

iReport Tutorial

February 2009

Creating and Displaying Reports Using BBJ and iReport 3.1.3	1
Overview	1
Getting Started.....	2
Access the Database	2
Make the Database Available	5
Import the Fields.....	6
Update the Title Band	8
Edit the Column Header Band.....	10
Edit the Detail Band	10
Add a Hyperlink.....	11
Edit the Page Footer Band	12
Edit the Summary Band	12
Viewing the Jasper Report	14
Using SCALL.....	14
Using Call.....	15
Using Object Syntax	15
Stored Procedures At-A-Glance	16

Creating and Displaying Reports Using BBJ and iReport 3.1.3

[back to top](#)

Creating and displaying reports in BBJ® 9.0 is now faster and easier with the inclusion of [JasperReports](#), the new BBjasper utility, and [iReport](#), the easy-to-use open source visual report designer for JasperReports. This tutorial will step you through how to setup a simple report shown below and assumes you have a working knowledge of BBJ, Enterprise Manager, and the BASIS IDE.



Cust #	Customer Name	Address	Monthly Sales
000100	Everest Industries	123 Main St. Suite 111, San Bernardino, CA	\$ 18,162.12
000200	Western Sport Distributors	Market Plaza 30021 Redhill Avenue, Tustin,	4,367.52
000300	Taylor Sporting Goods	1817 Augusta Circle Unit 412, San Juan	4,504.30
000400	Santa Monica Fitness Center	3481 Sunset Boulevard, Santa Monica, CA	3,847.96
000500	Ron Anderson And Company	17 Old Post Road, Palm Springs, CA	9,152.85
000600	Valley Cycle Stores	917 Ventura Boulevard, Sherman Oaks, CA	3,907.99
000700	Douglas Erickson & Company	1893 Monterey Court, Chula Vista, CA	5,218.94
000800	Trident Industries	781 Valencia Boulevard, Fullerton, CA	3,218.24
000900	Orange Coast Day Care, Inc.	9993 Pacific Coast Hwy, Corona Del Mar, CA	3,349.00
001000	Mile High Bike Rentals	9833 Main Street, Lake Arrowhead, CA	6,589.95
001001	Baker And Harrison	21300 North Trim Way Suite 128, Seattle, WA	7,027.00
001002	Robinson Enterprises	5883 Guliver Lane, San Diego, CA	1,125.30
999999	Cash Sale		19.61
Total Monthly Sales			\$ 70,490.78

Page 1 of 1

Overview

[back to top](#)

This tutorial demonstrates how easily you can create and display a monthly sales report with iReport.



A working sample of this demo is available when you install BBJ along with demos. To view the demo, run LaunchDock located in the Start menu under **BBJ > Demos**. The report demo is located under the Language/Interpreter section and is called Jasper Reports. Use the drop down box in the launcher application to select `addon_invoice` and click [Run Report]. All of the files used for this demo are in the `<BBj Install Dir>\demos\reports` directory.



For a quick start to reports, try [Using the iReport Wizard](#), the automatic report generator with customized layouts based on the data used in the report.

Using stored procedures (SPROCs) and SQL, we will retrieve the data from MKEYED files. Even though there is a data dictionary with SPROCs setup for the database in this example, you do not have to create a full-blown data dictionary. In fact, you do not even have to go through the effort of listing any of the tables in your database. Simply tell Enterprise Manager where your data files are and where it should store the data dictionary files and you are up and running! Even though the SPROCs are called using SQL, they utilize a standard READ RECORD loop to read the information in from the data files. You do not have to write any SQL code to generate the data for the report!

The report we will create uses the AddonDemoData Database, which is installed by default when you select "Demos" in the BBJ download.

Getting Started

[back to top](#)

1. Install the current version of BBJ, including the Demos option as shown below. All of the SPROCs used in these report are contained in the Demos installation. The BBJ installation is available at: <https://www.basis.com/products/bbj/download.html>

Optional Files (Please make at least one selection)

<input checked="" type="checkbox"/> BBJ Requires the latest 1.5 or greater JRE	<input checked="" type="checkbox"/> JDBC (0.5MB)	<input type="checkbox"/> Training Files (59MB)
<input checked="" type="checkbox"/> Thin Client	<input checked="" type="checkbox"/> ODBC (0.5MB)	<input checked="" type="checkbox"/> Demos (19MB) <i>Includes ClientObjects, embedded browser, and customized LaunchDock</i>
<input checked="" type="checkbox"/> BBJ Services	<input type="checkbox"/> BASIS IDE (40MB)♦	
<input checked="" type="checkbox"/> Admin Tools	<input checked="" type="checkbox"/> BASIS License Manager (2MB)	
<input checked="" type="checkbox"/> Utilities	<input type="checkbox"/> Documentation (28MB)	
<input checked="" type="checkbox"/> Additional Developer Tools		
<input checked="" type="checkbox"/> Barista 8.31 (35MB) Requires revision 8.31 of BBJ		
<input type="checkbox"/> AddonSoftware 8.31 (80MB) Requires latest build of Barista		

♦ Requires Java 5 JVM, due to IDE Options window incompatibilities with Java 6

2. Install iReport. The Windows version is available for download at:

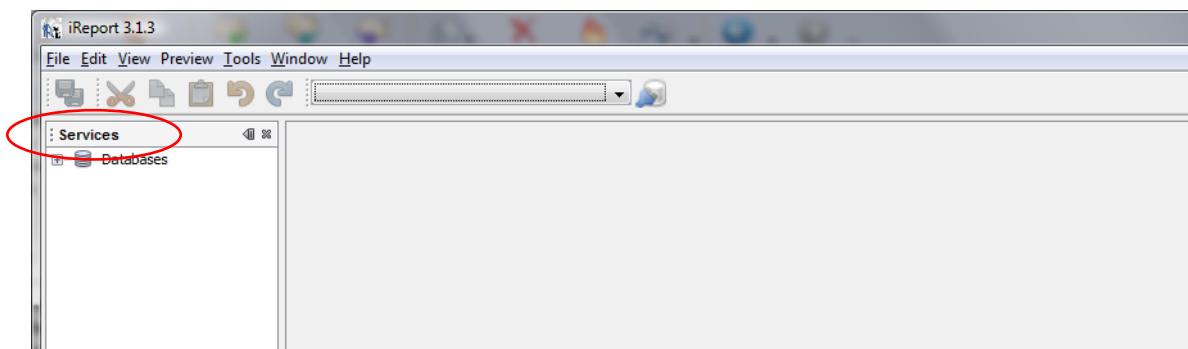
http://downloads.sourceforge.net/iReport/iReport-nb-3.1.3-windows-installer.exe?modtime=1231894247&big_mirror=0

Access the Database

[back to top](#)

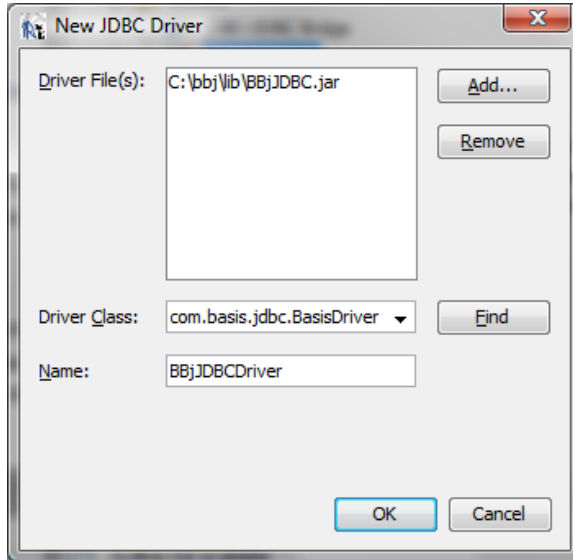
Before creating the actual report, we need to give iReport access to the database and be sure iReport recognizes the BBJ JDBC driver.

