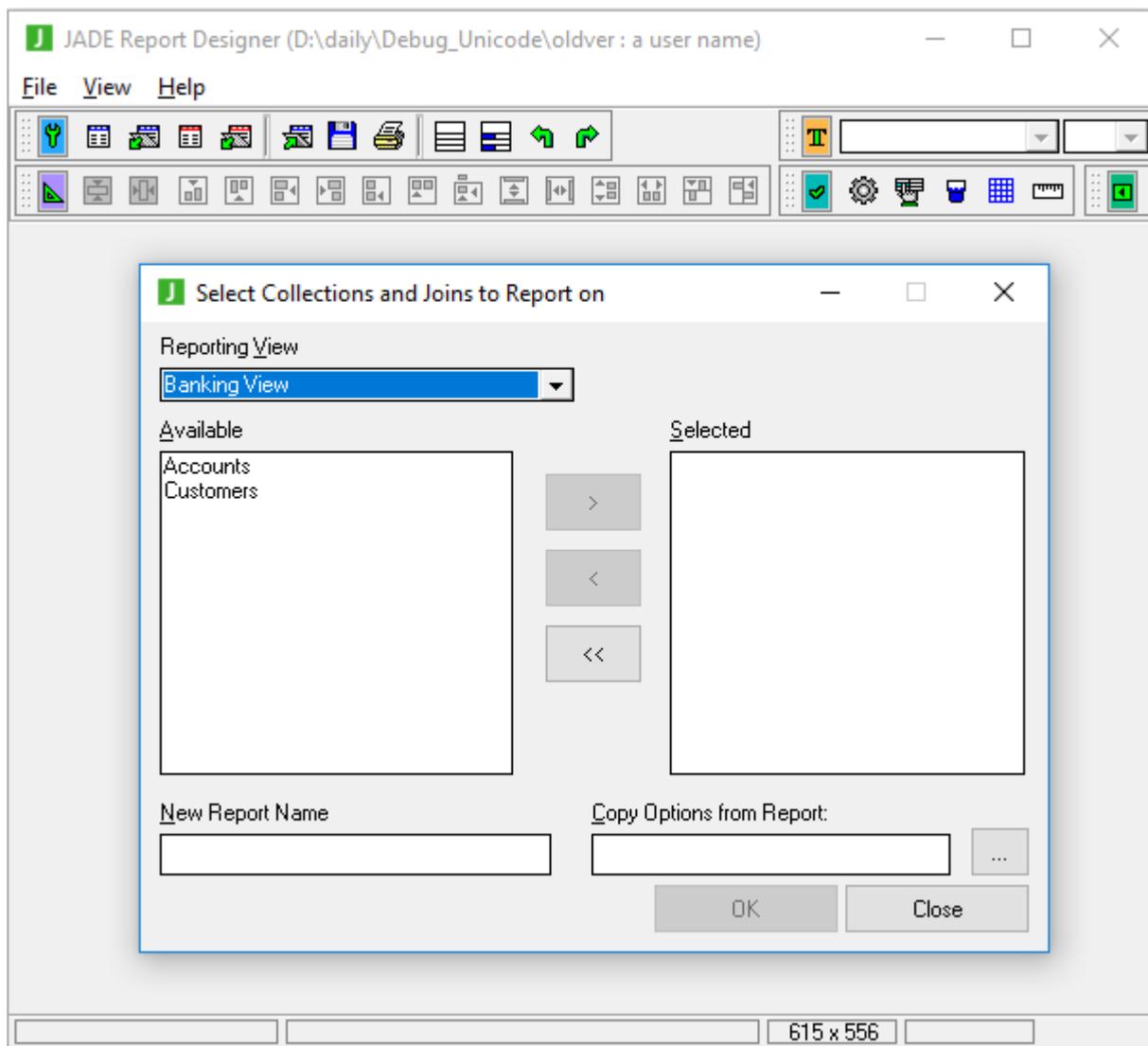
The Welcome dialog is then displayed. Click the **New Report** button at the left of the Welcome dialog and then click **OK** button.

If you want to apply properties from a previously created report, select an existing report in the **Open a Report** list box and then click the **Open a Report** button at the left of the dialog, before clicking **OK**.

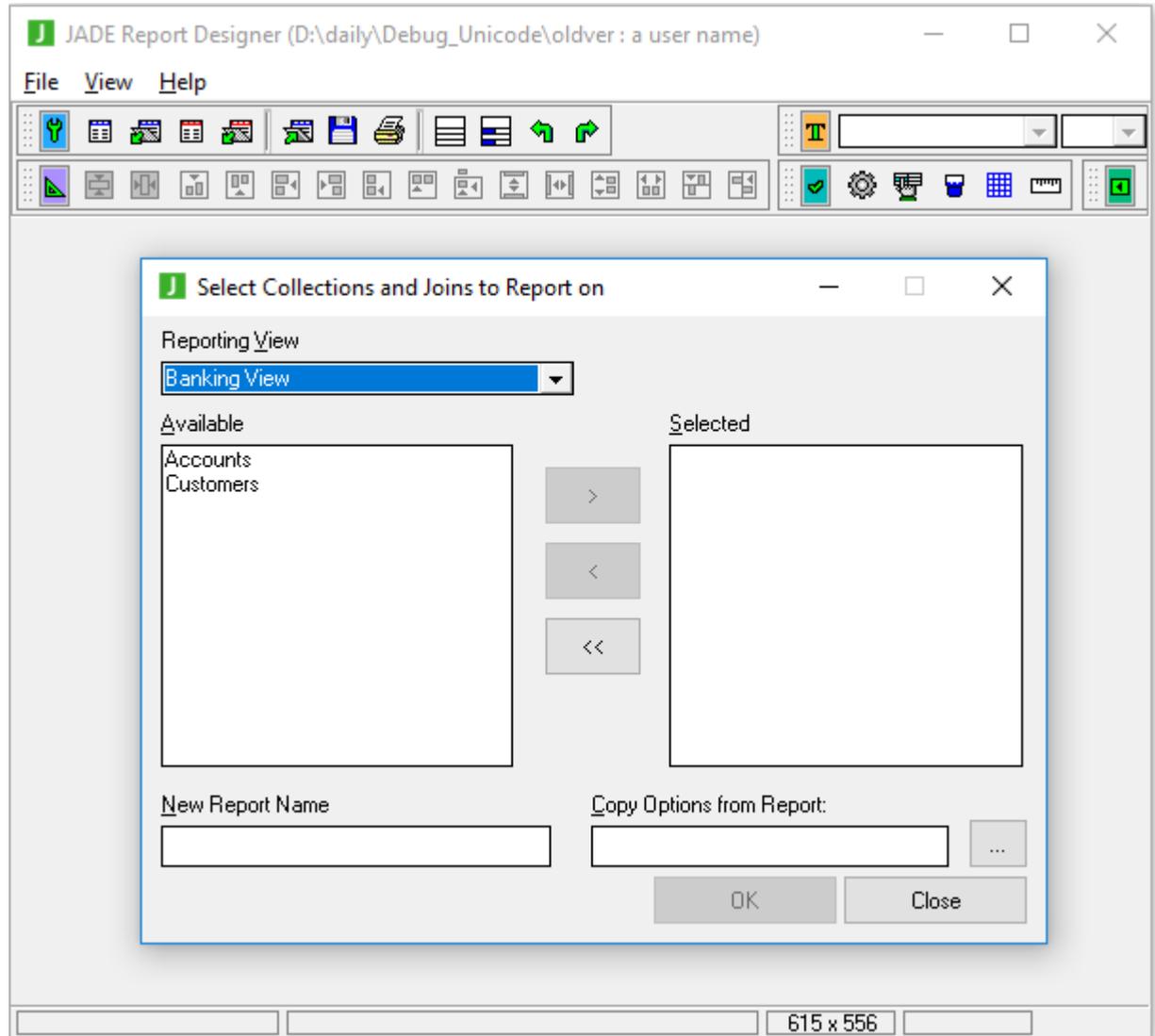
To create a report from within the JADE Report Writer Designer application, use any of the following actions.

- Select the **New Report** command in the File menu.
- Click **New Report** toolbar button.
- Press the Ctrl+N shortcut keys.

The Select Collections and Joins to Report on dialog is then displayed in the JADE Report Designer window.

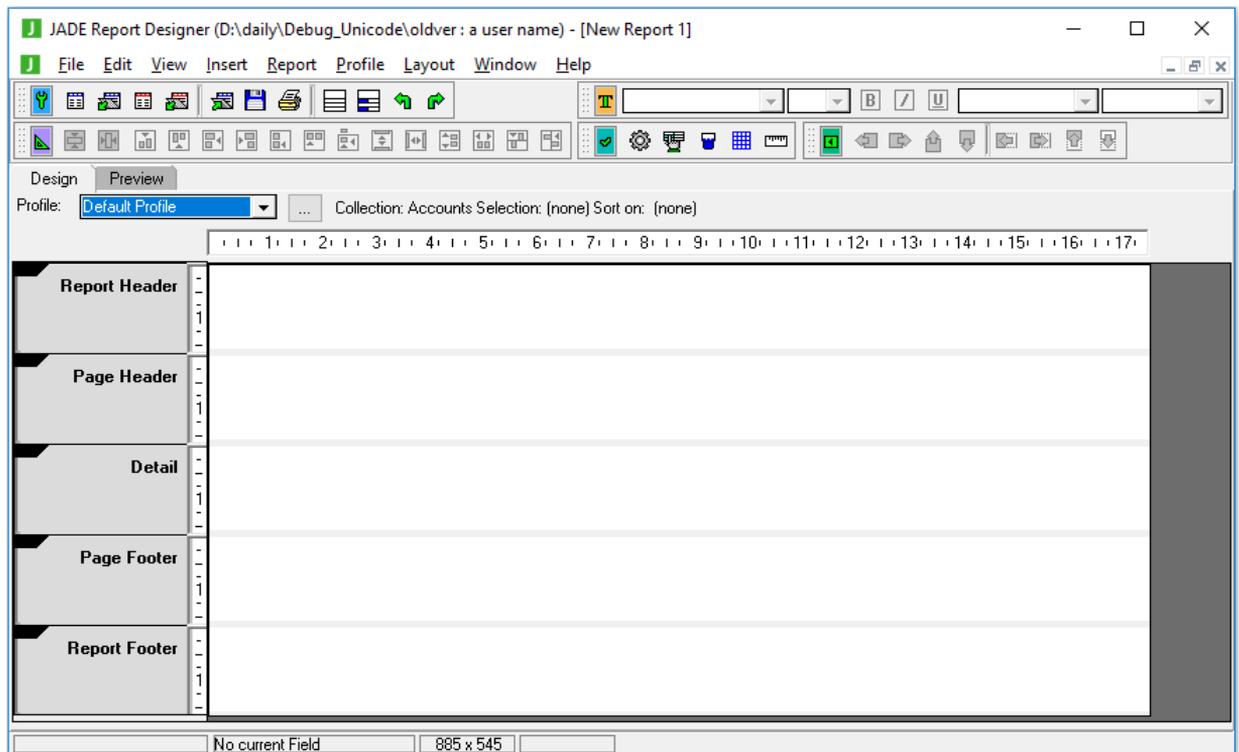


If you select a reporting view that has been previously defined, when you click **OK** to begin editing the report, you will need to specify whether to include any joins. (To not use any joins, uncheck the **Join Keys Required?** check box.)



**Note** While joins are a mainstay of relational databases, they are typically not required in JADE object-oriented databases if good object-oriented principles have been used in the design.

When the Select Collections and Joins to Report on dialog is closed, the Jade Report Designer window, which is the interface for creating reports, is displayed.



