

MODULE 5: RUNNING REPORTS

Module Overview

In Microsoft Dynamics® NAV 2013, reports can be incorporated into menus, or they can be called from an action on a page.

Sometimes, when you design reports, you might want to run them before they are integrated into the application. In the earlier modules, you used the object designer to run reports. This module will show how to integrate the reports into the RoleTailored client interface and how to create hyperlinks to reports.

Objectives

The objectives are:

- Use the features of the ReportViewer control.
- Describe how to run a report from the RoleTailored client.
- Examine how to add a report to a page and the Departments section.
- Examine how to create and run hyperlinks to reports at the command prompt, the Start menu, the Desktop and Microsoft Internet Explorer.
- Describe how to run a report from a SharePoint site.
- Describe how to create a hyperlink to a page.
- Describe how to create a hyperlink to a report.
- Review where to locate additional resources and information about reporting.
- Run a stand-alone report by using Visual Studio.

The ReportViewer Control

Microsoft Visual Studio 2010 includes report design functionality and ReportViewer controls so that you can add full-featured reports to custom applications. Reports can contain tabular, aggregated, and multidimensional data. ReportViewer controls are provided so that you can process and display the report in your application. There are two versions of the control. The ReportViewer web server control is used to host reports in ASP.NET projects. The ReportViewer Windows Forms control is used to host reports in Microsoft Windows application projects. The ReportViewer control processes .rdlc files.

Both controls can be configured to run in local processing mode or remote processing mode. How you configure the processing mode affects everything about the report from design to deployment.

- *Local processing mode* refers to report processing that is performed by the ReportViewer control in the client application. All report processing is performed as a local process by using data that your application provides.
- *Remote processing mode* refers to report processing that is performed by a Microsoft SQL Server Reporting Services report server. In remote processing mode, the ReportViewer control is used as a viewer to display a predefined report that is already published on a Reporting Services report server. All processing from data retrieval to report rendering is performed on the report server.

The Report Viewer control is part of the **Report Preview** Page.

ReportViewer Toolbar

The ReportViewer control includes a toolbar that provides navigation, search, export, and print functionality so that you can work with reports in a deployed application.

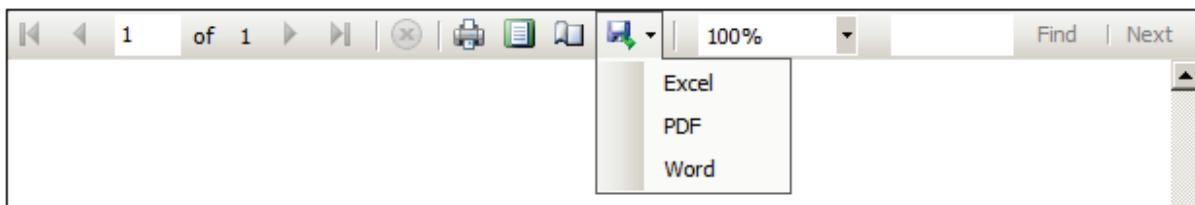


FIGURE 5.1: THE REPORTVIEWER TOOLBAR

Document Maps

A document map is a navigation area that is attached to the left side of the report view area. It includes a list of links that users can click to browse to a specific area of the report. Not all reports have a document map. You must define one in the report definition to use this feature. For more information about document maps and other interactive report features, refer to the "Interactive Features" module.

Stopping a Report

You can use the **Stop** button to stop report rendering. The **Stop** button can be used to stop a report when you test reports or when a report has slow performance.

Zooming the Report Page

The ReportViewer toolbar provides standard zoom functionality so that you can increase or decrease the report.

Searching a Report

The ReportViewer toolbar includes a search field so that you can find specific text in a report. You can search for content in the report by typing a word or phrase that you want to find. The search is case-insensitive and begins at the page or section that is selected. Wildcard characters and Boolean search operators are not supported. Only visible content is included in a search operation. If the report uses show / hide functionality, hidden content is not exposed through search operations. To search for the next occurrence of the same value, click **Next**.

Previewing a Report and Setting Page Sizes

The ReportViewer toolbar includes commands for viewing a report in print layout and setting page sizes. When you click **Print Layout**, the ReportViewer toolbar is updated to show just the commands that you can use during preview. **Print Layout** is a toggle command. You can switch between preview and print layout by clicking this button. When you click **Page Size**, you can specify page dimensions that are used only for print output. The page dimensions are initialized with values from the report definition. However, you can override the values when you print. You cannot save the values with the report.



Note: Note that if in the code of the report the function `CurrReport.PREVIEW` is used, then the print button will not be available in preview mode.

Exporting a Report

The ReportViewer toolbar provides export formats so that you can save a report as an Excel, PDF or Word program file.

The same report can have a different appearance and functionality, depending on the rendering format that you select. Reports that have links, document maps, and bookmarks might not function correctly after the report is saved to a file.

Depending on how items are aligned in a report, the report layout in a different file format might include additional pages or white space that you did not expect. For more information, refer to the "Design Considerations" module in the section about pagination.

 **Best Practice:** It's strongly encouraged to test the development with all render extension (Preview, Print, PDF, Excel, Word)

Printing a Report

The ReportViewer toolbar provides print support. You can specify page settings and page orientation for a printed report from the following user interface (UI) locations:

- **Report Properties** dialog box on the client report definition (RDLC) report layout in Visual Studio.
- **Properties** dialog box for the printer.
- **Print** dialog box that is displayed immediately before you print.
- **Page Setup** dialog box from the **Print Preview** page in the Reportviewer.
- **Print** dialog box from the **Print Preview** page in the Microsoft Dynamics NAV Portal Framework for Microsoft SharePoint 2010.

The page settings that are used for the printed report depend on the mode in which you print the report.

You can use the following modes to print a report:

- Run a report that has the UseRequestPage Property set to No, and does not display a request page.
- Print a report from the request page without previewing the report.
- Print a report from the **Print Preview** page.

The printer that you use determines what to do if the actual paper size in the printer differs from the page size that is specified.

Run a Report in Microsoft Dynamics NAV

After a report is created and designed by using Visual Studio, you can add the report to the interface of the application to make it available to the end-users.

There are several ways to add reports. These include the following:

- In the **Action Pane** of a page.
- In the **Role Center** menu.
- In the **Activities** page of a Role Center or in the Departments section.

Adding Reports to a Page

The following procedure explains how to make a report available in the application by adding the report to the **Action Pane** in the **Item List** page.

1. In the Microsoft Dynamics NAV development environment, open Object Designer, and then click **Page**.
2. Select the page to display the report, for example, Page 31, Item List.
3. Click **Design** to open the page in Page Designer.
4. Select an empty line in the designer. Click **View**, and then **Actions** to open Action Designer.
5. Scroll down to the ActionContainer that has **Reports** as its subtype.
6. Select the line that has **Inventory - List** as its caption.
7. Press **F3** to insert a new empty line (or select Edit, New).
8. On the new line, add an action to the list.
9. In the **Caption** field, enter a name for the action: **Inventory Matrix**.
10. In the **Type** field, select Action.
11. Click **View**, and then **Properties** to open the **Properties** window for the new action.
12. In the **Value** field of the RunObject property, click the drop-down arrow and select report 123456701 in the **Object List** window.
13. In the **Properties** window, set the **Promoted** property to **Yes**, and then set the **PromotedCategory** property to **Report**.

Reporting in Microsoft Dynamics® NAV 2013

14. Compile and save the page.
15. Open Dynamics NAV 2013 and then open the Item List page.

The screenshot shows the Microsoft Dynamics NAV Items screen. The top navigation bar includes Home, Actions, Navigate, Report - Inventory, Report - Sales, Report - Purchases, Report - Finance & Costing, Report - Manufacturing, and a search bar. Below the navigation bar is a toolbar with various icons for actions like New, Edit, View, Delete, and Reports. A red circle highlights the 'Inventory Matrix Report' icon. The main area displays a grid of item details, and on the right, there are two expandable sections titled 'Item Details - Invoic...' and 'Item Details - Plann...'. At the bottom left, the status bar shows 'CRONUS International Ltd.' and the date 'Thursday, January 23, 2014'. On the far left, the Action Pane is visible, showing items like Role Center, Sales Orders, Sales Quotes, and Sales Journals, with 'Items' selected.

FIGURE 5.2: THE NEW REPORT IS ADDED TO A PAGE.

The report is added to the list of promoted actions. Click the action to run the report. The report is also available in the **Reports** menu that appears over the **Action Pane**.

Adding Reports to a Role Center Menu

Ellen wants the Inventory Matrix report to be available for her colleague John. Mort will add it to the menu of John's Role Center.

1. In the Microsoft Dynamics NAV development environment, open Object Designer.
2. Click the **Page** button.
3. Select page **9009, Whse. Worker WMS Role Center**.
4. Click **Design** to open the page in Page Designer.
5. Select an empty line in the designer. Click **View**, and then **Page Actions** to open Action Designer.
6. Scroll down to the Action that has Customer Labels as its Caption.
7. Click the **Separator** button to insert a separator.
8. Move the pointer one line down.
9. Press **F3** to insert a new empty line (or select Edit, New).

Module 5: Running Reports

10. On the new line, in the **Caption** field, enter a name for the action: **Inventory Matrix**.
11. In the **Type** field, select Action.
12. Click **View**, and then **Properties** to open the **Properties** window for the new action.
13. In the **Value** field of the RunObject property, click the drop-down arrow and select **report 123456701** in the **Object List** window.
14. Compile and save the page.
15. Open the RoleTailored client.
16. In the **navigation** pane, click the **Role Center** page.

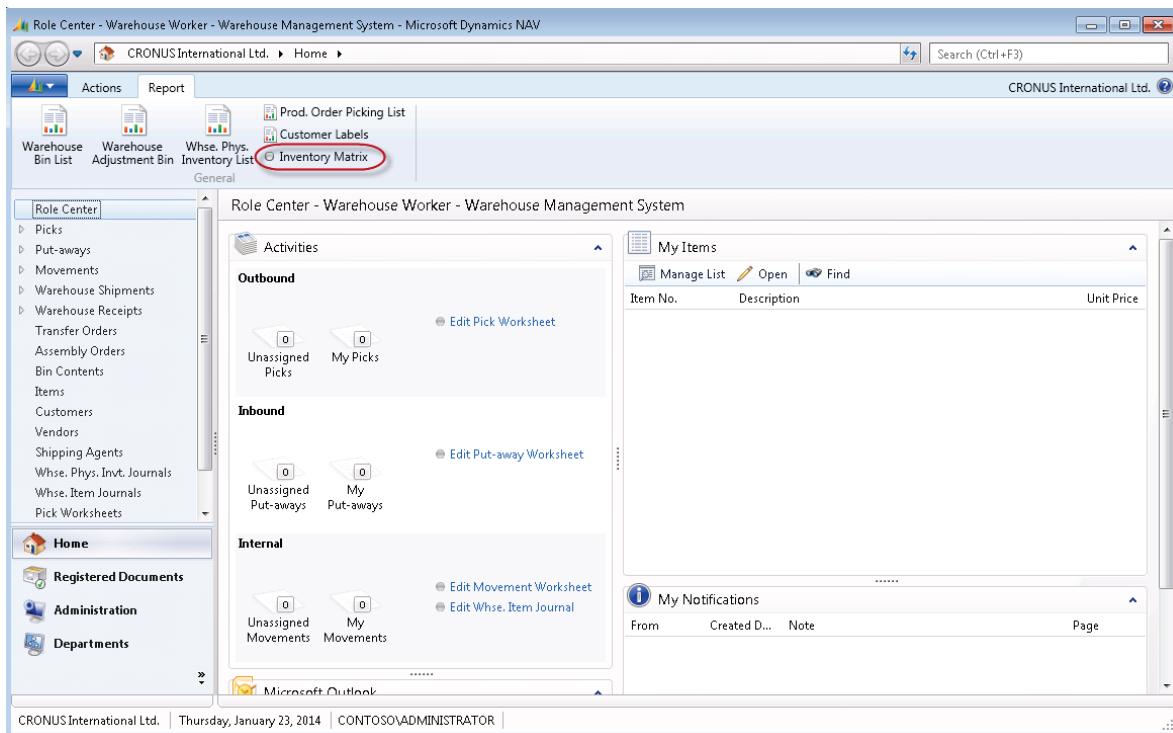


FIGURE 5.3: THE NEW REPORT IS ADDED TO THE ROLE CENTER MENU

Now the report is added to the **Role Center** menu. Click the action to run the report.