1. Open iReport.
2. In the **Services** pane inside iReport, expand the **Databases** tree.



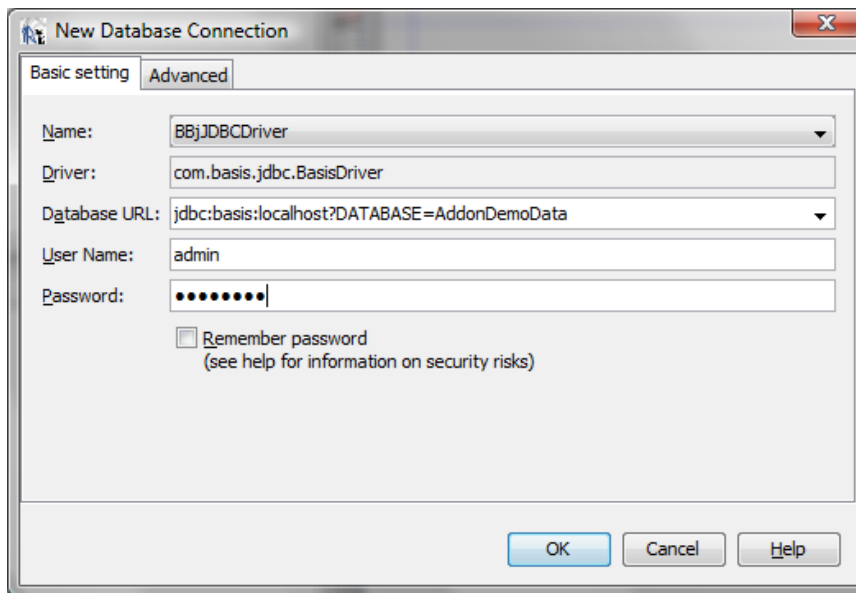
3. Right click on **Drivers** in the **Databases** tree and choose **New Driver**.

4. Click [Add], locate and select the `BBjJDBC.jar` file (found in the `<BBj Install Dir>\lib` directory) in the Driver File(s) section.
5. Click [Find] to locate the Driver Class `com.basis.jdbc.BasisDriver`.
6. Name it `BBjJDBCdriver` (as shown below) and click [OK].



7. Right click on the **Databases** in the **Services** pane and select **New Connection**.
8. Select the `BBjJDBCdriver` in the "Name" field and enter this connection string in the "Database URL" field:


```
jdbc:basis:localhost?DATABASE=AddonDemoData
```
9. Enter the "User Name" and "Password" for the `AddonDemoData` database; the defaults are `admin` and `admin123` respectively.

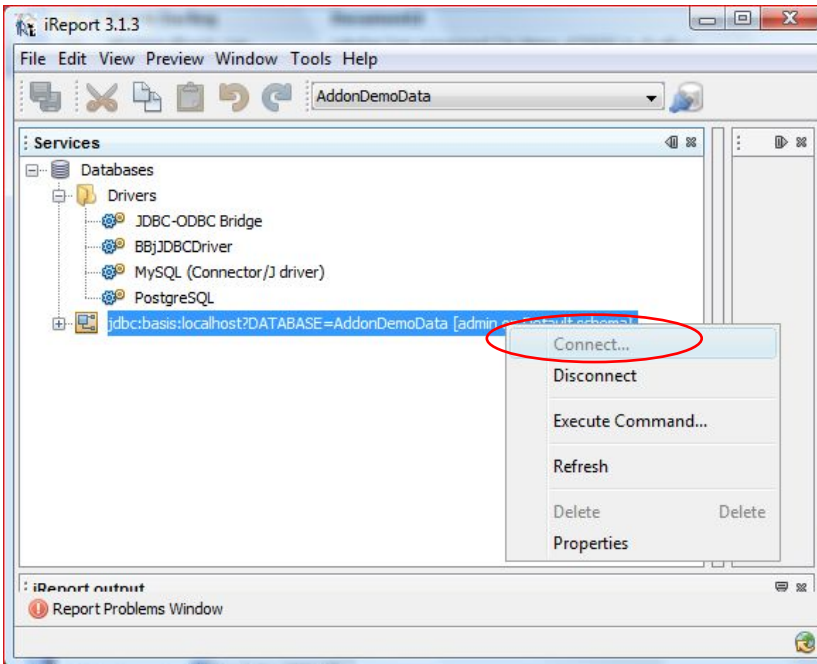


Note: You can alternatively include the user name and password in the connection string shown below but leave the "User Name" and "Password" fields blank in the dialog box.

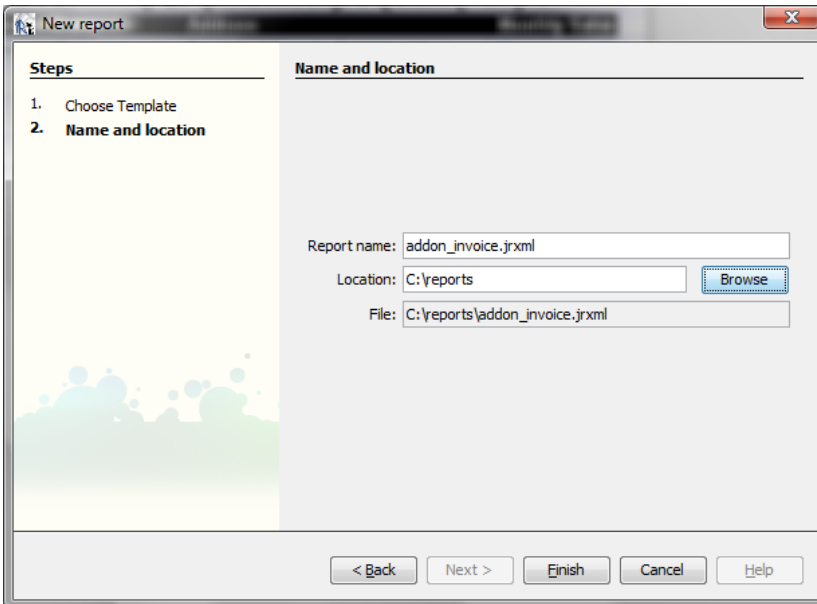
```
jdbc:basis:localhost?DATABASE=AddonDemoData&User=admin&Password=admin123
```