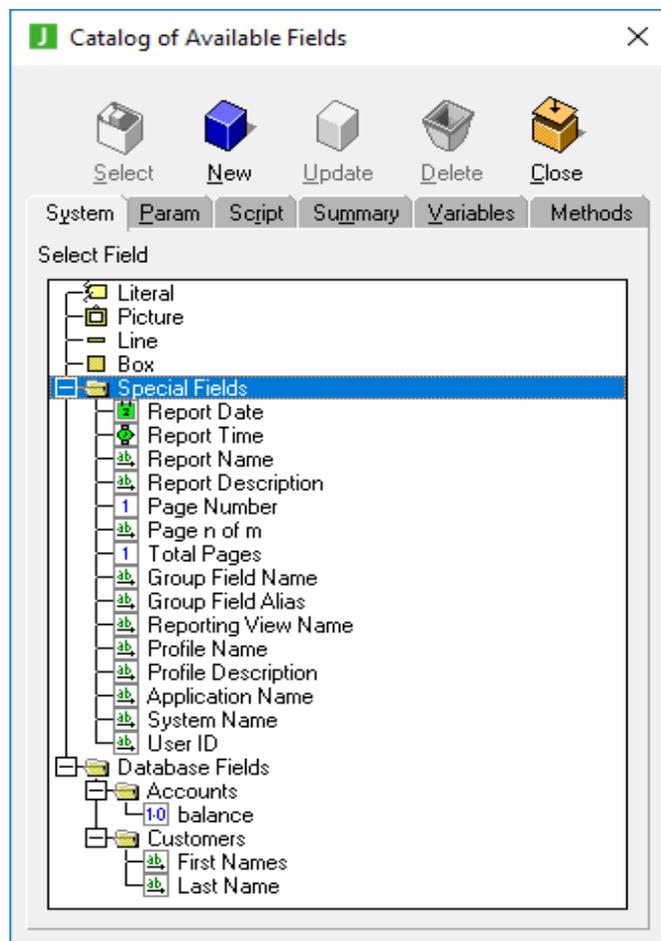
The Jade Report Designer window contains the following default sections.

| Section       | Description   |
|---------------|---|
| Report Header | The content in this section is displayed once, at the very beginning of the report. Use this section to print the title of the report and any other details you want to be displayed on the front of your report.   |
| Page Header   | The content in this section is displayed once on each page, at the top of each page. If the report header is not printed on its own page, the page header is printed below the report header on the first page of your report. Use this section to print details that you want to appear at the top of every page; for example, date, page number, or field headings. |
| Detail        | The content of this section is displayed for each of the input records provided by the specified root collection or collections. It fills up the space between the page headers and page footers until it presents all elements in the specified root collection or collections, using as many pages as necessary.  |
| Page Footer   | The content in this section is displayed once on each page, at the bottom of each page. Use this section to print details that you want to appear at the bottom of every page; for example, a report identification or date and page number if those are not displayed in the page headers.   |
| Report Footer | The content in this section is displayed once, at the very end of the report. Use this section to print information such as a grand total of numeric data or other static data that you want to print at the end of the report.   |

## Catalog of Available Fields

The Catalog of Available Fields dialog is used to insert fields into the report. You can open it by performing any of the following actions.

- Select the **Catalog** command from the View menu.
- Click the **Toggle Catalog** button on the **Quick Launch Tools** toolbar.
- Press the F6 shortcut key.



This is the major tool you should use to paint fields on your report layout. It also enables you to create, add, and update parameter fields, script fields, summary fields, and method fields and to display the usage of database field items, parameter fields, script fields, summary fields, and method fields.

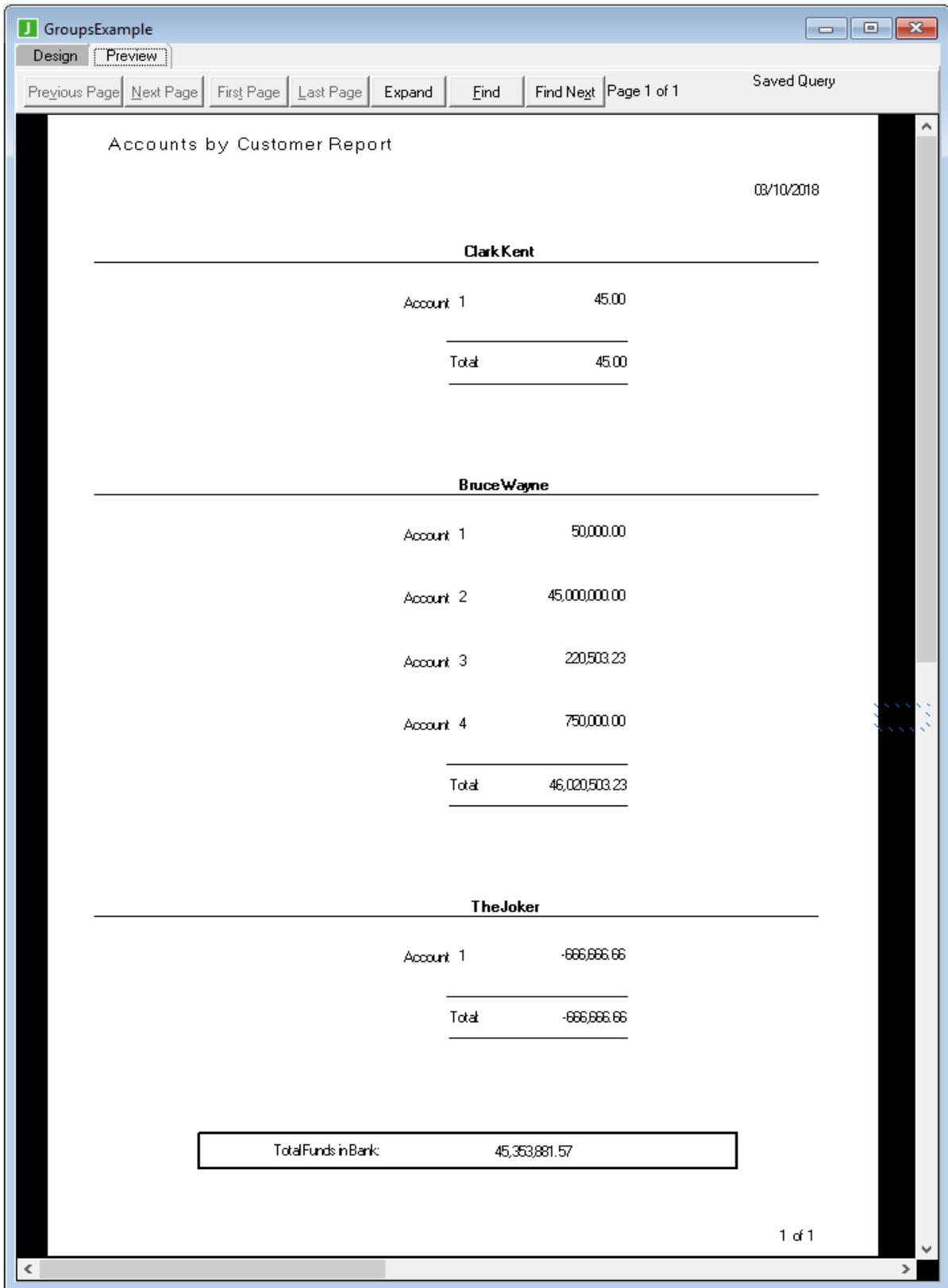
The catalog shows the elements that can be added to the report. These include **Literal** (static text that is not based on any database data), **Line** (a useful divider), **Special Fields** such as the current date and page numbers, and **Database Fields**.

To insert these fields into the report, click and drag them onto the report.

## Viewing the JADE Report

You can preview JADE Reports directly in the JADE Report Designer window, print to a printer or to PDF, or export to a variety of formats.

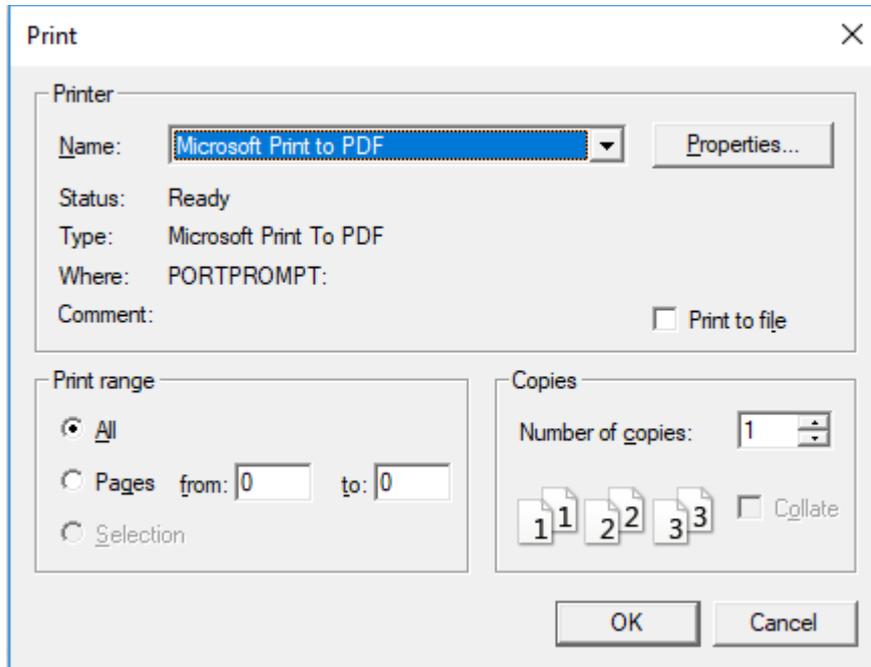
To preview a JADE Report from the JADE Report Designer window, simply select the **Preview** tab of the Jade Report Designer window. The output of the result is displayed in the current window.



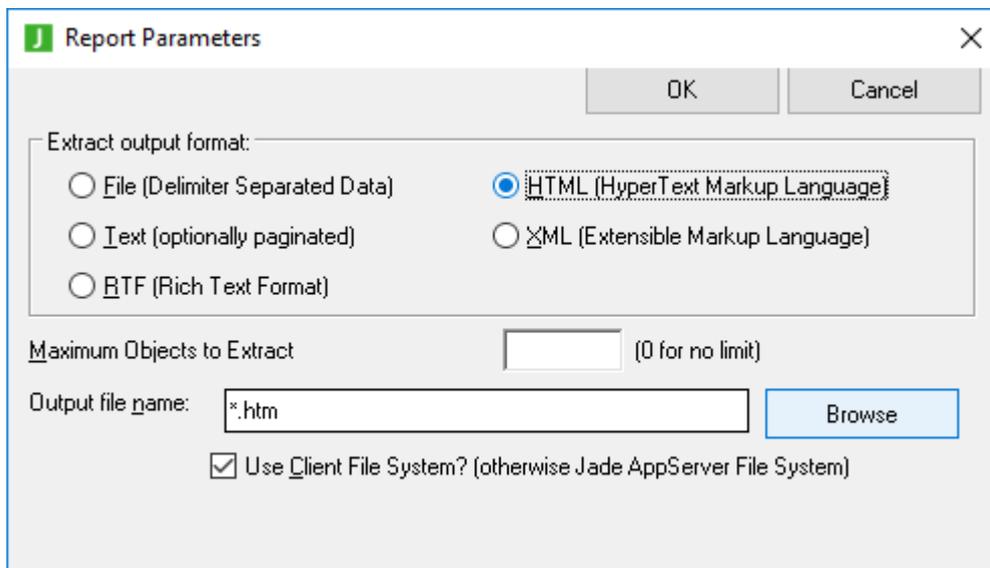
To print a JADE Report to printer or PDF file, open the print menu using one of the following actions.

- Select the **Print** option from the File menu.
- Click the **Print** toolbar button.
- Press the Ctrl+P shortcut keys.

The common Print dialog is then displayed, from which you can select where to print, along with the number of copies and the range of pages to print.



To export the report to another format (for example, a **.csv**, **.txt**, or **.htm** file), open the Report Parameters dialog by selecting the **Extract Data** command from the File menu.



From this dialog, you can select the output format from the following options.