Adding Reports to the Role Center Activities

John is not that experienced with computers. He regularly calls Mort to ask where he can find the **Inventory Matrix** report. To help John, Mort will make the report accessible as a link on the Role Center page.

1. In the Microsoft Dynamics NAV development environment, open Object Designer.
2. Click the **Page** button.

Reporting in Microsoft Dynamics® NAV 2013

3. Select page **9056, Warehouse Worker Activities**.
4. Click **Design** to open the page in Page Designer.
5. Scroll down to the Group that has Internal as its Caption.
6. Select **View, Control Actions**.
7. On the first empty line, in the **Caption** field, enter a name for the action: Print Inventory Matrix.
8. In the **Type** field, select Action.
9. Click **View**, and then **Properties** to open the **Properties** window for the new action.
10. In the **Value** field of the RunObject property, click the drop-down arrow and select **report 123456701** in the **Object List** window.
11. Compile and save the page.
12. Open the RoleTailored client and then open the **Role Center** page.

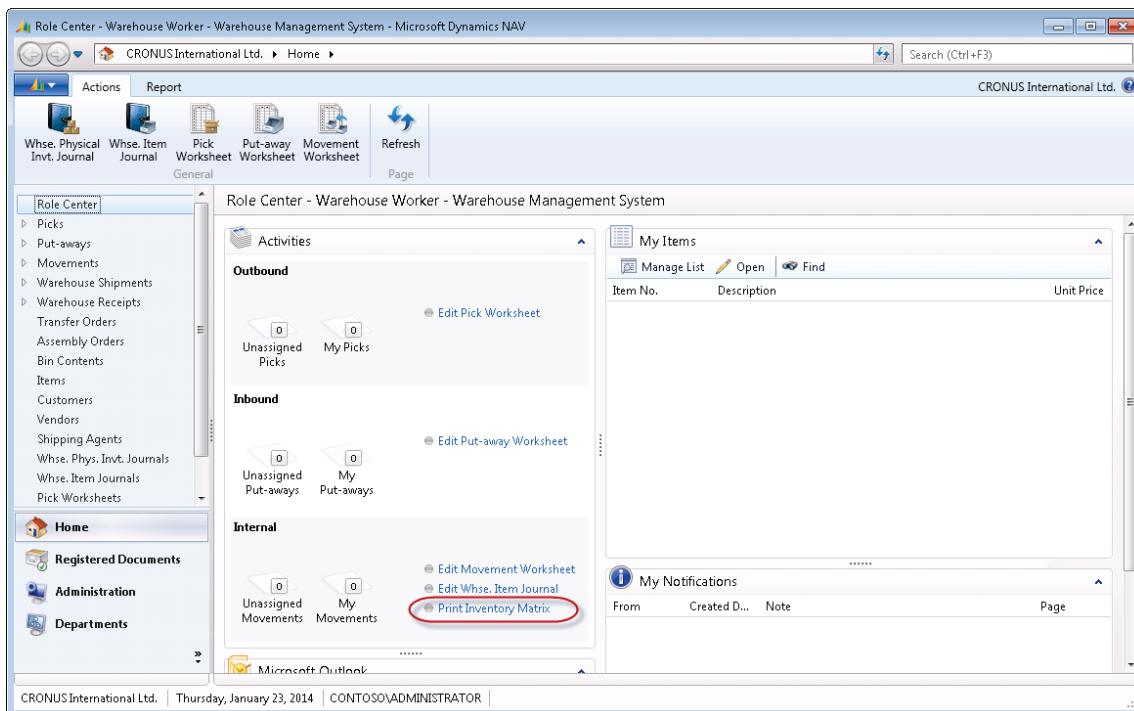


FIGURE 5.4: THE NEW REPORT IS ADDED TO THE ACTIVITIES PAGE

Now the report is added to the Internal group in the Role Center page. Click the action to run the report.

Adding Reports to a Departments Section

In Microsoft Dynamics NAV 2013, navigation through the application is performed by using the **navigation** pane. The **navigation** pane is made of MenuSuite objects.

MenuSuite objects contain the main menu content that is displayed in the **Departments** page and have a layered structure. In Dynamics NAV 2013, application objects, such as pages and reports, are accessed from either the Departments page or the Role Center page.

The Departments item in the **navigation** pane provides links to all pages. The links and pages under Departments are generated automatically based on the menu suite of the RoleTailored client user.

The **Departments** page provides access to reports and pages and it is built by using MenuSuite objects.

Unlike any other application objects, the ID of the MenuSuite is defined by the development environment, depending on the MenuSuite level. The ID of a MenuSuite is shown in the following table.

Level	MenuSuite Object
MBS	1010 Dept - MBS
Add-on 1	1051 Dept - Add-on 1
Company	1090 Dept - Company

Reporting in Microsoft Dynamics® NAV 2013

To modify the **navigation** pane or the **Departments** page, you can design the existing or create new MenuSuite objects. When you create a new MenuSuite object, you have to select the menu level.

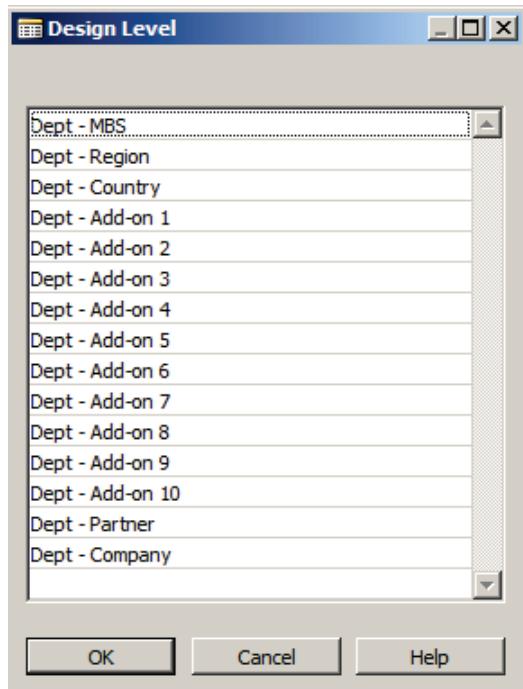
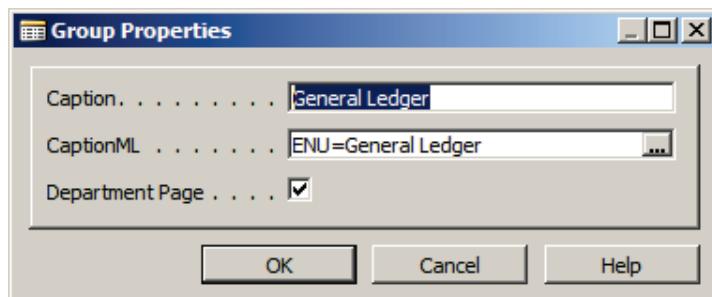


FIGURE 5.5: THE GROUP PROPERTIES WINDOW

When you create menu groups in the MenuSuite, a property **Department** page is available.



When you add menu items to a MenuSuite, you must specify the **Department** category to which the menu item belongs.

There are six categories:

- Lists
- Tasks

- Reports and Analysis
- Documents
- History
- Administration

The **Department** property is used to automatically generate the navigation structure.

Adding Reports to a Departments Section

Prakash wants to print the **Job Actual To Budget Chart** report (123456709). Instead of printing it by using the **Start** menu, he asks Mort to integrate the report in the **Job** menu on the departments section. Mort will first create a new MenuSuite object.

1. In the Microsoft Dynamics NAV development environment, open Object Designer.
2. Click the **MenuSuite** button.
3. Click the **New** button to open the **Design Level** window.
4. In the list, select the **Dept - Partner** level.
5. In the navigation pane designer, select the **Jobs** menu.
6. Expand the **Reports** menu group and select the last report in the group: Jobs per Item.
7. Right-click the menu item and select Create Item.
8. In the **Object Type** field, select Report.
9. In the **Object ID** field, click the drop-down arrow and select report 123456709 in the **Object List** window.
10. In the **Caption** field, change the caption to Job Actual to Budget - Chart.
11. In the **Department Category** field, select Reports and Analysis.
12. Click **OK** to close the **Create Item** window.
13. Compile and save the MenuSuite.
14. Open the RoleTailored client and then open the **Departments** page.
15. Click the **Jobs item**.

Reporting in Microsoft Dynamics® NAV 2013

Now the new report is added to the Reports and Analysis category.

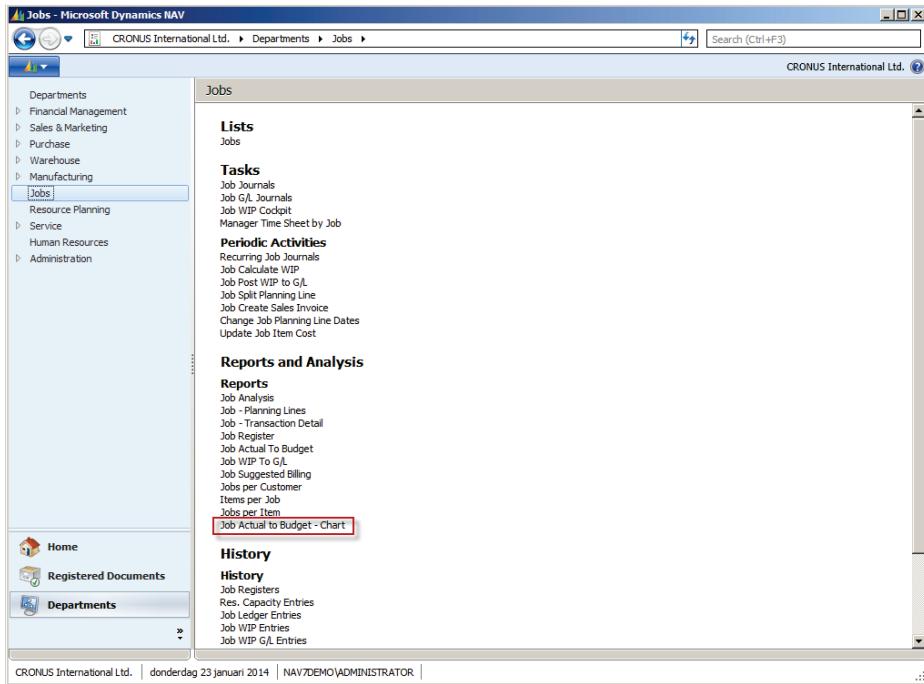


FIGURE 5.7: REPORT ADDED TO THE DEPARTMENTS SECTION

Hyperlinks to Reports

You can use hyperlinks to send or save quick links to specific pages in Microsoft Dynamics NAV. For example, you can create a hyperlink to a specific list page, such as **Customers**. This lesson explains how to create and run hyperlinks to reports.