10. Click [OK].

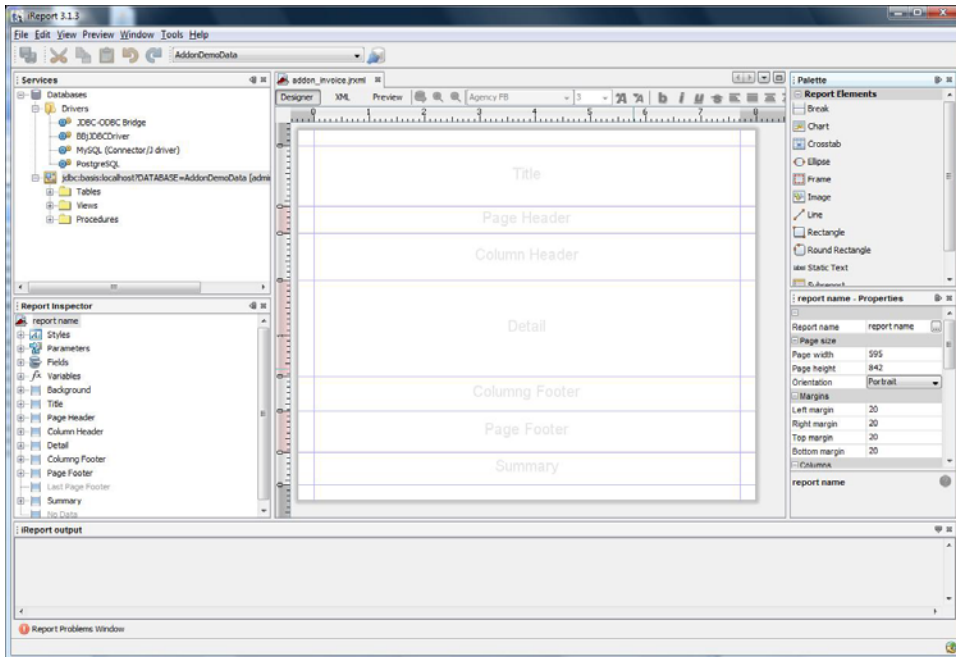
You should now have a BBJ database available to iReport in the **Services** pane (if BBJServices is not running, you will encounter an "unable to connect to the database" error). Verify that you are connected by right clicking on the database.



11. Create a new empty report via **File > New empty report** named `addon_invoice.jrxml` as shown below. Include a directory path for the report, and then click [Finish]. If you don't have a `C:\reports` directory, it is highly recommended you press [Browse] and create this directory so that you can run the `BBjasper.bbj` program at the end of this tutorial without modification.




Once you've successfully created a new report, your window should look like this:

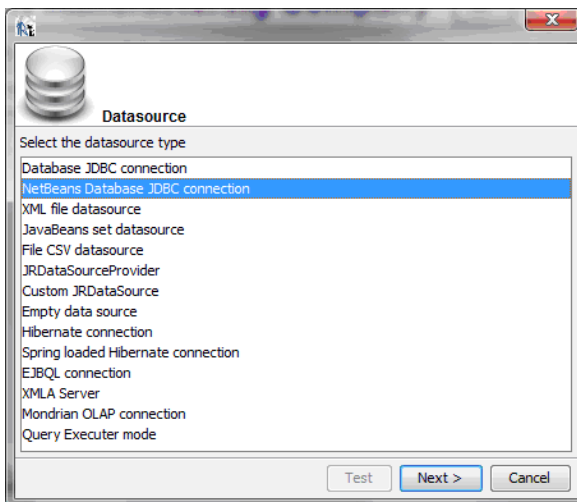


Make the Database Available

[back to top](#)

With the database now available in iReport, we need to make the database available to our new report.

1. Click on the **Report DataSources** tool button  and choose "New" in the **Connections/Datasources** dialog box.
2. Select **NetBeans Database JDBC connection** (shown below) and click [Next >].



3. Enter AddonDemoData as the name and use the dropdown list to select the connection string to connect to the AddonDemoData Database:

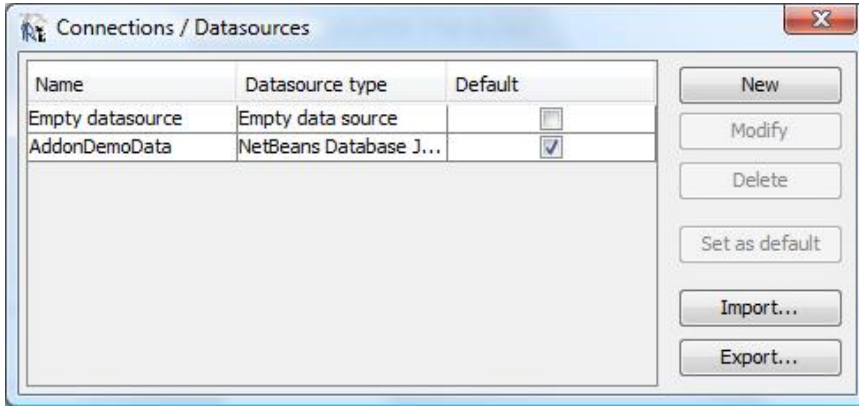
With User and Password in the dialog:

```
jdbc:basis:localhost?DATABASE=AddonDemoData
```

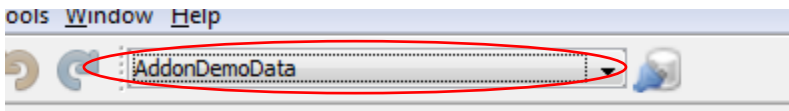
With User and Password entered in the connection string:

```
jdbc:basis:localhost?DATABASE=AddonDemoData&USER=admin&PASSWORD=admin123
```

- Click [Test] to verify the connection string works, then click [Save].
- Close the **Connections/Datasources** window shown below. The database is now available for use in a new report.

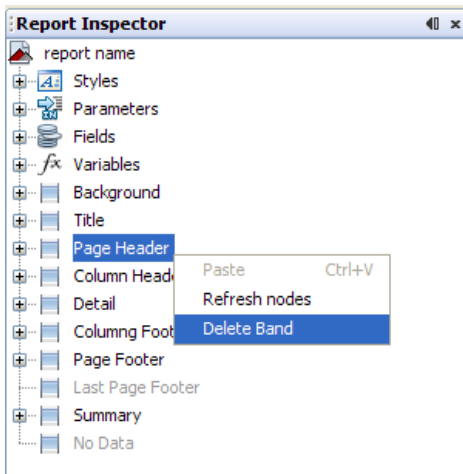


- Select AddonDemoData in the dropdown list for the report if it is not already selected.



- Inside the **Report Inspector** pane shown below, right click and select "Delete Band" for each of the following as we will not be using them in this report:

Page Header
Column Footer



- Select **report name** from the **Report Inspector** and using the **Properties** pane to the right (or right click on **report name**), change the report name to `addon_invoice`.

Import the Fields

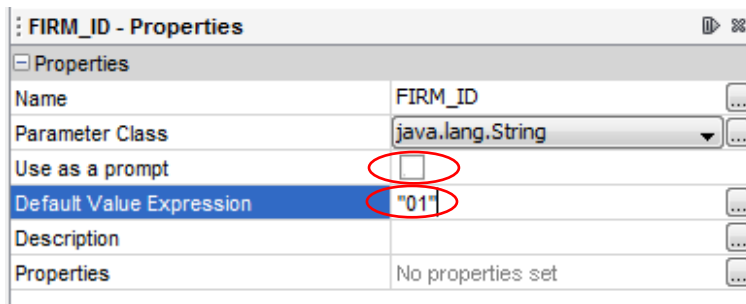
[back to top](#)

Now we want iReport to import all of the fields created by a query so that they are available to use. To import the fields, first we need to enter new parameters and our query.