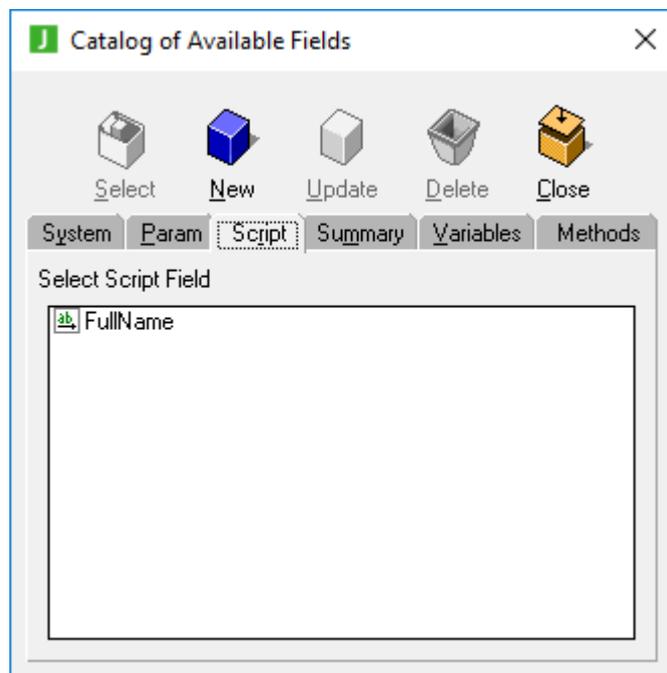
| Format                          | Extension | Description   |
|---------------------------------|-----------|---|
| File (Delimiter Separated Data) | .csv      | This format creates a Comma-Separated Values file (.csv) but does not require a comma specifically as the delimiter. You can select any symbol to use as a delimiter if commas are likely to be found in your data. |
| HTML                            | .htm      | This format applies HTML tags to your data to format it for display on the Internet as an .htm file.  |
| Text (optionally paginated)     | .txt      | This extracts your report as simple text. Any formatting is lost, and layout is approximated using spaces.  |
| XML                             | .xml      | This format applies XML tags to your data to format it for display on the Internet as an .xml file.   |
| RTF                             | .rtf      | This extracts your report as a Rich-Text Format file, which encodes formatted text and graphics in a universal format.  |

The **Maximum Objects to Extract** text box lets you set a limit on the size of the extraction for very large databases (for example, you may want to extract only the first ten thousand customers).

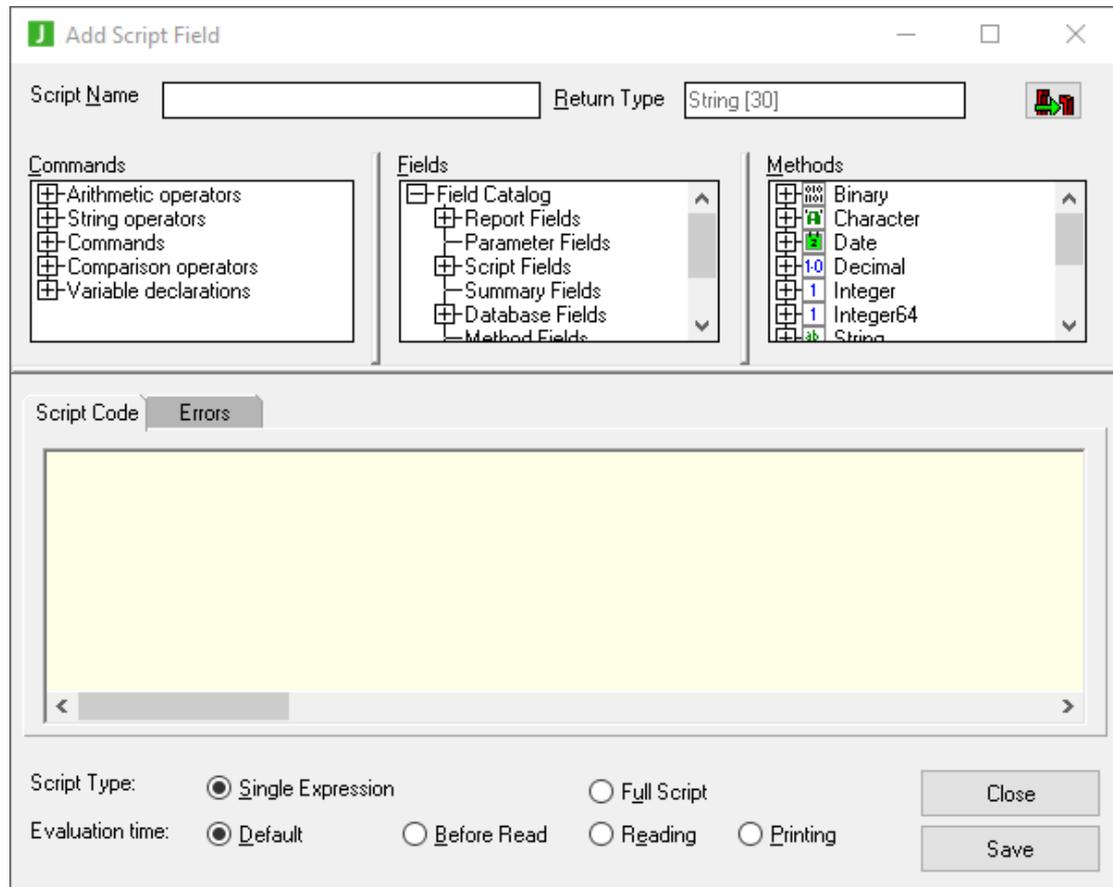
The **Use Client File System** check box is meaningful only when running the JADE Report Writer from a presentation (thin) client and that client is on a different file system to the application server. For single user and standard (fat) client systems, this will not be the case. However, if you want to extract a report from a remote presentation client to the file system of the application server, this check box enables you to do so.

## Report Scripts and Combining Data

When designing a report, it is often useful to create a reporting field that is composed of multiple database fields. To do this, the Catalog of Available Fields dialog has the **Script** sheet.



To add a new script to the Catalog, click the **New** button while the **Script** sheet is displayed. The **Add Script Field** dialog is then displayed.



While the **Full Script** option button for **Script Type** allows you to create powerful JadeScript methods directly from your report, for simple combining of database fields, it is faster and easier to use **Single Expression** option button.

The **Add Script Field** dialog combines the ability to write JADE syntax in the **Script Code** text box with easy-to-use GUI controls for users who are less familiar with JADE code.

---

**Tip** While the **Full Script** option button is selected in **Script Type**, the **Script Code** text box behaves like a JadeScript method.

---

## Exercise 4 – Creating a Report

In this exercise, you will create a report based on the banking system view created in the earlier exercises in this module.

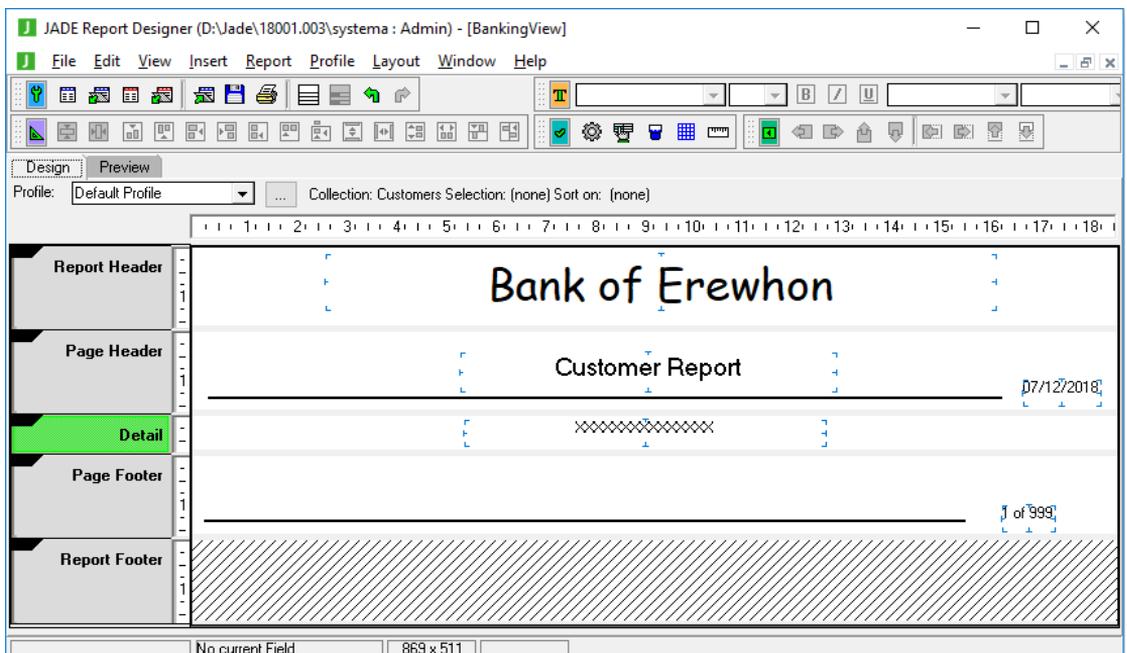
1. Add the following JadeScript method to the **BankingViewSchema**.

```

openDesigner();

vars
    rw : JadeReportWriterManager;
begin
    create rw transient;
    rw.startReportWriterDesigner("User", null);
epilog
    delete rw;
end;
    
```

2. Run the **openDesigner** JadeScript method and then select **New Report**.
3. Add **Customers** to the **Selected** list box, name the report **BankingView**, and then click **OK**.
4. Create the following form.



- a. Add String literals such as "Bank of Erewhon" or "Customer Report" by dragging a literal from the **System** sheet of the Catalog of Available Fields dialog to the report.
- b. Draw a line by dragging **Line** from the **System** sheet of the Catalog of Available Fields dialog to the report.  
The cursor will become a line-drawing cursor that you can click and drag across the report to draw lines.
- c. Add the current date to the report by dragging a **Report Date** from **Special Fields** on the **System** sheet of the Catalog of Available Fields dialog to the report.

- d. The "XXXXXXXXXXXXXXXXXXXX" element represents data from the database. Add data from the database by dragging one of the elements from **Database Fields** on the **System** sheet of the Catalog of Available Fields dialog.  
For this example, use **Last Name** in the **Customers** folder.
- e. Add the page number by dragging **Page n of m** from the **Special Fields** on the **System** sheet in the Catalog of Available Fields dialog.