Create Hyperlinks

To create a hyperlink, type the Microsoft Dynamics NAV Client Executable and Protocol handler and then type the syntax for the hyperlink. For example:

Code Example

```
Microsoft.Dynamics.Nav.Client.exe  
"DynamicsNAV://MyServer/DynamicsNAV70/CRONUS International  
Ltd./RunPage?Page=22"
```

This example specifies a server name, service, company, and page ID. If it is entered correctly, it opens the RoleTailored client and runs page 22, the **Customer List** page.

Building a URL

When you build a hyperlink URL, you can specify a page, a report, a query, an XMLport, a codeunit, or a node:

- DynamicsNAV://server/instance/company/runpage?page=22&bookmark=0ABA0700235752C7D1
- DynamicsNAV:// server/instance/company/runreport?report=901
- DynamicsNAV:// server/instance/company/runquery?query=50000
- DynamicsNAV://
server/instance/company/runxmlport?xmlport=99008500
- DynamicsNAV://
server/instance/company/runcodeunit?codeunit=50000
- DynamicsNAV://
server/instance/company/navigate?node=Home/Items

The first three parameters to a client URL are reserved for server, service, and company. You can omit certain parameters. However, if you do this, the default values are assumed. Therefore, you must leave the correct number of forward slashes to indicate the parameters that you have omitted. If you omit just one of these parameters, then it must be company. If you omit two parameters, then they must be company and instance. You can also omit all three parameters. For example, the following URL assumes the default instance and company on the MyServer server.

Code Example

```
DynamicsNAV://MyServer///RunPage?Page=22
```

You can also specify a server port:

Code Example

```
DynamicsNAV://MyServer:1234///RunPage?Page=22
```

This URL omits server, service, and company:

Code Example

```
DynamicsNAV:///navigate?node=Home/Items
```

Reporting in Microsoft Dynamics® NAV 2013

You can also specify the following additional parameters in a URL.

Parameter	Description
Personalization ID	<p>Specifies the unique identification used in personalization to store settings in the User Metadata table. If a personalization ID is not found, the page is started without personalization.</p> <p>Example:</p> <p><code>DynamicsNAV://localhost/DynamicsNAV70/CRONUS%20International%20Ltd./runpage?page=22&personalization=0000232e-0000-001a-0008-0000836bd2d2</code></p>
Bookmark	<p>Positions the pointer on a single record in a table.</p> <p>Only automatically generated bookmarks are used. If an incorrect bookmark is entered, you will receive an error message.</p> <p>Example:</p> <p><code>DynamicsNAV://localhost/DynamicsNAV70/CRONUS%20International%20Ltd./runpage?page=22&bookmark=120000000089083237343</code></p>
Mode	<p>Used to open a page in a specific mode.</p> <p>Other modes include view, edit, create, select, and delete.</p> <p>Example:</p> <p><code>DynamicsNAV://localhost/DynamicsNAV70/CRONUS%20International%20Ltd./runpage?page=22&mode=View</code></p>

Syntax for Creating Hyperlinks

The following table shows some examples of hyperlinks and provides information about how to specify parameters.

Parameters	Description	Synthax	Example
Company name	You can switch a company (case-sensitive). If you specify a company then you must also specify a server and an instance.	DynamicsNAV:///<CompanyName>/RunPage?Page=<page id>	DynamicsNAV:///CRO NUS%20International% 20Ltd./RunPage?Page=22
Navigate	Users can send or save quick links to specific pages.	DynamicsNAV://///n avigate?node=<service>	DynamicsNAV://///navi gate?node=Home/Ite ms
RunPage	You can run a specific page.	DynamicsNAV://///ru npage?page=<page id>	DynamicsNAV://///runp age?page=42
RunReport	You can run a specific report.	DynamicsNAV://///ru nreport?report=<re port id>	DynamicsNAV://///runr eport?report=50000
RunQuery	You can run the About This Query page for a specific query.	DynamicsNAV://///ru nquery?query=<que ry id>	DynamicsNAV://///runq uery?query=50000
RunXMLport	You can run a specific XMLport.	DynamicsNAV://///ru nxmlport?xmlport=<xmlport id>	DynamicsNAV://///runx mlport?xmlport=99008 500
RunCodeunit	You can run a specific codeunit.	DynamicsNAV://///ru ncodeunit?codeunit=<codeunit id>	DynamicsNAV://///runc odeunit?codeunit=500 00
Server name	You can specify servers.	DynamicsNAV://<Se rverName>///RunPa ge?Page=<pageid>	Microsoft.Dynamics.Na v.Client.exe DynamicsNAV://MySer ver///RunPage?Page=2 2
Server port	You can specify an optional server port number in the range 1-65535. The default port is 7046.	DynamicsNAV://<Se rverName><:Port>/ //RunPage?Page=<pageid>	DynamicsNAV://MySer ver:1234///RunPage?Pa ge=22
Service instance	You can specify a Service instance. You can find this value in the CustomSettings.config file. If you specify an instance you must also specify a server.	DynamicsNAV://<S ervice>//RunPage?P age=<pageid>	Microsoft.Dynamics.Na v.Client.exe DynamicsNAV://MySer ver//RunPage?Page=22

If you omit a parameter, such as Server or Service, do not remove the accompanying forward slash. If you remove the accompanying forward slash, it will change the overall structure of the hyperlink.

When you type a company name as a parameter, put the hyperlink in quotation marks if there are blank spaces in the name.

 **Best Practice:** When you enter a company name on the command line, enclose the quotation marks to make sure that spaces and other special characters are interpreted correctly. For URLs, use percent encoding to escape any characters greater than 127 in the ASCII character set.

Run Hyperlinks

In Microsoft Dynamics NAV, you can run report hyperlinks in different ways. You can run the report from the Run window, a Command Prompt, a shortcut or from a browser window. This is useful if, for example, you want to run a specific report for testing, or if you know the ID of a report that is frequently run.

The following demonstrations illustrate how to run hyperlinks.

Method	To run a hyperlink
Run window	Click Start , and then click Run . Enter the hyperlink in the Run window, and then click OK .
Command prompt window	At the command prompt, type the hyperlink.
Shortcut	Create a shortcut for the Microsoft Dynamics NAV executable on your Desktop. Right-click the shortcut , click Properties , and then enter the hyperlink URL in the Target field of the shortcut. Click OK , and then start Microsoft Dynamics NAV.
Browser window	Use a browser such as Internet Explorer and type the hyperlink URL directly into the address bar. To type hyperlinks into a browser, you must have Microsoft Dynamics NAV installed on your computer.

Module 5: Running Reports

Method	To run a hyperlink
Web Reference for a web service	Create a Codeunit that saves the report to PDF, Excel or Word. Publish the Codeunit as a web service. Create an application that includes the web service, for example an ASP.NET website.

Demonstration: Run a Report from the Run Window

Demonstration Steps

1. Run a report from the **Run** window.
 - a. In Microsoft Windows, click **Start**, and then click **Run**.
 - b. In the **Run** dialog box, enter the following command:

Code Example

```
"Microsoft.Dynamics.Nav.Client.exe"  
DynamicsNAV:///runreport?report=<ReportID>
```

Replace <ReportID> with the report ID that you want to use 111.

- c. Click **OK** to run the report.

Demonstration: Run a Report at the Command Prompt

Demonstration Steps

1. Run the report at the command prompt.
 - a. Click **Start**, and then click **Run**.
 - b. In the **Run** window, enter cmd.
 - c. In the **Command Prompt** window, enter the following command:: "C:\Program Files (x86)\Microsoft Dynamics NAV\70\RoleTailored Client\Microsoft.Dynamics.Nav.Client.exe" DynamicsNAV:///runreport?report=111
 - d. Press **Enter**.

Demonstration: Run a Report from a Shortcut

To run a report from the Windows Desktop, you can create a shortcut to the report.

Demonstration Steps

1. Use a shortcut to run a report.
 - a. On your Windows Desktop, create a shortcut to the **Microsoft.Dynamics.Nav.Client.exe**.
 - b. Right-click the shortcut and select Properties.
 - c. In the **Target** field, add the hyperlink URL. The complete target will resemble this: "C:\Program Files (x86)\Microsoft Dynamics NAV\70\RoleTailored Client\Microsoft.Dynamics.Nav.Client.exe" DynamicsNAV:///runreport?report=111

- d. Click **OK** to close the **Properties** window.
- e. Double-click the shortcut to run the report.

Demonstration: Run a Report from a Browser Window

Demonstration Steps

1. Run the report from the browser window.
 - a. Open Microsoft Internet Explorer.
 - b. In the address bar; enter the following URL:
dynamicsnav:///runreport?report=111
 - c. Press **Enter** to validate the URL.

The first time that you execute this command, you will receive a confirmation message, as is shown in the Run Report Confirmation figure.

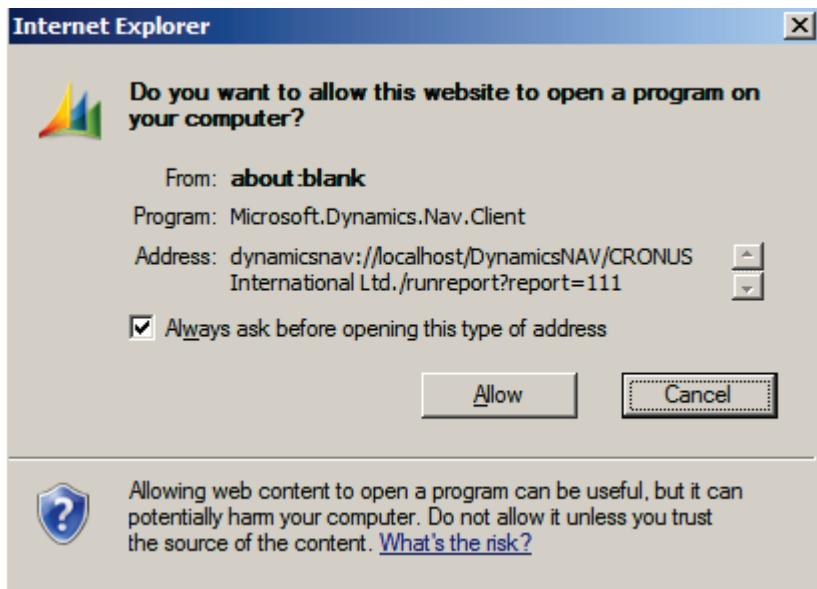


FIGURE 5.8: RUN REPORT CONFIRMATION WINDOW

Demonstration: Add a Report Hyperlink to a Webpage

Kevin has asked Tim, the IT manager, to conduct a small feasibility study for building a Customer Portal. A primary goal of this portal is to provide all outgoing sales documents electronically. By doing this, customers can download the documents, and they will also be able to print specific reports.

Tim checks whether he can start a report from a webpage.