- Using the **Report Inspector**, right click on **Parameters** and choose **Add Parameter**.
- Change the **Parameter Name** and **Default Value Expression** of the newly created/selected "parameter1 – Properties" in the **Properties** pane to FIRM_ID. Repeat twice more to create MONTH and YEAR with these settings:

Parameter Name	Parameter Class	Default Value Expression
FIRM_ID	java.lang.String	" 01 "
MONTH	java.lang.String	" 12 "
YEAR	java.lang.String	" 2007 "

Note: Be sure to type quotations around the **Default Value Expression** as shown and uncheck **Use as a prompt** for all three of the parameters.



- With `addon_invoice` selected in the **Report Inspector**, go to the properties box and enter this query for the report in the "Query Text" field:

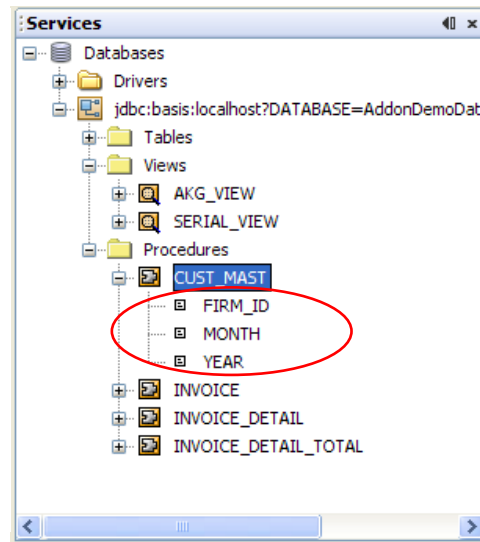
```
CALL CUST_MAST ( $P{FIRM_ID} , $P{MONTH} , $P{YEAR} )
```

This SQL statement calls a stored procedure* that is installed with the demos. You can view this procedure in Enterprise Manager, if desired, in the AddonDemo database SPROC.

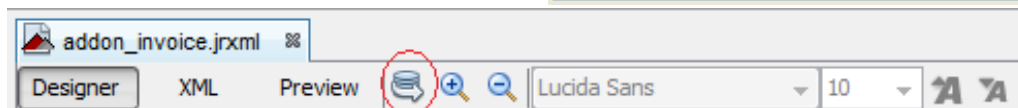
*For more information, see [Stored Procedures At-A-Glance](#) at the end of this document.

- In the **Services** pane, expand the AddonDemoData database we just created.

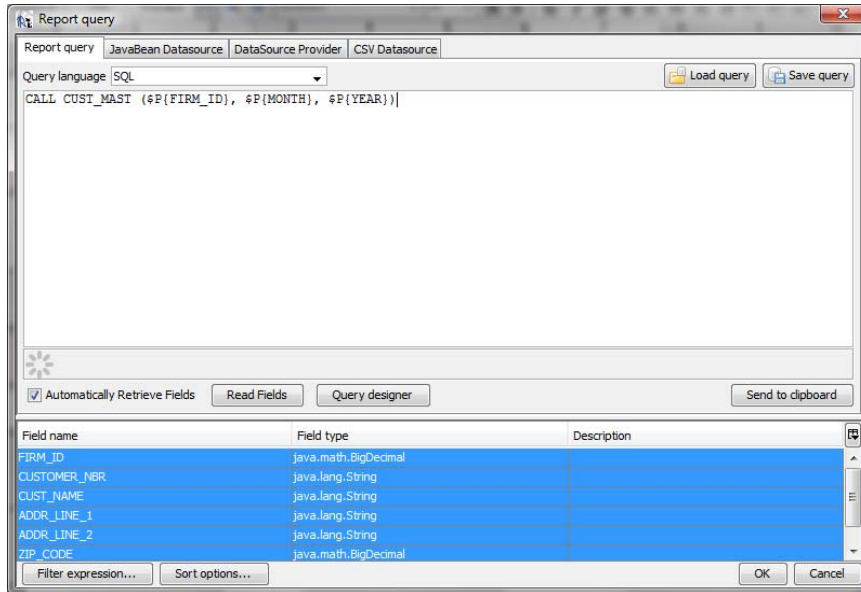
Expanding each of these **Tables**, and **Procedures** folders show what available in this database. For example, expand the **Procedures** see a list of all its available SPROC. on the `CUST_MAST` procedure to report parameter types required to this stored procedure (example to the right).



- In the Report Inspector pane, select `addon_invoice` and click the [Report Query] icon on the report toolbar as shown below.



6. Verify that "Automatically Retrieve Fields" is checked so that iReport will automatically import all of the fields that are returned from our query.
7. Click the [Read Fields] button.
8. After iReport populates the fields, close the window by clicking [OK].



Update the Title Band

[back to top](#)

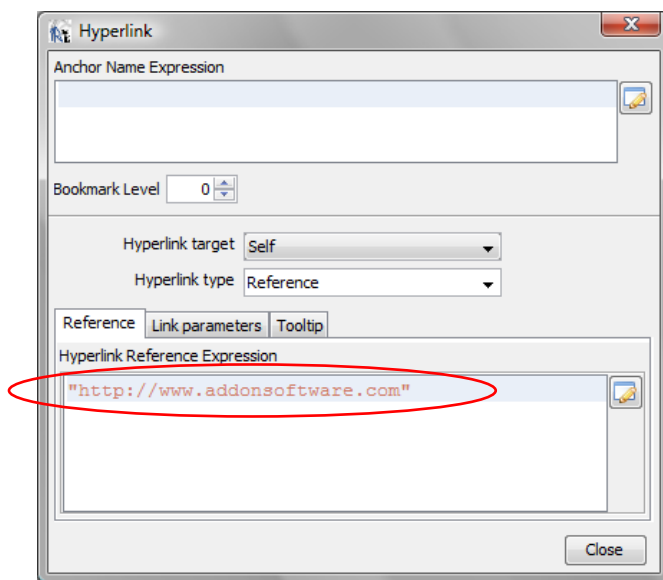
1. Locate the list of Report Elements in the Palette pane
2. Create a new static text in the Title Band by selecting [Static Text] and dragging it anywhere into the Title Band in the report **Designer** view.
3. Click on the Static text box to display the **Static text – Properties** pane. Enter `ADD+ON` Manufacturing Company as the field text, and adjust the other properties such as font family, style, size, color, as you like. Here is what our font selections look like:



- Drop an image element in the Title band. The path to the image file used is:
`<BBjInstallDir>\\demos\\reports\\images\\addon.png`

Note: Though the file browser will not display the PNG files by default, either change the display to "all files" or navigate to the directory and type in the name of the file to load the image.

- Click [OK].
- Adjust the size of the image to fit in the Title band and position it to the right of the title.
 Once the image is in place, we can add a hyperlink to it for use in the `BBjasper.bbj` utility. The hyperlink opens a browser with a specified URL.
- Right click on the image and choose "Hyperlink."
- Change **Hyperlink type** to "Reference." Next, type in this desired URL as the **Hyperlink Reference Expression with the quotation marks**: `"http://www.addonsoftware.com"`



- Click [Close].
 Note: When running the report using `BBjasper.bbj`, click on the image to open the default browser with the page referenced in the Hyperlink Reference Expression.
- Drag a new Text Field to the center of the Title Band and select it.
- Scroll down the **\$F{field} – Properties** pane to locate the "Text Field Expression." Click the to the right and paste in the following expression, replacing the `$F{field}` placeholder:

```
"Sales for " + ( $P{MONTH}.equals("1") ? "January" :
( $P{MONTH}.equals("2") ? "February" :
( $P{MONTH}.equals("3") ? "March" :
( $P{MONTH}.equals("4") ? "April" :
( $P{MONTH}.equals("5") ? "May" :
( $P{MONTH}.equals("6") ? "June" :
( $P{MONTH}.equals("7") ? "July" :
( $P{MONTH}.equals("8") ? "August" :
( $P{MONTH}.equals("9") ? "September" :
( $P{MONTH}.equals("10") ? "October" :
( $P{MONTH}.equals("11") ? "November" :
( $P{MONTH}.equals("12") ? "December" : "" ) ) ) ) ) ) ) ) ) ) + " " + $P{YEAR}
```

Note: This expression uses `$P{MONTH}` and converts it into a month name and is in the **Expression Class** `Java.lang.String`.