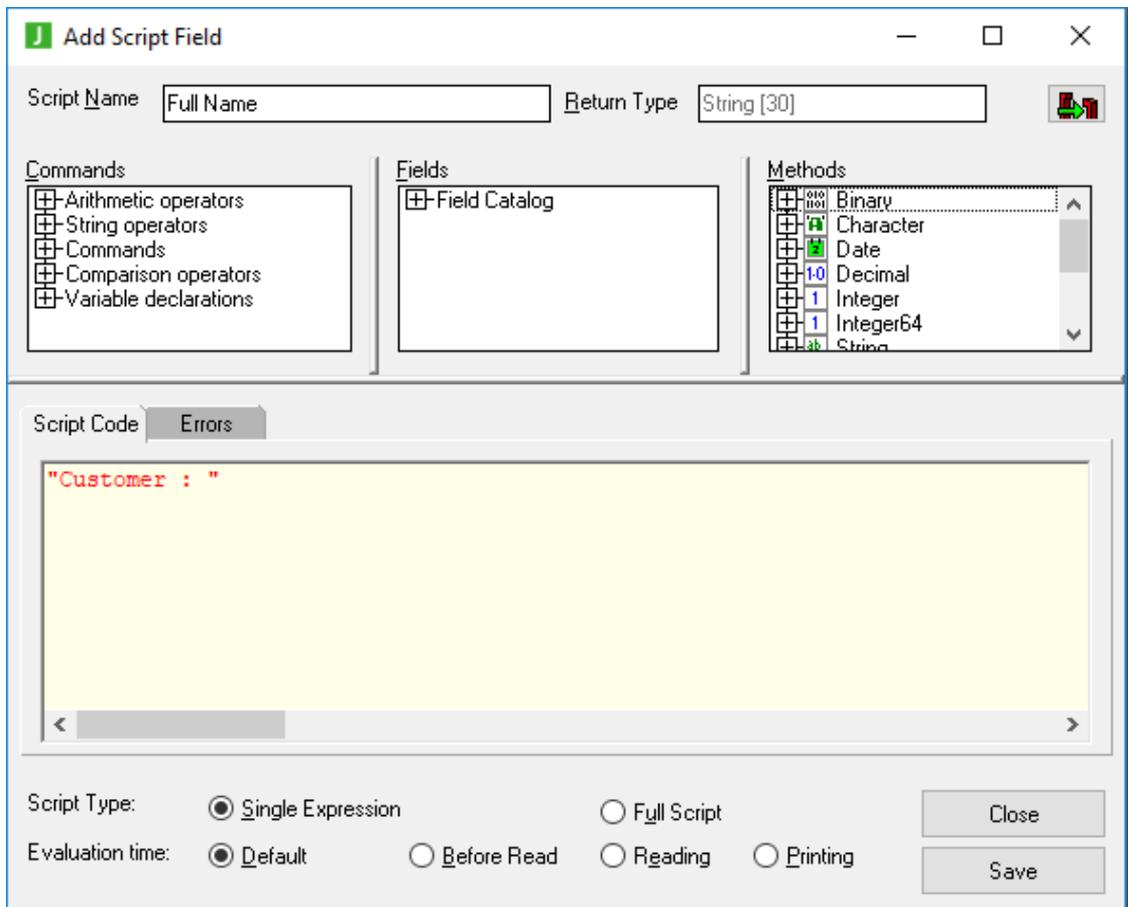
**Tip** If the Catalog of Available Fields dialog is not visible, enable it by pressing F6, clicking the **Toggle Catalog** button on the **Quick Launch Tools** toolbar, or by selecting the **Catalog** command from the View menu.

- 5. To view a preview of the report, select the Preview tab to display the **Preview** sheet of the Jade Report Designer window.

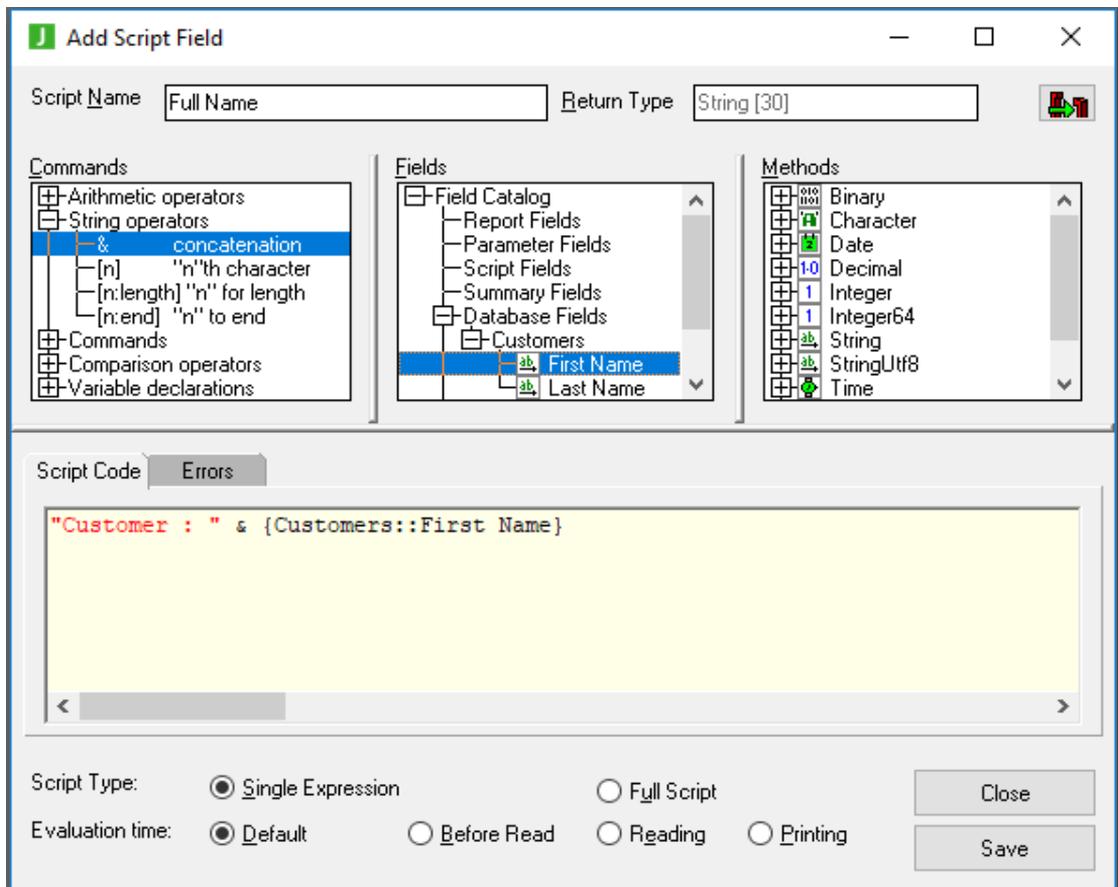
## Exercise 5 – Report Scripts

In this exercise, you will use a single expression script to create a composite of multiple database fields and a **String** literal to improve the details reported on each customer in the BankingView report. The goal is to output "**Customer : firstName lastName**".

- 1. Select the **Script** sheet of the Catalog of Available Fields dialog.
- 2. Click **New**, to create a new script.
- 3. Call the script **Full Name**.
- 4. In the **Script Code** text box, specify "**Customer :** ". Your form should look like the following.



5. Double-click the **& concatenation** option under the **String operators** entry in the list of **Commands**. This will add a **&** character to the script, which concatenates two strings together.
6. Double-click the **First Name** field in the **Customers** option within **Database Fields** in the list of **Fields**.



7. Add another string concatenation to the expression. You can select it from the **Commands** list or simply type **&**.

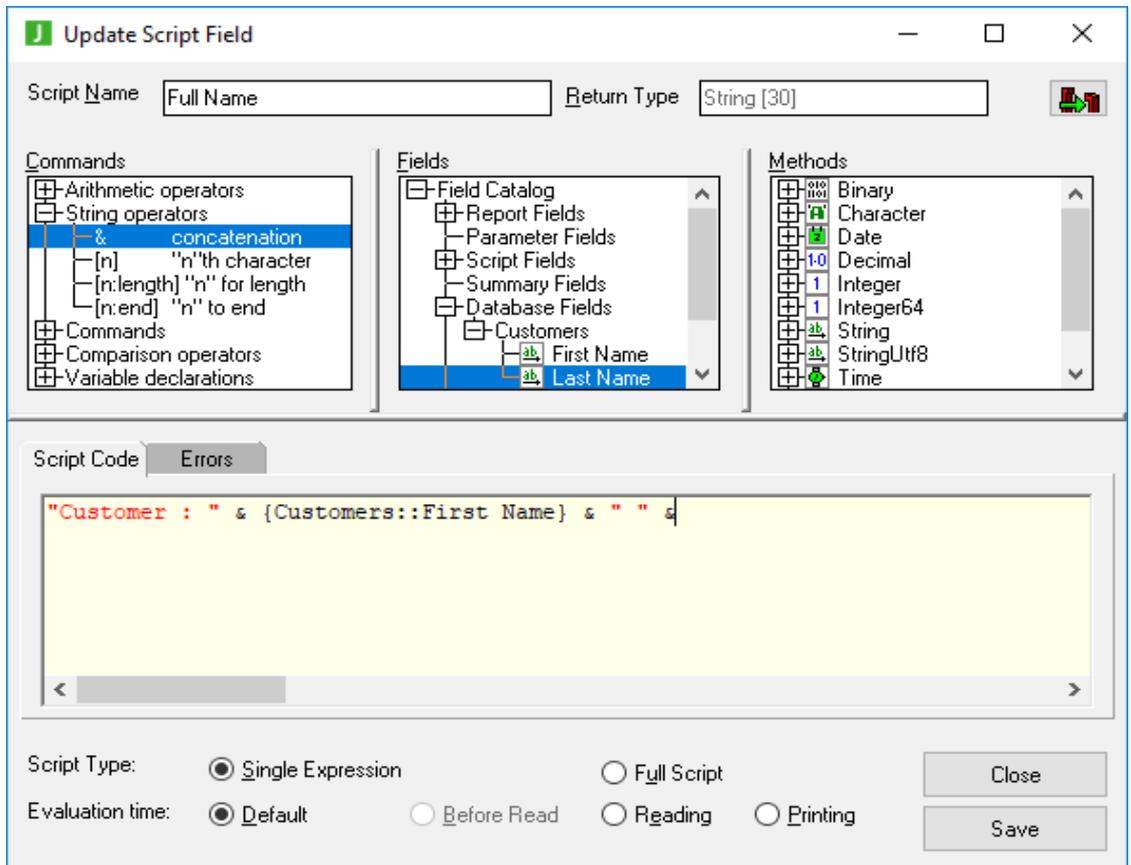
---

**Note** If you simply add **Last Name** now, **First Name** and **Last Name** will be concatenated together; for example, **"Customer : BarbaraBayton"**.

---

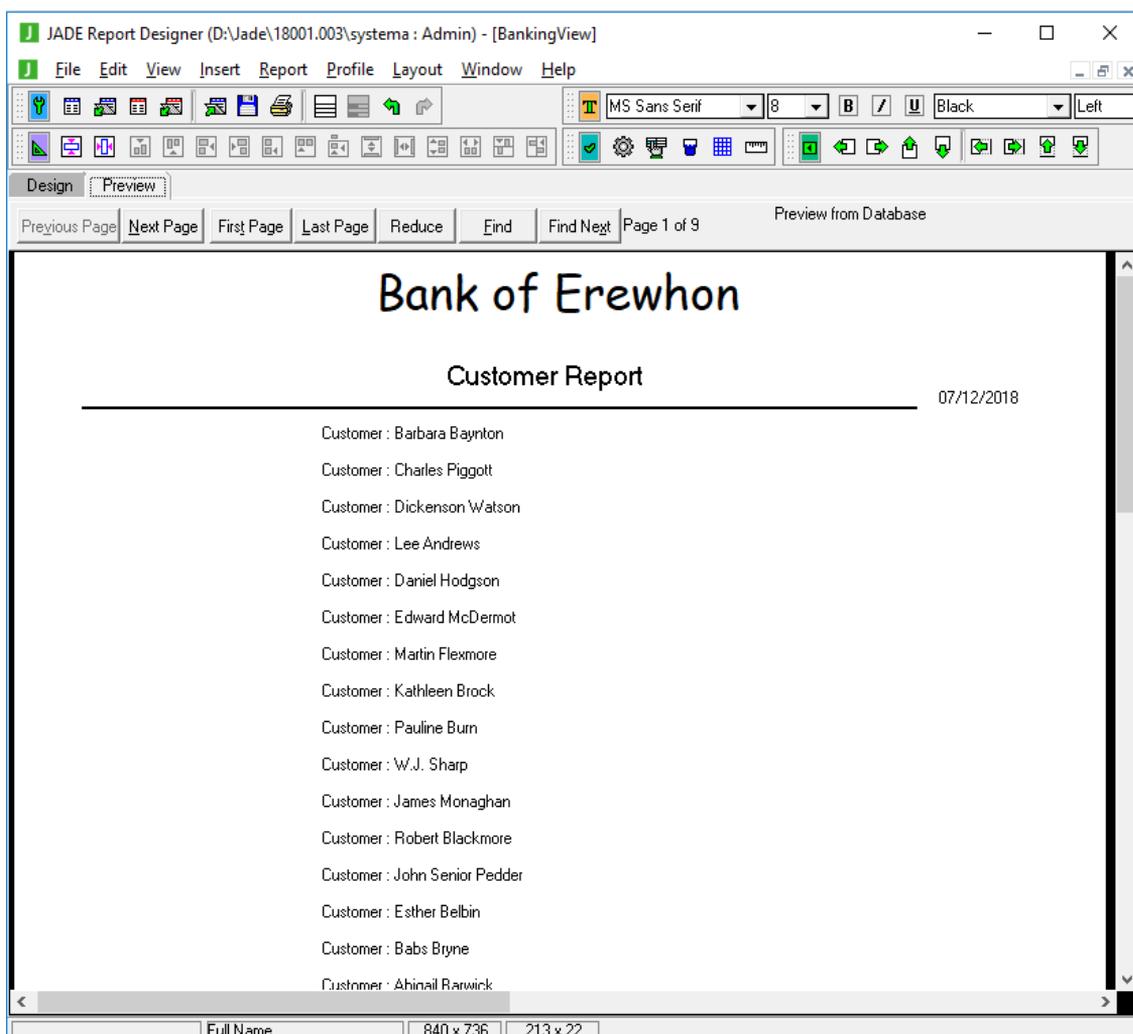
8. To add a space after the **First Name** and before the **Last Name**, type " " and another **&**.

