

# MODULE 1: CREATING REPORTS

## Module Overview

Microsoft Dynamics® NAV 2013 is one of the most innovative releases in the product's history.

This module overview describes the main differences between Microsoft Dynamics NAV 2013 and all previous versions. It also describes the competitive advantages of the Microsoft Dynamics NAV 2013 release.

Like the previous version, Microsoft Dynamics NAV 2013 is built on a three-tier architecture model. The introduction of Microsoft Dynamics NAV 2013 means the end of the two-tier architecture.

- The RoleTailored client is the client tier.
- In addition, Microsoft Dynamics NAV 2013 also supports additional client types, including web service clients, Microsoft Dynamics NAV Web client, a SharePoint client through Microsoft Dynamics NAV Portal Framework, and a NAS services client for programmatic access.
- Microsoft Dynamics NAV Server is the middle or server tier, managing all business logic and communication.
- SQL Server, augmented by Microsoft Dynamics NAV 2013 database components, is the data tier.

More information on the architecture is covered in the course *Installation and Configuration in Microsoft Dynamics NAV 2013*.

Although the two-tier architecture has proven its value through the years, the three-tier architecture offers many competitive advantages.

### Openness

The new architecture allows objects (pages and codeunits) to be exposed as web services. This means that the object (and its business logic) is made available as a .NET assembly and can be called from any external application speaking SOAP (Simple Object Access Protocol).

### Multiple Clients

This technological innovation had another impact. By making the business logic available on a separate tier and as a .NET assembly, Microsoft opened the gates for client applications other than the Classic client. It is now possible to develop applications on multiple software (WinForms, Web applications) or hardware platforms (computer, mobile devices, and so on).

The following can be considered Microsoft Dynamics NAV clients because they interact with the Microsoft Dynamics NAV database through Microsoft Dynamics NAV Server.

- RoleTailored client
- Microsoft Dynamics NAV Portal Framework
- Microsoft Dynamics NAV web client
- SOAP web service client
- OData web service client
- NAS services client

All of the clients mentioned here also share the same business logic.

### **Enhanced Productivity through the RoleTailored Interface**

The interface of the Dynamics NAV is role tailored, which means it is built around the way users work and the tasks they execute in their daily job. In the RoleTailored client, the interface is optimized for employees to fulfill their jobs. It offers functions that are relevant for a specific task. Instead of having to search for a specific function in the menus, users have instant access to all the relevant features they need, which results in higher productivity.

Although the new architecture made the product more open, it also implied certain challenges. Since the business logic is no longer executed on the client, some of the functionalities required (major) design changes. In addition, the new solutions needed to support multiple display targets in an easy way. The two areas severely impacted by this architecture are data presentation and reporting.

Microsoft has solved these challenges by introducing pages and by further integrating with other Microsoft products, such as Microsoft Visual Studio® and the ReportViewer control that supports a local processing mode that allows you to run Client Report Definition files (RDLC).

Pages are the RoleTailored client; the equivalent for the Classic form object. Unlike classic forms, pages have no fixed width and height, but resize automatically in the function of the display target.

The reporting solution in Microsoft Dynamics NAV 2013 is taking advantage of the enhanced functionalities of RDLC 2008.

This course describes the reporting solution.

### Objectives

The objectives are:

- Describe the roles of reports and know the anatomy of a report.
- Understand the way a report layout is designed and executed.
- Know how frequently used report properties are implemented in Report Definition Language Client (RDLC) report layouts.
- Know how to build a request options page for the RDLC report layout.
- Describe the prerequisites for report development in Microsoft Dynamics NAV 2013.
- List the software prerequisites for report development.
- List the hardware prerequisites for report development.
- Explain where to find more information about Microsoft Visual Studio 2010.
- Describe the roles of reports and know the anatomy of a report.
- Explain the structure of reports in Visual Studio Report Designer.
- Understand the rules for flattening data.
- Understand the steps involved in creating a basic report.
- Explain how to include a label, variable and expression.
- Explain the creation of the layout of a report.
- Document the User Experience (UX) guidelines for reports in Microsoft Dynamics NAV 2013.
- Demonstrate how to view the dataset at runtime.
- Describe how to set page options, paper size.
- Understand the creation process for the dataset.
- Know how frequently used report properties are implemented in RDLC report layouts.
- Know what the request options page in a report is used for and how to build a request options page for the RDLC report layout.
- Learn how to implement sorting and filtering in Microsoft Dynamics NAV 2013 reports.

## Prerequisites

The prerequisites for report development in Microsoft Dynamics NAV 2013. It will include an explanation of the supported version(s) of Visual Studio and how to obtain and install it.

### Visual Studio 2010 Professional

To develop RDLC reports in Microsoft Dynamics NAV 2013, Visual Studio 2010 Professional or above is required.

Visual Studio Report Designer offers several new options and features. Furthermore, due to its thorough integration with Microsoft SQL Server, it is possible to take advantage of the reporting capabilities of Microsoft Report Viewer, including the following.

- Richer formatting
- Interactive sorting
- Graphics and charts
- Export possibilities (PDF, Microsoft® Office Excel® and Microsoft® Office Word®)

When you install Microsoft Visual Studio 2010 Professional (or a higher stock keeping unit (SKU)) you must select at least the following option, Visual Web Developer, as shown as in the following illustration.

Visual Web Developer

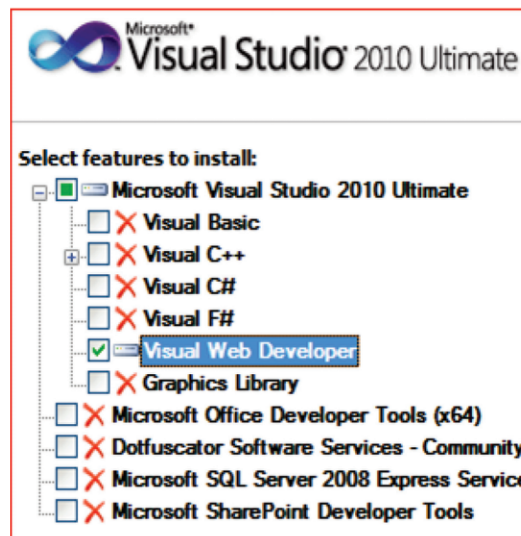


FIGURE 1.1: VISUAL STUDIO CHECKLIST

### Software Requirements

Visual Studio 2010 can be installed on the following operating systems.

- Windows Vista (x86 & x64) with Service Pack 2 - all editions except the Starter Edition
- Windows 7 (x86 & x64)
- Windows Server 2008 (x86 & x64) with Service Pack 2
- Windows Server 2008 R2 (x64)

Supported Architectures include the following.

- 32-Bit (x86)
- 64-Bit (x64)

### Hardware Requirements

To run Microsoft Visual Studio 2010, you will require the following hardware.

- A computer that has a 1.6 GHz or faster processor
- 1 GB (32 Bit) or 2 GB (64 Bit) RAM (Add 512 MB if running in a virtual machine)
- 3 GB of available hard disk space
- 5400 RPM hard disk drive
- DirectX 9 capable video card running at 1024 x 768 or higher-resolution display
- DVD-ROM Drive

### More information

More information about Microsoft Visual Studio 2010 is available at the Microsoft website <http://go.microsoft.com/fwlink/?LinkId=267132>

Here you can discover what is new in Microsoft Visual Studio Professional, version 2010, the versions that are available and how to obtain them.

For more information on Visual Studio products and solutions, including trials, MSDN subscriptions, partners, case studies and more, visit <http://go.microsoft.com/fwlink/?LinkId=267133>

For additional resources on getting started with Visual Studio products, visit <http://go.microsoft.com/fwlink/?LinkId=267134>

## Anatomy of Reports

This lesson describes the roles and components of reports in Microsoft Dynamics NAV 2013. It describes some general concepts related to report objects.

### Report Purposes

Reports can print information from a database and they can structure and summarize information. Reports can also include calculations or expressions. Reports are used to print documents such as invoices and to process data without printing anything.

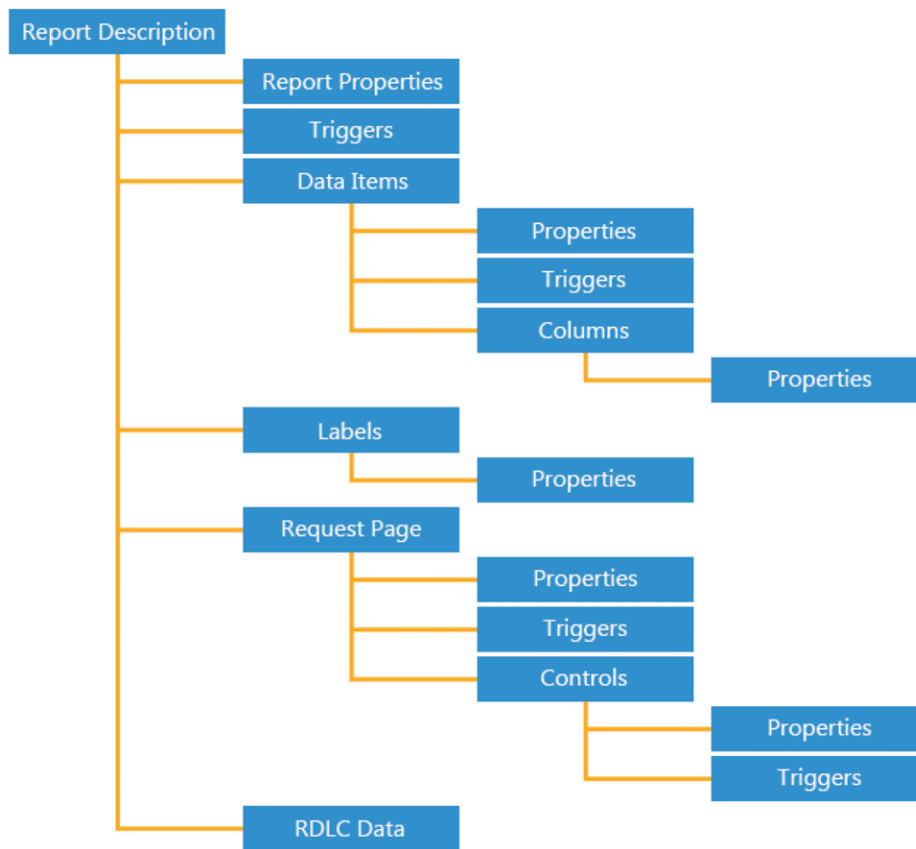
Creating reports includes designing both the business logic that covers the kind of information the report will contain, and the layout that deals with how the report will look when it is printed. Additionally, reports developed in Microsoft Dynamics have several purposes that include the following.

- Print information from a database in a structured way. Typical examples are the **Customer - List** and the **Customer - Order Detail** (reports 101 and 108).
- Print a company's official documents. For example, report 206, **Sales - Invoice** prints an invoice together with its relevant information.
- Automate recurring tasks, such as updating all prices in the item list. This can be performed by using C/AL code in a codeunit. However, using a processing-only report makes it easier because of the powerful data modeling available for report design. For example, report 794, **Adjust Item Costs/Prices**.

A report consists of a report description. The report description specifies how data is collected (the logical design or the dataset) and presented on screen or on paper (the visual design or layout).

The report description includes both the logical and the visual design and is not directly visible to the user.

The following illustration shows components of a report and how they are related in Microsoft Dynamics NAV 2013.



**FIGURE 1.2: REPORT DESCRIPTION DIAGRAM**

A report object is composed of a report dataset and a visual layout. You design a report by first defining the dataset and then designing the visual layout. Report objects also contain properties, triggers, code, and an optional request page.

### Summary

In Microsoft Dynamics NAV 2013, to design a client report definition (RDLC) report, you design the data model with Report Dataset Designer and you design the layout with Visual Studio Report Designer.

After you design reports, you can make them available in client applications. For example, you can add a report to the **Report** tab of a page. When you preview a report from a client computer, the report is displayed using Report Viewer.

## The Structure of Reports in Visual Studio

The starting point for designing RDLC report layouts is the Object Designer.

Reports run in Microsoft Dynamics NAV 2013 are designed in the Microsoft Dynamics NAV 2013 development environment. However, the RDLC report layout is built in Visual Studio.

The Visual Studio Report Designer relies completely on the dataset you define in the Report Dataset Designer.

### Defining the Dataset

You use Report Dataset Designer in the Microsoft Dynamics NAV development environment to define the dataset of a report. You build the report dataset from data items and columns. A data item is a table. A column can be one of the following.

- Field in a table
- Variable
- Expression
- Text constant

Typically, data items correspond to fields in a table. When the report is run, each data item is iterated for all records in the underlying table with an appropriate filter defined.

When a report is based on more than one table, you must set relations between the data items so that you can retrieve and organize the data. In Report Dataset Designer, you indent data items to establish a hierarchy of data items and control how the information is gathered.

For example To create a report that displays a list of customers and lists the sales orders that were placed by each customer, you must define the following data items:

- A data item that corresponds to the Customer table.
- A data item that corresponds to the Sales Order table.

You indent the second data item, which is the Sales Order table. As the report works through the records in the Customer table, it finds each customer's sales orders by examining the records in the Sales Order table.



### Properties

The report description contains properties that are related to the report and the properties that are related to the other components of the report (dataset, columns, request page and controls).

When designing reports in Visual Studio, you can also define a number of properties. These properties are saved in the RDLC data.

### Triggers

Triggers represent predefined events that occur during report execution and that cause the system to execute specific user-definable C/AL statements.

Triggers in a report can be divided into the following categories.

- Report triggers
- Dataset triggers
- Request page triggers
- Request page control triggers

### Designing the Visual Layout

You build the visual layout of a report by arranging data items. A report that is displayed or printed must have a client report definition (RDLC) layout. You use Visual Studio Report Designer to design the RDLC layout. Microsoft Dynamics NAV 2013 supports RDLC 2008. You generally display most data in the body of a report, and you use the header to display information before any data item record is displayed. For example, you can display a report title, company, and user information in the header of a report.

With Visual Studio Report Designer, you can add useful features to your report layouts, such as:

- Links from a field on a report to either a page or another report.
- Images and graphs.
- The ability to toggle columns so that you can hide or display data.
- The ability for users to interactively change the column on which data in the report is sorted.
- The ability to display RTF text.

A report in Visual Studio always has exactly one Body, and it is not possible to add more than one. Optionally it can have one Page Header and one Page Footer. Extra headers or footers cannot be added.

You can enable the Header and Footer sections by right-clicking in a blank report area, then enabling or disabling the Page Header and Page Footer sections. When disabling a Page Header or Page Footer, the entire section is deleted.

When the report runs, it first runs the Page Header, then the Page Body, and then the Page Footer. It will not run the Page Body for each record. Looping through records is done by using a **Data Region** in the body section. More information on Data Regions is discussed later in this material.

However, now it is possible in RDLC2008, in the Page Header sections, to work with data fields. This is different from RDLC2005. The module on adding code to a report describes how to use expressions and custom coding to work with data fields in the header and footer.

Reports use a variety of report items to organize data on a report page. Report items are available in the toolbox. You drag report items to the report design surface and then drag report data from the **Report Data** pane to the report items on the design surface. The design surface is not What You See Is What You Get (WYSIWIG). The report items have an initial layout position that can change when the report is processed. The following list describes typical uses for different report items:

### **Textbox**

Use for titles, date stamps, and report names. Placeholder text appears in text boxes where you have linked data and layout items. The placeholder text is display text for an underlying report expression. During report processing, report data is substituted for the report expressions and the calculated values replace the placeholder text.

### **Table, Matrix**

Use to display tabular or crosstab data from a report dataset. Table and Matrix are templates of a Tablix data region and provide a starting grid layout for data from a report dataset.

### **Chart**

Use to graphically display data from a report dataset. You can also nest a chart inside a Tablix data region.

### **Gauge**

Use to present a visual image for a single value within a range of values.

### List

Use to create a free-form layout such as the forms on a webpage. A List is a template of a Tablix data region and provides a starting layout for data from a report dataset.

### Image

Use to add existing images to a report.

### Line

Use lines as graphical elements.

### Rectangle

A rectangle can be used as a container for other report items. Use a rectangle to help control how report items such as data regions appear on a report page when the report is rendered.

A typical report uses Tablix data regions. You drag fields from the **Report Data** pane and drop them on a Tablix cell for a data region, or to the **Grouping** pane to define groups for the selected data region.

## Rules for Flattening Data

The RoleTailored client report works with a flattened dataset. This means that data from multiple tables is joined before it is sent to the client.

In the report dataset designer there can be multiple data items that might be indented or not. Depending on the indentation and relations between data items, a resultset is generated at runtime. This resultset is the result of the flattening of the indented data items into a two dimensional resultset.

The rules for flattening data can be described as follows.

- **Reports with one data item:** This corresponds to a single SELECT Column FROM Table statement. The result can be displayed in the body section.
- **Reports with indented data items:** This is similar to SELECT Column FROM Table A INNER/OUTER JOIN Table B ON Table A.Column = Table B.Column. If PrintOnlyIfDetail is set to True on the parent, an INNER join is used. If PrintOnlyIfDetail is set to False on the parent data item, an OUTER JOIN is used. The resulting set can then be displayed using grouping and filtering.

- **Reports with multiple data items on the same level:** This is similar to a UNION statement. Again, grouping and filtering can be needed to display the correct records. The difference with a UNION statement is that both data items will have different fields and a different number of columns. Actually, it is more like a concatenation of multiple resultsets.

The following dataset generation in both clients is explained based on an example. In this example, three posted sales invoices will be printed. The first sales invoice header, Invoice No. 1001, has four sales invoice lines of (Type = Item) to which two different value added tax (VAT) percentages apply. The second sales invoice (1002) has one line and is sold without VAT. The third invoice (1003) has one line, to which one single VAT percentage applies.

The RoleTailored client works with a flattened dataset; this means that tables will be joined (in reports with multiple tables). The result of the queries will be one single dataset containing information from multiple tables. Once the dataset is available, the dataset can be processed using filtering and sorting and the report can be rendered. In the following illustration, the dataset will look like this (not all fields from the sales invoice report are included):

No	Description	No	Description	Line Amount	VAT Type	VAT Amount
1001	Invoice 1001	1	Line 1	400	null	null
1001	Invoice 1001	2	Line 2	300	null	null
1001	Invoice 1001	3	Line 3	200	null	null
1001	Invoice 1001	4	Line 4	100	null	null
1001	Invoice 1001	null	null	null	VAT10	100
1001	Invoice 1001	null	null	null	VAT25	250
1002	Invoice 1002	1	Line 1	1500	null	null
1003	Invoice 1003	1	Line 1	2000	null	null
1003	Invoice 1003	null	null	null	VAT10	200

**FIGURE 1.3: DATASET GENERATION**

The dataset combines information from three tables. For invoice 1001 six lines are included. Invoice 1002 has only one line, and invoice 1003 has two lines.

The first four lines for invoice 1001 relate to the four sales invoice lines. The two other lines represent the VAT specification lines for the invoice. To generate the dataset, the logical model of the report is used.

To select records from a single table, a simple SELECT statement is used.

In this case, if the report contains several unindented data items, the tables will be merged using a **UNION** statement. The resulting recordset contains the fields from all data items in the report.