Demonstration Steps

1. Run a report from a webpage.
 - a. Click **Start**, and then open Microsoft Visual Studio 2010.
 - b. Select File, New, Web Site.
 - c. In the new website window, select the ASP.NET website template.
 - d. In the upper-right corner, leave the **.NET Framework 4** option.
 - e. In the Location field, enter the following path:
C:\Users\%USERNAME%\Documents\Visual Studio
2010\WebSites\HyperlinkTest (where %USERNAME% represents
your Windows account).
 - f. In the **Language** field, select Visual Basic.
 - g. Click the **OK** button to create a new website.
 - h. Click the **Source** button at the bottom of the window.
 - i. Modify the code until it resembles the following line of code:

Code Example

```
<%@ Page Title="Home Page" Language="VB" MasterPageFile="~/Site.Master"
AutoEventWireup="false"

CodeFile="Default.aspx.vb" Inherits="_Default" %>

<asp:Content ID="HeaderContent" runat="server"
ContentPlaceHolderID="HeadContent">

</asp:Content>

<asp:Content ID="BodyContent" runat="server"
ContentPlaceHolderID="MainContent">

<div>

    <a href="dynamicsnav://localhost/DynamicsNAV70/CRONUS International
Ltd./runreport?report=111">Customer Top 10 List</a>

</div>

</asp:Content>
```

- j. Press **F5** to run the website.
- k. If the **Debugging Not Enabled** dialog box appears, select the **Run without debugging** option and then click the **OK** button.

The result will resemble this:

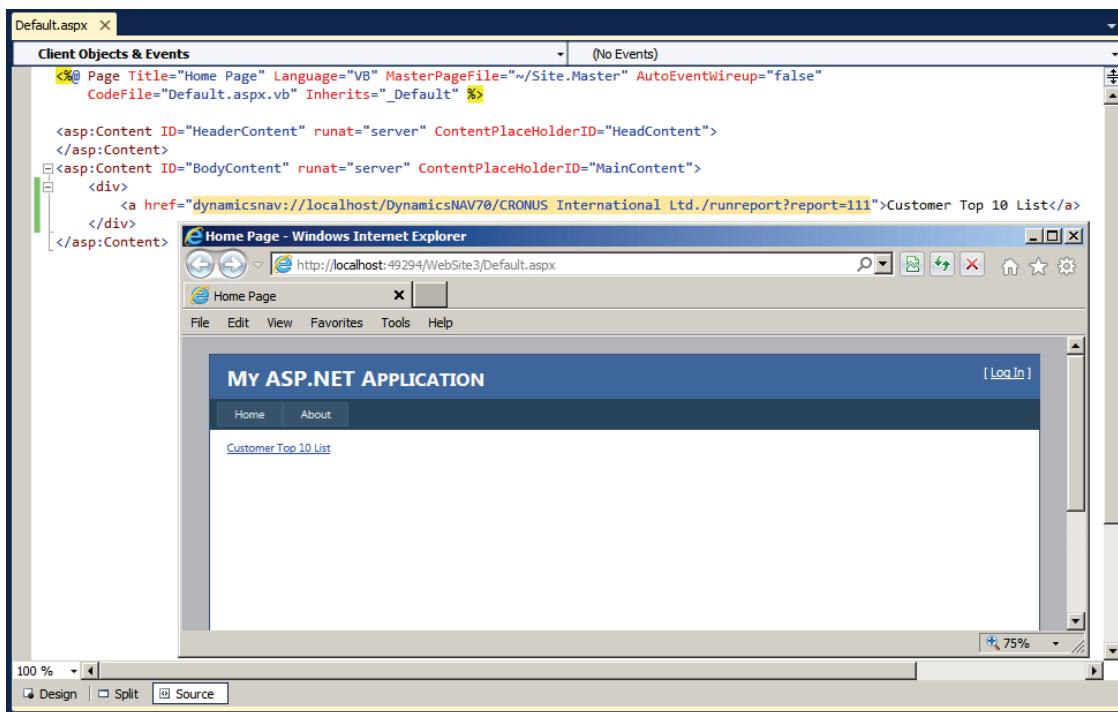


FIGURE 5.9: THE WEBSITE PAGE IN SOURCE VIEW WINDOW

Demonstration: Add a Report Hyperlink to Favorites

Now that Tim knows that links to a report can be included on a webpage, he decides do to more testing. Tim wants to know whether links to reports can be added to the **Favorites** menu in Internet Explorer, and he begins to run the **HyperlinkTest** site just created.

Demonstration Steps

1. Add a report to the **Favorites** menu.
 - a. Open Microsoft Internet Explorer.
 - b. In the address bar; enter the following URL:
`dynamicsnav:///runreport?report=111`
 - c. Press Enter to validate the URL.
 - d. Right-click the **The Customer Top 10 List** link.
 - e. Select Add to Favorites.
 - f. In the **Add a Favorite** window, enter the following name for the report: Customer - Top 10 List.
 - g. Click the **New Folder** button.

Reporting in Microsoft Dynamics® NAV 2013

- h. In the Create a Folder window, enter the following Folder Name: DynamicsNAV Reports.
- i. In the **Create In** drop-down list, select Favorites.
- j. Click the **Create** button to create the folder and to return to the **Add a Favorite** window. Notice that the **Create In** field points to the newly created folder.

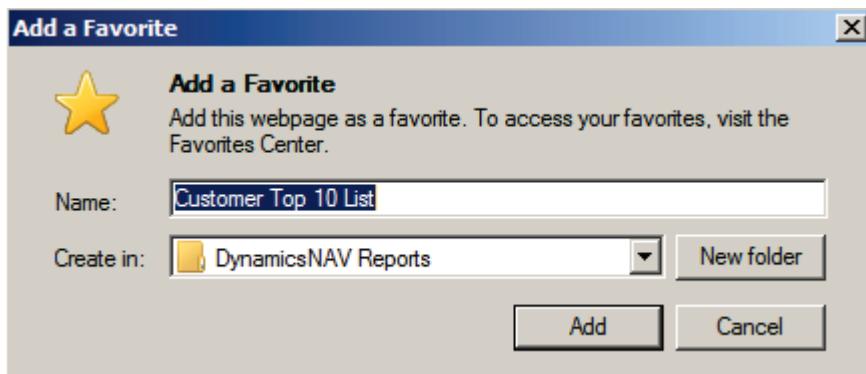


FIGURE 5.10: CUSTOMER TOP 10 LIST PROPERTIES WINDOW

- k. Click the **Add** button to close the **Add a Favorite** window.
- l. In Internet Explorer, select Favorites, DynamicsNAV Reports.
- m. Click the **Customer Top 10 List** link. The RoleTailored client will be started and the **Request Options** page for the report is displayed.

Tim is satisfied with the results of his tests. He decides to check one more thing.

1. Verify the URL.
 - a. In Internet Explorer, select Favorites, DynamicsNAV Reports.
 - b. Right-click the **Customer Top 10 List** link.
 - c. Select Properties.

- d. In the **Customer Top 10 List Properties** window, click the **Web Document** tab.

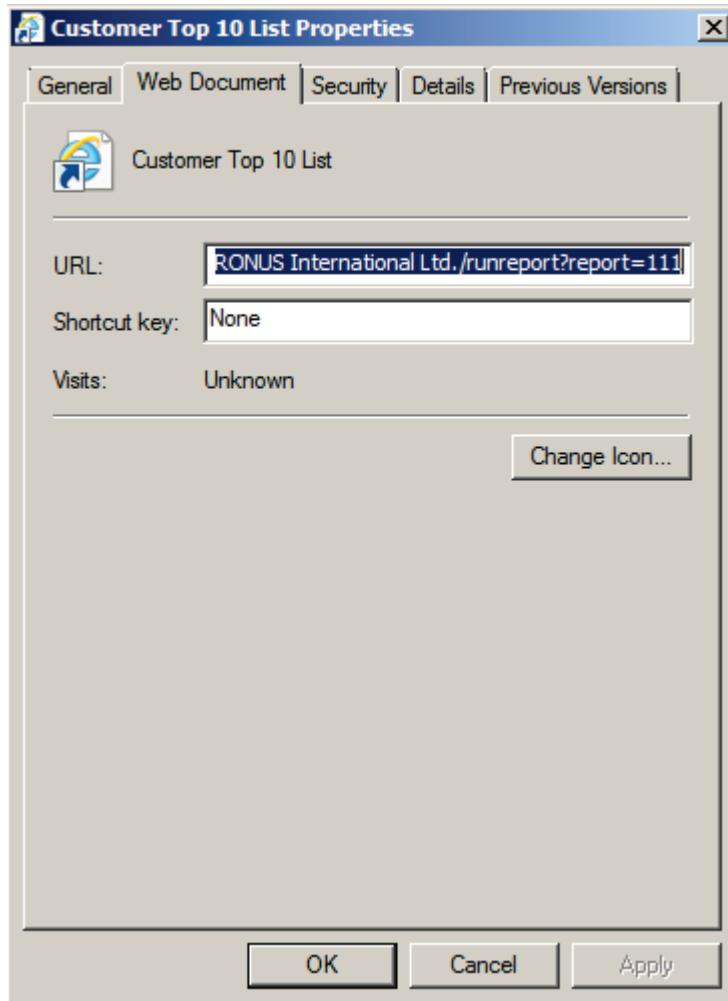


FIGURE 5.11: URL PROPERTIES WINDOW

Tim knows that the **URL** field contains the editable link to the report.



Run a Report from the SharePoint Site

You can use the Microsoft Dynamics NAV Portal Framework to display Microsoft Dynamics NAV reports on Microsoft SharePoint sites. There are two ways to display pages and reports:

- Display reports in application pages by using a URL.

You can display a report on a SharePoint site by using a URL. To display the report, you can type the URL directly in the address of a web browser or use it to create links to the report from other SharePoint pages. Reports open in an application of the SharePoint site.

- Create a page action that runs the report.

By using the Microsoft Dynamics NAV development environment, you can create an action to a page, and then set the action to run the report.



Additional Reading: For more information, refer to *How to Add Actions to a Page*: <http://go.microsoft.com/fwlink/?LinkId=267285>

Limitations of Reports on a SharePoint Site

Limitations exist when you create reports by using the Microsoft Dynamics NAV Portal Framework compared to when you use the RoleTailored client. These limitations include the following:

- Filtering

There is a limit on the number of filters a user can set. The user can only specify one filter in addition to the number of filter fields that are set by the ReqFilterFields Property.

Filtering is not available on the **Request** page for reports that are displayed in SharePoint. If you want to filter a report, then add C/AL code to the report's **Request** page in the Microsoft Dynamics NAV development environment.

- Printing

Client-side printing is not supported. You must set up reports to print from Microsoft Dynamics NAV Server. For more information, refer to How to Specify a Printer Selection for Reports and the STARTSESSION Function.

Unlike pages, reports are not supported in Microsoft Dynamics NAV web parts.

Demonstration: Displaying Reports in SharePoint Sites By Using URLs

When you use the Microsoft Dynamics NAV Portal Framework for Microsoft SharePoint 2010, instead of opening a SharePoint site and browsing to a report, you can open a report directly by using its URL. When you have the URL of a report, you can use it as a hyperlink to the report. Then, you can include it in other sources, such as email messages or Word documents, or send it to other people.

The following link displays report 5 **Receivables – Payables** for the CRONUS International Ltd. company. The report is displayed in a Microsoft Dynamics NAV web client that is running on port 8080 of a computer that has the name MySharePointServer.