The form should now look like the following.



- Finally, select **Last Name** from the **Fields** list and then click **Save**.

- To see your changes, select the Preview tab to display the **Preview** sheet of the Jade Report Designer window.



## Exercise 6 – Extracting a Report

In this exercise, you will extract the BankingView report to a variety of formats.

- Select the **Extract Data** command from the File menu.

The following dialog is then displayed.

**J** Report Parameters

OK Cancel

Extract output format:

File (Delimiter Separated Data)  HTML (HyperText Markup Language)

Text (optionally paginated)  XML (Extensible Markup Language)

RTF (Rich Text Format)

Maximum Objects to Extract  (0 for no limit)

Output file name:  Browse

Use Client File System? (otherwise Jade AppServer File System)

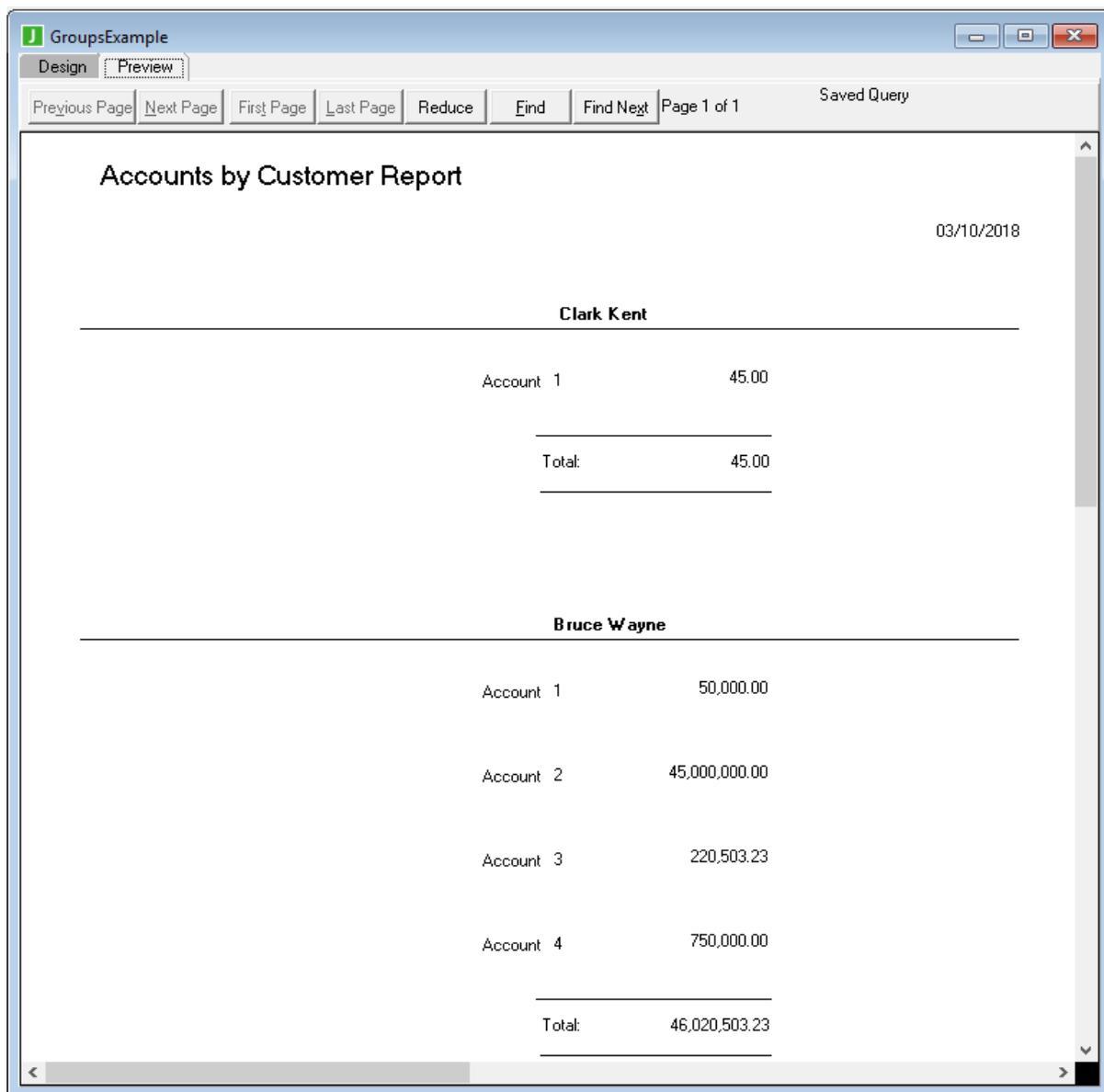
2. Select the **File (Delimiter Separated Data)** option button.
3. Click **Browse** at the right of the **Output file name** text box, to locate a convenient folder; for example, **C:\Reports**.
4. Ignore the **Maximum Objects to Extract** text box and **Use Client File System** check box.
5. Click **OK**, to extract the report as a CSV file.
6. Repeat steps 1 through 5 of this instruction for each of the available output formats.
7. View the generated files.

You will see that some look similar to the preview (for example, HTML output when viewed in a browser) and some different (for example, CSV output).

## Report Writer Groups

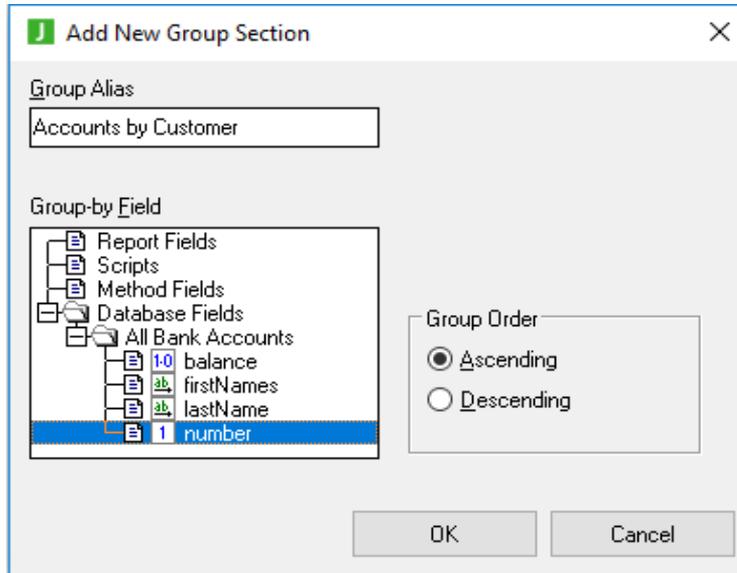
Report groups enable you to group data by subcategory and optionally produce summaries of the data within each group.

This can be useful for reporting on one-to-many relationships, grouping all data of the *many* property by a unique identifier of the *one*. For example, you can group all bank accounts by their customer's ID number, to display the bank accounts of each customer in the bank.

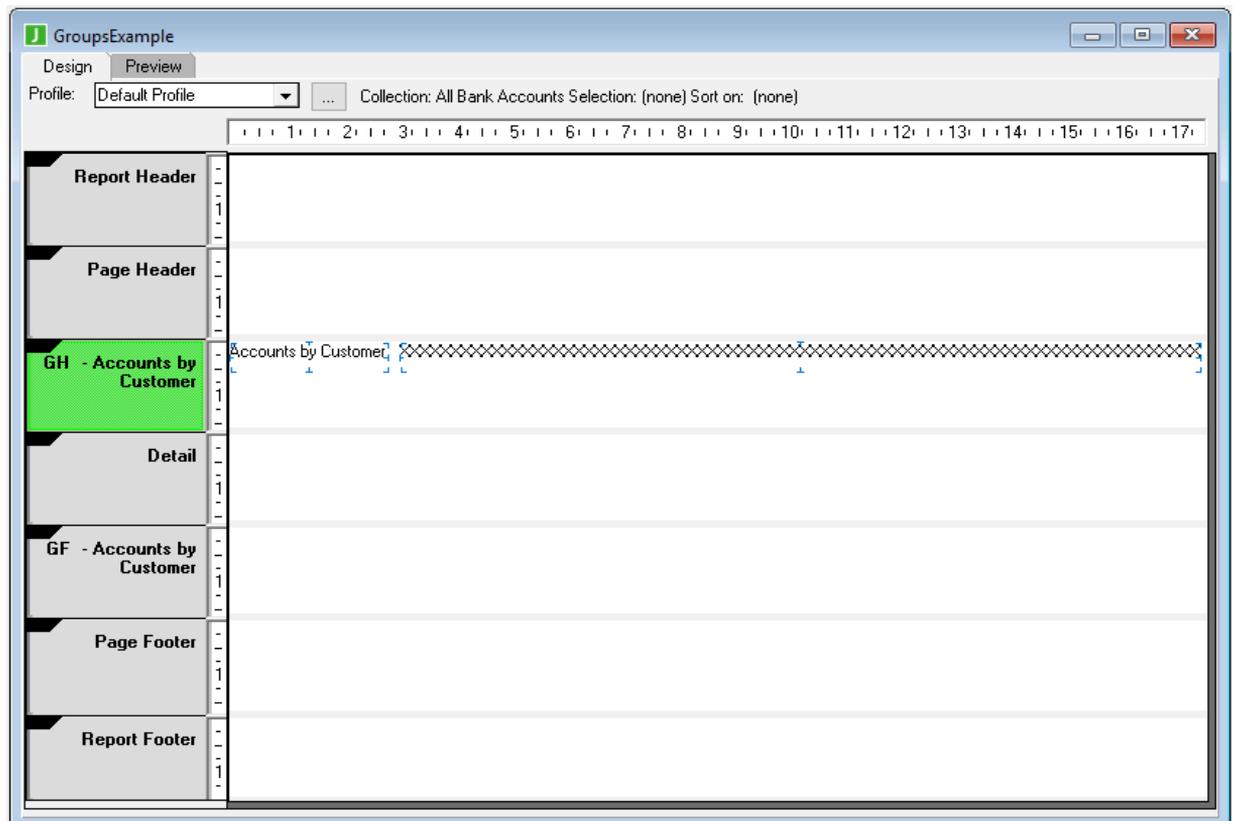


To create a group, select the **Group** command in the Insert menu.

The Add New Group Section dialog is then displayed.



You can name the group with an alias and select any of the features selected in the reporting view.