- Click [OK].
- Click the [Preview] tab to verify you entered the information correctly and with the required quotation marks. The hyperlinks will not be functional in this preview step but will fully operate in BBjasper.



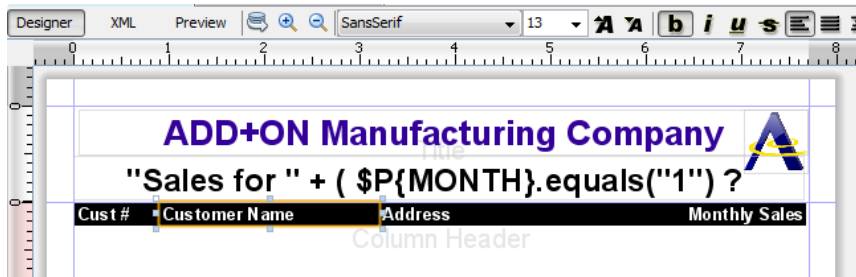
iReport will display any error generated in the Report Problems window and will also link you to the section that generated the error. The iReport output is often helpful in locating the source of any problems.

- To exit out of the preview, click the Designer tab. The hyperlinks that we added earlier will not be functional within the Ireport viewer utility.

Edit the Column Header Band

[back to top](#)

- Add the following Static Text fields to the Column Header Band:
 - Cust#*
 - Customer Name*
 - Address*
 - Monthly Sales*
- Adjust the position, font, forecolor, and bgcolor, etc. as you desire, to resemble this:



- Click on the [Preview] button verify the layout looks the way you want.

Edit the Detail Band

[back to top](#)

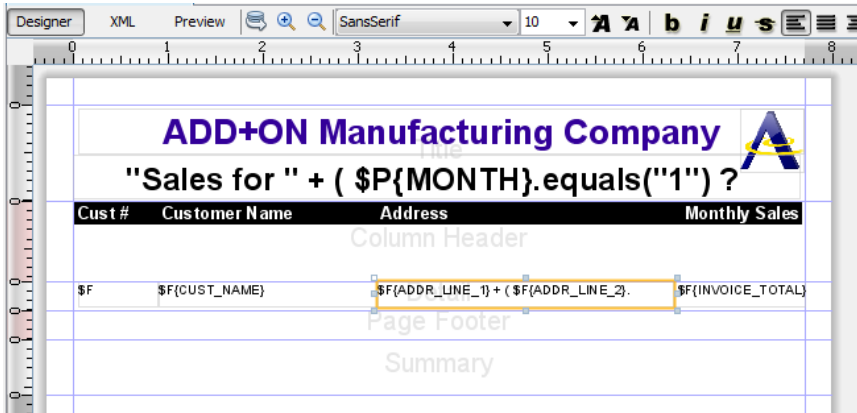
- Add the following fields to the detail band by choosing "Fields" in the Report Inspector and dragging them into the Detail Band.
 - CUSTOMER_NBR*
 - CUST_NAME*
 - ADDR_LINE_1*
 - INVOICE_TOTAL*

This creates a new text field in the detail band with the Text Field Expression set to the correct field (i.e. for CUSTOMER_NBR, set the Text Field Expression to $\$F\{CUSTOMER_NBR}$, which the query that we entered earlier returns).

2. Modify the **Pattern** in INVOICE_TOTAL's properties to \$#,##0.00 for proper formatting.
3. Verify that the **Expression Class** for the INVOICE_TOTAL field is java.math.BigDecimal.
4. Adjust the **Text Field Expression** for ADDR_LINE_1 to include ADDR_LINE_2 within the same text field by going to the ADDR_LINE_1 properties and changing the **Text Field Expression** to:

```
$F{ADDR_LINE_1}+" "+$F{ADDR_LINE_2}
```

5. Position the fields and decrease the size of the Detail Band as shown below:



6. Click on the [Preview] button to verify the layout; adjust as needed.

Add a Hyperlink

[back to top](#)

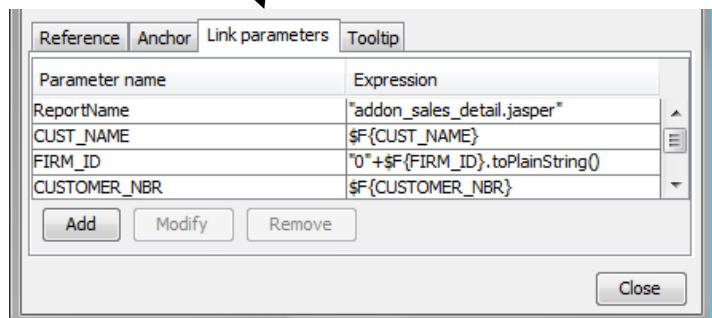
To add a link to an additional report, use the Hyperlink option. This example uses a previously created report that installed with the BBj "Demos" you selected at download.

1. Right click on the INVOICE_TOTAL field and choose "Hyperlink."
2. Set the hyperlink target to "Blank" and the Hyperlink type to "RemoteAnchor."
3. Go to the Link parameters tab and add the following parameters as java.lang.String types, and including the quotation marks as listed below:

Parameter Name	Expression Values
ReportName	"<BBjInstallDir>\demos\reports\addon_sales_detail.jasper"
CUST_NAME	\$F{CUST_NAME}
FIRM_ID	"0"+\$F{FIRM_ID}.toString()
CUSTOMER_NBR	\$F{CUSTOMER_NBR}
MONTH	\$P{MONTH}
YEAR	\$P{YEAR}

When used with the BBjasper.bb utility, this displays the specified report (passing in the parameters we added) in a new window when clicking a monthly sales value in the report.

4. Click the [Close] button.
5. Click the [Preview] button to verify you entered the expressions correctly; adjust as needed.



Edit the Page Footer Band

[back to top](#)

1. Select and drag a Text Field from the **Report Elements** pane into the Page Footer Band to display the date.
2. Set the **Expression Class** to `java.util.Date`.
3. Change the **Text Field Expression** to `new Date()`.

We will also add an additional Text Field to number the pages in the report.