When selecting records from multiple linked tables, the results of the SELECT statements must be concatenated. As a result, in reports with indented tables, data items will first be joined using the conditions specified in the **DataItemLink** property. In this case, the Sales Invoice Header and the Sales Invoice Line table will be joined, and all relevant fields will be selected.

After handling the sales invoice lines, the report will proceed to the next data item. (In this example, it is assumed that no dimensions are specified.) Next, it will try to retrieve the VAT Amount Lines. Again, the Sales Invoice Header and the VAT Amount Line data items will be joined and the relevant fields will be selected from the data item.

So far, several recordsets are now generated in the example. As a final step, these smaller recordsets will be merged through a UNION statement. As a result, the Sales Invoice Line and the VAT Amount Line recordsets will be merged into a recordset containing fields from all three tables, which results in the dataset that is shown.

## Creating the Dataset

This lesson explains how to add data items and columns in the dataset designer.

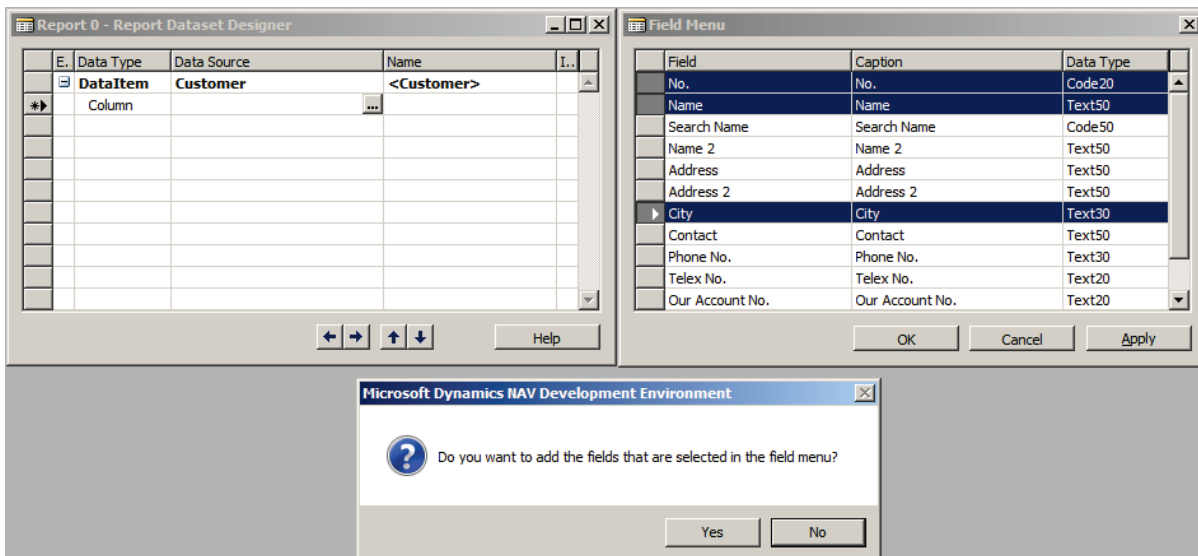
### Creating a New Report

To create a new report you will create a blank report and use Report Dataset Designer to add items to the dataset. Then you will use Visual Studio Report Designer to design the layout of the report.

As an example, you will start by creating a simple list report,

1. In the development environment, on the Tools menu, select Object Designer, or use the shortcut Shift-F12.
2. In the Object Designer, click the **Report** button on the left side of the window, and then click the **New** button at the bottom of the Object Designer.
3. The report dataset designer now opens. In the report dataset designer, the first line contains **DataItem** in the **Data Type** column. In the next column, **Data Source**, you can enter a table name or table number directly or use the lookup button to select the table from the list of tables.

4. In the next column, **Name**, you can enter the name of the data item. You can leave the default or enter a new name.
5. For example, you can enter 18 as the table number as the Data Source. The system will, after you leave the column, update the table number to the table name: Customer. In the **Name** column it will set <Customer>.
6. To define the fields that you want to include from the table into the report dataset, you must create a line for every field from the table that you want to include.
7. In the designer, on the first empty line, in the **Data Type** column, select **Column** from the drop-down list.
8. In the **View** menu, select the **Field** menu. The **Field Menu** window opens, and it displays a list of fields that you can add to the report dataset. You can select one or multiple fields in the field menu by holding down the SHIFT or CONTROL key. Click the **OK** button to add the selected fields in the **Field** menu to the report dataset.



**FIGURE 1.4: FIELD MENU WINDOW**

9. To include the caption of a column in the dataset, select the **Include Caption** check box. It is the last column in the report dataset designer. The caption will become available as a parameter in Visual Studio Report Designer.
10. Repeat steps 1 through 9 in the report dataset designer to add data items and columns until you have added all the fields you want to include in the report. For example, select the **No.**, **Name**, **Address**, **Phone No.** and **Country/Region Code** fields.
11. Make sure to select the Include Caption for all of the rows in the Report Dataset Designer.

12. On the **File** menu click **Save**.
13. In the **Save As** dialog box, in the **ID** field, enter an ID for the report. In the **Name** field, enter a Name for the report.
14. Click **OK**.

### Include a Label

You can add a label to the dataset, for example, if you want to add a title to the report, or a caption for a table.

You can either add labels that are related to another data item, such as the caption for a table field, or labels that are not related to another data item, such as the title of the report.

To add a label that is related to a column, use the following steps.

1. In Report Dataset Designer, do one of the following:
  - a. Select the **Include Caption** check box for a Column data type row to add a label for the column.
  - b. Select a Column data type row for which you want to add a label to the dataset, and on the **View** menu, choose Properties. In the **Properties** window, set the **IncludeCaption** property to Yes.
2. Compile and save the report.

To add a label that is not related to a column follow the next steps.

1. In Report Dataset Designer, on the **View** menu, choose Labels.
2. In Report Label Designer, in the **Name** field, add a Common Language Specification (CLS)-compliant name for the label. This name is used in the Visual Studio Parameters.
3. In the **Caption** field, enter the caption.
4. On the **View** menu, choose Properties.
5. If you have a Multilanguage application, then in the **Properties** window specify the values for the **CaptionML** property.
6. Compile and save the report.



**Note:** *The Common Language Specification (CLS), is a set of basic language features needed by many applications to fully interact with other objects regardless of the language in which they are implemented.*

*More information about the Common Language Specification, visit:*

<http://go.microsoft.com/fwlink/?LinkId=267137>

---

### Include a Variable

If you want to include a variable in the report dataset, then do the following.

1. In Report Dataset Designer, on a blank line, in the Data Type column, select Column from the drop-down list.
2. In the Data Source column, enter the variable. You can use the **Lookup** button to open the C/AL Symbol Menu dialog box and view the variables for the report.
3. In Report Dataset Designer, in the Name column, enter a CLS-compliant name for the variable. This name is used in the dataset in Visual Studio.

### Include an Expression

If you want to include a calculation or other source expression in the report dataset, then do the following.

1. In Report Dataset Designer, on a blank line, in the Data Type column, select Column from the drop-down list.
2. In the Data Source column, enter the source expression, such as `FORMAT(TODAY,0,4)`.
3. In the Name column, enter a CLS-compliant name for the variable. This name is used in the dataset in Visual Studio.

## Creating the Layout

This lesson covers how to create a layout in Visual Studio – (RDLC) It will show this by demonstrating how to create a simple list report.

### Creating an RDLC Layout

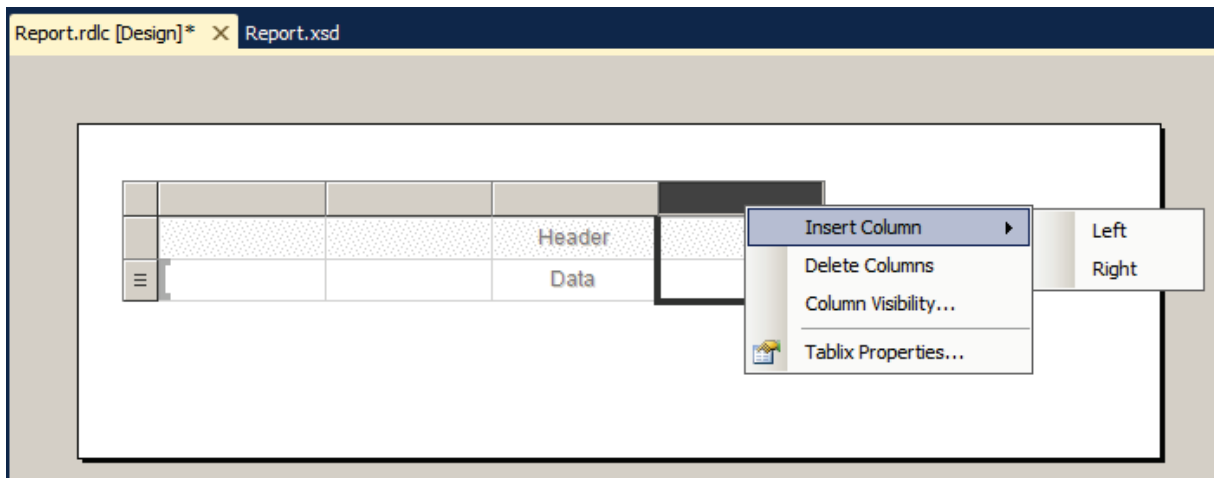
The previous lesson showed how to create the dataset for a report, which includes the following fields: **No.**, **Name**, **Address**, **Phone No.** and **Country/Region Code** from the Customer table.

After you save and compile the report in Report Dataset Designer, you must now design the layout in Visual Studio. For this report, you will display all the fields that you selected and display the company name in a larger font and then set it to bold. You will also display the date, time and username of the person that executed the report.



To create the RDLC layout for the report, follow these steps.

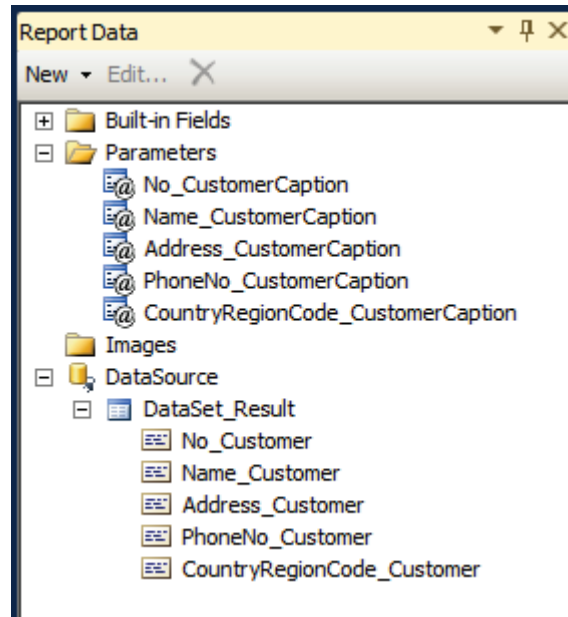
1. On the **View** menu, choose Layout.  
Visual Studio opens. The dataset in Visual Studio contains the dataset items that you added in Report Dataset Designer.
2. From the toolbox, drag a Table control to the report layout.
3. The table has only three columns so you will add two more columns to accommodate all the fields in the dataset.
4. Right-click the third column and select Insert Column and then select Right to add a new column.
5. Repeat step 4 to add another column.



**FIGURE 1.5: ADD A COLUMN TO A TABLE WINDOW**


The next step is to populate the table with the data from the dataset.

6. On the View menu, click Report Data. The Report Data window opens.

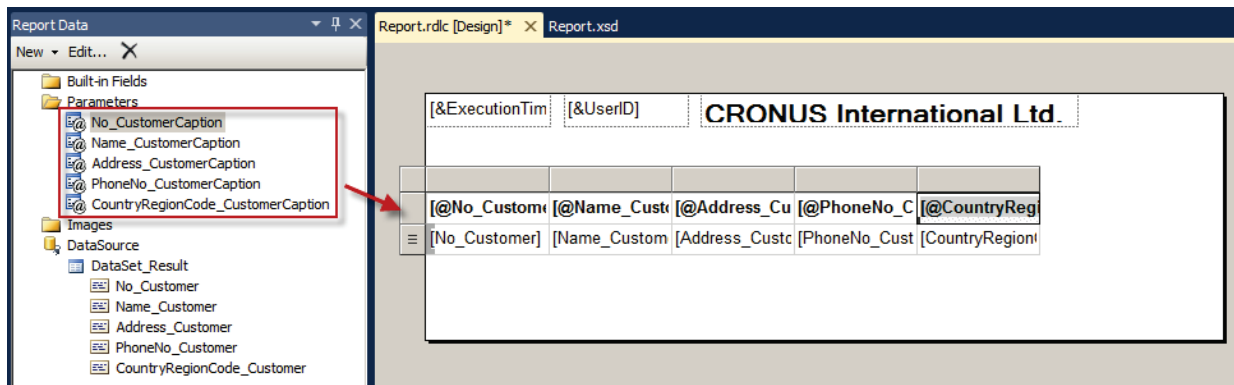


**FIGURE 1.6: REPORT DATA WINDOW**

7. From the **Report Data** window, under **DataSource**, and under **DataSet\_Result**, drag the **No.** field to the first column, second row of the table in the Data section of the report layout.
8. Under **Data Sources**, and under **DataSet\_Result**, drag the **Name** field to the second column, second row of the table in the Data section.
9. Under **Data Sources**, and under **DataSet\_Result**, drag the **Address** field to the third column, second row of the table in the Data section.
10. Under **Data Sources**, and under **DataSet\_Result**, drag the **Phone No.** field to the fourth column, second row of the table in the Data section.
11. Under **Data Sources**, and under **DataSet\_Result**, drag the **Country\_Region\_Code** field to the fifth column, second row of the table in the Data section.
12. From the toolbox drag three text boxes and place them above the table in the designer. These text boxes will display the date and time, the username and the company name.
13. Select and right-click the first text box and click **Expression** to open the **Expression Properties** window.
14. Under **Category**, click **Built-in Fields**, double-click **ExecutionTime** and click **OK**. This expression displays the time the report is executed.
15. Select and right-click the second text box and click **Expression** to open the **Expression Properties** window.
16. Under **Category**, click **Built-in Fields**, double-click **UserID** and click **OK**. This expression displays the user name for the user that executed the report.


 **Note:** It is recommended to add the UserID as a column to the dataset. By doing this, the Microsoft Dynamics NAV user ID will display, instead of the Windows user ID of the service tier, in a three tier environment.

17. Select and right-click the third text box and then click **Text Box Properties** to open the **Text Box properties** window.
18. Under the **General** tab, in the **Value** drop-down list, enter the company name. For this walkthrough, you can enter CRONUS International Ltd. and click **OK**.
19. Click **Font** and set the font size to a larger font and under Style, select Bold and click **OK**. For example, you can select 16 for the font size.
20. On the header row of the table, populate the text boxes with the Captions.
21. In the report dataset window, expand the Parameters folder.
22. Select the No\_CustomerCaption and drag it onto the first text box of the header row.
23. Repeat this step for the other Captions as in the following screenshot:

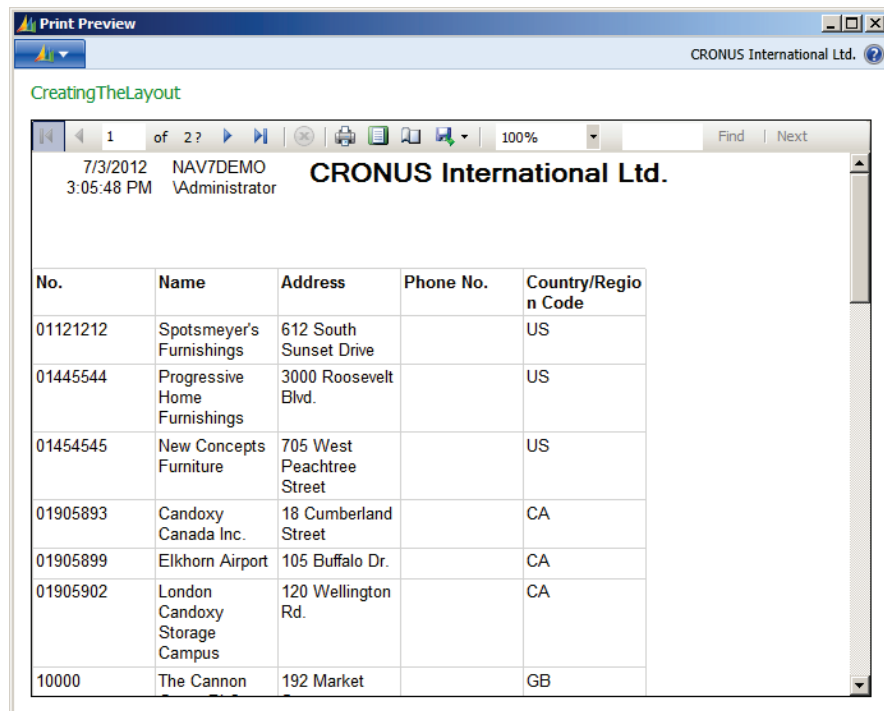


**FIGURE 1.7: ADD CAPTIONS TO THE TABLE WINDOW**

24. You will format the text boxes for the field captions in the table to display in bold font.
25. Select and right-click the **No. caption** text box (first column and first row) and click **Text Box Properties** to open the **Properties** window.
26. Click the **Font** tab, and under Style, select the **Bold** check box and then click **OK**.
27. For the text boxes in the first row, set the captions to bold.
28. Close Visual Studio to go back to C/Side.

 **Note:** Visual Studio can remain open in case report development is not completely finished and you want to test and run the report.

29. In Report Dataset Designer, select a new row. A dialog box shows the following message:  
The layout of report id: ' <report ID> ' is changed by another application.  
Filename of the changed RDLC: ' <filename> '  
Do you want to load the changes?
30. In the dialog box, choose **Yes** to load the changes that you made to the RDLC layout into the report.
31. In the development environment, on the **File** menu, choose **Save**.
32. In the **Save** dialog box, select **Compiled**, and then choose **OK**.
33. To run the new report, in Object Designer, select the report and then choose **Run**.
34. The result will look like this:



No.	Name	Address	Phone No.	Country/Region Code
01121212	Spotsmeyer's Furnishings	612 South Sunset Drive		US
01445544	Progressive Home Furnishings	3000 Roosevelt Blvd.		US
01454545	New Concepts Furniture	705 West Peachtree Street		US
01905893	Candoxy Canada Inc.	18 Cumberland Street		CA
01905899	Elkhorn Airport	105 Buffalo Dr.		CA
01905902	London Candoxy Storage Campus	120 Wellington Rd.		CA
10000	The Cannon	192 Market		GB

**FIGURE 1.8: CUSTOMER LIST REPORT WINDOW**

## NAV 2013 UX Guidelines for Microsoft Dynamics NAV 2013

The following information is to be considered as general guidelines and recommendations regarding the user interface of Microsoft Dynamics NAV 2013 reports. The information used in the following table is shown in centimeters (cm), inches (in) and points (pt).