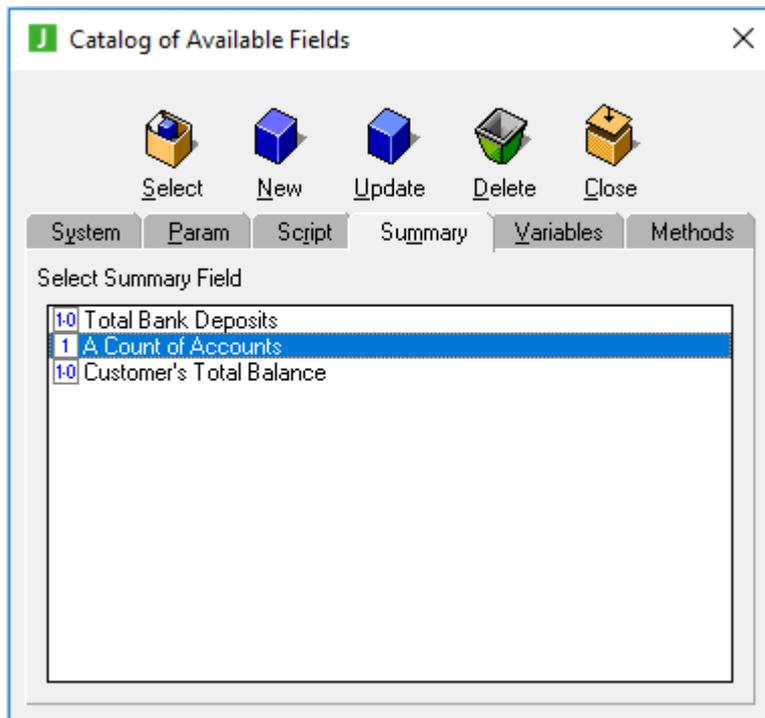


The report will surround the **Detail** section with new sections for the group - a group header above it and a group footer below it. By default, the header contains the alias of the group and the value of the selected feature for each group entry. You can modify this to meet your requirements; for example, if you want to put the name of the customer instead of the ID number, or to change the font. Although the group footer is empty, by default, it is however is a good place to put data summaries; for example, the total balance of the bank accounts of each customer.

In the detail section, you should put the data that you want shown in each grouping; for example, the balance of each account.

## The Summary Sheet

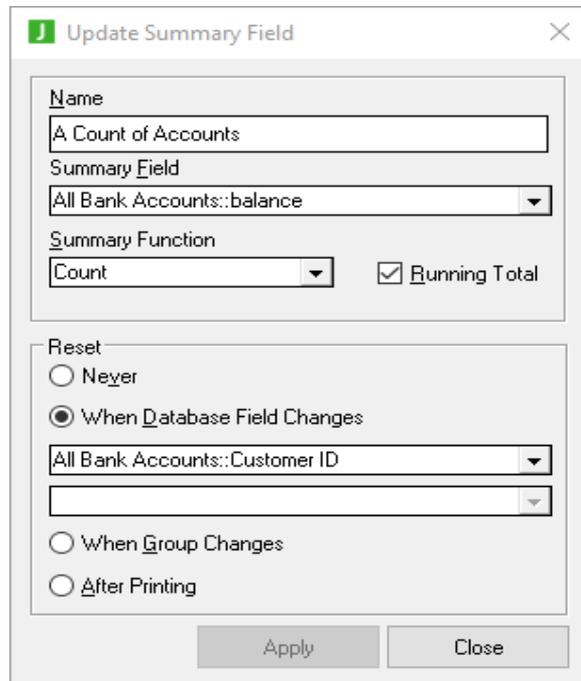
The **Summary** sheet of the Catalog of Available Fields dialog is used for producing totals and averages of sets of data. These summaries can be inserted into the report like any other element and can provide useful information such as the total balance of the accounts of a customer or a running total of how many accounts each customer has.



To create a new summary, use any of the following actions from the Catalog of Available Fields dialog.

- Click **New** button.
- Press the Alt+N shortcut keys.

The Add Summary Field dialog is then displayed, from which you can create the new summary.



The value of the **Summary Field** combo box is the feature of the collection objects to be summarized. Although this can be any feature, the entities available in the **Summary Functions** combo box depend on the type of the value selected in the **Summary Field** combo box.

The **Summary Function** is the manner in which the **Summary Field** is to be summarized, and can be one of the following values.

| Function         | Description  |
|------------------|--|
| Count            | Total number of instances within the collection of the selected field.   |
| Distinct Count   | Number of unique values amongst the instances within the collection of the selected field.   |
| Non-null Count   | Number of instances that are not null within the collection of the selected field.   |
| Max              | Highest value found amongst the instances within the collection of the selected field. This can be a numerical maximum or an alphabetical maximum, depending on the type of the field. |
| Min              | Lowest value found amongst the instances within the collection of the selected field. This can be a numerical minimum or an alphabetical minimum, depending on the type of the field.  |
| Sum              | Total of all the values of the target field within the collection. This can be applied only to numerical fields.   |
| Average          | Mean value of all the values of the target field within the collection. This can be applied only to numerical fields.  |
| Non-Zero Average | Mean value of all the values of the target field within the collection, with zero-values skipped. This can be applied only to numerical fields.  |

The **Running Total** check box is used if the summary needs to be updated continuously between each item of the collection. As this incurs a performance cost, you should not check this if you are presenting the summary only at the end of a group or at the end of the report.

The Reset group box in the lower half of the dialog is used to select when the summary is reset. For a summary of the whole report, select **Never** or **After Printing**. These values behave identically except when performing multiple print runs within the same JADE Report Writer Designer session, in which case the **After Printing** option button resets the total between print runs.

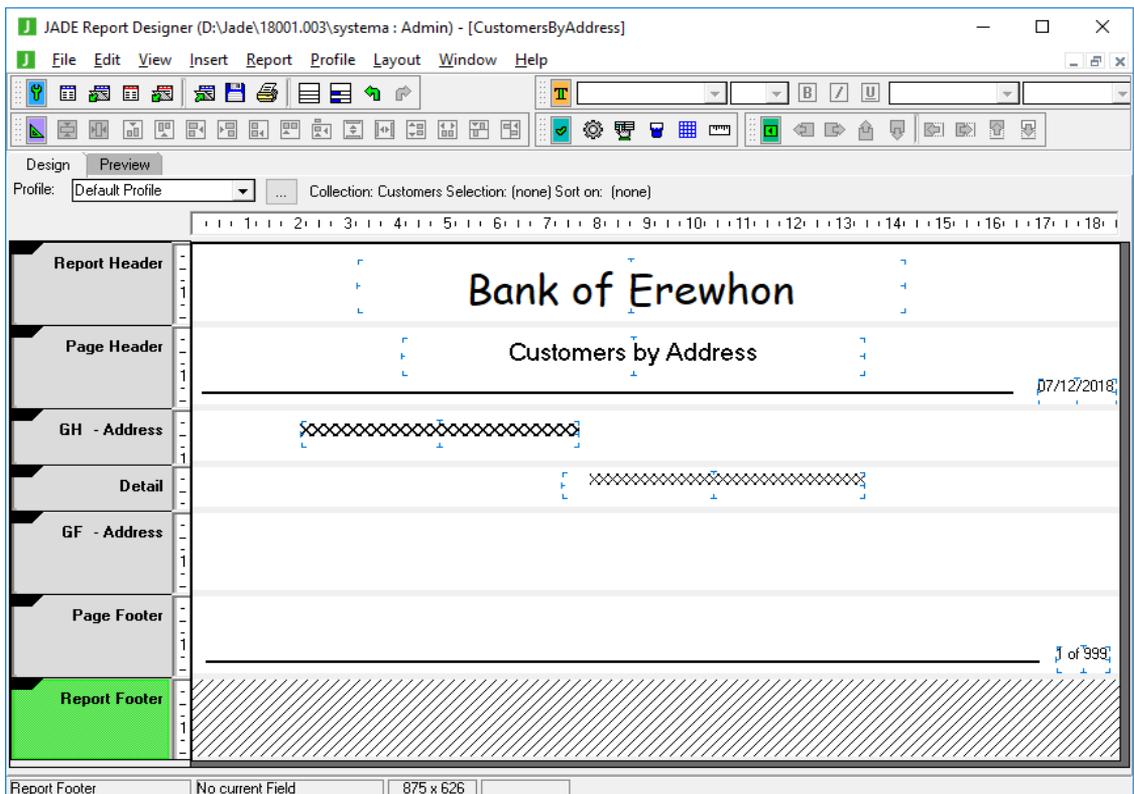
To create a summary for each group when using the **Groups** feature, select the **When Group Changes** option button, which resets the summary between each group of the collection.

You can also specify the summary to reset when a specific database field changes, by selecting the **When Database Field Changes** option button.

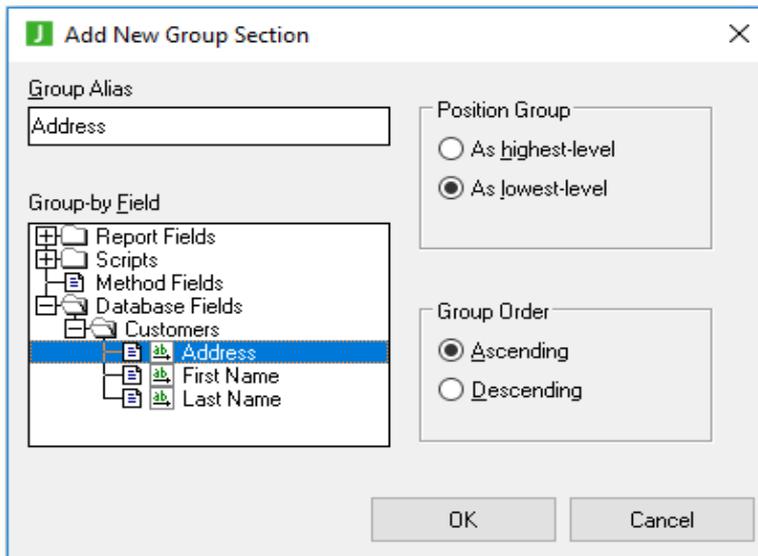
## Exercise 7 – Grouping Data

In this exercise, you will design a new report of all **Customers**, grouped by **Address**.

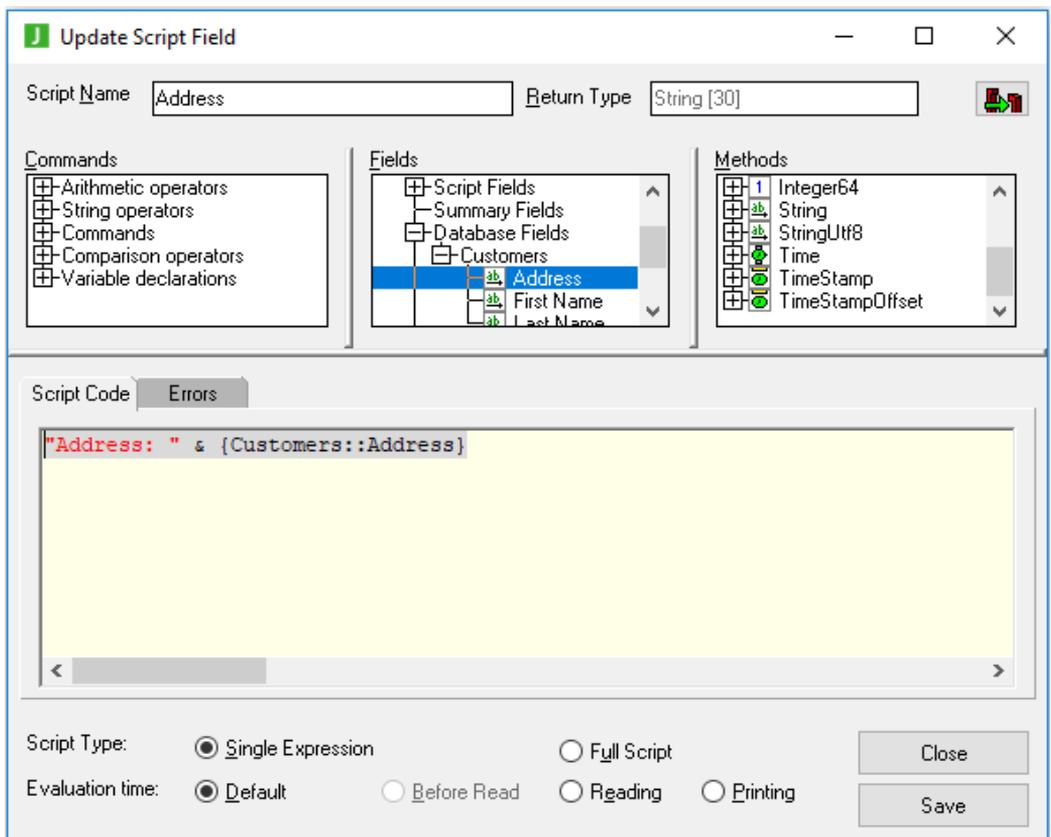
1. Open the JADE Report Configuration application by running the JadeScript **startConfiguration** method.
2. Select the **Open** command from the **View** menu.
3. Open the **BankingView** configuration.
4. In the **Customer** class, add the **address** feature and change its alias to **Address**.
5. Close the JADE Report Configuration window and then open the JADE Report Designer window (by running the JadeScript **startDesigner** method).
6. Create a new report called **CustomersByAddress**, with **Customers** selected.