4. From **Report Inspector**, expand **Variables** and drag the `PAGE_NUMBER` variable to the Page Footer Band.
5. Set the **Expression Class** as a `java.lang.String` and set the **Text Field Expression** to:

`"Page " + $V{PAGE_NUMBER} + " of "`

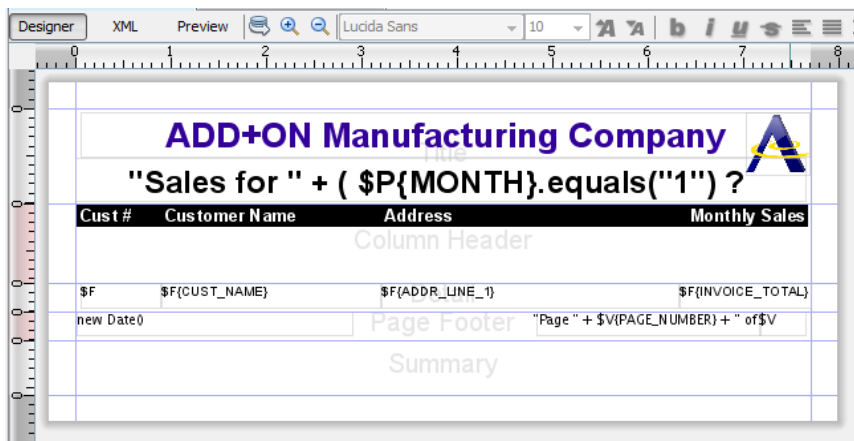
Note: We are taking advantage of a variable created automatically within iReport when a new report is created.

We also need to list the total number of pages for the report.

6. Add another Text Field at the end of this one with an **Expression Class** of `java.lang.Integer` and set the **Text Field Expression** to:

`$V{PAGE_NUMBER}`

7. Adjust the size of the fields and adjust the band to the proper size as shown below.



6. Click on the [Preview] button to verify the report appears the way you intended.

Edit the Summary Band

[back to top](#)

The summary band will contain the Total Sales figure for the report.

1. Add a new Static Text and set the text property to:

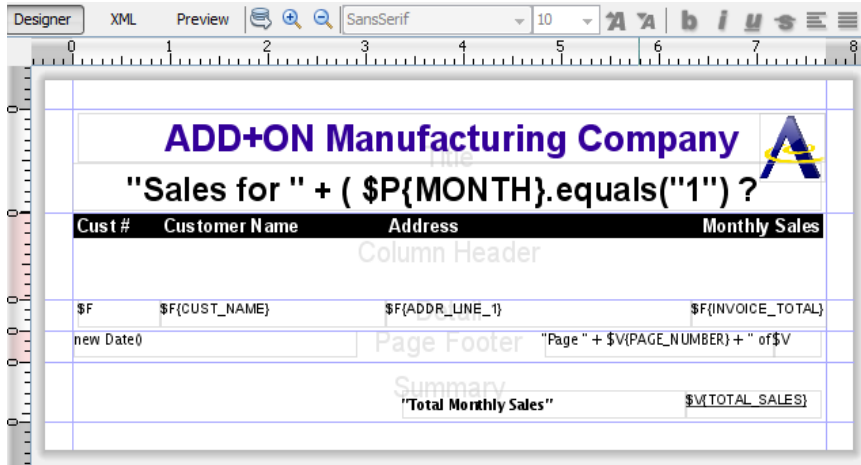
Total Monthly Sales

To get the Total Sales, we need to add a new variable.

2. Go to the **Report Inspector**, right click on Variables, and choose "Add Variable."

3. Change the name to TOTAL_SALES; in **Properties** change the Variable Class to `java.math.BigDecimal` and the Calculation type to Sum.
4. Set the Variable Expression to:

$$\$(INVOICE_TOTAL)$$
5. Drag the new variable into the Summary Band.
6. Change **Pattern** to `$.#,##0.00` for proper formatting.
7. Position both text fields in the summary band and adjust the size of the fields as well as the band itself as follows:



8. The basics are now in place. Verify that the report runs correctly by clicking on the [Preview] tab to display the report in the Report Viewer that is included with iReport.



9. To exit iReport, select **File >Exit**.

Congratulations on your first Report!



A working sample of this demo is available when you install BBJ along with demos. To view the demo, run LaunchDock located in the Start menu under **BBj > Demos**. The report demo is located under the Language/Interpreter section and is called Jasper Reports. Use the drop down box in the launcher application to select `addon_invoice` and click [Run Report]. All of the files used for this demo are in the `<BBj Install Dir>\demos\reports` directory.

Viewing the Jasper Report

[back to top](#)

The Jasper Reports Demo programs utilize the new BBjasper utility, which makes displaying reports in your BBJ or Visual PRO/5® application possible. Depending on the number of parameters that you pass into your report, you can show a report in less than 10 lines of code.

In all of the examples below, we are setting report parameters for the `Firm_ID` and date range to select a subset of the entire dataset.

Using SCALL

[back to top](#)

You can run the BBjasper utility via the command line. by a Visual PRO/5 or BBJ system call. The following sample code demonstrates the SCALL Interface built into BBjasper.bbj.

Here is a summary of what the code does in each section.

Set variables for creating the report – sets the report name along with the connection string and locale. BBjasper uses the Report name to determine the location of the report. The connection string determines where to connect to the database and provides user authentication, if desired. Note that the connection string requires quotes around it when used in an SCALL.

Parameters for the ADD+ON Monthly Report – sets the parameters that are passed in as key value pairs. All of the parameters that the stored procedure requires are set in this section and passed to BBjasper. Next, the codes builds a string used for the SCALL.

SCALL the report – SCALLs BBjasper.bbj using the information passed in to create/display the specified report.

```

10 REM =====
20 REM Set variables for creating the report
30 REM =====
40 report_name$="c:/ reports/addon_invoice.jasper"
50 connect_string$=chr(34)+"jdbc:basis:localhost?database=AddonDemoData&user=guest "+chr(34)
60 lang$="en"
70 REM =====
80 REM Parameters for the ADD+ON Monthly Report
90 REM =====
100 param1$="FIRM_ID"
110 value1$="01"
120 param2$="MONTH"
130 value2$="1"
140 param3$="YEAR"
150 value3$="2008"
160 reportcmd$="c:/program files/basis/bin/bbj.exe c:/program
files/basis/utils/reporting/bbjasper/bbjasper.bbj - " + report_name$ + " " + connect_string$ + "
" + lang$
180 reportcmd$ = reportcmd$ + " "+param1$ + " " + value1$
190 reportcmd$ = reportcmd$ + " "+param2$ + " " + value2$
200 reportcmd$ = reportcmd$+" "+param3$ + " " + value3$
210 REM =====
220 REM SCALL the report
230 REM =====
240 a=scall(reportcmd$)

```

Using Call

[back to top](#)

The call interface gives additional control over using SCALL and should be a familiar feature to most BBx® programmers.

Set BBjasper, reportFile connection string, and locale variables – sets variables for BBjasper, the report we built using iReport, and the connection string that we will use to access the data in the report. We also set a variable for the locale here.

Set the parameters – creates a hashmap and inputs the parameters needed for the report.

Create the report, viewer window, and show the window – uses the call interface in BBjasper.bbj to create the report. The first call creates the report and returns the report object to the program. The second call fills the report. The third call creates the window for the report and the last call shows the window.

```
REM =====
REM Set BBjasper, reportFile connection string, and locale variables
REM =====
bbjasper$ = "c:/bbj/utils/reporting/bbjasper/bbjasper.bbj"
reportFile$ = "c:/bbj/demos/reports/addon_invoice.jasper"
connectstring$ = "jdbc:basis:localhost?database=AddonDemoData&user=admin&password=admin123"
locale$="en"
REM =====
REM Set the parameters
REM =====
params! = new java.util.HashMap()
params!.put("FIRM_ID","01")
params!.put("MONTH","12")
params!.put("YEAR","2007")
REM =====
REM Create the report, viewer window, and show the window
REM =====
call bbjasper$+"::createReport",report!,reportFile$,connectstring$,params!,locale$
call bbjasper$+"::fillReport",report!
call bbjasper$+"::createViewWindow",viewerWindow!,report!
call bbjasper$+"::showViewerWindow",viewerWindow!,1
release
```

Using Object Syntax

[back to top](#)

Object syntax is used in cases where one requires complete control, giving a full range of options over the report.