Demonstration Steps

1. Run the report.
 - a. Open your browser and type the following address:
[http://localhost:8080>List.aspx?company=CRONUS%20International%20Ltd.&report=5](http://localhost:8080/List.aspx?company=CRONUS%20International%20Ltd.&report=5)

Report URL Syntax

The URL to open a report in the Microsoft Dynamics NAV web client has the following syntax:

Code Example

```
<http|https>://<sharepointserver>[:<port>]/<site>/report.aspx?report=<ID>&[company=<companynamen>]
```

The following table describes the notation used to indicate address syntax.

Notation	Description
Text without brackets	Parameters that you must type as shown.
<>	A placeholder for values that you must supply. Do not include the brackets in the address.
[]	Optional parameters. Do not include the brackets in the address.
	A set of values from which to select. Use one of the options and do not include this option in the address.

Remember the following:

- You can put parameters in any order after report.aspx? For example, report.aspx?company=CRONUS%20International%20Ltd.&report=5 yields the same as report.aspx?report=5&company=CRONUS%20International%20Ltd.
- Separate parameters after aspx? with &.
- Use %20 for any spaces in values and names.

 More information about the report URL parameters is available here:
<http://go.microsoft.com/fwlink/?LinkId=267287>

Hyperlinks in a Report

In Microsoft Dynamics NAV 2013, you can create reports that provide dynamic data. Microsoft Dynamics NAV 2013 reports can include images, graphics, and interactive sorting. Additionally, you can toggle on data columns and link to other data in the system. This lesson shows how to set up a report to include a link to the **Customer Card** page. So, when the **Customer Card** page is run in preview mode, it can search any existing customer directly from the report.

By including this functionality, the report no longer serves as a static, printed list. It becomes a dynamic list that can be used for direct drill-down into system data.

Demonstration: Adding a Link in from a Report to a Page

When Michael prints the **Customer Top 10 List** report, he notices that only the customer number and name are included in the list. To find the other customer information, Michael must open the Customer Card. So, Michael asks Mort to include a link to the Customer Card in the report. However, before Mort can include the link in Visual Studio he must add and include a variable in a hidden text box on a section.

Demonstration Steps

1. Add a Variable and Code to the Report.
 - 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 **R123456711.fob**.
 - f. Click **OK** to start the import.

Module 5: Running Reports

- 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.
- j. In Object Designer, click **Report**, and then locate the **Customer - Top 10 List** report by typing object ID 123456711.
- k. Select the **Customer Top 10 List** report, and then click the **Design** button.
- l. In the **C/AL Globals** of this report there's a variable CustomerRecRef of **DataType** RecordRef and in the OnPreDataItem() trigger, the following line of code was added:

Code Example

```
CustomerRecRef.OPEN(18);
```

- m. In the OnAfterGetRecord() trigger, the following line of code was added at the bottom of the trigger:

Code Example

```
CustomerRecRef.SETPOSITION(Customer.GETPOSITION);
```

This C/AL code will make sure the recordref variable points to the record of the customer being fetched in the report.

2. Add the text box and add the line to the report.
To add the line to the report, follow these steps.

Mort must create a new line in the Report Dataset Designer to add the bookmark of the customer record to the dataset. This line will be used to locate the value of the current customer represented in the dataset in Visual Studio after the layout is opened there.

- a. In the **Report Dataset Designer** window, add a new line at the end of the Integer dataitem.
- b. In the **Data Type** column select **Column**.
- c. In the **Data Source** column enter the following:

Code Example

```
FORMAT(CustomerRecRef.RECORDID,0,10)
```

- d. In the Name column type, CustomerBookmark.

The value 10 in this expression only formats RECORDID into a text representation that is compatible with the URL handler of reports and pages.

- e. Save and compile the report.

After the bookmark is added to the Report Dataset Designer, Mort can open the layout of the report and continue to modify it in Visual Studio. The following steps will add a link to the customer number on the report that opens the customer card for the selected customer.

Now the newly created bookmark is part of the dataset in Visual Studio under Data Sources and its value is being referenced in the link that you are adding.

- f. In the development environment, with the **Customer - Top 10 List** report open in Report Designer, click **View**, and then select Layout.
- g. In Visual Studio, in the **Body** of the report, locate the text box **Customer_No**. The value of the text box is **[Customer_No]**.
- h. Right-click the text box and select **Properties**.
- i. In the **Properties** window, locate the **Action** tab.
- j. Select the **Go to URL** option and enter the following expression in the field:

Code Example

```
= "dynamicsnav:///runpage?page=21&mode=edit&bookmark=" + Fields!CustomerBookmark.Value
```

Because the mode parameter is used, the user can edit the customer page.

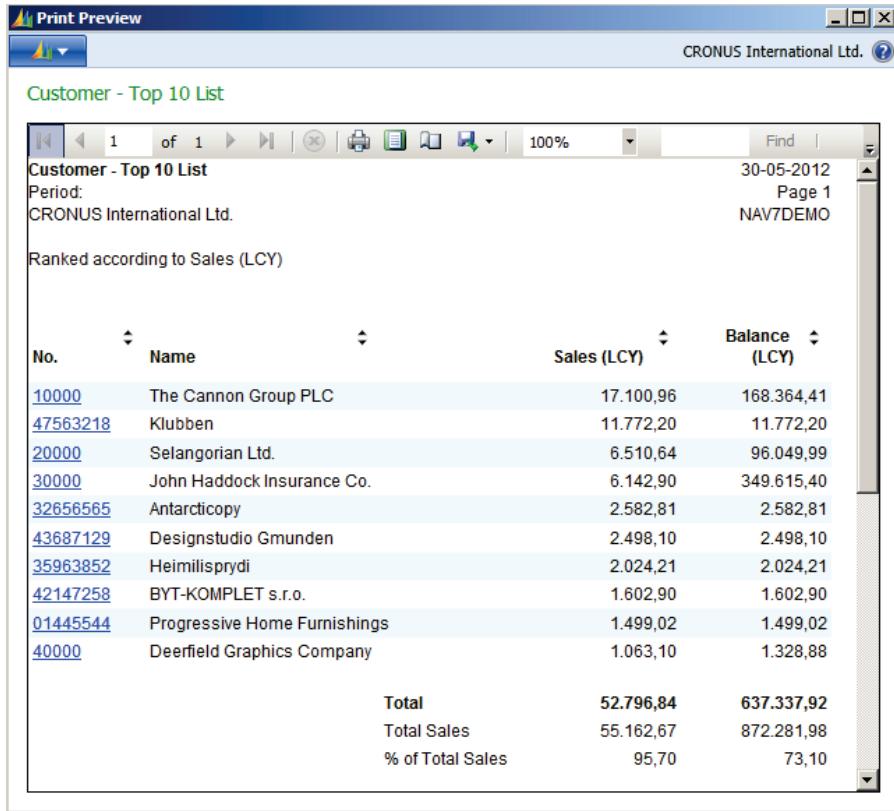
- k. In the **Font** tab, set the **Effects** to **Underline** and set the **Color** to **Blue**.
- l. Click **OK** to close the window.
- m. Save the report in Visual Studio and return to the development environment.
- n. Save and compile the report.

In the development environment, Mort saves the **Customer - Top 10 List** report. A message informs him that the .rdlc file for this report has changed and asks if he wants to load the changes. He clicks **Yes** to save the changes in the database.

To compile and save the report, Mort must set the **EnableHyperLinks** property.

 **Note:** Microsoft Dynamics NAV 2013 cannot verify URLs and protect against malicious sites that can be harmful to your computer. You should set the **EnableHyperlinks** property to **Yes** only if you can make sure that hyperlinks on the report target trusted sites.

- o. In the Report Dataset Designer, go to the last row and then click the down arrow.
- p. In the **Properties** window, enter **Yes** in the **EnableHyperLinks** property.
- q. Select Tools, and then click **Compile** to compile the report.
- r. Click **Run**.
- s. Click **Preview** to view the report. The report lists the top 10 list of customers.



The screenshot shows the Microsoft Dynamics NAV Print Preview window titled "Customer - Top 10 List". The report header includes the title, period (30-05-2012), page number (Page 1), and system name (NAV7DEMO). It also specifies that the results are "Ranked according to Sales (LCY)". The main content is a table with columns: No., Name, Sales (LCY), and Balance (LCY). The table lists ten customers with their respective sales and balances. At the bottom, there are summary rows for Total, Total Sales, and % of Total Sales.

No.	Name	Sales (LCY)	Balance (LCY)
10000	The Cannon Group PLC	17.100,96	168.364,41
47563218	Klubben	11.772,20	11.772,20
20000	Selangorian Ltd.	6.510,64	96.049,99
30000	John Haddock Insurance Co.	6.142,90	349.615,40
32656565	Antarcticopy	2.582,81	2.582,81
43687129	Designstudio Gmunden	2.498,10	2.498,10
35963852	Heimilisprydi	2.024,21	2.024,21
42147258	BYT-KOMPLET s.r.o.	1.602,90	1.602,90
01445544	Progressive Home Furnishings	1.499,02	1.499,02
40000	Deerfield Graphics Company	1.063,10	1.328,88
Total		52.796,84	637.337,92
Total Sales		55.162,67	872.281,98
% of Total Sales		95,70	73,10

FIGURE 5.12: CUSTOMER TOP TEN REPORT WINDOW

If you are resting on the customer number, you will receive a link to open the specific customer on a separate customer card, and then you can drill down into the customer's data.



Demonstration: Creating a Link from a Report to a Report

This demonstration will show how to set up a report to include a link from the **Customer - Top 10 List** report to the **Customer - Detail Trial Bal.** report. When you run the **Customer - Top 10 List** report in preview mode and show balance data, you can select a value in the Balance (LCY) column or a section of the pie chart to view detailed balance information. By including this functionality, you have a dynamic report, and now you can drill down into data.

Before you can add a link from a report to another report, you must enable hyperlinks on the report.

Demonstration Steps

1. Enable hyperlinks on a report.
 - a. Open the Microsoft Dynamics NAV 2013 demonstration database.
 - b. Select Tools, Object Designer to open the Object Designer.
 - c. Select **File, Import**.
 - d. Browse to the report **R123456711.fob**.
 - e. Click **OK** to start the import.
 - f. In the dialog box, choose to open the Import Worksheet window.
 - g. In the Action column, verify that the action is set to **Create**.
 - h. Click **OK** to import the object in the database.
 - i. In **Object Designer**, elect report 123456711, the **Customer - Top 10 List** report, and then select **Design**. The Report Dataset Designer window opens.
 - j. In **Report Dataset Designer**, select an empty row.
 - k. On the **View** menu, select Properties.
 - l. In the **Report - Properties** window, in the **EnableHyperlinks** field, set the **Value** field to **Yes** if it's not already set to **Yes**.
 - m. On the **File** menu, select Save, in the **Save** dialog box, make sure Compile is selected, and then select **OK** to compile and save the report.
 - n. Close the Properties and the Report Dataset Designer windows.

Module 5: Running Reports

2. Add a link from the Balance text box on the report to another report. Then, add a link from the Balance field.

In the following procedure, you add a link from the balance text box in report 111, the **Customer - Top 10 List** report to report 104, the **Customer Detail Trial Bal.** report.