### Page Setup:

	Left	Right	Top	Bottom
<b>A4 Portrait</b>	2 cm	1 cm	1 cm	1 cm
<b>A4 Landscape</b>	1 cm	1 cm	2 cm	1 cm
<b>Letter Portrait</b>	0,5 in	0,5 in	0,5 in	0,5 in
<b>Letter Landscape</b>	0,5 in	0,5 in	0,5 in	0,5 in
<b>Tabloid Portrait</b>	0,5 in	0,5 in	0,5 in	0,5 in
<b>Tabloid Landscape</b>	0,5 in	0,5 in	0,5 in	0,5 in

### Body size:

	Width	Height	
<b>A4 Portrait</b>	18 cm	27,7 cm	(785 pt)
<b>A4 Landscape</b>	27,7 cm	18 cm	(510 pt)
<b>Letter Portrait</b>	7,5 in	10 in	(720 pt)
<b>Letter Landscape</b>	10 in	7,5 in	(540 pt)
<b>Tabloid Portrait</b>	10 in	16 in	(1152 pt)
<b>Tabloid Landscape</b>	16 in	10 in	(720 pt)

### Report title:

Font	Segoe UI 14 pt Bold
Cell height	20 pt
Padding	Cell padding: left: 5 pt, right: 5 pt, top: 0 pt, bottom: 0 pt
Text alignment	Document reports: Right; All other reports: Left

## Other text in the header:

Font	Segoe UI 8 pts Regular
Cell height	10 pt
Padding	Cell padding: left: 5 pt, right: 5 pt, top: 0 pt, bottom: 0 pt

## Space to Body:

- Height: 20 pt

## Headers in the body:

Font	Segoe UI 8 pt Bold
Cell height	10 pt
Padding	Cell padding: left: 5 pt, right: 5 pt, top: 0 pt, bottom: 0 pt

## All other text:

Font	Segoe UI 8 pts Regular
Cell height	10 pt
Padding	Cell padding: left: 5 pt, right: 5 pt, top: 0 pt, bottom: 0 pt

## Line spacing:

- Grid 10 pt. Multiplication of 10s. (As a designer/developer, you can use your discretion to space the content.)

## Table Header:

- Row1: Title: Bold
- Row2: Height: 2,5pt: Border bottom; black; solid; width: 1 pt
- Row3: Height: 2,5pt
- Row4: Text
- Header row can be up to 30 pt in height if required (allowed to be up to 3 lines)
- Vertical align: bottom

### Table content:

- Cell height is 10 pt
- Cell padding: left: 5 pt, right: 5 pt, top :0 pt, bottom: 0 pt
- Vertical align for cells: top (default)
- If row below needs to be indented, add +10 to Left Padding

### Cell colors:

- List reports: Alternate BackgroundColor= mod 2, "WhiteSmoke", "White"

### Logo:

- Height: 40 pt
- Space below Logo: 10 pt

### Text alignment:

Numeric/decimal fields	right aligned (including header)
Text fields	left aligned
Numeric text	left aligned (eg. customer no)
Date/time	left aligned

### Link color:

- Blue (no underline)

### Chart color:

- Palette: BrightPastel

---

 **Additional Reading:** More information, including examples, can be found here: <http://go.microsoft.com/fwlink/?LinkId=267138>

---

## Demonstration: How to View the Dataset Using the "About This Report" Feature

When you run a report, to view the underlying, flattened, dataset that is generated by Microsoft Dynamics NAV, you can enable an option at runtime in the report viewer.

## Demonstration Steps

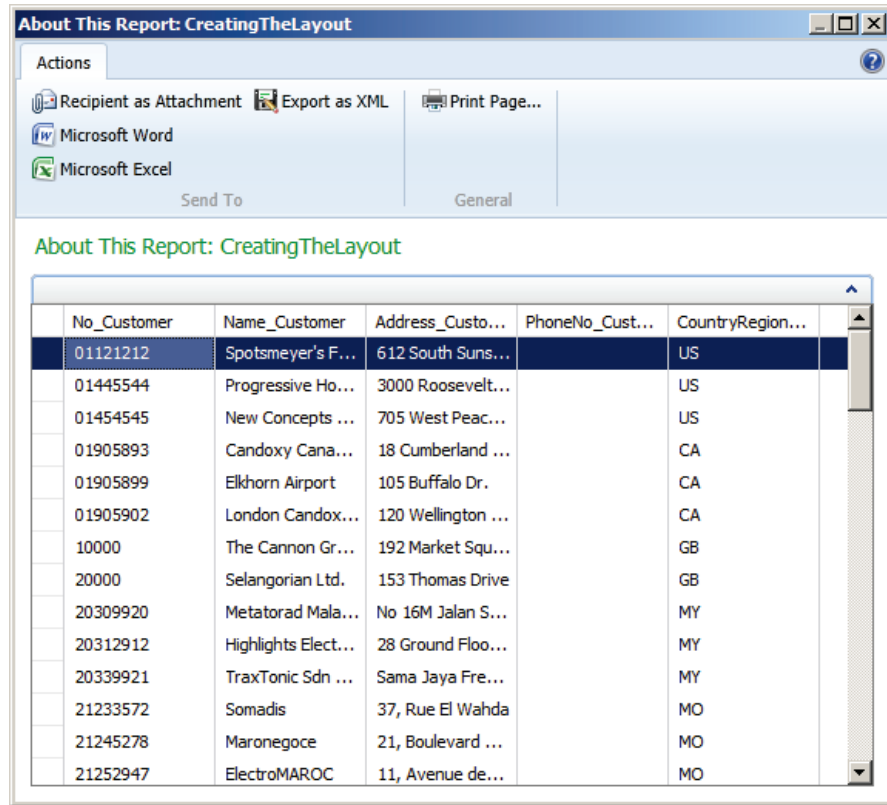
1. First, you must run the report. Then when the report is being executed and displayed in the report viewer, you must enable the About This Report feature.  
This can be achieved by performing the following procedure.
  - a. In the report viewer, click the menu at the top-left of the window.
  - b. A drop-down menu appears.
  - c. In this drop-down menu select the Help, About This Report option. Alternatively, you can use the shortcut: Ctrl+Alt+F1.

2. A popup window now appears containing the following message:

*The About This Report feature is now active. Run your report again to view the dataset.*

- a. Close the popup window.
- b. Close the report viewer.
- c. Run the report again.
- d. Repeat steps 1 through 3.
- e. A popup window now appears that displays the dataset of the report.





**FIGURE 1.9: ABOUT THIS REPORT WINDOW**

- In the popup window, a column will appear for every field from the report dataset and a row for every record.

In the Action Pane of the popup window you will find the following actions.

  - Recipient as Attachment:

This action will create a new email and include the dataset as an HTML attachment.
  - Export as XML:

This action will export the dataset to an XML file.
  - Microsoft Word:

This action will export the dataset to a Microsoft Office Word document.
  - Microsoft Excel:

This action will export the dataset to a Microsoft Office Excel document.

- Print Page:

This option will send the dataset to the printer.



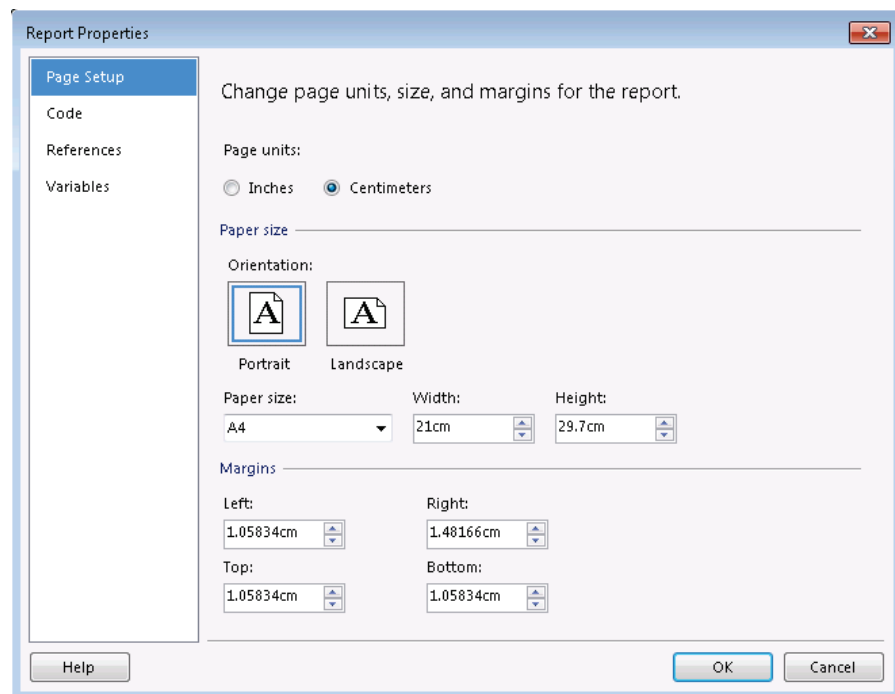
**Note:** Once the feature is enabled for a report, it will remain so until you close the client. It is recommended not to enable this feature for every report, because the dataset is kept in memory and will consume extra resources on the client.

## Demonstration: How to Set Page Options

To change the dimensions of the design area for a report, specify the settings you want to use in the Page Setup.

### Demonstration Steps

1. To Set Page Options:
  - a. Select Page Setup on the Report Properties dialog box to set page options for the entire report.
  - b. You can do this via the Menu, Report, Report Properties and then select Page Setup on the left.



**FIGURE 1.10: PAGE SETUP WINDOW**

- **Inches:** Specifies the page size in inches. By default, your local settings determine the page unit that is selected.
- **Centimeters:** Specifies the page size in centimeters. By default, your local settings determine the page unit that is selected.

- **Portrait:** Specifies the page orientation of the report. Portrait indicates that the page is longer vertically.
- **Landscape:** Specifies the page orientation of the report. Landscape indicates that the page is longer horizontally.
- **Paper size:** Select a size from the list that automatically enters the dimensions in the Width and Height controls. Alternatively, select Custom Size and then change the Width and Height controls to reflect the size you want to use.
- **Width:** Displays the width of the paper size that is selected in the Paper type list.
- **Height:** Displays the height of the paper size that is selected in the Paper type list.
- **Top:** Sets the distance between the top of the report and the top of the first line of the report.
- **Bottom:** Sets the distance between the bottom of the report and the bottom of the last line of the report.
- **Left:** Sets the distance between the left edge of the report and the left edge of the report rows.
- **Right:** Sets the distance between the right edge of the report and the right edge of the report rows.

### Demonstration: How to Set the Paper Size

When you print a report, you can specify the size of the paper to use. The size of the paper determines the number of pages in a report and the report data that fits on each page. Paper size affects only reports that are rendered with the hard page-break renderers: PDF, Image, and Print. Setting the paper size has no effect on other renderers.

From the **Report Viewer** toolbar, you can export a report to a hard page-break renderer or click the **Print** button to print a copy of the report. You might need to set the paper size or other page setup properties. Use the **Report Properties** dialog box to change page setup properties, including the paper size.

More information about pagination is provided in module 4, Design Considerations.

### Demonstration Steps

1. To change the paper size, perform the following steps.
  - a. Right-click outside of the report body and click **Report Properties**.
  - b. In **Page Setup**, select a value from the **Paper Size** list.
  - c. Each option populates the **Width** and **Height** properties.

- d. You can also specify a custom size by typing numeric values in the **Width** and **Height** boxes.
- e. Click **OK**.

Size values have a default unit based on the user's local settings. To designate a different unit, type a physical unit designator such as **cm**, **mm**, **pt**, or **pc** after the numeric value.

### **Demonstration: How to Set the Body to Fit**

As demonstrated in the previous topic, you can set the paper size of the report in the report properties. It is recommended to also set the size of the Body to fit the report size.

When you are adding report items to the body of a report, you will notice that the Body of the report might grow. Also, when you add a column to a Tablix, the Tablix data region will grow horizontally. This might increase the size of the body, and the size of the report body might become greater than the size of the report. As a result, when you run the report, the body will be split over multiple pages.

More information about report pagination is available in the Interactive Features module.

### **Demonstration Steps**

1. To do this, perform the following steps.
  - a. In the drop-down list at the top, select **Report**.
  - b. Copy the **PageSize** properties using your mouse or **CTRL-C**.
  - c. In the drop-down list at the top of the Properties window, select **Body**.
  - d. In the **Size** properties, paste the copied values using your mouse or **CTRL-V**.

## **The Microsoft Visual Studio Integrated Development**

Since the creation of the RDLC layout for reports in Microsoft Dynamics NAV 2013 is managed in Microsoft Visual Studio 2010 it is important to have a good understanding of Microsoft Visual Studio 2010.

The Visual Studio product family shares a single integrated development environment (IDE) that is composed of several elements: the **Menu** toolbar, **Standard** toolbar, various tool windows docked or auto-hidden on the left, bottom, and right sides, and the editor space. The tool windows, menus, and toolbars available depend on the type of project or file you are working in.

### Menu Bar

The menu bar appears at the top of the IDE and gives access to all features and commands in the Visual Studio IDE. Some familiar menu names are **File**, **Edit**, **View**, **Tools**, **Window** and **Help**. Each menu contains commands and command groups. Commands are functions that either launch submenus or accomplish tasks, such as printing a document, refreshing a view, or creating a new file. Depending on the type of project you are creating, new menus can appear (Format, Website, Data, and Report).

### Toolbars

Toolbars are normally displayed at the top of the IDE, under the menu bar, but can be moved and docked to the left, right or bottom side of the IDE.

Toolbars typically are rows of buttons and other controls (such as combination boxes, list boxes, text boxes, and splitter controls) that are associated with commands. When you click a toolbar button, its associated command is activated. Toolbar buttons usually have icons that suggest the underlying commands, such as a printer for a Print command.

Like menus, toolbars are convenient graphical ways to present commands to users. Typically, related commands are clustered together on the same menu or toolbar.

By default, not all available toolbars are displayed. To add additional toolbars, select **View**, **Toolbars** and check the toolbars to display. From the same menu, you can also choose to create new toolbars.

### Editors and Designers

Microsoft Visual Studio includes multiple editors and designers that depend on the type of file or document that you are authoring. The Text editor is the basic word processor of the IDE, while the Code editor is the basic source code editor.

Other editors, such as the CSS editor, the HTML Designer, and the Web Page Designer, share many of the features found in the Code editor, along with enhancements specific to the type of code or markup supported.

Editors and designers usually have two views: a graphical design view and the code behind view (or source view). Design View lets you specify the location of controls and other items on the user interface or webpage. You can easily drag a control from the toolbox and place it on the design surface.

Source View displays the source code for the file or document. This view supports coding help such as IntelliSense, collapsible code sections, code refactoring, and code snippet insertion. Other features include bookmarks and displaying line numbers.

Some editors, such as the Web Page Designer and HTML Designer, also provide a hybrid view where both the graphical and code view of a file are visible simultaneously. This view is called the Split View.

In the editor, you can have multiple files opened at a time. For each file that is opened in a project or solution, the code editor will contain a tab. You can easily switch from one file to another by clicking the tab.

### **Solution Explorer**

The Solution Explorer displays solutions, their projects, and the items in those projects. From Solution Explorer, you can open files for editing, add new files to a project, and view solution, project, and item properties.

### **Properties Editor**

The **Properties** window contains all properties for the item that is selected in the editors or designers. At startup, the window will show the properties of the field that is selected in the Solution Explorer. The contents of the window will be refreshed when you select another file in the Solution Explorer, or any object in the designers (text box, table row, and so on). When working in the code editor, the name of the code trigger being edited is visible.

### **Toolbox**

The toolbox contains a list of controls that you can add to the design view of the project. The contents of the toolbox will vary depending on the type of project you are editing. To add controls from the toolbox to your project, drag and drop it to the Design view. You can add new controls to the toolbox by right-clicking the **toolbox** and selecting **Choose Items**. The toolbox window is often auto-hidden.

## **Dataset Patterns**

Reports in Dynamics NAV 2013 are executed in two steps, which reflect the two steps in the report design: first data set design in CSIDE followed by report layout design in Visual Studio:

1. The C/AL runtime retrieves the data from the involved source tables, performs necessary calculations and combines the data in a single flattened dataset. This is performed by the NAV server.
2. The produced dataset is transferred to the report viewer's run-time hosted on the NAV client, which in turn renders the dataset data according to the report layout definition.

This lesson focuses on the first part, where report developers confront a particular challenge while having multiple source tables as the input, but a single consolidated data set as the output.

The following examples illustrate and explain most of the typical patterns for data set design in Dynamics NAV reports in an easy to understand way. Various combinations of these patterns appear in real-life Dynamics NAV reports.

### Demonstration Data

To illustrate we introduce two simple NAV tables, which resemble the popular header-line pattern. The tables are populated with several rows of sample data.

The "Header" table represents a simplified document header table, which has an integer "No." field as the primary key and "Customer" fields to hold the customer name.

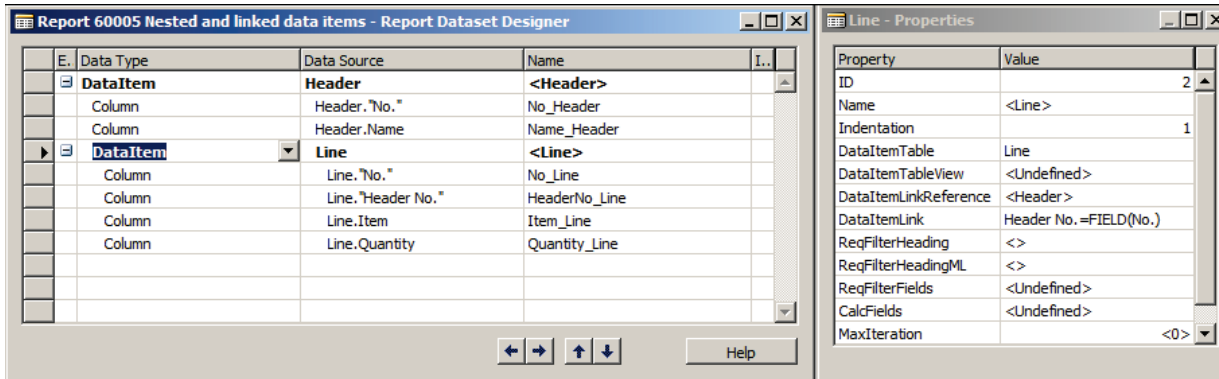
No.	Name
11	Selangorian Ltd.
22	London Postmaster
33	Meyers Design
44	Total Solutions Inc.

The "Line" table represents a simplified order line with the "No." field as the primary key; "Header No." is a reference to the "Header" table, while the "Item" and "Quantity" fields represent the payload of the order line.