Determine the desired report filename, build the connection string, and create the report – determines the report name, builds a connection string to the database that contains the report data, and then creates a new BBjasperReport object based off of that information. This is fairly standard, but could be enhanced to ask the user to provide database authentication.

Fill the report with data – demonstrates how to set report parameters. Most reports are dynamic in the sense that they can report on different data given predetermined criteria such as a date range, customer type, and so on. This makes it possible to create a generic report that can be used again and again against the same database to report on different aspects of the dataset. The last line of the section executes the fill() method on the report object, which results in the creation of the report based off of the live data in the database.

Customize the report's title and placement, and show it – displays the graphical report in a customized viewer window. Creating the JasperViewerWindow is similar to creating a standard BBjWindow in having control over the window's size, placement, title, and even flags to determine initial state and properties such as the ability to be minimized and maximized. The example code creates the window 'initially invisible,' but then centers it via a method call and then shows the window. The JasperViewerWindow allows the end user to zoom in and out of the report, size it according to the width/height, and navigate through the available pages. Toolbar buttons on the window also facilitate refreshing the report data (requering the database to get a new snapshot of the data), printing the report, and saving the report to disk.

All of these operations are also available programmatically, given the BBjJasperReport object. With the object, your program can export the report to various formats such as HTML and PDF. It can also refresh the report, if desired, and print it. This makes it easy for your application to generate the reports automatically and print or e-mail them as desired.

```

REM =====
REM Use statements
REM =====
use :../utils/reporting/bbjasper/bbjasper.bbj::BBJasperReport
use :../utils/reporting/bbjasper/bbjasper.bbj::BBJasperViewerWindow

REM =====
REM Determine the desired report filename, build the connection string, and create the report
REM =====
reportName$ = "c:\reports\addon_invoice.jasper"
connectString$ = "jdbc:basis:localhost?database=AddonDemoData&user=guest"
addonSalesReport! = new BBJasperReport(reportName$, connectString$)

REM =====
REM Fill the report with data
REM =====
addonSalesReport!.putParam("FIRM_ID", "01")
addonSalesReport!.putParam("MONTH", "12")
addonSalesReport!.putParam("YEAR", "2007")
addonSalesReport!.fill()

REM =====
REM Customize the report's title and placement, and show it
REM =====
x=10
y=10
width=800
height=800
flags$ = $00000093$
title$="ADD+ON Sales Report"
reportWindow! = new BBJasperViewerWindow(addonSalesReport!,x,y,width,height,title$,flags$)
reportWindow!.show(1)

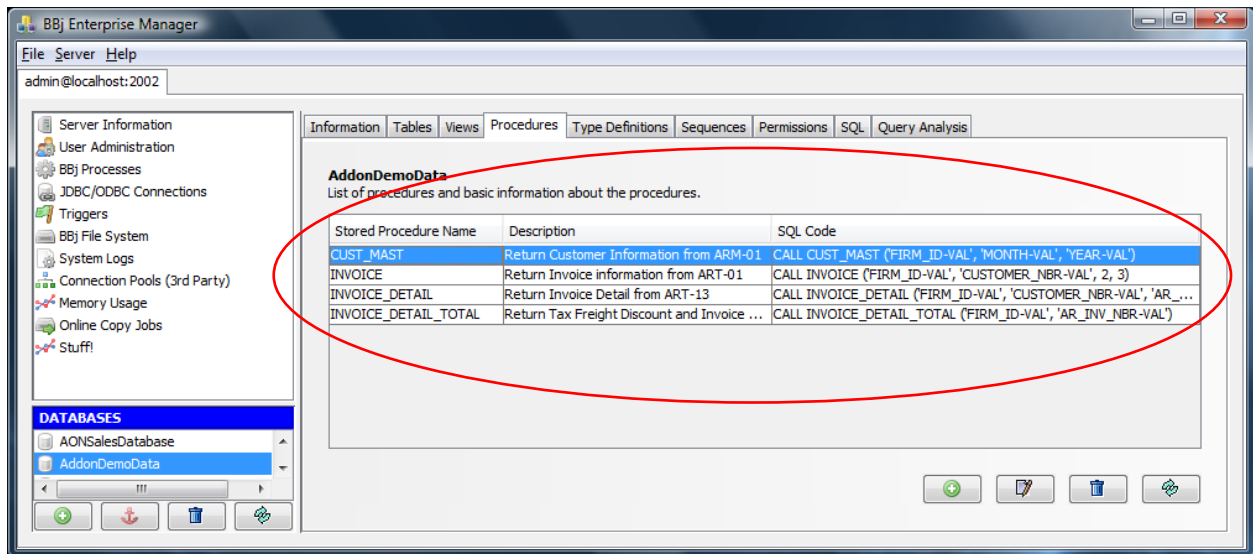
```

A working sample of this sample report appears in <BBj Install Dir>/demos/reports for reference. Using iReport, select **File > Open** and choose `addon_invoice.jrxml` from the reports directory.