To add a link from the **Balance** field:

- a. On the **View** menu, select Layout. Visual Studio opens.
- b. In Visual Studio, in the Body section of the report, locate the **Customer_Balance_LCY_** text box. The value of the text box is **=Fields!Customer_Balance_LCY_.Value**. You can confirm this value by selecting the text box and viewing the Value property in the Visual Studio **Properties** window.
- c. Select the text box, and then select **Placeholder Properties**.
- d. In the **Placeholder Properties** window, click the **Action** tab.
- e. Select the **Go to URL** check box and enter the following expression in the Select URL text box underneath the **Go to URL** check box:

```
= "dynamicsnav:///runreport?report=104&filter=Customer.%22No.%22:" + Fields!Customer_No_.Value
```

- f. Click **OK** to close the window.
 - g. On the **File** menu, select Save to save the report in Visual Studio.
3. Format the Balance text box to indicate that it is a link.

After you add the link from the balance text box to report 104 (Customer - Detail Trial Bal.), you might want to change the format of the balance text box to indicate that the data in the text box is a link.

- a. In Visual Studio, in the Body section of the report, select the **Customer_Balance_LCY_** text box.
- b. On the **View** menu, select the **Properties** window.
- c. In the **Properties** window, in the **Color** field, select **Blue**.
- d. Expand Font and set Text Decoration, to Underline.
- e. Save the report in Visual Studio.

4. Add a link from the pie chart to report 123456711.
 - a. In Visual Studio, select the pie chart, and then select Series Properties.
 - b. In the **Series Properties** window, select the **Action** tab, then select Go to URL and enter the following expression in the **Select URL** text box underneath the **Go to URL** check box:

```
=“dynamicsnav:///runreport?report=104&filter=Customer.%22No.%22:+Fields!Customer_No_.Value”
```

- c. Select the **OK** button to close the **Series Properties** window.
- d. In Visual Studio, on the **File** menu, select Save Report.rldc to save and then close the report in Visual Studio.

5. Run the modified report.

In the development environment, you must now save the modified report and compile it before it you can run it.

- a. In the development environment, save the **Customer - Top 10 List** report.
- b. A message informs you that the .rdlc file for this report has changed and asks whether you want to load the changes. Click **Yes** to save the changes in the database.
- c. In the **Save** window, verify that the **Compiled** check box is selected, and then click **OK**.
- d. In the development environment, select the **Customer Top 10 List** report and click **Run**. The **Edit-Customer- Top 10 List** page for report 123456711 opens in the RoleTailored client.
- e. Under **Options**, in the **Show** drop-down list, click **Balance (LCY)** and in the **Chart Type** drop-down list, select **Pie chart**.
- f. Click **Preview** to view the report. The report lists the top 10 list of customers. The Balance (LCY) column is formatted as a hyperlink. If you click the pie chart it is also formatted as a hyperlink.
- g. Select a value in a Balance (LCY) column. The **Customer - Detail Trial Bal.** report for the specified customer opens.
- h. Click a sector in the pie chart that represents a customer. The **Customer - Detail Trial Bal.** report for the specified customer opens.

Run a Stand-Alone Report in Visual Studio

In Microsoft Dynamics NAV 2013, you can share both the layout and the dataset to debug and, or change the report without having the whole Microsoft Dynamics NAV database. This new functionality is explained in this section.

This can be useful when the report does not include a particular dataset or it has rendering issues. Additionally, it can be useful when a developer does not have access to the Microsoft Dynamics NAV installation.

Run an Offline Report

Viktor is a Microsoft Certified Partner working for CRONUS International Ltd. He is working on some improvements to the **Customer - Top 10 List** report. However, during his development phase, he has encountered some problems. Viktor wants to send the report to Microsoft Support to help him troubleshoot the problems. The Microsoft Support engineer wants to be able to view the report and report data offline instead of on a computer that has Microsoft Dynamics NAV installed. Viktor saves the report dataset and the report layout and sends them to the Microsoft Support engineer, who runs the report on his computer by using Visual Studio.

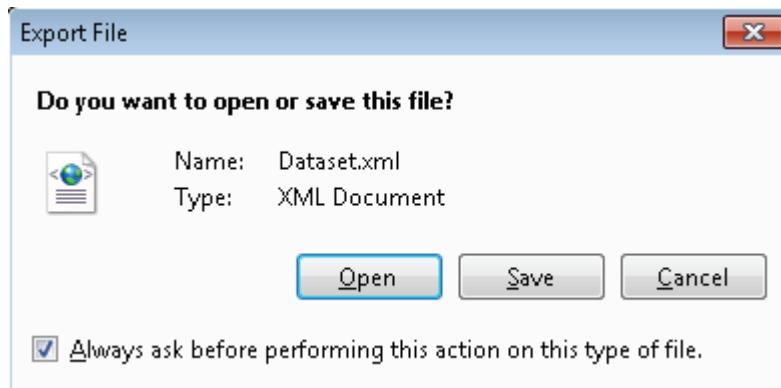
Saving the report dataset and layout

First, Viktor exports the report dataset to an .xml file.

To export the report dataset to an .xml file, follow these steps.

1. On the computer that has Microsoft Dynamics NAV installed, open the RoleTailored client.
2. In the RoleTailored client, on the **Report** tab, select **Customer - Top 10 List**.
3. On the request page for the **Customer - Top 10 List** report, on the **Application** menu, click Help, and then select About This Page to start the About This Report feature.
4. Close the **About This Page** page.
5. On the **Request** page for the **Customer - Top 10 List** report, click Preview to open the **Customer - Top 10 List** report in Report Viewer.

6. On the report preview, on the **Application** menu, click Help, and then select About This Report to display the report dataset.
7. On the **About This Report** page, select Export as XML.
8. In the Export File dialog box, click Save.



9. In the **Export File** dialog box, browse to your desktop, enter Dataset for the file name, set the Save type field to XML (*.xml), and then click Save.

Next, Viktor must save the report layout as an .rdlc file.

To copy the report layout .rdlc file, follow these steps.

1. In the development environment, click **Tools**, and then click **Object Designer**.
2. In Object Designer, click Report, select report 111, **Customer - Top 10 List**, and then click Design.
3. On the **View** menu, click Layout to open the report layout in Visual Studio.
4. In Visual Studio, in Solution Explorer, select Report.rdlc, and then in the Properties window, note the path where the Report.rdlc file is located.



5. Copy the Report.rdlc file from the location in the **Properties** window to your desktop.

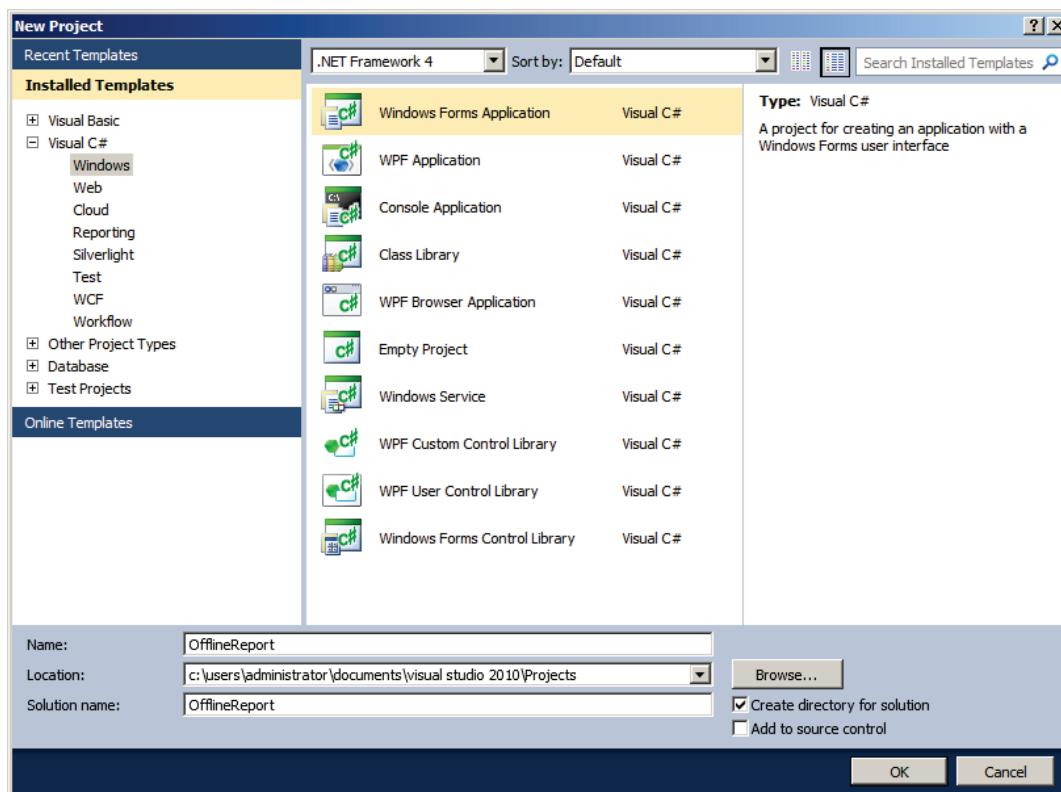
Viktor sends both the Dataset.xml file and the Report.rdlc file to the Microsoft Support engineer who is helping him troubleshoot the report.

Running the Report Offline

The Microsoft Support engineer has received the Dataset.xml file and the Report.rdlc file from Viktor and now he or she runs the report on his or her computer by using Visual Studio.

To run the report by using Visual Studio, follow these steps.

1. On the second computer, open Visual Studio.
2. In Visual Studio, in the **File** menu, click **New**, and then click **Project**.
3. In the **New Project** window, under Installed Templates, expand Visual C#, select Windows, and then select Windows Forms Application.
4. In the Name field, enter OfflineReport, and then click the OK button to create the solution.



5. In Solution Explorer, right-click the **OfflineReport** project, select Add, and then select Existing Item.

6. In the **Add Existing Item** dialog box, browse to your desktop, set the file type to All Files (*.*), select the Dataset.xml file and the Report.rdlc file, and then click Add.
 7. In Solution Explorer, under OfflineReport, select Dataset.xml. In the **Properties** window, in the **Copy to Output Directory** field, select Copy Always from the drop-down list.
 8. In the **Toolbox** dialog box, under Reporting, drag a new Report Viewer control to Form1.
 9. Select the report viewer control, and then in the Properties window, under Misc, expand Local Report, and then set the ReportEmbeddedResource property, to OfflineReport.Report.rdlc.
-



Note: If your report includes external images, then in the **Properties** window of the report viewer control, you must set the EnableExternalImages property to True.



Note: If your report includes hyperlinks, then in the **Properties** window of the report viewer control, you must set the EnableHyperlinks property to True.