No.	Header No.	Item	Quantity
111	11	Bicycle	1
222	11	Armchair	1
333	11	Toolkit	1
444	22	Armchair	2
555	22	Bicycle	2
666	44	Toolkit	4

## Linked Data Items

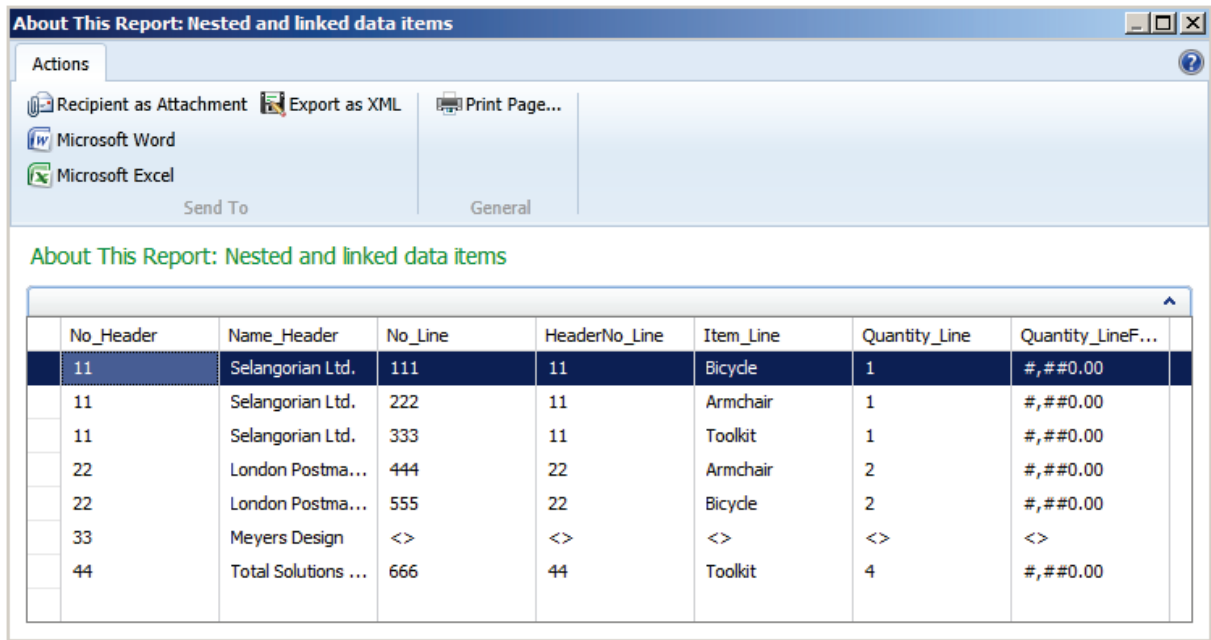
This is a very common pattern for the datasets in Dynamics NAV reports. As shown on the following screenshot, the “Line” data item is nested in “Header” data item:



**FIGURE 1.11: DATA ITEM DESIGNER WINDOW**

**Note:** The table relation, which connects Line.“Header No.” field with Header.“No.” field is defined as: Header No.=FIELD(No.)

The resulting dataset at runtime contains 7 rows as shown in the “About this Report” feature:



**FIGURE 1.12: DATASET OF REPORT WINDOW**



The fields from the header data item are repeated on the records from the related line data item. For example, header 11 is connected to 3 lines and as a consequence in each of the 3 records the header fields are repeated.

In the resulting dataset certain fields might have no values. In this example, since the header row "33" – "Meyers Design" has no single linked line, all the fields from "Line" for this header are empty (marked as null).

The produced data set is much equivalent to SQL LEFT Join operation on "Header"-"Line" tables.

### Linked Data Items with PrintOnlyIfDetails

This example has the same data set design as above, but the data item property **PrintOnlyIfDetails** property of the header data item is set to Yes.

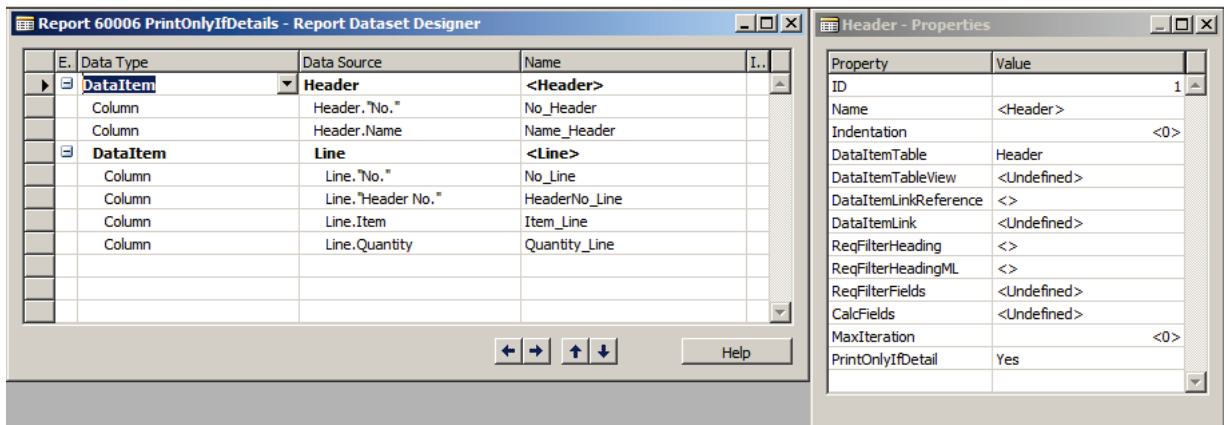


FIGURE 1.13: USING PRINTONLYIFDETAILS WINDOW

The property **PrintOnlyIfDetails** specifies whether to print data in a report for the parent data item when the child data item does not generate any output.

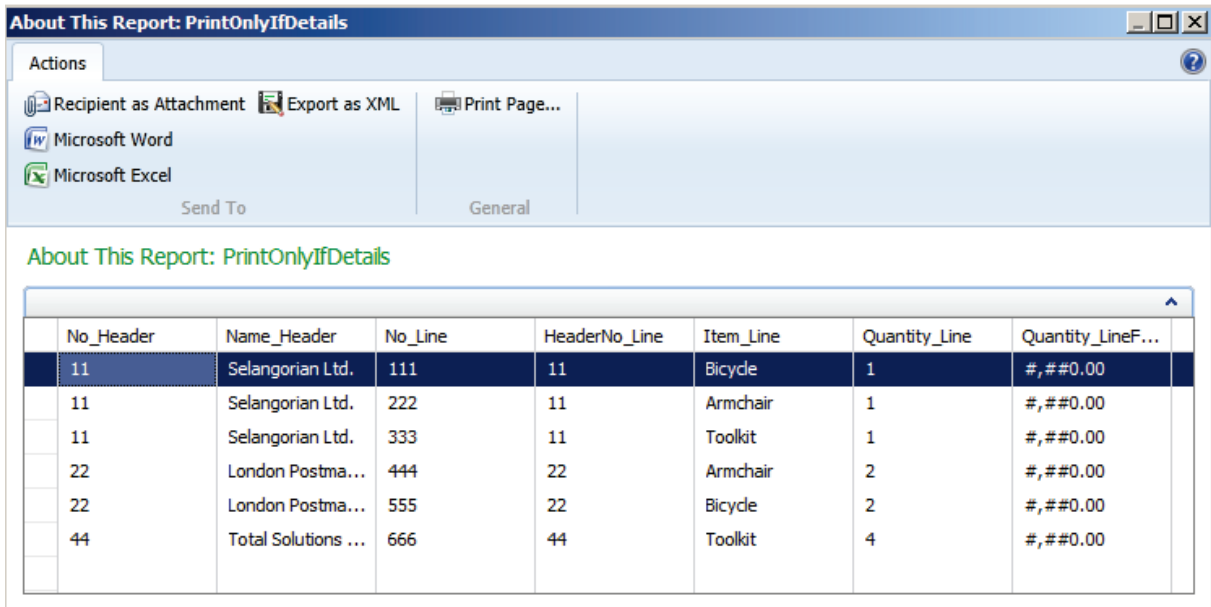
**Note:** The **PrintOnlyIfDetails** property has no effect on a data item that does not have any child data items. If this property is No and there is no record in the child data item that corresponds to the current record in the parent data item, then the report prints data from the current record in the parent data item, even though there is no data for the child data item. If this property is Yes and there is no record in the child data item that corresponds to the current record in the parent data item, then the report does not print data from the current record in the parent data item.

If there are more than two data items, then the report iterates through each parent-child relationship in the same way.

Another way to achieve the result of not printing blank lines is to add a filter on the table. For example using the **SETFILTER** Function (Record).

# Reporting in Microsoft Dynamics® NAV 2013

The resulting data set has fewer rows than above, since the Header lines without related Line records (e.g. "33" – "Meyers Design") are excluded. Subsequently there are also no "null" fields anymore.



The screenshot shows a window titled "About This Report: PrintOnlyIfDetails". It has an "Actions" bar with options like "Recipient as Attachment", "Export as XML", "Print Page...", "Microsoft Word", and "Microsoft Excel". Below this is a table with the following data:

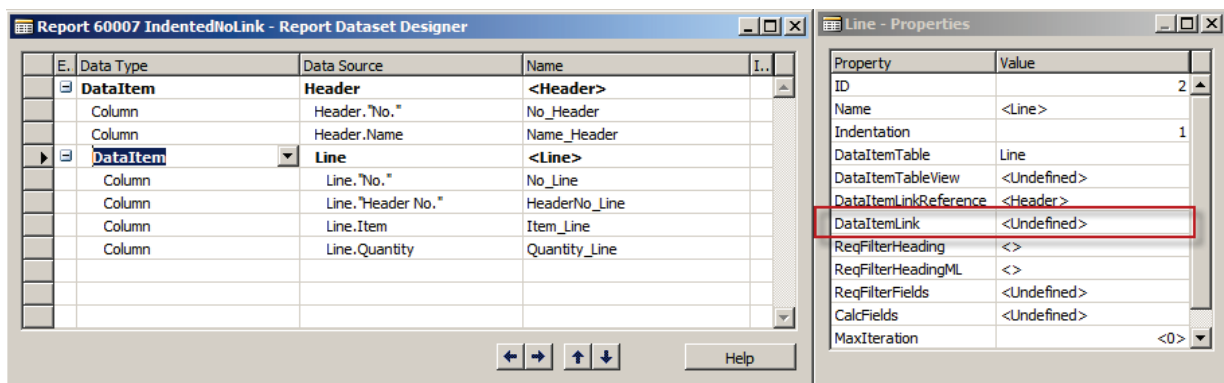
No_Header	Name_Header	No_Line	HeaderNo_Line	Item_Line	Quantity_Line	Quantity_LineF...
11	Selangorian Ltd.	111	11	Bicycle	1	#,##0.00
11	Selangorian Ltd.	222	11	Armchair	1	#,##0.00
11	Selangorian Ltd.	333	11	Toolkit	1	#,##0.00
22	London Postma...	444	22	Armchair	2	#,##0.00
22	London Postma...	555	22	Bicycle	2	#,##0.00
44	Total Solutions ...	666	44	Toolkit	4	#,##0.00

FIGURE 1.14: PRINT ONLY IF DETAILS WINDOW

The produced data set is now similar to a SQL INNER Join operation on "Header"- "Line" tables.

## Nested Data Items without Link

In this example there are still two nested data items, however the properties DataItemLinkReference and Data Item Link aren't set, which effectively disconnects lines from the headers.



The screenshot shows the "Report 60007 IndentedNoLink - Report Dataset Designer" window. It displays a tree view of data items and a "Line - Properties" window. The tree view shows a "DataItem" named "Header" with two columns, and a nested "DataItem" named "Line" with four columns. The "Line - Properties" window shows the following properties:

Property	Value
ID	2
Name	<Line>
Indentation	1
DataItemTable	Line
DataItemTableView	<Undefined>
DataItemLinkReference	<Header>
DataItemLink	<Undefined>
ReqFilterHeading	<>
ReqFilterHeadingML	<>
ReqFilterFields	<Undefined>
CalcFields	<Undefined>
MaxIteration	<0>

FIGURE 1.15: NO DATAITEMLINK WINDOW

The resulting data set is much larger, since it includes all possible combinations of headers and lines, with absolutely no regard to their possible table relations. Since in our example there are 4 headers and 6 lines, the resulting dataset will contain 24 rows.

The screenshot shows a window titled "About This Report: IndentedNoLink". It has an "Actions" menu with options: "Recipient as Attachment", "Export as XML", and "Print Page...". Below the menu are icons for "Microsoft Word" and "Microsoft Excel". The main area displays a table with the following data:

No_Header	Name_Header	No_Line	HeaderNo_Line	Item_Line	Quantity_Line	Quantity_LineFormat
11	Selangorian Ltd.	111	11	Bicycle	1	#,##0.00
11	Selangorian Ltd.	222	11	Armchair	1	#,##0.00
11	Selangorian Ltd.	333	11	Toolkit	1	#,##0.00
11	Selangorian Ltd.	444	22	Armchair	2	#,##0.00
11	Selangorian Ltd.	555	22	Bicycle	2	#,##0.00
11	Selangorian Ltd.	666	44	Toolkit	4	#,##0.00
22	London Postma...	111	11	Bicycle	1	#,##0.00
22	London Postma...	222	11	Armchair	1	#,##0.00
22	London Postma...	333	11	Toolkit	1	#,##0.00
22	London Postma...	444	22	Armchair	2	#,##0.00
22	London Postma...	555	22	Bicycle	2	#,##0.00
22	London Postma...	666	44	Toolkit	4	#,##0.00
33	Meyers Design	111	11	Bicycle	1	#,##0.00
33	Meyers Design	222	11	Armchair	1	#,##0.00
33	Meyers Design	333	11	Toolkit	1	#,##0.00
33	Meyers Design	444	22	Armchair	2	#,##0.00
33	Meyers Design	555	22	Bicycle	2	#,##0.00
33	Meyers Design	666	44	Toolkit	4	#,##0.00
44	Total Solutions ...	111	11	Bicycle	1	#,##0.00
44	Total Solutions ...	222	11	Armchair	1	#,##0.00
44	Total Solutions ...	333	11	Toolkit	1	#,##0.00
44	Total Solutions ...	444	22	Armchair	2	#,##0.00
44	Total Solutions ...	555	22	Bicycle	2	#,##0.00
44	Total Solutions ...	666	44	Toolkit	4	#,##0.00

**FIGURE 1.16: DATASET NO DATAITELINK WINDOW**

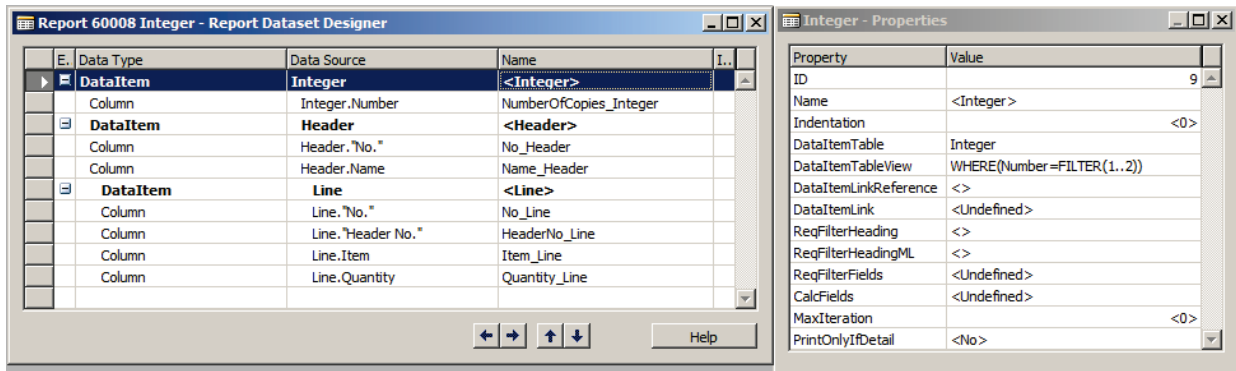
### Linked Data Items with an Integer Table

The data set design from previous examples is not very practical for header-line reports, however it is widely used in case if one of the data items refers an Integer table, for example when producing several copies of certain rows or groups of rows in the resulting dataset.

In this example we introduce another data item from the Integer table, which encloses the Header-Line data items ones. The purpose of such a design is to repeat Header-Line rows multiple times, for example to provide several copies of the same document.

# Reporting in Microsoft Dynamics® NAV 2013

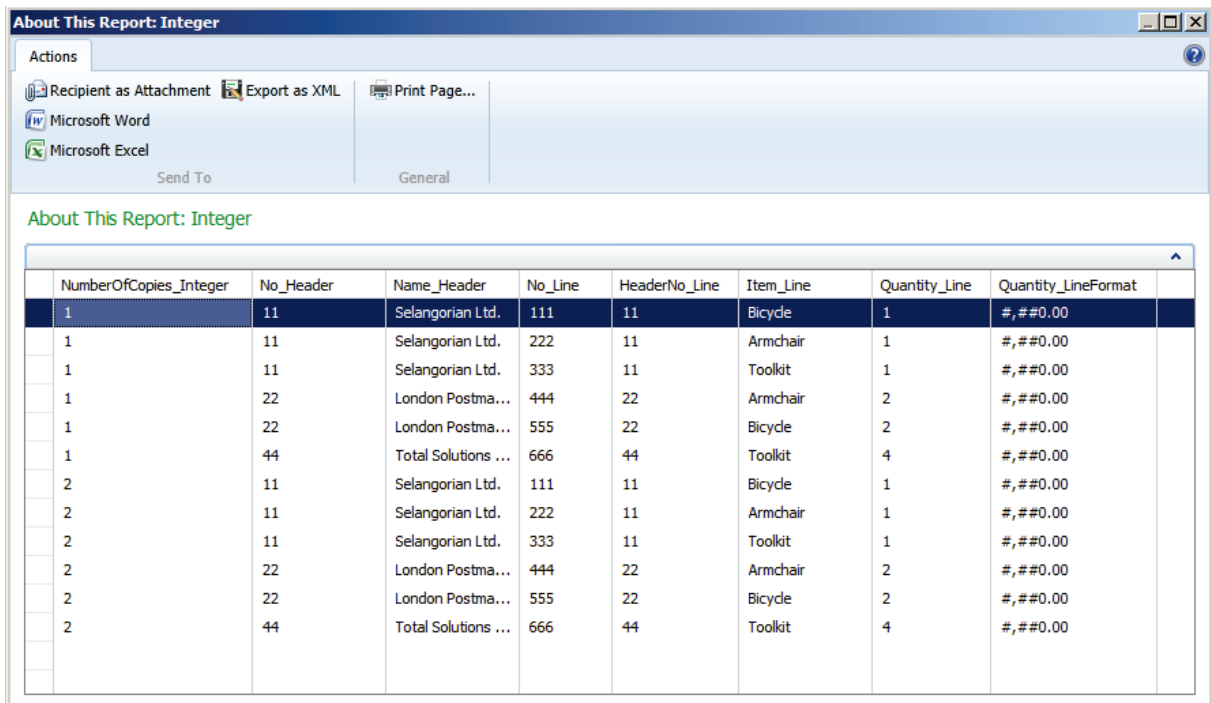
The following is a screenshot of the Report Dataset Designer of the report:



**FIGURE 1.17: INTEGER DATAITEM IN DATASET WINDOW**

**Note:** The expression `WHERE(Number=FILTER(1..2))` in **DataItemTableView** property of the **Integer** data item makes sure, that the outer data item will produce exactly two rows from the **Integer** table.