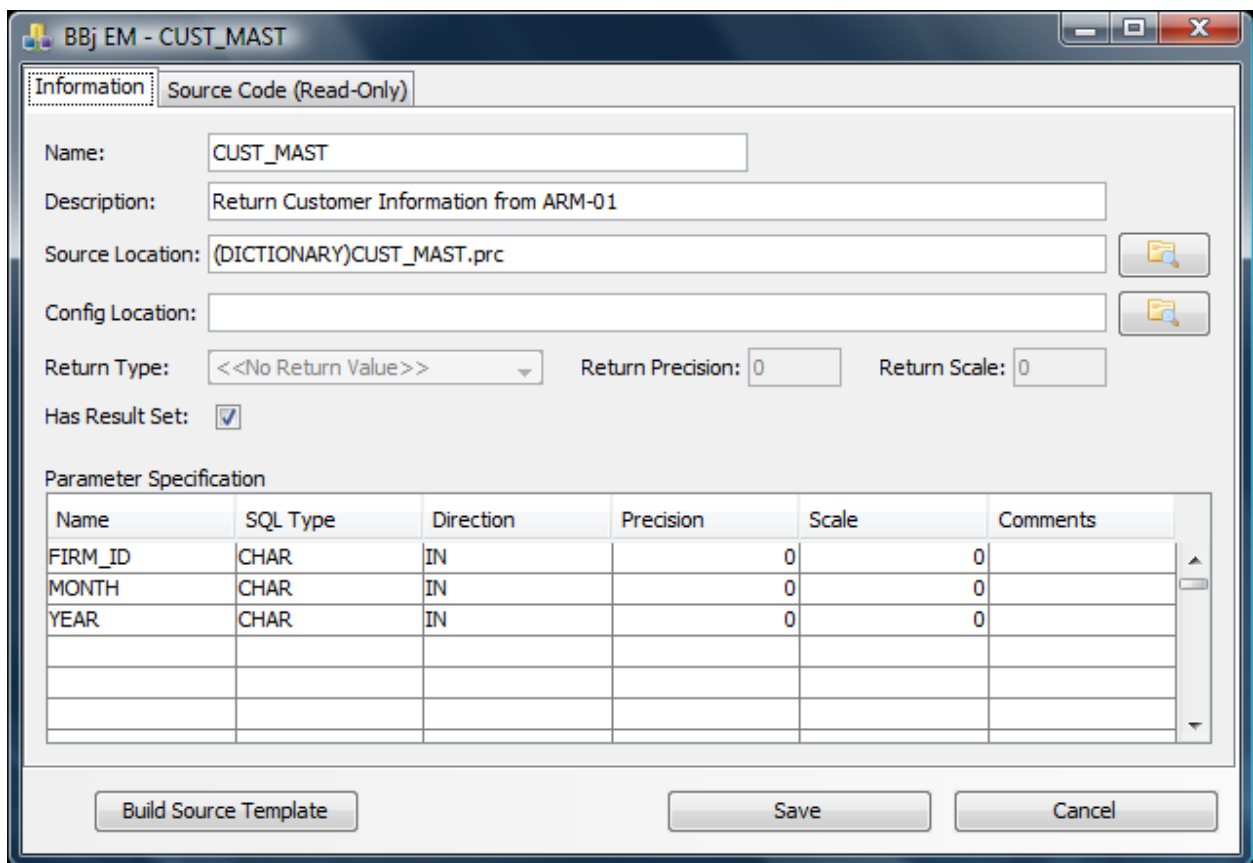
Stored Procedures At-A-Glance

[back to top](#)

The report for this example makes use of a pre-defined stored procedure that is installed along with the AddonDemoData database. To see all of the available SPROCs, launch the Enterprise Manager (EM), select the "AddonDemoData" database and then click on the "Procedures" tab as shown in the next screen shot. Here, EM displays a list of the SPROCs, along with a description of each and sample SQL Code demonstrating how to call the procedure.

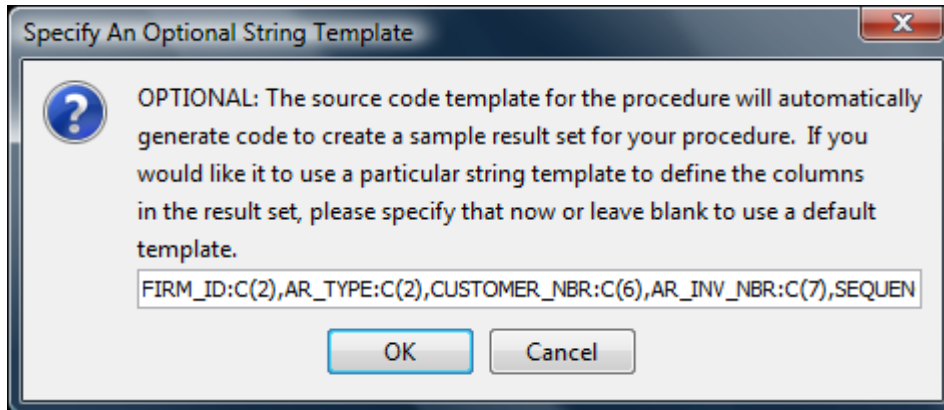


To see the property sheet for an individual stored procedure, select it in the list and double click the entry. Below is the property sheet for the `CUST_MAST` procedure that details all of the properties, including "Name," "Description," "Source location," and the complete set of parameters that the stored procedure utilizes.

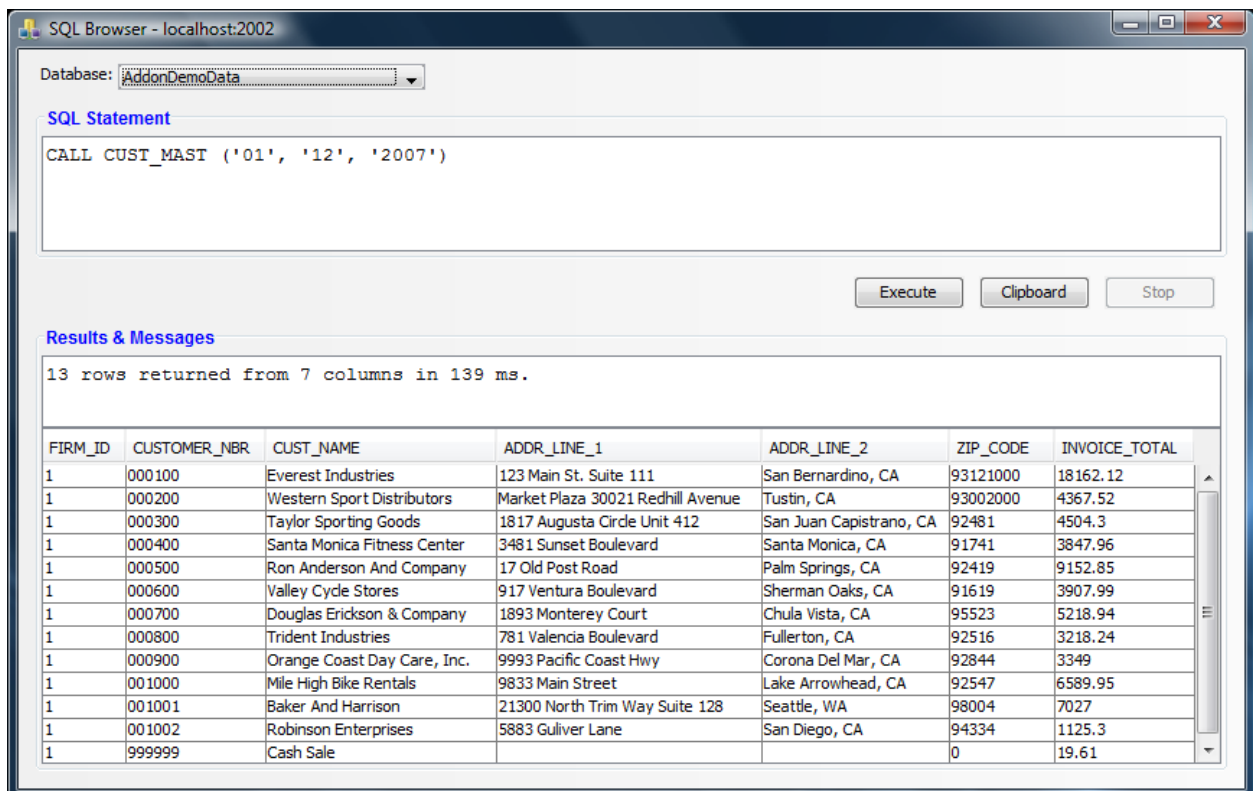


The [Build Source Template] button in the bottom left corner facilitates the creation of a new stored procedure by writing out a customized program that will be used as the procedure's source code. By specifying an optional string template, developers can build a program that contains all of the code

necessary to retrieve the stored procedure object, fill a result set based off of the template via traditional READ RECORDs from the data file, and return the result set to the client.



After modifying the Source Template program to ensure that it is fully functional, EM makes it easy to test the stored procedure via its SQL tab. Simply copy the SQL Code given for the desired procedure and paste it into the SQL Browser. Replacing the default parameters of 'FIRM_ID-VAL', 'MONTH-VAL', 'YEAR-VAL' with real data values such as '01', '12', '2007' and executing the query shows the following result set indicating that the stored procedure was created and setup correctly:



[back to top](#)