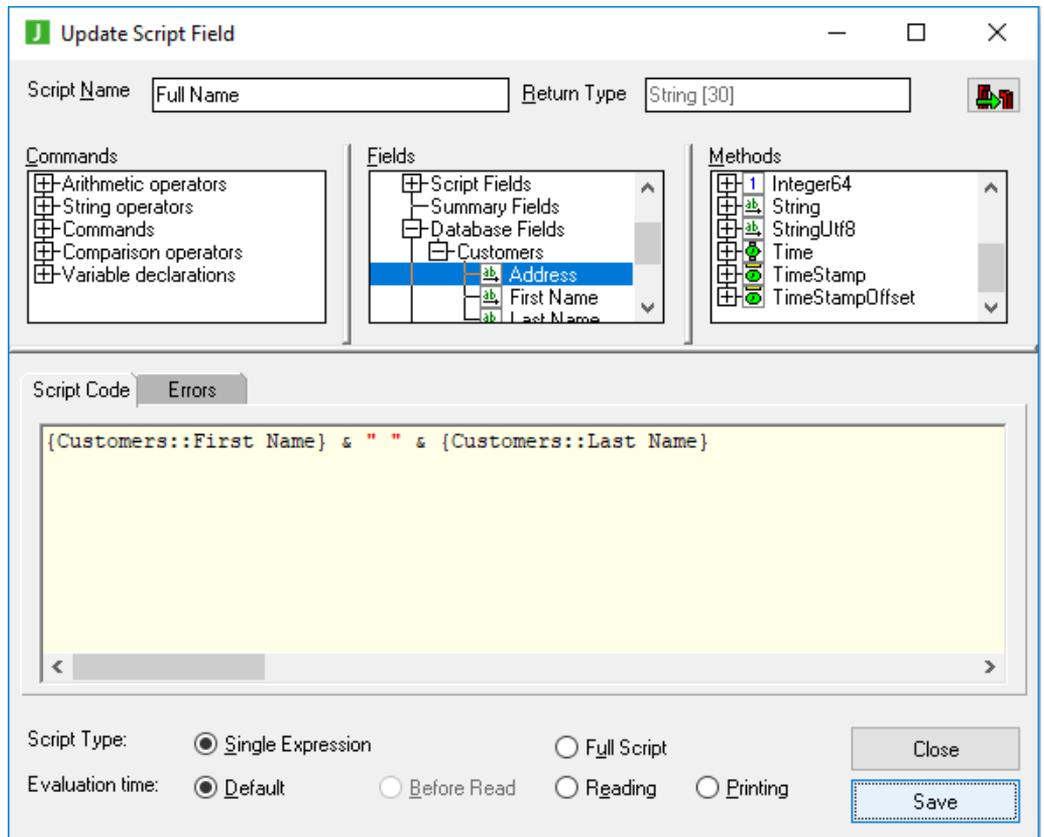
7. Design the report, as follows.
  - a. The **GH - Address** and **GF - Address** are the header and footer for the **Address** grouping. To add the **Address** grouping, select the **Group** command from the Insert menu and then fill it out as shown in the following dialog.



- b. By default, the **Group Header** contains a String literal of the name of the field (**Address**, in our case) and the selected **Database Field** value. These should be replaced by a **Single Expression Script Field**, to format the data more appropriately.



- c. The **Detail** section should contain the following **Single Expression Script Field** value.



8. To view the report, select the **Preview** tab of the Jade Report Designer window.

## Exercise 8 – Summarizing Data

In this exercise, you will use the **Summary** feature to count the number of customers at each address.

1. Select the **Summary** tab on the Catalog of Available Fields dialog, to display the **Summary** sheet.
2. Add a new summary by clicking the **New** button when the **Summary** sheet is displayed.

- Fill out the form, as follows.

**Update Summary Field**

Name: Total Customers in Address

Summary Field: Customers::Address

Summary Function: Count  Running Total

Reset:

- Never
- When Database Field Changes
- When Group Changes
- After Printing

Group #1 [Address]: Group #1 [Address]

Apply Close

**Note** We could add this **Summary** field directly to the report. However, it can also be used within a **Single Expression Script Field** as a convenient way to include a label.

- Create a new script, as follows.

**Update Script Field**

Script Name: Total Customers Return Type: String [60]

Commands:

- Arithmetic operators
- String operators
- Commands
- Comparison operators
- Variable declarations

Fields:

- Parameter Fields
- Script Fields
- Summary Fields: Total Customers in Address
- Database Fields

Methods:

- Binary
- Character
- Date
- Decimal
- Integer
- Integer64
- String

Script Code: "Total Customers in " & {Customers::Address} & ": " & {!Total Customers in Address}.String

Script Type:  Single Expression  Full Script

Evaluation time:  Default  Before Read  Reading  Printing

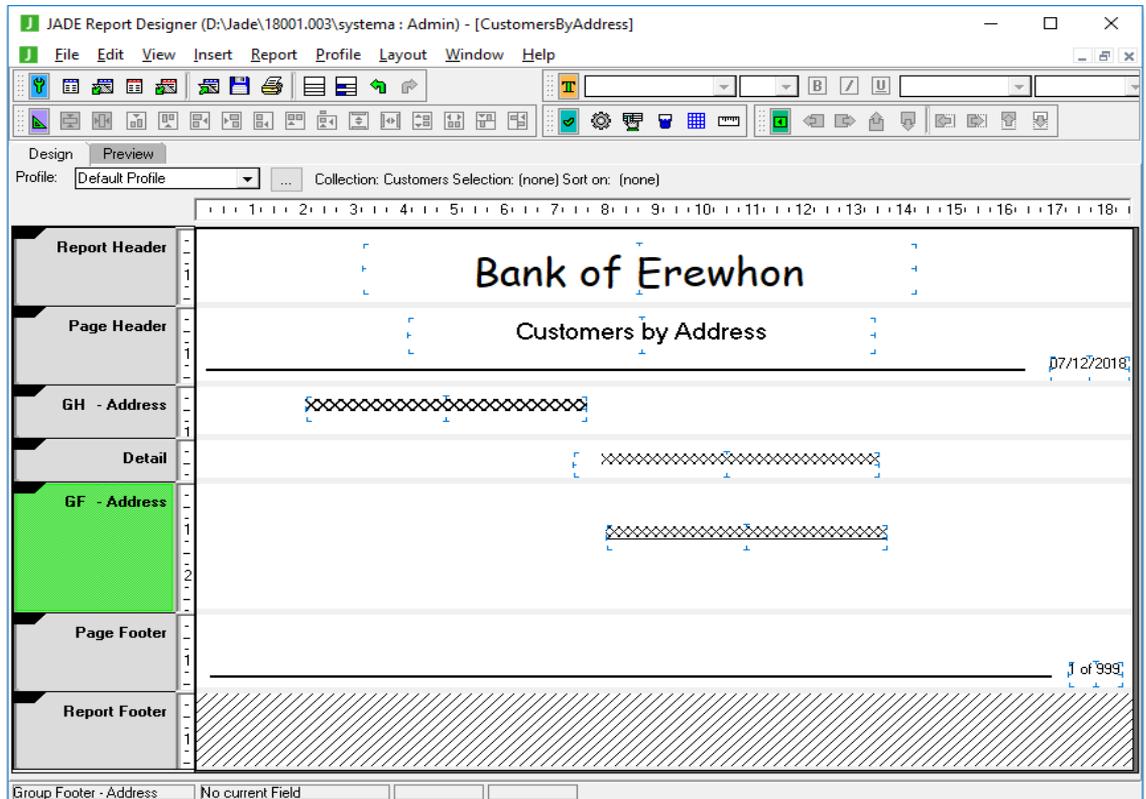
Close Save

**Note** The value in the **Return Type** text box has been changed from **String[30]** to **String[60]**, as the results of this script may be longer than 30 characters.

To modify the return type, click the button at the right of the **Return Type** text box.

The **Summary** field has also been cast to a String type, by appending **.String** at the end.

5. Add this new script to the report in **GF - Address** and set it to underlined.



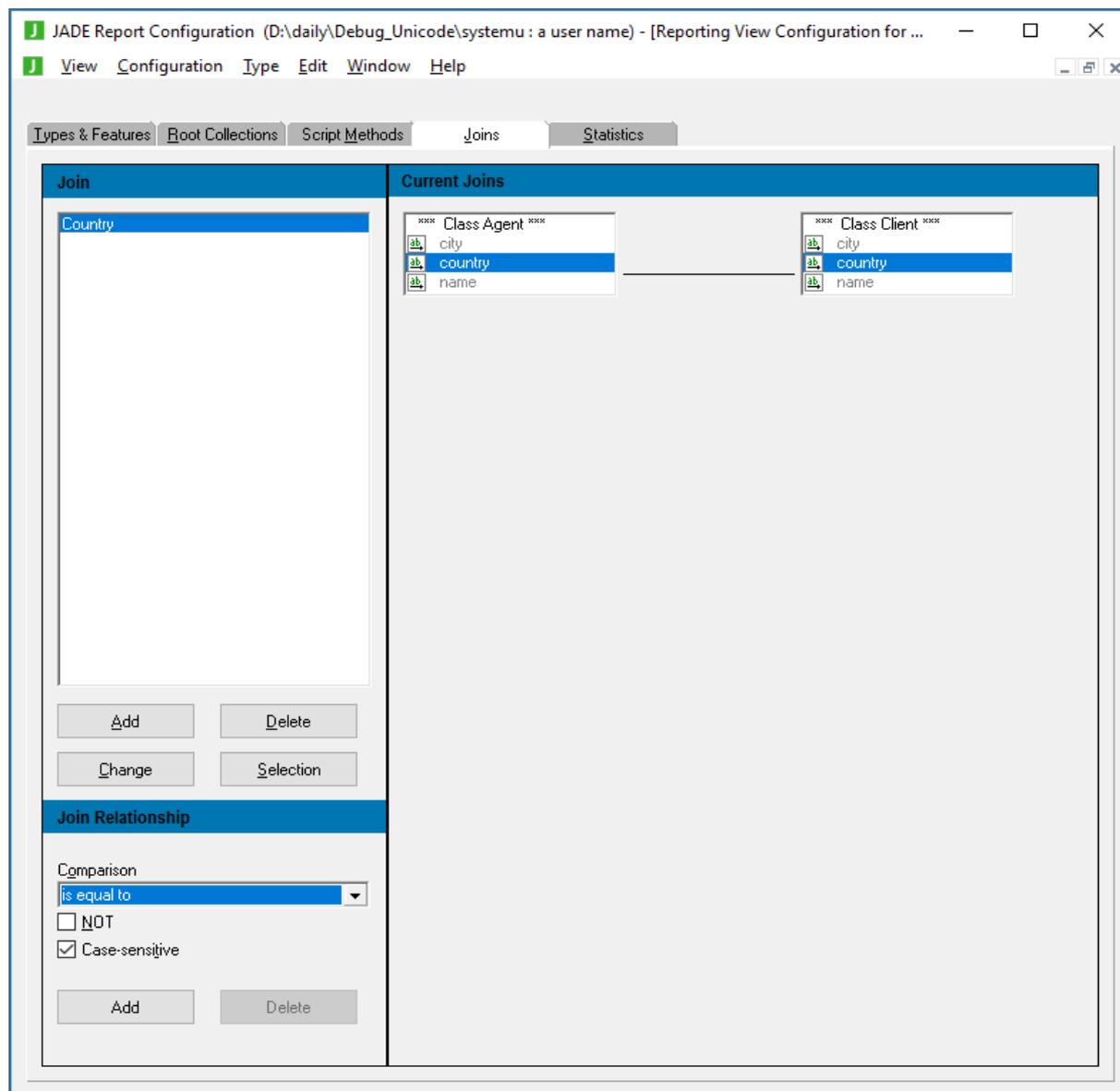
6. Select the **Preview** tab of the Jade Report Designer window, to view the report.