The resulting data set now contains two identical set of Header-Line rows complemented by the **NoOfCopy\_Integer** field:

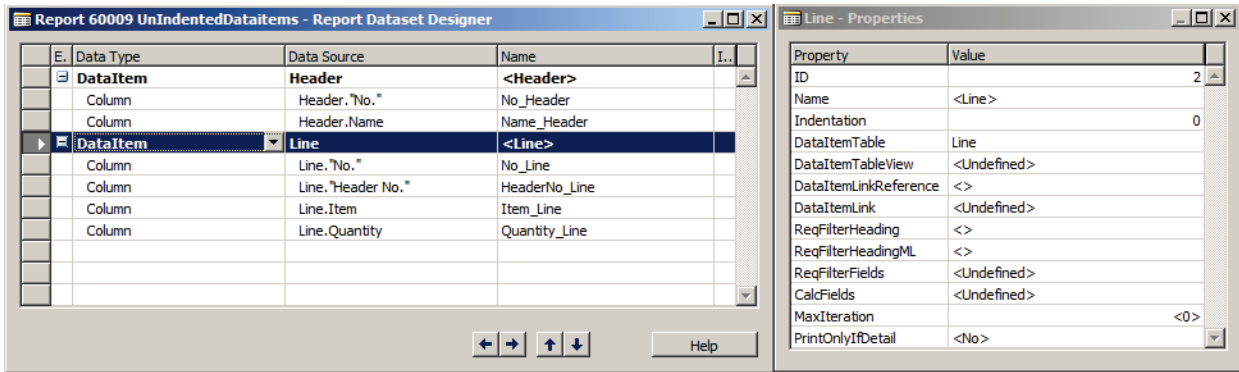


**FIGURE 1.18: DATASET OF INTEGER DATAITEM WINDOW**

The field **NumberOfCopies\_Integer** is available in the dataset and can be used in the group expression of a data region, for example a List or Table.

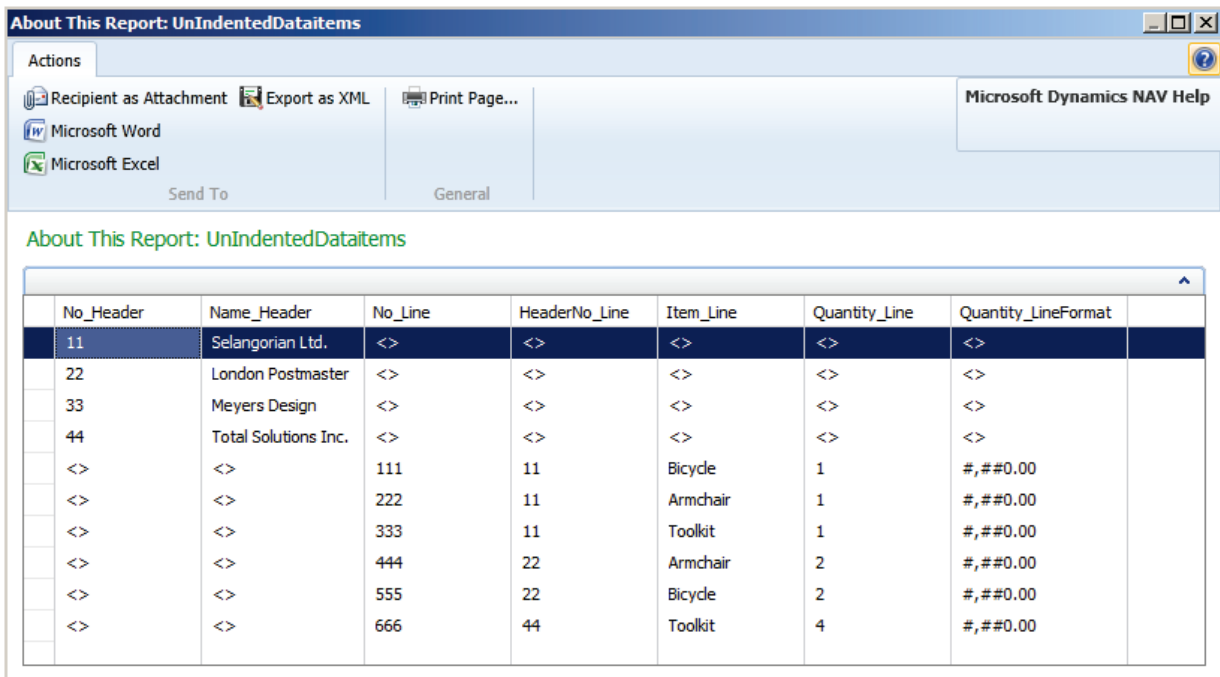
## Equally Nested Data Items

In this example the same indentation level is used for both data items:



**FIGURE 1.19: UNINDENTED DATAITEMS WINDOW**

The resulting data set combines the rows from both data items as in the following screenshot.



**FIGURE 1.20: RESULTSET OF UNINDENTED DATAITEMS WINDOW**

All the rows from the Header table are included, but with empty fields for the Line table, followed by all the Line rows, but with empty fields for the Header table.

The data set output is similar to SQL UNION operation on "Header"- "Line" tables.

## Report Properties

When designing reports in Microsoft Dynamics NAV, you can achieve good results without writing a single letter of code. Much can be done using properties. This lesson describes some properties that are frequently used in the report.

### Report Properties

Properties occur on many different levels in the report description. The following list shows the properties on the report level that are frequently used when designing reports.

#### UseRequestPage

You can use this property to decide whether the user of the report will be presented with a request page or not.

If UseRequestPage is set to No, no request page will be shown. This implies that the user will not be able to choose a sorting order or to set filters and that the print dialog box will not be shown (that is, a report will be printed immediately).

This property overrules any request page from the report.

#### ProcessingOnly

This property is used to set whether a report will produce graphical output, or will only be able to process data.

If ProcessingOnly is set to Yes, it is not possible to insert an RDLC layout. Additionally, the **Print** and **Preview** buttons in the request options form or page are replaced by an **OK** button. No graphical output is produced. (Any RDLC data part will be removed when setting the property to Yes.)

#### PaperSourceFirstPage, PaperSourceDefaultPage and PaperSourceLastPage:

In the **Report - Properties** window, in the **PaperSourceFirstPage** field, choose the paper source. The list contains 13 standard trays and 16 printer-specific trays.

If the printer that is used for this report does not support the specified paper source, then the printer uses the default tray that is specified by the printer driver.

If you do not specify a value for the PaperSourceFirstPage property, then the printer uses the tray that is specified in the PaperSourceDefaultPage property. If you do not specify a value for either the PaperSourceFirstPage property or the PaperSourceDefaultPage property, then the printer uses the default printer tray that is defined for the operating system. If you do not specify a value for the PaperSourceDefaultPage property, then the printer uses the default printer tray that is defined for the operating system. If you do not specify a value for the PaperSourceFirstPage property or the PaperSourceLastPage property, then the printer uses the tray that is specified in the PaperSourceDefaultPage property.

The printer specific trays that are contained in the list are intended for use with printers that have other paper sources than the standard ones that are listed in the property.

The settings in the **Print** and **Page Setup** dialog boxes override the value that is specified for the PaperSourceDefaultPage property.

If you do not specify a value for the PaperSourceFirstPage/PaperSourceLastPage property, then the printer uses the tray that is specified in the PaperSourceDefaultPage property. If you do not specify a value for either the PaperSourceFirstPage/PaperSourceLastPage property or the PaperSourceDefaultPage property, then the printer uses the default printer tray that is defined for the operating system.

### **Data Item Properties**

The following list shows the properties on a data item level that are frequently used when designing reports.

#### **DataltemTableView**

Use this property to select one of the existing keys and the related, sort order and the filter for this data item.

If a key is selected, the users of the report will not be able to select any key, sorting order or filters for that data item when the report is executed. As a result, there will be no tab for the current data item in the request page.

If a sorting order is selected, it is suggested as the default if the users of the report select a sorting order.

If a filter is selected, it is used together with the filters that the user specifies in the request page.

Selecting a sorting order or a filter, or both, does not inhibit the user from selecting a key or adding another filter.

The DataltemTableView property is a part of the logical design and does not apply to the RDLC report layout.

### **DataltemLinkReference**

Use this property to specify the data item on a higher level for the indented data item it will be linked to. When you specify DataltemLinkReference, use the DataltemLink property to specify one or more fields from each data item to base the link on.

This property is part of the logical design and impacts the dataset.

### **DataltemLink**

Use this property to specify the corresponding fields from two data items that are linked by the DataltemLinkReference property. The link is defined as a property of the indented data item.

This property is part of the logical design and impacts the dataset.

### **PrintOnlyIfDetail**

Use the PrintOnlyIfDetail property to limit printing a data item that does not have a detailed data item. This property is evaluated when flattening the data. If the property is shown as False on the parent data item, the system uses an outer join select statement.

### **ReqFilterFields**

Use this property to indicate those fields that will automatically be included on the tab of the request page that pertains to this data item. The user can set filters on these fields.

When the request page is in use, the user will be able to add other fields and set filters on these fields. Even so, it is still recommended to use this property. If the underlying table has many fields, the occasional user of the report might not easily find the relevant fields to filter.

If the user is not going to be able to put filters on other fields, you must remove the request page altogether by using the UseRequestPage property of the report, or remove the tab for this data item, by setting a specific key in the DataltemTableView property.

### **ReqFilterHeading**

Use this property to create a caption for the request form tab that pertains to this data item. You can specify captions in different languages using the ReqFilterHeadingML property.



### Report Functions

Apart from the properties, there are a number of C/AL statements that are often used in reports.

#### **CurrReport.Preview**

This property indicates whether a report is being printed in preview mode.

If you run a client report definition (RDLC) report layout in preview mode and then call the **CurrReport.PREVIEW** function, then the Print icon, Print Layout icon, Page Setup icon, and Save As icon are not displayed.

If you run a client report definition (RDLC) report layout in preview mode and do not call the **CurrReport.PREVIEW** function, then you can print from the **Print Preview** window.

#### **CurrReport.SKIP**

You can use the **SKIP** function to conditionally skip the processing of the current record. If the processing occurs inside a loop, the processing continues with the next record after the **SKIP** function is called.

A typical situation in which you will use the **SKIP** function is when you want to retrieve records from a related table by using values in the current record for forming a filter. If the values in the current record already indicate that no records from the related table will be retrieved, there is no need to perform this processing and you can use the **SKIP function** to avoid the processing.

#### **CurrReport.BREAK**

The **BREAK** function exits from a loop or a trigger in a data item trigger of a report.

**BREAK** causes the current trigger to end. It breaks the iteration of the current data item.

#### **CurrReport.QUIT**

When you use the **QUIT** function, the report is exited without committing any changes that are already made to the database during the execution. The OnPostReport trigger will not be called.

### **CurrReport.LANGUAGE**

For the documents that you want to print in the language of the recipient, rather than in your own working language, you can add a single line of code in the document. This functionality is already enabled for most reports in the standard Microsoft Dynamics NAV database. The document is printed in the language that is specified in the **Language Code** field in the **Customer Card** window.

In reports that need the multiple document languages functionality, you must insert the following C/AL code as the first line in the OnAfterGet Record() trigger.

```
CurrReport.LANGUAGE := Language.GetLanguageID("Language Code")
```

For each of these reports, you must create a new variable, Language, with the data type Record pointing to the Language table (table 8). When you have compiled the object, it will no longer print in the user's working application language if another language is specified in the **Customer Card** window.

## Building the Request Page

When a report is run, the **request page** is displayed. The **request page** is a special window the user can use to define output options for the report.

In some cases there is an extra tab called **Options** that is not related to a data item. It contains general information that determines the output and the execution of the report.

The **request page** is displayed when the report is run. Its purpose is to let the user provide additional input before previewing or printing the report.

### **The Request Page**

A **request page** is a page that is run before the report starts to execute. Request pages enable end users to specify options and filters for a report. When a new report is created, a **request page** is automatically created for the report. Additionally, custom options can be created in the **request page**. These options are designed by using the Request Options Page Designer that is accessed from the Report Designer. The following steps show how to open the Request Options Page Designer.

1. Open the Report Designer from the Object Designer.
2. In the Report Designer, click **View, Request Page**.

The Request Options Page Designer is similar to the Page Designer. Here developers can add controls, such as FastTabs and fields to the request page. Controls that are created in the Request Options Page Designer are always displayed above the FastTabs that correspond to data items.

You design the **Options** FastTab of a request page by using Request Options Page Designer in the development environment. To access Request Options Page Designer, open **Report Dataset Designer**, choose **View**, and then choose **Request Page**.

You design the filters on request pages by using the following report properties.

- ReqFilterHeading
- ReqFilterHeadingML
- ReqFilterFields

In addition to specifying options and filters, users can choose from the following actions on a **request page**,

- Print
  - Print
  - PDF
  - Microsoft Word
  - Microsoft Excel
- Preview
- Cancel

For more information on how to design and edit the **Request Page**, refer to the *C/SIDE Introduction in Microsoft Dynamics NAV 2013*.

The fields that you define as ReqFilterFields are shown on the **request page**. Additionally, an end user can add more fields on which to filter to the **request page**. It is recommended that you add fields that the end users of the report will frequently set filters on.

By default, for every data item in the report, a FastTab for defining filters and sorting is created on the **request page**. To remove a FastTab from a **request page**, do not define any ReqFilterFields for the data item and set the DataItemTableView property to define sorting. The **request page** is displayed but there is no tab for this data item. If a DataItemTableView is not defined, then end users can select a sort field and sort order at runtime. If you set UseReqPage to No, then the report will start to print as soon as it is run. In this case, end users cannot cancel the report run. It is still possible to cancel the report, but some pages might print. In a complex report that uses data from several tables, the functionality can depend on a specific key and sort order.

Design your reports so that end users cannot change the sort order in a way that affects the functionality of the report.

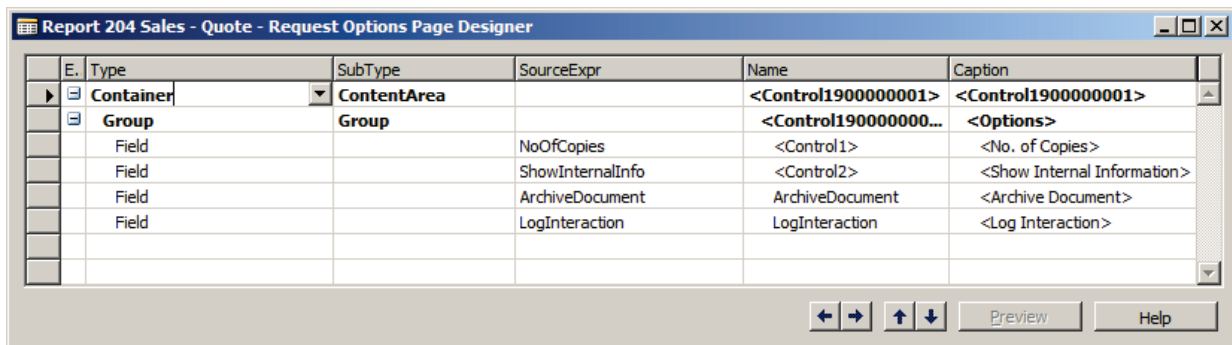
## Demonstration: Open the Request Page

Susan talks to many customers during her daily job, and now she has another request for Mort. Some customers want to have item pictures included on the sales quote document. She asks Mort if it is possible to include this as an option on the **Request Options Page**. Mort agrees to change the **Request Options Page**.

### Demonstration Steps

1. To open the Request Page.
  - a. Open the Microsoft Dynamics NAV 2013 demonstration database.
  - b. Open the CRONUS International Ltd. demonstration company.
  - c. Select Tools, Object Designer to open the Object Designer.
  - d. Select File, Import.
  - e. Browse to the report **R123456704.fob**.
  - f. Click OK to start the import.
  - g. In the dialog box, choose to open the Import Worksheet window.
  - h. In the Action column, verify that the action is set to Create.
  - i. Click Open to import the object in the database.
  - j. Select **Tools, Object Designer**.
  - k. Click the **Report** button.
  - l. Select report **123456704, Sales Quote**.
  - m. Click the **Design** button.
  - n. Select **View, Request Page**.

Mort is now ready to change the Request Options Page.



**FIGURE 1.21: REQUEST OPTION WINDOW**



**Note:** Note that adding a field in the request options page will not change the report layout. In order for the images to show/hide the RDLC layout needs to be updated accordingly.

---

## Sorting and Filtering

When you develop reports for Microsoft Dynamics NAV 2013, you must keep performance in mind. Because a company's database will grow, and as a result, reports will have to aggregate more data over time. Helping users locate the data they need is key and integrating sorting and filtering in reports will provide an added value.

### Sorting Data

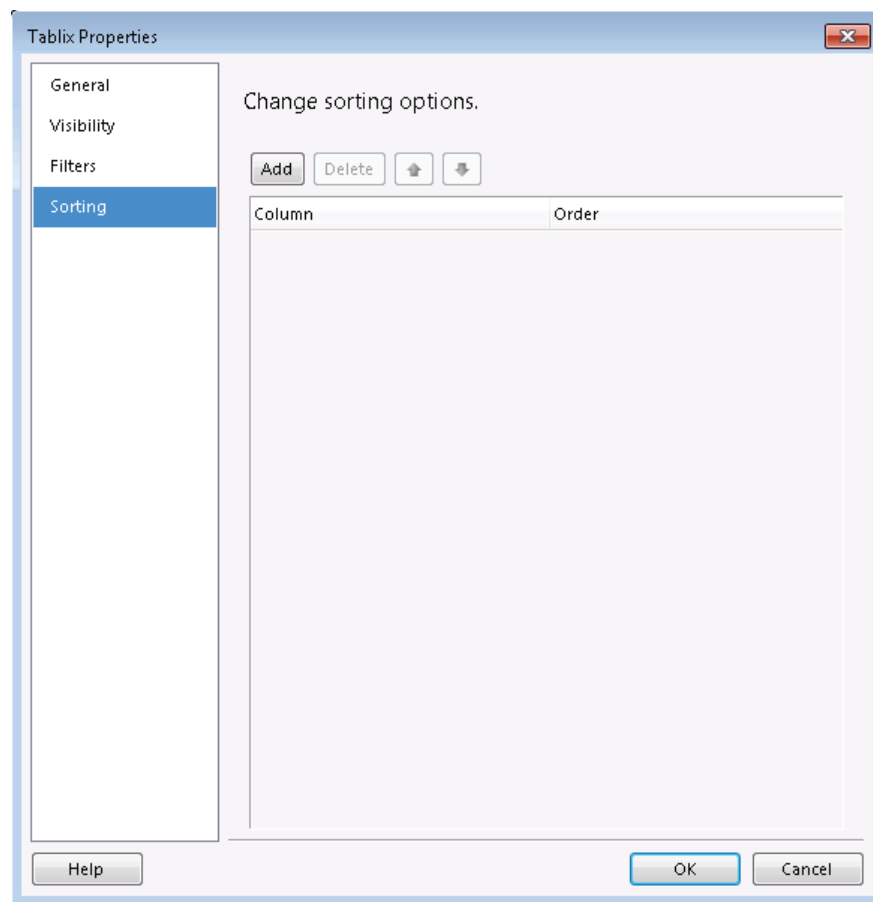
Data can be sorted in many ways. In Microsoft Dynamics NAV, each sort order is directly related to a table key. Meaning, you can have as many sorting orders as there are keys in your table. You can define up to 40 keys (one primary, 39 secondary keys) in a table.