10. In Solution Explorer, select the OfflineReport project, click Add, and then click Class.
11. In the **Add New Item** window, under Installed Templates, select Visual C# Items, and then click Class.
12. In the **Name** field, enter CreateDataSource.cs, and then select Add.
13. In the **CreateDataSource.cs** tab, replace the code with the following:

Code Example

```
using System;
using System.Collections.Generic;
using System.Data;
using System.Linq;
using System.Text;
namespace OfflineReport
{
    public static class CreateDataSource
    {
        public static DataTable CreateTable()
        {
            return null;
        }
    }
}
```

```
{  
    DataSet dataSet = new DataSet();  
  
    dataSet.ReadXml("Dataset.xml");  
  
    return dataSet.Tables[0];  
}  
}  
}
```

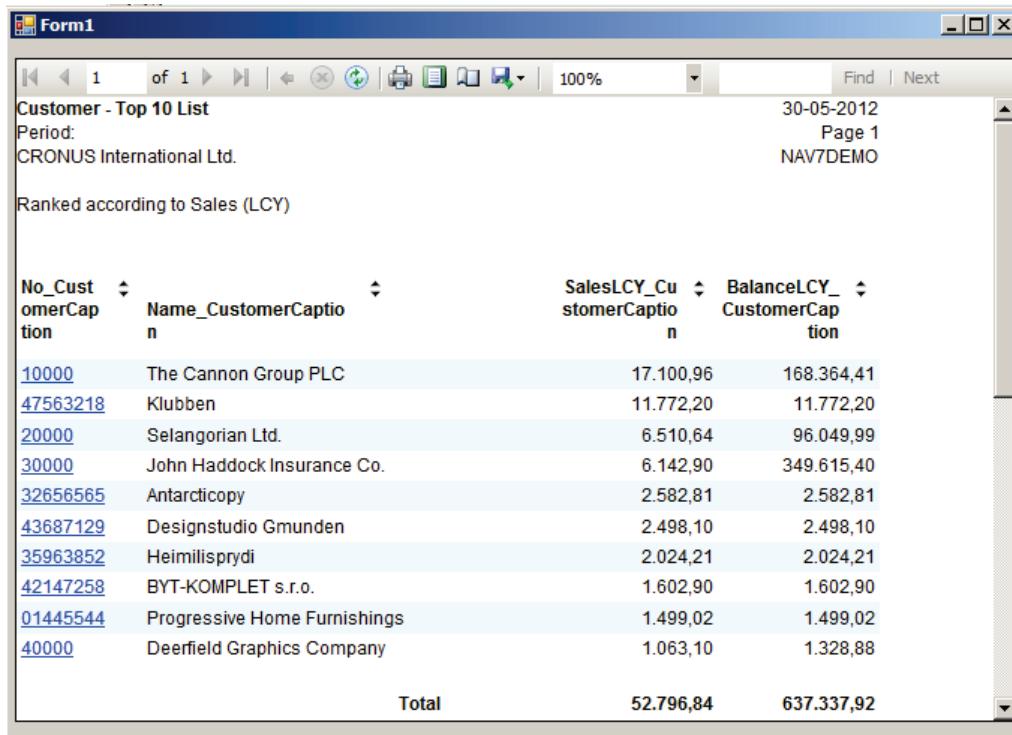
14. On the **File** menu, click Save CreateDataSource.cs.
15. Select **the Form1.cs [Design]** tab, and then on the **View** menu, click **Code**.
16. In the **Form1.cs** tab, replace the code with the following:

Code Example

```
using System;  
  
using System.Collections.Generic;  
  
using System.ComponentModel;  
  
using System.Data;  
  
using System.Drawing;  
  
using System.Linq;  
  
using System.Text;  
  
using System.Windows.Forms;  
  
using Microsoft.Reporting.WinForms;  
  
namespace OfflineReport  
{  
    public partial class Form1 : Form  
    {  
        public Form1()  
    }  
}
```

```
{  
    InitializeComponent();  
  
}  
  
private void Form1_Load(object sender, EventArgs e)  
  
{  
  
    this.reportViewer1.LocalReport.DataSources.Add(new  
ReportDataSource("DataSet_Result", CreateDataSource.CreateTable()));  
  
    this.reportViewer1.RefreshReport();  
  
}  
  
}  
}
```

17. On the **Build** menu, click Build Solution. The solution builds successfully.
18. On the **Debug** menu, click Start Debugging to view the report in the report viewer control.



The screenshot shows a Windows application window titled "Form1". Inside, a report viewer displays a table titled "Customer - Top 10 List". The report includes header information: "Period: 30-05-2012", "Page 1", and "CRONUS International Ltd. NAV7DEMO". Below the header, it says "Ranked according to Sales (LCY)". The table has four columns: "No_Cust", "Name_CustomerCaption", "SalesLCY_Cu", and "BalanceLCY_Cu". The data is as follows:

No_Cust	Name_CustomerCaption	SalesLCY_Cu	BalanceLCY_Cu
10000	The Cannon Group PLC	17.100,96	168.364,41
47563218	Klubben	11.772,20	11.772,20
20000	Selangorian Ltd.	6.510,64	96.049,99
30000	John Haddock Insurance Co.	6.142,90	349.615,40
32656565	Antarcticopy	2.582,81	2.582,81
43687129	Designstudio Gmunden	2.498,10	2.498,10
35963852	Heimilisprydi	2.024,21	2.024,21
42147258	BYT-KOMPLET s.r.o.	1.602,90	1.602,90
01445544	Progressive Home Furnishings	1.499,02	1.499,02
40000	Deerfield Graphics Company	1.063,10	1.328,88
Total		52.796,84	637.337,92

19. To make a change to the report layout, in Solution Explorer, right-click **Report.rdlc**, and then click Open.



Note: Now that you have created the OfflineReport solution, to run other reports from Visual Studio, you have to replace the Dataset.xml and Report.rdlc files.

Interesting Links

This section provides a list of hyperlinks and resources to use to obtain information about Microsoft Dynamics NAV 2013 and reporting.

The Microsoft Dynamics NAV Team Blog

This blog site of the Microsoft Dynamics NAV Product team contains technical articles and resources about Microsoft Dynamics NAV.

<http://go.microsoft.com/fwlink/?LinkId=267288>

Microsoft Dynamics Developer Center

The Microsoft Developer Network (MSDN) Developer Centers and Resource Centers collect content and resources related to specific products and technologies. They connect you to code samples, community sites, technical articles and documentation, upcoming events, and much more.

The Microsoft Dynamics Developer Center provides access to all kinds of development resources for all five Microsoft Dynamics products: (Microsoft Dynamics® AX, Microsoft Dynamics® CRM, Microsoft Dynamics® GP, Microsoft Dynamics® NAV, and Microsoft Dynamics® SL). It offers general information such as product highlights, getting started and learning information. Additionally, it contains links to newsgroups, communities and (team) blogs.

<http://go.microsoft.com/fwlink/?LinkId=267289>

Microsoft SQL Server Development Center

The Microsoft SQL Server Development Center contains all kinds of resources and information about development in Microsoft SQL Server®.

<http://go.microsoft.com/fwlink/?LinkId=267290>

In the Microsoft SQL Server Reporting Services (SSRS) section of the SQL Server Developer Center, you can find all kinds of resources related to reporting in SSRS. To go directly to the SSRS section, click the following link:

<http://go.microsoft.com/fwlink/?LinkId=267291>

Reporting in Microsoft Dynamics® NAV 2013

Microsoft Visual Studio

The following is the home page of the Microsoft Visual Studio product family.

<http://go.microsoft.com/fwlink/?LinkId=267292>

Microsoft Visual Studio Express Editions

This site contains resources for the Microsoft Visual Studio Express Editions. The page includes software downloads, samples and resources, information about Frequently Asked Questions (FAQ), system requirements, help resources, installation topics, and forums.

<http://go.microsoft.com/fwlink/?LinkId=267293>

Creating and Consuming a Codeunit Web Service

This walkthrough provides an overview of how to create and consume a simple web service that uses Microsoft Dynamics NAV.

<http://go.microsoft.com/fwlink/?LinkId=267294>

Microsoft Dynamics NAV 2013 Launch Portal

The Launch Portal references tools, training, and content that you can use to kick start Microsoft Dynamics NAV 2013 readiness. This material supplements the communications and processes that are provided by your local Microsoft country or regional organization. A PartnerSource sign in is required.

<http://go.microsoft.com/fwlink/?LinkId=267296>

Microsoft Dynamics NAV 2013 Developer and IT Pro Help

The Microsoft Dynamics NAV 2013 Developer and IT Pro Help provides information about developing for, installing, and managing Microsoft Dynamics NAV 2013.

<http://go.microsoft.com/fwlink/?linkid=126283>

Updates to the developer and IT Pro Help, with both new content and improvements to existing content can be downloaded from the following page:

<http://go.microsoft.com/fwlink/?linkid=126282>

User Experience Guidelines for Microsoft Dynamics NAV 2013

The Microsoft Dynamics NAV 2013 User Experience Guidelines (or “UX Guide” for short) is a tool to help Microsoft Dynamics NAV Independent Software Vendors (ISVs) in building more user-friendly applications. By using the Microsoft UX Guide proactively during your development process, you can save time and avoid design

violations.

<http://go.microsoft.com/fwlink/?LinkId=267297>

Microsoft Dynamics NAV Developer Documentation

This page contains links to several technical articles related to development in Microsoft Dynamics NAV.

<http://go.microsoft.com/fwlink/?LinkId=267300>

The Microsoft Dynamics NAV Technical Community

This site offers a several tools to help you connect and collaborate with other Microsoft Dynamics NAV users, developers and implementers.

<http://go.microsoft.com/fwlink/?LinkId=267298>

SQL Server 2008 Report Definition Language Specification

This document specifies the structure and semantics of the SQL Server 2008 Report Definition Language (RDL), and XML Schema for presenting reports.

<http://go.microsoft.com/fwlink/?LinkId=267299>

Lab 5.1: Add a Report to the RoleTailored Client

Scenario

Prakash wants his **Resource - Utilization (Chart) Report** (report 123456706) to be available from the Role Center menu so that he and other project managers can access it from the RoleTailored client. He asks Mort to make it available in the menu on the Role Center page. Additionally, he wants it to be available from the **Resources** page in the Departments section (under the Reports and Analysis category).

Objectives

In this lab, you will add one of the reports that you built in earlier modules, to the RoleTailored client interface.

Exercise 1: Add a Report to the RoleTailored Client

Exercise Scenario

Add report 123456706 that is created in Lab 4.2 (Creating a TOP X Report), to the Role Center menu for Prakash and the other project managers. Additionally, create a new MenuSuite object to make the report available under the Reports and Analysis category in the Resources section of the **Departments** page.

Task 1: Add the Report to the Job Project Manager RC Page.

High Level Steps

1. Add report 123456706 that is created in Lab 4.2, to the **Role Center** menu for Prakash and the other project managers.

Detailed Steps

1. Add report 123456706 that is created in Lab 4.2, to the **Role Center** menu for Prakash and the other project managers.
 - 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, in case it's not already open.
 - d. Select File, Import.
 - e. Browse to the object **P123456700.fob**.
 - f. Click **Yes** 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**.

Module 5: Running Reports

- i. Click **OK** to import the object in the database.
- j. In the Microsoft Dynamics NAV development environment, open Object Designer.
- k. Click the **Page** button.
- l. Select page **123456700**, Job Project Manager RC2.
- m. Click **Design** to open the page in Page Designer.
- n. Select an empty line in the designer. In the menu, Select **View**, and then **Page Actions** to open Action Designer.
- o. Scroll down to the Action that has **Jobs per Item** as its Caption.
- p. Click the **Separator** button to insert a separator.
- q. Scroll down to the **ActionContainer** that has **HomeItems** as its **Subtype**.
- r. Press **F3** to insert a new empty line (or select Edit, New).
- s. On the new line, in the **Caption** field, enter a name for the action: **Resource - Utilization (Chart)**.
- t. In the **Type** field, select Action.
- u. Click **View**, and then **Properties** to open the **Properties** window for the new action.
- v. In the **Value** field of the **RunObject** property, click the drop-down arrow and select **report 123456706** in the **Object List** window.
- w. Compile and save the page.

Task 2: Create a New MenuSuite Object

High Level Steps

1. Create the Menusuite object.

Detailed Steps

1. Create the Menusuite object.
 - a. In the Microsoft Dynamics NAV development environment, open Object Designer.
 - b. Click the **MenuSuite** button.
 - c. Click the **New** button to open the **Design Level** window.
 - d. In the list box, select **Dept - Partner** level.