## Root Collection Joins

You can use the JADE Report Writer joins functionality to report on data that exists in more than one root collection. Although the use of joins is a common technique when working with relational databases, a well-designed JADE system can normally use JADE's relationship protocols to remove the need for collection joins. The most usual use case for joins in a JADE database is therefore when it has been converted from a relational database.

When joining two collections, they must have at least one property that contains common values; for example, a **Salesperson** collection and a **Customer** collection for an international company can both have a **country** property. If there was no **Country** class defined in the database, you could instead use a join to find which salespeople and customers live in the same country.

To specify a join from the JADE Report Configuration application, you must have defined at least two root collections for the reporting view, and of the objects contained in those collections, you must have defined at least one property that is common to both.



In this example, **Agents** and **Clients** in the Erewhon system both have a **country** property. To create a collection containing the join of clients and agents by country, the **country** property in both classes has been selected, and the **Join Relationship** has been set to **is equal to**.

The resulting collection contains only the agents that share a country with a client and the clients that share the country with the agent.

By referencing the client and agent names in the **Detail** section of a report, we can generate a list of all country matches.

| Erewhon Investments Ltd.   | 04/10/2018 |
|--|------------|
| <b>Match 1</b>   |            |
| The Agent Angela Bettersfield and the Client Christine Montgomery both live in England |            |
| <b>Match 2</b>   |            |
| The Agent Angela Bettersfield and the Client Howard Ellis both live in England         |            |
| <b>Match 3</b>   |            |
| The Agent Angela Bettersfield and the Client Peter Morrissey both live in England      |            |
| <b>Match 4</b>   |            |
| The Agent Airl Hamada and the Client Elaine Lee both live in Japan                     |            |
| <b>Match 5</b>   |            |
| The Agent Peter Smallsmith and the Client Pauline Wild both live in New Zealand        |            |
|     | 1 of 3     |

## Setting Security Options

When designing reports in the JADE Report Writer, it is often necessary to consider which users should have access to which reports and put security measures in place to enforce access rules. By default, all users have full access to all reports.

To allow for the definition of access rules, the **JadeReportWriterSecurity** class of RootSchema provides the following methods.

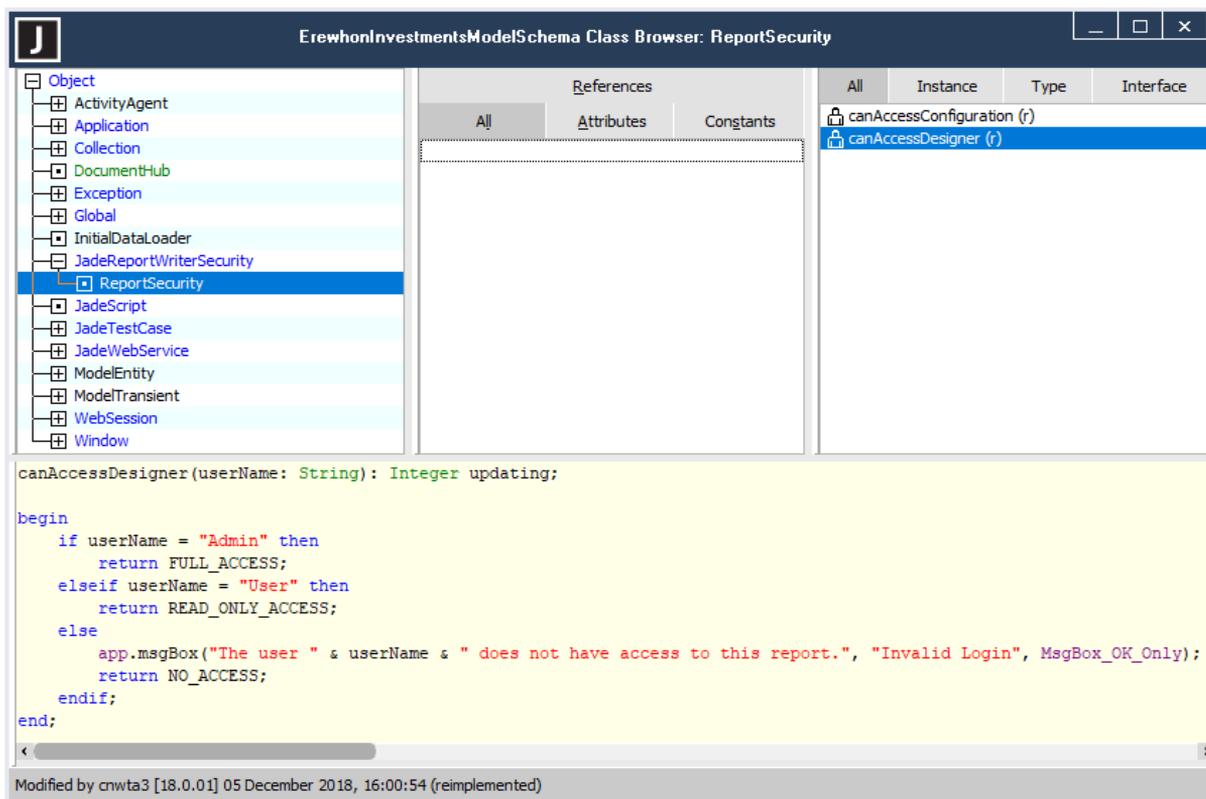
| Method                   | Description  |
|--------------------------|--|
| canAccessConfiguration   | Returns the type of access that the specified user has to the configuration application.   |
| canAccessDesigner        | Returns the type of access that the specified user has to the designer application.  |
| canAccessFolder          | Returns the type of access that the specified user has to the specified folder in the configuration application.   |
| canAccessReport          | Returns the type of access that the specified user has to the specified report when reports are listed for selection.  |
| canAccessView            | Returns the type of access that the specified user has to the specified view when reports are listed for selection or extraction.  |
| canAccessViewClass       | Controls visibility of view classes in the designer application.   |
| canAccessViewFeature     | Controls visibility of view features in the designer application.  |
| canDeleteReport          | Controls which reports the user can delete.  |
| canMaintainFolders       | Returns the type of access that the specified user has to folders in the configuration application.  |
| canMaintainSystemOptions | Returns the type of access that the specified user has to system options in the configuration application.   |
| canMaintainViews         | Returns the type of access that the specified user has to views in the configuration application.  |
| folderDeleted            | Called when a folder is deleted, with the specified folder path in the same format as that of the <b>newFolderAdded</b> method. Enables synchronization of user security details when a folder is deleted.   |
| folderPathChanged        | Called when a folder path is changed, with the specified folder paths in the same formats as that of the <b>newFolderAdded</b> method. Enables synchronization of user security details when a folder name is changed or the folder is moved to another parent folder. |
| isViewFeatureAccessSet   | Specifies whether the user can access the view.  |
| newFolderAdded           | Enables the user who created a new report to access that report when security is set.  |
| newReportAdded           | Enables the user who created a new report to access that report when security is set.  |

| Method            | Description   |
|-------------------|---|
| newViewAdded      | Enables the user who created a new view to access that view when security is set.                                     |
| reportDeleted     | Called when a report is deleted. Enables synchronization of user security details when a report is deleted.           |
| reportNameChanged | Called when a report name is changed. Enables synchronization of user security details when a report name is changed. |
| viewDeleted       | Called when a view is deleted. Enables synchronization of user security details when a view is deleted.               |
| viewNameChanged   | Called when a view name is changed.   |

These methods will return one of the following constants, as defined on the JadeReportWriterSecurity class.

| Class Constant   | Value | Description  |
|------------------|-------|--|
| FULL_ACCESS      | 2     | Allows full access to the report for definition and use                  |
| NO_ACCESS        | 0     | No access is allowed to the report                                       |
| READ_ONLY_ACCESS | 1     | The report can be accessed and run but the definitions cannot be changed |

To change the access rules from the default (that is, all users can access all reports), the methods can be reimplemented on a user-defined subclass of **JadeReportWriterSecurity**.



In this example, whenever the Report Writer Designer window is opened, it checks the user name that was provided. If it is **"Admin"**, the user has full access to reports. A user name of **"User"** returns read-only access. An error message box is displayed and the Jade Report Writer Designer application will not open for any other user name.

**Note** This is a simplified example, normally there would be a greater level of security defined in the method than simply checking for an expected user name.

## Exercise 9 – Specifying Designer Security Settings

In this exercise, you will add a password to the Report Writer Designer application, to prevent unauthorized access.

1. Open the **BankingViewSchema** in the Class Browser.
2. Press F4, to display the Find Type dialog and then search for **JadeReportWriterSecurity**, opening it in the current browser.
3. Add a **JadeReportWriterSecurity** subclass called **BankingReportSecurity**.
4. Add a method called **canAccessDesigner** to the **BankingReportSecurity** class and click **Yes** on the message box warning that you are reimplementing a superclass method.

5. Code the method as follows.

```
canAccessDesigner(userName: String): Integer updating;  
  
vars  
    form : Logon;  
begin  
    create form transient;  
    form.showModal();  
    if form.txtPassword.text = "secret" then  
        return FULL_ACCESS;  
    else  
        app.msgBox("Incorrect Password.", "Logon Failed", MsgBox_OK_Only);  
        return NO_ACCESS;  
    endif;  
epilog  
    delete form;  
  
end;
```

6. Modify the **startDesigner** JadeScript method in **BankingViewSchema** as follows.

```
startDesigner();  
  
vars  
    rw : JadeReportWriterManager;  
begin  
    create rw transient;  
    // rw.startReportWriterDesigner("User", null);  
    rw.startReportWriterDesigner("User", BankingReportSecurity);  
epilog  
    delete rw;  
  
end;
```

7. Run the JadeScript method. If you enter **secret** as the password, the JADE Report Writer Designer application starts as normal; otherwise you are denied entry.

## Exercise 10 – Specifying Configuration Security Settings

In this exercise, you will add a user name requirement to the JADE Report Writer Configuration application.

1. Add a method to **BankingReportSecurity** called **canAccessConfiguration** and then click **Yes** in the message box warning that you are reimplementing a superclass method.

2. Code the method as follows.

```
canAccessConfiguration(userName: String): Integer updating;  
begin  
    if userName = "Admin" then  
        return FULL_ACCESS;  
    elseif userName = "User" then  
        return READ_ONLY_ACCESS;  
    else  
        return NO_ACCESS;  
    endif;  
end;
```

3. Modify the **startDesigner** JadeScript method in **BankingViewSchema**, as follows.

```
startConfiguration();  
  
vars  
    rw : JadeReportWriterManager;  
begin  
    create rw transient;  
    // rw.startReportWriterConfiguration("User", null);  
    rw.startReportWriterConfiguration("Admin", BankingReportSecurity);  
epilog  
    delete rw;  
end;
```

4. Run the JadeScript method, which should allow you full access.

Try changing the user name from **Admin** to **User** for read-only access, or any other user name for denial of access.