Sorting can be defined several ways, including the following.

- Use the **DataItemTableView** property to define the sorting of a data item. Select any of the keys that are defined in the table and also determine the sorting order (ascending or descending). Sorting methods that are defined here cannot be modified by the user. When specifying a Key in the DataItemTableView property, the **Sort** button is removed from the **Request Options** page. Additionally, the tab for the corresponding data item is removed from the **Request Options** page (unless request filter fields are specified in the **ReqFilterFields** property of the same data item).
- The user can decide how the data will be sorted. In this instance, do not specify a Key in the DataItemTableView property of the data item. (Specify the Order if necessary.) When a user prints a report, he or she can click the **Sort** button and define both the Key and the Order to use in the **Sort** window. Remember that the Order selected in the **Sort** window will overrule the Order specified in the DataItemTableView.

- Use coding to define the sorting programmatically. One way is to define the sorting hardcoded in the C/AL Code using the SETCURRENTKEY statement. An alternative and more flexible solution is for the user to select from a limited set of sorting methods on the **Request Options** page (and then tie the options to a specific SETCURRENTKEY statement). Remember that programmatic sorting overrules the sorting defined in the **Sort** window or the DataItemTableView property.

The RDLC report layout offers another option to sort data. In Visual Studio Report Designer, you can specify additional sorting methods for any matrix or table control. To do this, right-click a **table or matrix control** and select **Properties**. On the **Sorting** tab, define a custom sorting order.



**FIGURE 1.22: ADDITIONAL SORTING DEFINED IN THE TABLIX PROPERTIES WINDOW**

You can specify sorting options that do not match an existing key. For example, you can have your sales orders list sorted on **Order Date**, although the Sales Header table does not contain a corresponding key.



**Note:** In the RDLC reports, it is not necessary for the Microsoft Dynamics NAV keys to sort. In this instance the sorting will occur in the Report Viewer and might take longer to complete.

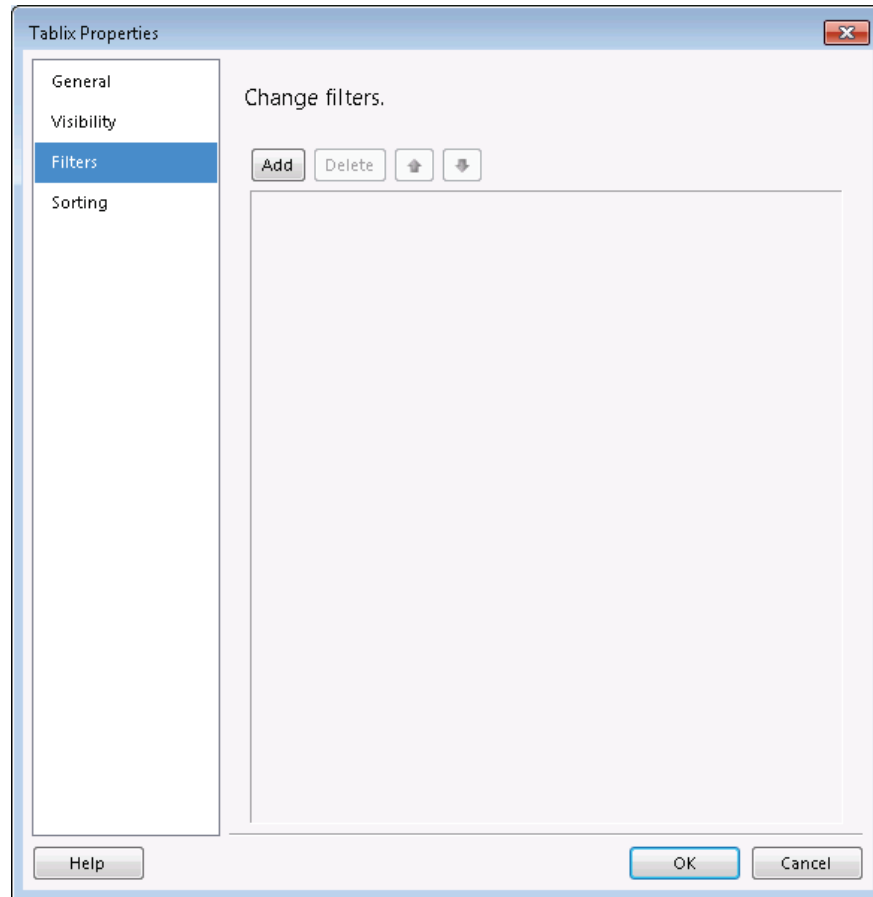
---

### Filtering Data

A user can use filtering to limit the data that is included in a report. Several ways to perform filtering include the following.

- **Use request filter fields:** Request filter fields offer a variable way of filtering information. The user can decide whether or not to apply any filters. If any filter is specified, it is applied. If no filters are specified, all information is available. The advantage is that the designer can solve multiple reporting needs with one report. The disadvantage is that the user can set any filters and possibly manipulate the report output.
- **Use the DataItemTableView property to set a filter on a data item inside the report:** Filters that are defined inside a report cannot be removed. The filter in the DataItemTableView is not translated into explicit filtering criteria in the RDLC report layout. Instead, the filter is applied on the service tier, before the dataset is sent to the client.
- **Use code to set some filters:** Using code to set filters requires both programming skills and access to the C/AL Editor. Hardcoded filters can be set with the SETRANGE and SETFILTER statements. The disadvantage is that these filters are inside in the report and cannot be removed by the user. The report can only serve specific purposes. The advantage is that the report output cannot be manipulated.

Inside the Visual Studio IDE there is another way to define some filters. Additional filtering can be defined by setting the properties of any Table or Matrix control. To define additional filter criteria, right-click any **table or matrix control**, and select **Properties**. On the **Filters** tab, define the filters.



**FIGURE 1.23: TABLIX FILTER PROPERTIES WINDOW**

Filters that are defined inside Visual Studio cannot be removed by the user.

An alternative for defining hardcoded filters in the C/AL Editor is to use variables on the **Request Options** page and use these variables as a filter. To do this, the variables must be part of the dataset, meaning they must be added to the Report Dataset Designer.

There is no option to specify parameters at runtime. Parameters are used only for Captions and Labels.





**Best Practice:** *It is considered a best practice to avoid filtering in the RDLC layout of the report. When you apply a filter in the RDLC layout, it means the data will be fetched from the database, then sent to the report viewer and then it will be filtered. If the filter is applied in the Report Dataset Designer, then only the required data is fetched from the database and sent to the report viewer.*

*Sometimes, when you know the values that will be used for the filters in advance, it is more advantageous to apply the filter(s) in the report dataset instead of in the RDLC layout.*

*However, with the RDLC layout, you can use variables and expressions for dynamic filtering. If, when you design the report, you do not know the exact values that will be used for filtering, then you can use expressions, variables and parameters to filter in the RDLC layout.*

---

The dataset is sent from Microsoft Dynamics NAV Server into the memory on the client machine.

If the dataset is larger than the amount of memory that is available, then the operating system might stop allocating more memory and the client might stop responding. To avoid this scenario you can try to reduce the dataset by applying appropriate filters that reduce the amount of data that needs to be retrieved and processed in a report.

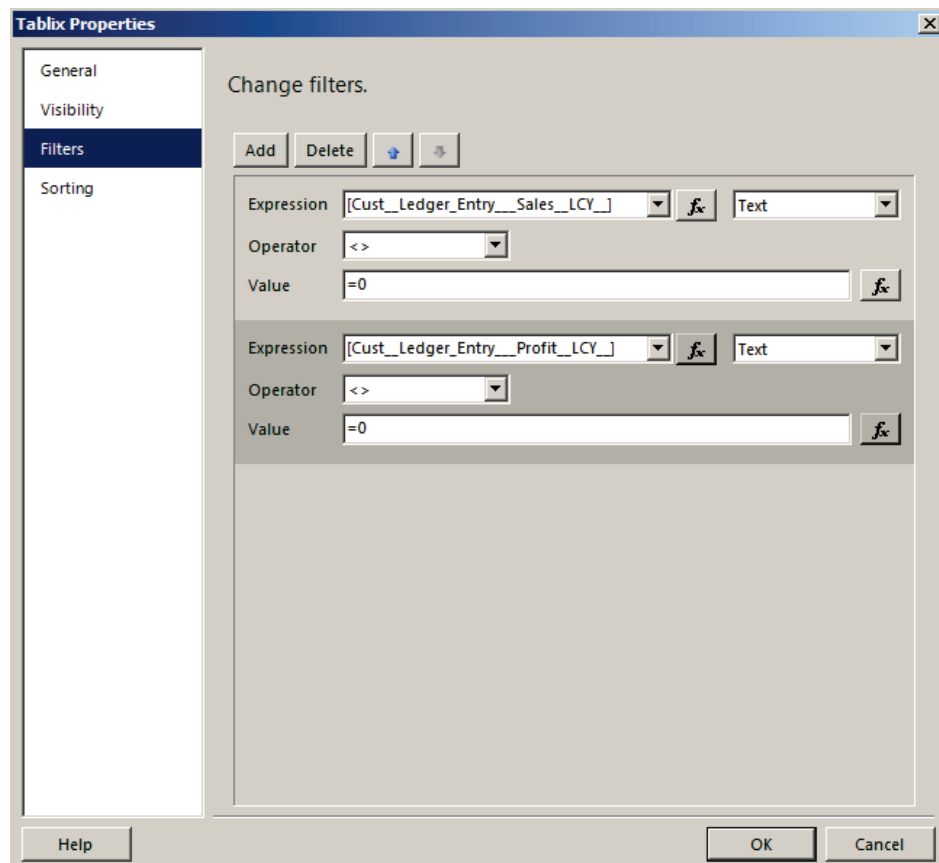
### **Demonstration: Adding Sorting and Filtering**

Kevin, the Sales Manager, is using report **123456715, Salesperson - Commission**, to calculate and monitor the performance of his salespersons. He notices that the report contains lines where all the amounts are zero. He asks Mort to filter the report so that it only contains lines where both the **Sales (LCY)** and **Profit (LCY)** fields are different from zero. Additionally, he wants the details sorted by posting date, in descending order.

#### **Demonstration Steps**

1. To add sorting and filtering.
  - a. Open the Microsoft Dynamics NAV 2013 demonstration database.
  - b. Open the CRONUS International Ltd. demonstration company.
  - c. Select Tools, Object Designer to open the Object Designer.
  - d. Select File, Import.
  - e. Browse to the report R123456715.fob.
  - f. Click OK to start the import.
  - g. In the dialog box, choose to open the Import Worksheet window.
  - h. In the Action column, verify that the action is set to Create.

- i. Click Open to import the object in the database.
- j. Click the **Report** button.
- k. Select report **123456715, Salesperson Commission**.
- l. Click the **Design** button.
- m. Select **View, Layout**.
- n. Select **Table1** in the **Body** section. Right-click any cell in the table and choose **Select 'Table1'**.
- o. Right-click the shaded border and select **Properties**.
- p. Select the **Sorting** tab.
- q. Click the **Add** button.
- r. In the **Sort by** field, select **[Cust\_Ledger\_Entry\_Posting\_Date\_]**.
- s. In the **Order** column, select **Z-A**.
- t. Select the **Filters** tab.
- u. In the Filter list, add the criteria as shown in the following illustration.



**FIGURE 1.24: CHANGE FILTERS WINDOW**

- v. Click **OK** to close the Tablix Properties window.
- w. Exit Visual Studio.
- x. Load the changes.
- y. Save and compile the report.

## Lab 1.1: Creating a Table Report

### Scenario

Your company has implemented Microsoft Dynamics NAV 2013 and some additional reports need to be created.

Throughout the labs, you will be assigned the role of Mort, a customization developer in the Information Technology (IT) Department. Mort is mainly responsible for fixing problems and implementing features and customizations in the internal IT applications.

Kevin, the Sales Manager, has asked for an overview of the customer sales and profit (in local currency) by salesperson. The list needs to contain totals by salesperson, by country/region code, by customer and by document type. The list also needs to contain a grand total for all salespersons.

The report is not finished yet. Only a draft version of the object exists, containing the correct data model and two sections. Your task is to build the report for the RoleTailored client, from scratch.

### Objectives

The purpose of this lab is to create a RoleTailored client report with a table layout from scratch. You will learn how to add formatting to specific fields, and will add grouping and totaling.

In the next modules, you will add some additional functionality to the report.

### Exercise 1: Create the Report

#### *Exercise Scenario*

Build a report showing the following.

- Total sales (LCY) and total profit (LCY) by document type, by customer no., by country/region code and by salesperson
- Subtotals for each group
- Grand total for all salespersons

Dates and totals in the report must be formatted as defined in Microsoft Dynamics NAV.

### **Task 1: Import Report 123456700**

#### ***High Level Steps***

1. To import the object in the demonstration database, use the following procedure.

#### ***Detailed Steps***

1. To import the object in the demonstration database, use the following procedure.
  - a. Open the Microsoft Dynamics NAV 2013 demonstration database.
  - b. Open the CRONUS International Ltd. demonstration company.
  - c. Select Tools, Object Designer to open the Object Designer.
  - d. Select File, Import.
  - e. Browse to the report R123456700.fob.
  - f. Click OK to start the import.
  - g. In the dialog box, choose to open the Import Worksheet window.
  - h. In the Action column, verify that the action is set to Create.
  - i. Click Open to import the object in the database.

### **Task 2: Open the Object in Visual Studio**

#### ***High Level Steps***

1. First, start the Visual Studio development environment to create the RDLC report layout.

#### ***Detailed Steps***

1. First, start the Visual Studio development environment to create the RDLC report layout.
  - a. In the Object Designer, open report 123456700, Sales by Salesperson - L1.1 in design mode.
  - b. Select View, Layout to open Microsoft Visual Studio.
  - c. Visual Studio will open and show a blank Report Layout.

### **Task 3: Add a Table to the Report**

#### ***High Level Steps***

1. Next, add some controls from the Toolbox window to the Report Layout window.

The Toolbox window is normally displayed on the left side of the Visual Studio IDE. If not, it can be made visible by pressing Ctrl+Alt+X.

### **Detailed Steps**

1. Next, add some controls from the Toolbox window to the Report Layout window.

The Toolbox window is normally displayed on the left side of the Visual Studio IDE. If not, it can be made visible by pressing Ctrl+Alt+X.

- a. In the **Toolbox** window, click the **Table control**.
- b. Click in the Body section to add a table to the report. (You can also drag the Table control to the Body section.)

The report now contains a table with two rows (header, data) and three columns.

### **Task 4: Add Columns to the Table**

#### **High Level Steps**

1. The following steps show how to add additional columns to the table.

#### **Detailed Steps**

1. The following steps show how to add additional columns to the table.
  - a. Select any cell in the table. Now a grey bar is displayed above each column: the column handle. The column handle can be used to select and/or resize a column.
  - b. Right-click the column handle above the third column and select "Insert column to the right." A new column will be added to the table. The new column will have the same width as the selected column.
  - c. Repeat steps 1 and 2 three times.

As a result, the table will have two rows and seven columns.

### **Task 5: Add Dataset Elements to the Table**

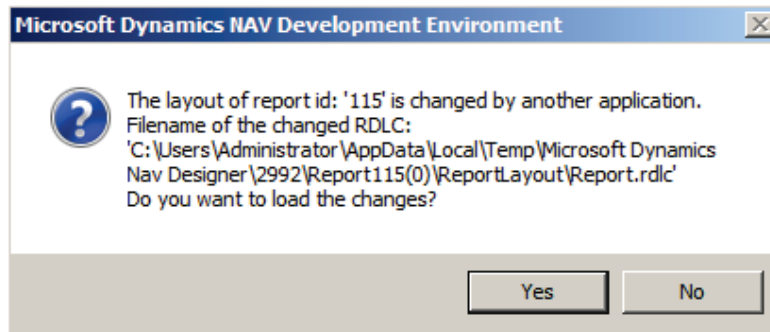
#### **High Level Steps**

1. The dataset is generated based on the report dataset designer. The following steps show how to add dataset elements to the table.

#### **Detailed Steps**

1. The dataset is generated based on the report dataset designer. The following steps show how to add dataset elements to the table.
  - a. In the **Report Data** window, select an item from the dataset. For example, select Customer\_\_Salesperson\_Code\_.
  - b. Drag the selected element to a cell in the detail row of the table. In this case drag it to the first column in the Detail row.
  - c. Select the Customer\_\_Country\_Region\_Code\_ and drag it to the second column on the Detail row.

- d. Select the `Cust_Ledger_Entry_Customer_No_` and drag it to the third column on the Detail row.
- e. Select the `Cust_Ledger_Entry_Document_Type_` and drag it to the fourth column on the Detail row.
- f. Select the `Cust_Ledger_Entry_Posting_Date_` and drag it to the fifth column on the Detail row.
- g. Select the `Cust_Ledger_Entry_Sales_LCY_` and drag it to the sixth column on the Detail row.
- h. Select the `Cust_Ledger_Entry_Profit_LCY_` and drag it to the seventh column on the Detail row.
- i. Save and close Visual Studio. You will be redirected to the Report Designer. Press **Esc** to leave the Report Designer. The following dialog box will be displayed.



**FIGURE 1.25: MICROSOFT DYNAMICS NAV DIALOG BOX**

- j. Click **Yes** to load the RDLC layout in the report object. (If you select **No**, all changes made in the previous steps will be discarded, and the previous RDLC layout will be retained.)
- k. Click **Yes** to save the object.

### **Task 6: Run the Report**

#### **High Level Steps**

1. To check the layout of the report, run the report.

#### **Detailed Steps**

1. To check the layout of the report, run the report.
  - a. In the Object Designer, select the report and click **Run**. The RoleTailored client will be started and the **request page** of the report will be displayed.
  - b. Click the **Preview** button to run the report. The result will appear similar to the following image :

Customer Salesperson Code	Customer Country Region	Cust Ledger Entry Customer No	Cust Ledger Entry Document Type	Cust Ledger Entry Posting Date	Cust Ledger Entry Sales LCY	Cust Ledger Entry Profit LCY
JR	US	01445544	Invoice	20/01/2014 0:00:00	1499,02	305,12
JR	US	01454545	Invoice	31/12/2013 0:00:00	0	0
PS	GB	10000	Invoice	31/12/2013 0:00:00	0	0
PS	GB	10000	Invoice	31/12/2013 0:00:00	0	0
PS	GB	10000	Invoice	31/12/2013 0:00:00	0	0
PS	GB	10000	Invoice	31/12/2013 0:00:00	0	0
PS	GB	10000	Invoice	31/12/2013 0:00:00	0	0
PS	GB	10000	Invoice	31/12/2013 0:00:00	0	0
PS	GB	10000	Invoice	5/01/2014 0:00:00	6615,23	1184,33
PS	GB	10000	Credit Memo	12/01/2014 0:00:00	-234,27	-42,07
PS	GB	10000	Payment	12/01/2014 0:00:00	0	0
PS	GB	10000	Payment	12/01/2014 0:00:00	0	0

**FIGURE 1.26: SALES REPORT PREVIEW WINDOW**

## Task 7: Add Currency Formatting to the Report

### High Level Steps

1. In the Print Preview notice, some of the fields that show amounts (decimal values) are not formatted correctly—some have decimals—some do not, to reflect the same formatting in the RoleTailored client, you will need to change the formatting of the Sales (LCY) and Profit (LCY) fields in the report.

### Detailed Steps

1. In the Print Preview notice, some of the fields that show amounts (decimal values) are not formatted correctly—some have decimals—some do not,  
To reflect the same formatting in the RoleTailored client, you will need to change the formatting of the Sales (LCY) and Profit (LCY) fields in the report.
  - a. Open report **123456700** in Visual Studio.
  - b. Select the **Sales (LCY)** field in the detail row.
  - c. Right-click the cell and select **Placeholder Properties**.



- d. Select the Number tab.  
On this tab, there are two options to define a format: by using a format, or by using an expression.  
If a format is being used, you can either select a standard format, or build your own custom format.  
However, you want to use the same formatting that is defined in the G/L Setup window. From previous steps, you know that the dataset also contains special formatting elements for decimal fields. These elements can only be selected using an expression.
- e. Select the **Custom** option in the Category column.
- f. Click the **[Fx]** button next to the **Custom Format** field. The **Expression** window is displayed.
- g. Double-click the **Cust\_Ledger\_Entry\_Sales\_LCY\_Format** element in the Fields (DataSet\_Result) category.
- h. Click **OK** to close the **Expression** window.
- i. Click **OK** to close the **Placeholder Properties** window.
- j. Repeat steps 2 through 9 to format the **Profit (LCY)** field using the **Cust\_Ledger\_Entry\_Profit\_LCY\_Format** element.
- k. Save and Exit Visual Studio, import the new RDLC layout in the object, save the object and run it.

The result will appear similar to the following image.

Customer Salesperson Code	Customer Country Region	Cust Ledger Entry Customer No	Cust Ledger Entry Document Type	Cust Ledger Entry Posting Date	Cust Ledger Entry Sales LCY	Cust Ledger Entry Profit LCY
JR	US	01445544	Invoice	20/01/2014 0:00:00	1,499.02	305.12
JR	US	01454545	Invoice	31/12/2013 0:00:00	0.00	0.00
PS	GB	10000	Invoice	31/12/2013 0:00:00	0.00	0.00
PS	GB	10000	Invoice	31/12/2013 0:00:00	0.00	0.00
PS	GB	10000	Invoice	31/12/2013 0:00:00	0.00	0.00
PS	GB	10000	Invoice	31/12/2013 0:00:00	0.00	0.00
PS	GB	10000	Invoice	31/12/2013 0:00:00	0.00	0.00
PS	GB	10000	Invoice	31/12/2013 0:00:00	0.00	0.00
PS	GB	10000	Invoice	5/01/2014 0:00:00	6,615.23	1,184.33
PS	GB	10000	Credit Memo	12/01/2014 0:00:00	-234.27	-42.07
PS	GB	10000	Payment	12/01/2014 0:00:00	0.00	0.00
PS	GB	10000	Payment	12/01/2014 0:00:00	0.00	0.00

**FIGURE 1.27: SALES REPORT PREVIEW WINDOW**

### Task 8: Decimal and Thousand Separators

#### **High Level Steps**

1. By default, the decimal separator is "." while the thousand separator is ",". Both separators are not defined hardcoded; they depend on the language of the report. The language can be defined using the Language property of the report. As with decimal formatting, you can use either a specific value (that applies for all users in all cases) or use an expression to make it user-dependent.

#### **Detailed Steps**

1. By default, the decimal separator is "." while the thousand separator is ",". Both separators are not defined hardcoded; they depend on the language of the report. The language can be defined using the Language property of the report. As with decimal formatting, you can use either a specific value (that applies for all users in all cases) or use an expression to make it user-dependent.
  - a. Open the report in Visual Studio.
  - b. In the **Properties** window, make sure **Report** is selected in the drop-down list.
  - c. Select the Language property. Note that the default value is set to "=User!Language" which explains the current decimal and thousand separator.
  - d. Save and Exit Visual Studio, import the new RDLC layout in the object, save the object and run it.
  - e. The result will appear similar to the following image.

Customer Salesperson Code	Customer Country Region	Cust Ledger Entry Customer No	Cust Ledger Entry Document Type	Cust Ledger Entry Posting Date	Cust Ledger Entry Sales LCY	Cust Ledger Entry Profit LCY
JR	US	01445544	Invoice	20/01/2014 0:00:00	1,499.02	305.12
JR	US	01454545	Invoice	31/12/2013 0:00:00	0.00	0.00
PS	GB	10000	Invoice	31/12/2013 0:00:00	0.00	0.00
PS	GB	10000	Invoice	31/12/2013 0:00:00	0.00	0.00
PS	GB	10000	Invoice	31/12/2013 0:00:00	0.00	0.00
PS	GB	10000	Invoice	31/12/2013 0:00:00	0.00	0.00
PS	GB	10000	Invoice	31/12/2013 0:00:00	0.00	0.00
PS	GB	10000	Invoice	31/12/2013 0:00:00	0.00	0.00
PS	GB	10000	Invoice	5/01/2014 0:00:00	6,615.23	1,184.33
PS	GB	10000	Credit Memo	12/01/2014 0:00:00	-234.27	-42.07
PS	GB	10000	Payment	12/01/2014 0:00:00	0.00	0.00
PS	GB	10000	Payment	12/01/2014 0:00:00	0.00	0.00

**FIGURE 1.28: SALES REPORT WITH SEPERATORS PREVIEW WINDOW**

## Task 9: Add Date Formatting to the Report

### High Level Steps

1. In the Print Preview notice that has the dates in the report still displayed as a Date Time field, you will need to change the formatting of the Posting Date field. Perform the following steps to add date formatting.
2. Run the RoleTailored client report. Notice that all posting dates are formatted as defined in the report.

You can also define the date format in Visual Studio. To have the date formatted as MM-DD-YYYY, perform the following steps.

### Detailed Steps

1. In the Print Preview notice that has the dates in the report still displayed as a Date Time field, you will need to change the formatting of the Posting Date field. Perform the following steps to add date formatting.
  - a. Open the report in the Report Dataset Designer.
  - b. Select the line containing the **Posting Date** field and select **Properties**.
  - c. In the Properties window, set the SourceExpr property to: `FORMAT("Cust. Ledger Entry"."Posting Date", 0, '<Day,2>-<Month,2>-<Year4>')`
  - d. Save and compile the report in the Report Designer.

2. Run the RoleTailored client report. Notice that all posting dates are formatted as defined in the report.

You can also define the date format in Visual Studio. To have the date formatted as MM-DD-YYYY, perform the following steps.

- a. Open the report in the Report Designer.
- b. Select the line containing the **Posting Date** field and select **Properties**.
- c. In the Properties window, set the SourceExpr property to "Cust.Ledger.Entry"."Posting Date". (Remove the format instruction entered earlier.)
- d. Open the report in Visual Studio.
- e. On the detail row, right-click the text box containing the **Posting Date** field and select **Textbox Properties**.
- f. In the **Textbox Properties** window, click the **Number** tab.
- g. In the **Category**, select the **Custom** option.
- h. In the text box, enter **MM-dd-yyyy**.
- i. Click **OK** to close the **Textbox Properties** window. The specified date format will be copied to the **Format code** field in the **Textbox Properties** window.
- j. Save and Exit Visual Studio.

The dates are now formatted according to the specified formula.

Date formats specified in the Report Dataset Designer will not overrule any format defined in Visual Studio.

### Task 10: Add Grouping to a Report

#### *High Level Steps*

1. Add a grouping to the table.
2. Add a new grouping for the **Country/Region Code** field.
3. Add a new grouping for the **Customer No.** field.
4. Add a new grouping for the **Document Type** field.
5. Delete the detail columns that were added as groups.

#### *Detailed Steps*

1. Add a grouping to the table.
  - a. Open the object in the RDLC design mode.
  - b. Select any cell in the table. The column and row handles appear.
  - c. Select the detail row by clicking the **row handle**.
  - d. Right-click the **row handle** for the detail row and select **Add group, Parent Group**.

- e. In the **Group by** list box, select the [Customer\_\_Salesperson\_Code\_] element.
- f. Select the check boxes: **Add group header** and **Add group footer**.
- g. Click OK to close the Grouping Properties window.

Notice two new rows are added to the table: a group header row (above the detail row) and a group footer row (below the detail row).

Also notice that at the bottom of the screen, Group1 and Details are listed in the **Row Groups** window.

2. Add a new grouping for the **Country/Region Code** field.
  - a. Click the **Group1** drop-down list in the **Row Groups** window.
  - b. Select **Add group, Child Group**.
  - c. In the **Group by** list box, select the [Customer\_\_Country\_Region\_Code\_] element.
  - d. Select the check boxes: **Add group header** and **Add group footer**.
  - e. Click OK to close the Grouping Properties window.

Notice that two new rows are added to the table: a group header row (above the detail row) and a group footer row (below the detail row).

Also notice that a Group2 is added below Group1 in the Row Groups window.

3. Add a new grouping for the **Customer No.** field.
  - a. Click the **Group2** drop-down list in the **Row Groups** window.
  - b. Select **Add group, Child Group**.
  - c. In the **Group by** list box, select the [Customer\_\_No\_] element.
  - d. Select the check boxes: **Add group header** and **Add group footer**.
  - e. Click OK to close the Grouping Properties window.

Notice that two new rows are added to the table: a group header row (above the detail row) and a group footer row (below the detail row).

Also notice that a Group3 is added below Group2 in the Row Groups window.

4. Add a new grouping for the **Document Type** field.
  - a. Click the **Group3** drop-down list in the **Row Groups** window.
  - b. Select **Add group, Child Group**.
  - c. In the **Group by** list box, select the [Cust\_\_Ledger\_Entry\_\_Document\_Type\_] element.

- d. Select the check boxes: **Add group header** and **Add group footer**.
  - e. Click OK to close the Grouping Properties window.  
Notice that two new rows are added to the table: a group header row (above the detail row) and a group footer row (below the detail row).  
Also notice that a Group4 is added below Group3 in the Row Groups window.
5. Delete the detail columns that were added as groups.
    - a. Since we have added the Salesperson Code and Country Region Code as groups they are no longer required at the detail level of the report.
    - b. Select the column header containing the Salesperson Code fields at the detail level (column 5).
    - c. Right click the column header and select Delete.
    - d. Repeat the last to steps for the column that contains the Country Region Code field (column 6).
    - e. Save and Exit Visual Studio, import the new RDLC layout in the object, save the object and run it.

The result will appear similar to the following image.

Group1	Group2	Group3	Group4	Cust Ledger Entry Posting Date	Cust Ledger Entry Sales LCY	Cust Ledger Entry Profit LCY
JR	AT	43687129	Invoice	11-01-2014	2.498,10	549,90
	BE	32656565	Invoice	15-01-2014	2.582,81	726,01
	CZ	42147258	Invoice	11-01-2014	1.602,90	353,60

**FIGURE 1.29: SALES REPORT WITH DATE FORMATTING PREVIEW WINDOW**

### **Task 11: Add Subtotals to a Report**

#### ***High Level Steps***

1. The following steps show how to add totaling to a report.

#### ***Detailed Steps***

1. The following steps show how to add totaling to a report.
  - a. Open the object in the RDLC design mode.
  - b. Right-click in the text box of the sixth column on the group footer row for the Country/Region Code group.
  - c. Select Expression in the drop-down list that appears.
  - d. In the Expression Editor that opens, type:  
`=Sum(Fields!Cust_Ledger_Entry_Sales_LCY__.Value)`.
  - e. Specify the correct format for the subtotal.
  - f. Right-click in the text box of the seventh column on the group footer row for the Country/Region Code group.
  - g. Select Expression in the drop-down list that appears.
  - h. In the Expression Editor that opens, type:  
`=Sum(Fields!Cust_Ledger_Entry_Profit_LCY__.Value)`.
  - i. Specify the correct format for the subtotal.
  - j. Save and Exit Visual Studio, import the new RDLC layout in the object, save the object and run it.

The result will appear similar to the following image.

Group1	Group2	Group3	Group4	Cust Ledger Entry Posting Date	Cust Ledger Entry Sales LCY	Cust Ledger Entry Profit LCY
JR	AT	43687129	Invoice	11-01-2014	2,498,10	549,90
					2498,1	549,9
	BE	32656665	Invoice	15-01-2014	2,582,81	726,01
					2582,81	726,01
	CZ	42147258	Invoice	11-01-2014	1,602,90	353,60

**FIGURE 1.30: SALES REPORT WITH SUBTOTALS PREVIEW WINDOW**

## Task 12: Add Grand Totals to a Report

### High Level Steps

1. The following steps show how to add grand totals to a report.

### Detailed Steps

1. The following steps show how to add grand totals to a report.
  - a. Open the object in the RDLC design mode.
  - b. Right-click in the text box of the sixth column on the table footer row.
  - c. Select Expression in the drop-down list that appears.
  - d. In the Expression Editor that opens, type:  
=Sum(Fields!Cust\_Ledger\_Entry\_Sales\_LCY\_.Value).
  - e. Specify the correct format for the grand total.
  - f. Repeat steps 2 through 5 for the **Cust\_Ledger\_Entry\_Profit\_LCY\_** field.
  - g. Save and Exit Visual Studio, import the new RDLC layout in the object, save the object and run it.



The result will appear similar to the following image.

The screenshot shows a 'Print Preview' window for a report titled 'Sales by Salesperson - L1.1'. The window contains a table with the following data:

Group1	Group2	Group3	Group4	Cust Ledger Entry Posting Date	Cust Ledger Entry Sales LCY	Cust Ledger Entry Profit LCY
JR						
	AT	43687129	Invoice	11-01-2014	2,498.10	549.90
					2498,1	549,9
	BE	32656565	Invoice	15-01-2014	2,582.81	726.01
					2582,81	726,01
	CZ	42147258	Invoice	11-01-2014	1,602.90	353.60

**FIGURE 1.31: SALES REPORT WITH GRAND TOTALS PREVIEW WINDOW**

The layout at the current state of development is not user friendly. In the next module you will further enhance the layout and fine-tune the report.

## Lab 1.2: Creating a Matrix Report

### Scenario

Ellen is the Warehouse Manager, and one of her tasks is to optimize the entire warehouse. She ensures that item inventory levels are accurate and that periodic physical inventory counts occur.

Ellen asks Mort to build a report showing the item inventory by location.

### Objectives

The purpose of this lab is to create a RoleTailored client report with a matrix layout from scratch. You will learn how to add formatting to specific fields, and how to add grouping and totaling.

The classic layout is available as a separate object on the training DVD, namely report R123456701.fob. Use this as a starting point.

In the next modules, you will add some additional functionality to the report.

### Exercise 1: Build a Matrix Report.

#### *Exercise Scenario*

Build a matrix report for report 123456701 showing the total Inventory by Location Code. Location codes need to be printed as bold white text on a dark blue background. Item numbers need to be printed as black text on an orange background.

#### **Task 1: Import Report 123456701**

#### *High Level Steps*

1. To import the object in the demonstration database, use the following procedure.

#### *Detailed Steps*

1. To import the object in the demonstration database, use the following procedure.
  - a. Open the Microsoft Dynamics NAV 2013 demonstration database.
  - b. Open the CRONUS International Ltd. demonstration company.
  - c. Select **Tools, Object Designer** to open the Object Designer.
  - d. Select **File, Import**.
  - e. Browse to the report R123456701.
  - f. Click **OK** to start the import.

- g. In the dialog box, choose to open the **Import Worksheet** window.
- h. In the Action column, verify that the action is set to Create.
- i. Click **OK** to import the object in the database.

### **Task 2: Open the Object in Visual Studio**

#### ***High Level Steps***

1. The first thing to do is to start the Visual Studio development environment to create the RDLC report layout of the report.

#### ***Detailed Steps***

1. The first thing to do is to start the Visual Studio development environment to create the RDLC report layout of the report.
  - a. In the Object Designer, open report **123456701, Item Matrix - L1.2** in design mode.
  - b. Select **View, Layout** to open Microsoft Visual Studio.

Visual Studio will open and show a blank Report Layout.

### **Task 3: Add a Matrix Control to the Report**

#### ***High Level Steps***

1. To add the Matrix to the report follow these steps.

#### ***Detailed Steps***

1. To add the Matrix to the report follow these steps.
  - a. In the **Toolbox** window, click the **Matrix control**.
  - b. Click in the Body section to add a matrix to the report. (You can also drag the Matrix control to the Body section).

The report now contains a matrix with two rows and two columns.

### **Task 4: Change the Position of the Matrix Control**

#### ***High Level Steps***

1. When you place a matrix control on the Body section, it might not have the correct position. The position of the control can be changed either by dragging the control to the correct position or by changing the control properties.

## Detailed Steps

1. When you place a matrix control on the Body section, it might not have the correct position. The position of the control can be changed either by dragging the control to the correct position or by changing the control properties.
  - a. Select any cell in the matrix.
  - b. Press **Esc**. The size handles (white blocks) will appear.
  - c. Use the size handles to drag the control to the desired position (by preference in the top left corner). Eventually you can resize the control.

As an alternative for step 3, you can change the position of the control by changing the properties of the matrix.

In the **Properties** window, select the matrix1 control. The properties for the matrix will be displayed (and the matrix control will be selected.)

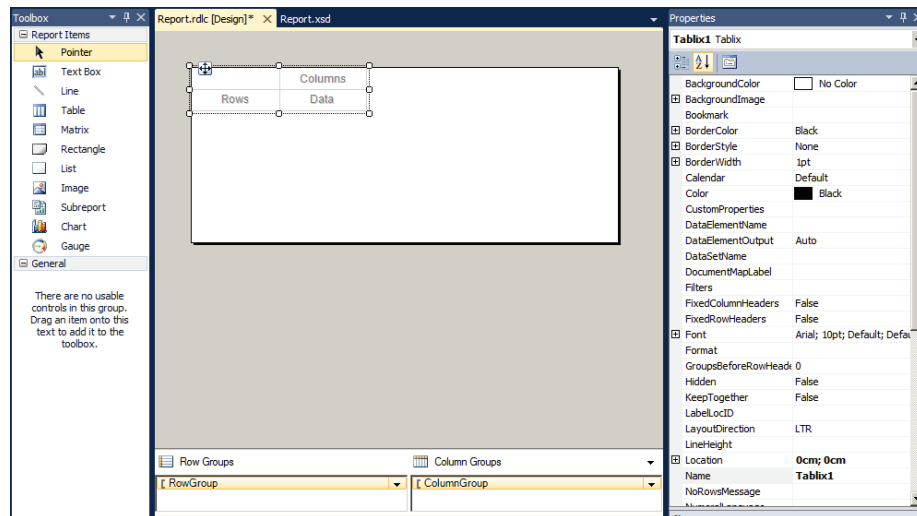


FIGURE 1.32: REPORT LAYOUT WINDOW

## Task 5: Add Dataset Elements for the Items to the Matrix

### High Level Steps

1. To add Dataset Elements for the Items to the Matrix.

### Detailed Steps

1. To add Dataset Elements for the Items to the Matrix.
  - a. In the dataset, select the Item\_\_No\_\_ element and drag it to the first cell on the second row of the matrix control.

### Task 6: Add Dataset Elements for the Locations to the Matrix

#### *High Level Steps*

1. To add Dataset Elements for the Locations to the Matrix.

#### *Detailed Steps*

1. To add Dataset Elements for the Locations to the Matrix.
  - a. In the dataset, select the Item\_Ledger\_Entry\_\_Location\_Code\_ element and drag it to the second cell on the first row of the matrix control.

### Task 7: Add Totals to the Matrix

#### *High Level Steps*

1. To add Totals to the Matrix.

#### *Detailed Steps*

1. To add Totals to the Matrix.
  - a. In the dataset, select the Item\_Ledger\_Entry\_Quantity element and drag it to the second cell on the second row of the matrix control.

### Task 8: Add Formatting to the Matrix

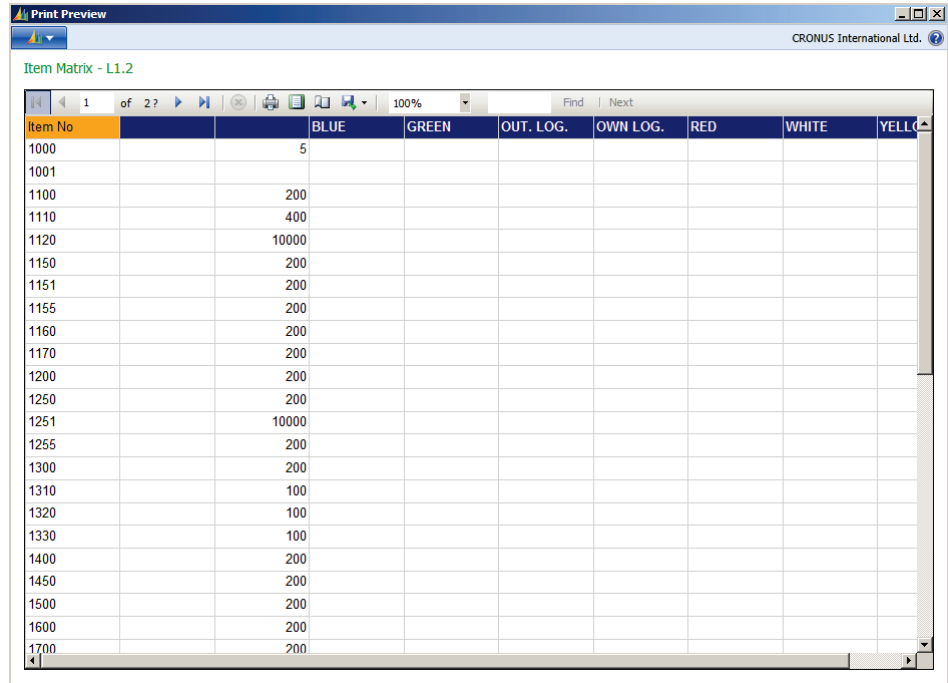
#### *High Level Steps*

1. To add Formatting to the Matrix.

#### *Detailed Steps*

1. To add Formatting to the Matrix.
  - a. Select the text box containing the **Item No.** field.
  - b. In the **Properties** window, set the BackgroundColor property to Orange.
  - c. Select the text box containing the **Location Code** field.
  - d. In the **Properties** window, set the BackgroundColor property to Dark Blue.
  - e. In the **Properties** window, set the Color property to White.
  - f. In the **Properties** window, set the FontWeight property to Bold.
  - g. Select the text box containing the **Quantity** field.
  - h. In the **Properties** window, in the BorderStyle property collection, set the Left, Right, Top, Bottom and Default properties to Solid.
  - i. Save and Exit Visual Studio, import the new RDLC layout in the object, save the object and run it.

The result will appear similar to the following image.



Item No	BLUE	GREEN	OUT. LOG.	OWN LOG.	RED	WHITE	YELL
1000	5						
1001							
1100	200						
1110	400						
1120	10000						
1150	200						
1151	200						
1155	200						
1160	200						
1170	200						
1200	200						
1250	200						
1251	10000						
1255	200						
1300	200						
1310	100						
1320	100						
1330	100						
1400	200						
1450	200						
1500	200						
1600	200						
1700	200						

**FIGURE 1.33: MATRIX REPORT PREVIEW WINDOW**

## Lab 1.3: Adding a Chart

### Scenario

As a Project Manager, Prakash is using report 1009, **Job Actual to Budget** to compare the budgeted and the realized amounts of the different jobs. To have an overall view that includes all jobs, Prakash asks Mort to present the figures in the report by means of a chart.

To avoid destroying the original report, Mort will start from report 123456709, which is a copy of the standard report 1009.

### Objectives

The purpose of this lab is to add a chart to an RDLC report layout. The chart will be based on variables from the dataset (instead of fields). Finally, you will fine-tune the chart by adding some basic formatting.

### Exercise 1: Adding a Chart to an RDLC Report Layout.

#### *Exercise Scenario*

Add a column chart to report 123456709. The chart needs to display the following pricing.

- Total budgeted cost price (Light Coral)
- Total realized cost price (Red)
- Total budgeted sales price (Light Green)
- Total realized sales price (Forest Green)

### Task 1: Import Report 123456709

#### *High Level Steps*

1. To import the object in the demonstration database, use the following procedure.

#### *Detailed Steps*

1. To import the object in the demonstration database, use the following procedure.
  - a. Open the Microsoft Dynamics NAV 2013 demonstration database.
  - b. Open the CRONUS International Ltd. demonstration company.
  - c. Select **Tools, Object Designer** to open the Object Designer.
  - d. Select **File, Import**.
  - e. Browse to the report R123456709.

- f. Click **OK** to start the import.
- g. In the dialog box, choose to open the **Import Worksheet** window.
- h. In the Action column, verify that the action is set to Create.
- i. Click **OK** to import the object in the database.

### **Task 2: Design Report 123456709**

#### ***High Level Steps***

1. Now, start designing the RDLC report layout.

#### ***Detailed Steps***

1. Now, start designing the RDLC report layout.
  - a. In the Object Designer, open report 123456709, **Job Actual to Budget - Chart** in design mode.
  - b. Select **View, Layout** to open Microsoft Visual Studio.

Since the report is a copy from the standard report 1009, it already has an RDLC layout.

### **Task 3: Add a Chart to the Report**

#### ***High Level Steps***

1. The following steps show how to add a chart to the report.

#### ***Detailed Steps***

1. The following steps show how to add a chart to the report.
  - a. Select the Body section by clicking the right grey bar or by selecting the **Body** option in the **Properties** window.
  - b. Change the height of the Body section by changing its Height property (in the Size property collection). Set the Height property to 18 cm.
  - c. Use the vertical scrollbar to visualize the bottom end of the Body section.
  - d. In the **Toolbox** window, select the **Chart control**.
  - e. Click the **Body** section to insert a Chart control.
  - f. The Select Chart Type window opens.
  - g. In this window select the Ok button to add a 3D-Column chart to the Body of the report.
  - h. Click the **Report Data** window to display the dataset.
  - i. In the dataset, select No\_Job and drag it to the Category area below of the chart. When you drag a field from the dataset over the Chart, the areas will display.



- j. Next, add the fields containing the job totals to the Data area. However, in this case, the totals are displayed using variables. In the dataset, you must select the variables that are used in the sections to display the job totals and drag them to the Data area of the Chart.
- k. Select JobTotalAmt4 and drag it to the Data area.
- l. Select JobTotalAmt5 and drag it to the Data area.
- m. Select JobTotalAmt7 and drag it to the Data area.
- n. Select JobTotalAmt8 and drag it to the Data area.
- o. Change the size of the chart by changing its properties: set the Height property to 7 cm and set the Width property to 18 cm.

### Task 4: Add Formatting to the Chart

#### High Level Steps

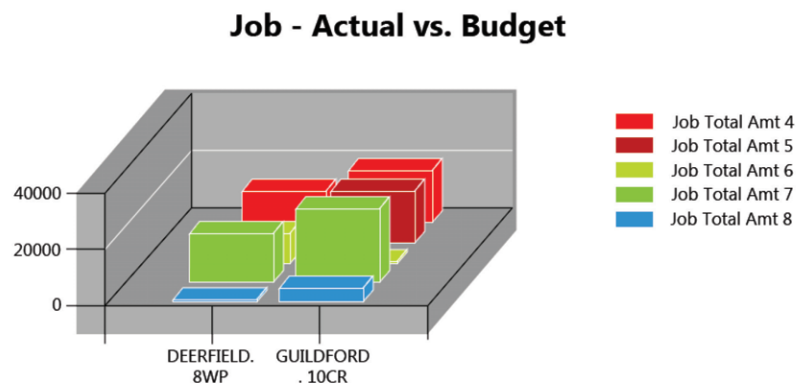
1. Now it is time to add some formatting to the report. Add a chart title, change the names in the legend, change the default colors of the columns and add some effects.  
The following steps show how to add a title to the chart.
2. The following steps show how to change the default colors of the columns.
3. Now, add labels to clarify the elements shown on the X and Y axis of the chart.
4. Add Visual Effects to the Chart.

#### Detailed Steps

1. Now it is time to add some formatting to the report. Add a chart title, change the names in the legend, change the default colors of the columns and add some effects.  
The following steps show how to add a title to the chart.
  - a. Right-click the **Title control** in the chart and select Title Properties.
  - b. On the **Title Text** field, enter Job - Actual vs. Budget.
  - c. On the **Font** tab format the title to Verdana, 12 pt, Bold and Red.
  - d. Click **OK** to close the **Properties** window.
2. The following steps show how to change the default colors of the columns.
  - a. Right-click the field **JobTotalAmt4** in the data fields at the top of the chart and select Series Properties.
  - b. On the **Fill** tab, in the **Pick Color** field: select Light Coral.
  - c. Click **OK** to close the **Properties** window.
  - d. Repeat steps 1 through 3 for the **JobTotalAmt5** value. Set the Border and Fill Color to Red.

- e. Repeat steps 1 through 3 for the **JobTotalAmt6** value. Set the Border and Fill Color to Light Green.
  - f. Repeat steps 1 through 3 for the **JobTotalAmt7** value. Set the Border and Fill Color to Forest Green.
  - g. Click **OK** to close the **Chart Properties** window.
3. Now, add labels to clarify the elements shown on the X and Y axis of the chart.
- a. Right-click the chart control and select Properties.
  - b. On the **Legend** tab, you can choose to show or hide the legend, and to change the layout and the position of the legend.
  - c. Click **OK** to close the **Chart Properties** window.
4. Add Visual Effects to the Chart.
- a. Right-click the **chart control** and select Chart Area Properties.
  - b. On the **3-D Options** tab, check the **enable 3D** field. By checking the field, the other options on the tab will be enabled.
  - c. Change the Horizontal rotation to 5°.
  - d. Change the Wall Thickness to 30 percent.
  - e. Change the Inclination to 15.
  - f. Click **OK** to close the **Chart Properties** window.
  - g. Now that the chart is finished, save and exit Visual Studio, import the new RDLC layout in the object, save the object and run it.

The result will appear similar to the following image.



**FIGURE 1.34: JOB ACTUAL TO BUDGET REPORT PREVIEW WINDOW**

## Lab 1.4: Using a Gauge

### Scenario

Susan, the Sales Manager, has asked Mort to enhance the existing report **Salesperson - Sales Statistics** so that the information presented in the report is more visually clear. This report shows amounts for sales, profit, invoice discount, payment discount, and the profit percentage, for each salesperson for a selected period. The report also shows the adjusted profit and adjusted profit percentage, that reflect any changes to the original costs of the items in the sales.

Visualizing the information in this report more clearly will ensure that Susan will have a better understanding of the figures and their ratio.

As a best practice, Mort will make a copy of the existing report and make the changes to the new report.

### Objectives

The purpose of this lab is to introduce the Gauge control and learn how to add it to a report.

### Exercise 1: Adding a Gauge Control to a Report

#### *Exercise Scenario*

- Import report 123456710, **Salesperson - Sales Statistics- Gauge**
- Add a gauge to the report
- Add dataset elements to the gauge
- Add formatting to the gauge

#### **Task 1: Import the Report**

##### *High Level Steps*

1. To import the object in the demonstration database, use the following procedure.

##### *Detailed Steps*

1. To import the object in the demonstration database, use the following procedure.
  - a. Open the Microsoft Dynamics NAV 2009 demonstration database.
  - b. Open the CRONUS International Ltd. demonstration company.
  - c. Select **Tools, Object Designer** to open the Object Designer.
  - d. Select **File, Import**.

- e. Browse to the report R123456710.
- f. Click **OK** to start the import.
- g. In the dialog box, choose to open the **Import Worksheet** window.
- h. In the Action column, verify that the action is set to Create.
- i. Click **OK** to import the object in the database.

### Task 2: Design the Report

#### High Level Steps

1. Now start designing the RDLC report layout.

#### Detailed Steps

1. Now start designing the RDLC report layout.
  - a. In the Object Designer, open report 123456710, **Salesperson - Sales Statistics- Gauge** in design mode.
  - b. Before you continue, review how the existing report is developed. The report dataset designer shows there are two data items: Salesperson/Purchaser and Cust. Ledger Entry. In the C/AL code view of the report notice the code in the OnAfterGetRecord trigger of the Cust. Ledger Entry. Data item. This is the trigger of the report where the following values are calculated.
    - SalesLCY
    - ProfitLCY
    - InvDiscLCY
    - PmtDiscGivenLCY
    - PmtToleranceLCY
    - AdjProfit
  - c. Now, select **View, Layout** to open Microsoft Visual Studio.
  - d. Since the report is a copy from the standard report 114, it already has an RDLC layout.

### Task 3: Add a Gauge to the Report

#### High Level Steps

1. The following steps show how to add a Gauge to the report.

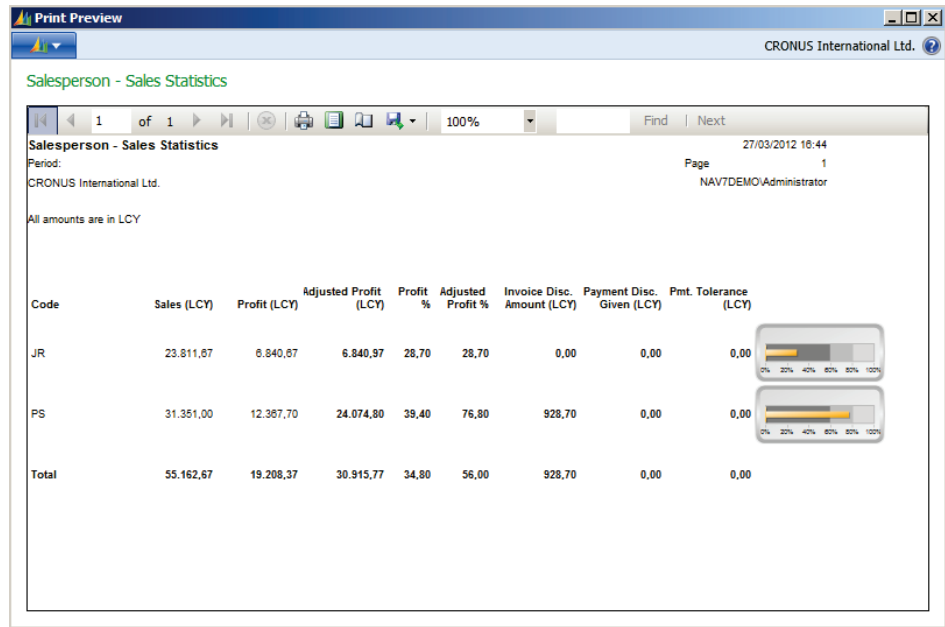
#### Detailed Steps

1. The following steps show how to add a Gauge to the report.
  - a. Select the last column of the Tablix and right-click.
  - b. Select the option: **Insert column, right**.
  - c. A new column is now added to the Tablix.

- d. Click and select the cell in the detail level of the tablix, in the newly created column.
- e. From the toolbox, double-click the **Gauge data region**, or drag it to the design surface into the selected cell. The **Select a Gauge Type** dialog box appears.
- f. Select the Bullet Graph gauge.
- g. Click **OK**.
- h. Click the **gauge** to display the **Data** pane.
- i. In the **Data** pane, right-click the **Linear Pointer 1** box, and in the drop-down menu that opens select Pointer properties.
- j. A popup window opens displaying the Linear Pointer Properties.
- k. In this window, click the **Fx** button, next to the value field.
- l. In the **Expression Designer** window type the following expression.
- m. =Code.CalcPct(Fields!Cust\_Ledger\_Entry\_\_Sales\_LCY\_\_.Value, Fields!AdjProfit.Value).
- n. Click **OK**.
- o. Click **OK**.
- p. Now select the cell in which the Gauge is added.
- q. In the **Properties** window set the Size properties to the following values: Width: 3 cm, Height: 1,4 cm.
- r. Now that the report is finished, save and exit Visual Studio, import the new RDLC layout in the object, save the object and run it.

# Reporting in Microsoft Dynamics® NAV 2013

The report should now appear similar to the following image.



**FIGURE 1.35: REPORT WITH GAUGE**

## Module Review

### ***Module Review and Takeaways***

This module explained the architecture of Microsoft Dynamics NAV 2013 and the Visual Studio integrated development environment. It also examined the integration between the Report Dataset Designer and the Visual Studio Report Designer in relation to report creation, report execution, dataset generation, sorting and filtering data. Additionally, the module discussed some of the most frequently used properties and functions.

### **Test Your Knowledge**

Test your knowledge with the following questions.

1. Which report items are not available in the Visual Studio Report Designer?
  - ( ) Text box
  - ( ) Chart
  - ( ) Gauge
  - ( ) Drop-down list
2. Which property can be set on a column to influence the name of the corresponding element in the dataset?
  - ( ) DataSetName
  - ( ) Name
  - ( ) DataSetSourceExpr
  - ( ) DataSetID
3. What does RDL stand for?
  - ( ) Report Dataset Language
  - ( ) Report Definition Language
  - ( ) Report Design Language
  - ( ) Report Data Language

4. Which export formats are offered by the report viewer control?
- XML
  - PDF
  - Microsoft Office Excel
  - HTML
5. Which data item properties are supported by the report designer in the Microsoft Dynamics NAV 2013 ?
- DataItemTableView
  - DataItemLink
  - NewPagePerRecord
  - TotalFields



## Test Your Knowledge Solutions

### Module Review and Takeaways

1. Which report items are not available in the Visual Studio Report Designer?
  - Text box
  - Chart
  - Gauge
  - Drop-down list
2. Which property can be set on a column to influence the name of the corresponding element in the dataset?
  - DataSetName
  - Name
  - DataSetSourceExpr
  - DataSetID
3. What does RDL stand for?
  - Report Dataset Language
  - Report Definition Language
  - Report Design Language
  - Report Data Language
4. Which export formats are offered by the report viewer control?
  - XML
  - PDF
  - Microsoft Office Excel
  - HTML

5. Which data item properties are supported by the report designer in the Microsoft Dynamics NAV 2013 ?

DataItemTableView

DataItemLink

NewPagePerRecord

TotalFields