Task 3: Change the MenuSuite Object to Add the Report to the Departments Section

High Level Steps

1. Add the Report to the Departments Section.

Detailed Steps

1. Add the Report to the Departments Section.
 - a. In the Navigation Pane Designer, select the **Resource Planning** menu.
 - b. Expand the Reports group.
 - c. Right-click the last report in the Reports group (**Resource - Price List**) and select Create Item.
 - d. In the **Create Item** window, select Report in the **Object Type** field.
 - e. In the **Object ID** field, type **123456706**.
 - f. In the **Caption** field, change the caption to Resource - Utilization (Chart).
 - g. In the **Department Category** field, select Reports and Analysis.
 - h. Click **OK** to close the **Create Item** window.
 - i. Compile and save the MenuSuite.

Task 4: Set Permission to Configure the Report

High Level Steps

1. To set permissions to the profile, follow these steps.

Detailed Steps

1. To set permissions to the profile, follow these steps.
 - a. Open the Dynamics NAV Client.
 - b. In the search box at the right top of the screen type: **Profiles**.
 - c. Select the profile of the **PROJECT MANAGER** and double click.
 - d. The profile card of the PROJECT MANAGER now opens.
 - e. Select the button to the right of the **Owner ID** field.
 - f. The **Users** window opens.
 - g. Select the **Administrator** user and click **OK**.
 - h. Click **OK** to close the profile card of the PROJECT MANAGER.

Task 5: Create a New Shortcut for the RoleTailored client

High Level Steps

1. Create a New Shortcut for the RoleTailored client.

Detailed Steps

1. Create a New Shortcut for the RoleTailored client.
 - a. Right-click the **Windows Desktop** and select **New > Shortcut**.
 - b. In the **Create Shortcut** window, click the **Browse** button.
 - c. In the Browse for Folder window, browse to the following path and application:
C:\Program Files(x86)\Microsoft Dynamics NAV\70\RoleTailored Client\Microsoft.Dynamics.Nav.Client.exe.
 - d. Click the **OK** button to close the **Browse for Folder** window.
 - e. In the **Create Shortcut** window, click the **Next** button.
 - f. Enter the following name for the shortcut: Project Manager Profile.
 - g. Click the **Finish** button.

Task 6: Change the New Shortcut for the RoleTailored Client

High Level Steps

1. Modify the New Shortcut for the RoleTailored client.

Detailed Steps

1. Modify the New Shortcut for the RoleTailored client.
 - a. Right-click the newly created shortcut.
 - b. Select Properties.
 - c. Change the Target field to:
"C:\Program Files(x86)\Microsoft Dynamics NAV\70\RoleTailored Client\Microsoft.Dynamics.Nav.Client.exe" -configure -profile:"Project Manager"
 - d. Click the **Apply** button.
 - e. Click the **OK** button.

Task 7: Start the RoleTailored Client that has the Project Manager Profile (in Configuration Mode)

High Level Steps

1. Use the shortcut to run the RoleTailored client that has the Project Manager profile.

Detailed Steps

1. Use the shortcut to run the RoleTailored client that has the Project Manager profile.
 - a. On the Windows Desktop, double-click the **Project Manager Profile** icon.
 - b. On the Role Center page, click the **Report** menu.

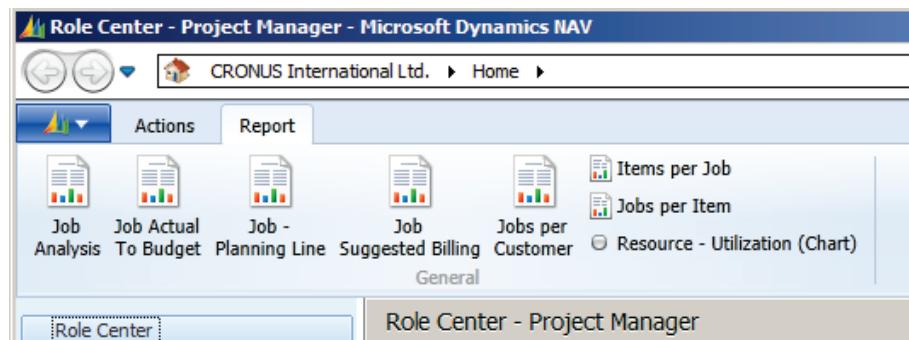


FIGURE 5.17: ROLE CENTER PAGE WINDOW

- c. Click the **Departments** menu.
- d. Click the **Resource Planning** section.

Now the **Resource - Utilization (Chart) Report** is listed under the Reports and Analysis category (as the bottom report in the Reports group).

Lab 5.2: Call a Report from a Web Service

Scenario

Susan regularly receives requests and questions from customers about sales invoices and shipments. After Susan records all the details, she sends the information to the representative who is responsible for handling the requests and questions.

At the same time, Rebecca receives several requests to resend lost or missing documents to the customers.

To better and faster serve the customers, Susan and Rebecca want to quickly search a document (by number), and print a copy of it so that they can add it to the case or send it to the customers.

Mort views this as an excellent opportunity to develop a self-serving customer portal.

Objectives

In this lab, you will build a codeunit and expose it as a web service. Although this course is not a web development course, it will show how to build a small web application that includes the web service.

Although this lab is educational, it does not focus on how to define the appropriate application security. So, it is highly recommended not to implement the lab in a live production environment.

Exercise 1: Run a Report from a Web Service

Task 1: Create a New Codeunit.

High Level Steps

1. Create a New Codeunit.

Detailed Steps

1. Create a New Codeunit.
 - a. In the Microsoft Dynamics NAV development environment, open Object Designer.
 - b. Click the **Codeunit** button.
 - c. Click the **New** button.
 - d. Select File, Save As.
 - e. In the **ID** field, enter 123456701.

- f. In the **Name** field, enter Get Sales Document WS.
- g. Check the **Compiled** field.
- h. Click **OK**.

Task 2: Add Global Variables to the Codeunit

High Level Steps

1. Add global variables to the codeunit.

Detailed Steps

1. Add global variables to the codeunit.
 - a. Select View, C/AL Globals to open the **C/AL Globals** window.
 - b. On the **Variables** tab, select the first (empty) line.
 - c. In the **Name** field, enter FileName.
 - d. In the **DataType** field, select Text.
 - e. In the **Length** field, enter 250.
 - f. Select the next line.
 - g. In the **Name** field, enter SalesInvHdr.
 - h. In the **DataType** field, select Record.
 - i. In the **Subtype** field, click the button and select the **Sales Invoice Header** table from the **Table List** window.
 - j. Select the next line.
 - k. In the **Name** field, enter SalesCMHdr.
 - l. In the **DataType** field, select Record.
 - m. In the **Subtype** field, click the button and select the **Sales Cr.Memo Header** table from the **Table List** window.
 - n. Select the next line.
 - o. In the **Name** field, enter SalesShptHdr.

- p. In the **DataType** field, select Record.
 - q. In the **Subtype** field, click the button and select the **Sales Shipment Header** table from the **Table List** window.

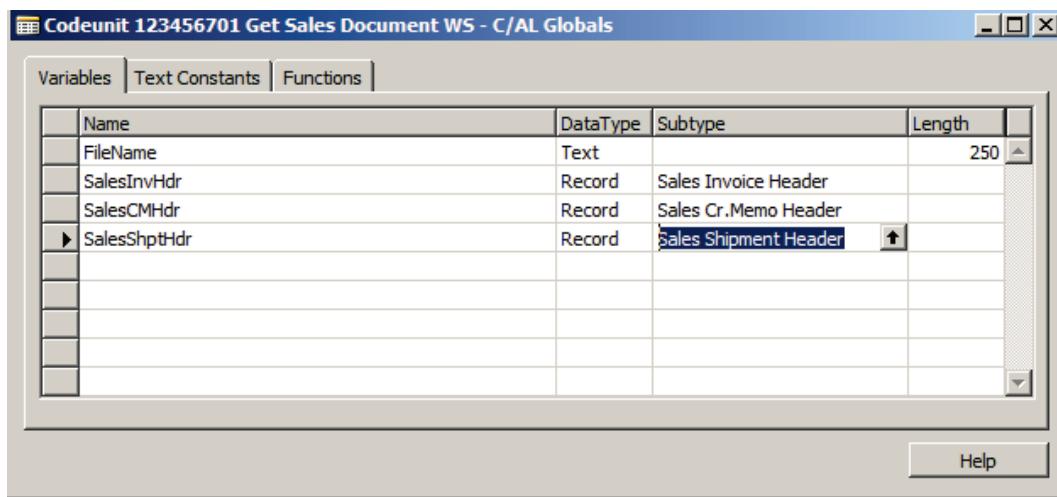


FIGURE 5.18: THE C/AL GLOBALS WINDOW

- r. Click File, Save to save and compile the codeunit.

Task 3: Add the CheckInvoiceNo function to the Codeunit

High Level Steps

1. Add the CheckInvoiceNo function to the codeunit.

Detailed Steps

1. Add the CheckInvoiceNo function to the codeunit.
 - a. In the **C/AL Globals** window, click the **Functions** tab.
 - b. On the first empty line, enter **CheckDocNo**.
 - c. Place the pointer on the line that contains the CheckDocNo function and then click the **Locals** button to open the **C/AL Locals** window.
 - d. On the **Parameters** tab, select the first empty line.
 - e. In the **Name** field, enter DocType.
 - f. In the **DataType** field, select Integer.
 - g. Select the next line.
 - h. In the **Name** field, enter DocNo.
 - i. In the **DataType** field, select Code.
 - j. In the **Length** field, enter 20.
 - k. Click the **Return Value** tab.
 - l. In the **Return Type** field, select Integer.
 - m. Close the **C/AL Locals** window.

Task 4: Add Code to the CheckDocNo Function

High Level Steps

1. Add code to the CheckDocNo function.

Detailed Steps

1. Add code to the CheckDocNo function.
 - a. In the C/AL Editor, enter the following code in the CheckDocNo() function trigger:

Code Example

```
CASE DocType OF  
    110: BEGIN  
        IF SalesShptHdr.GET(DocNo) THEN BEGIN  
            EXIT(1)  
        END ELSE  
        EXIT(0);  
    END;  
  
    112: BEGIN  
        IF SalesInvHdr.GET(DocNo) THEN BEGIN  
            EXIT(1)  
        END ELSE  
        EXIT(0);  
    END;  
  
    114: BEGIN  
        IF SalesCMHdr.GET(DocNo) THEN BEGIN
```

```

    EXIT(1)

    END ELSE

    EXIT(0);

    END;

END;

```

- b. Click File and then click Save to compile the codeunit. The function trigger will resemble this:

```

Codeunit 123456701 Get Sales Document WS - C/AL Editor
CheckDocNo(DocType : Integer;DocNo : Code[20]) : Integer
CASE DocType OF
  110: BEGIN
    IF SalesShptHdr.GET(DocNo) THEN BEGIN
      EXIT(1)
    END ELSE
      EXIT(0);
  END;
  112: BEGIN
    IF SalesInvHdr.GET(DocNo) THEN BEGIN
      EXIT(1)
    END ELSE
      EXIT(0);
  END;
  114: BEGIN
    IF SalesCHMhdr.GET(DocNo) THEN BEGIN
      EXIT(1)
    END ELSE
      EXIT(0);
  END;
END;

```

FIGURE 5.19: SALES INVOICE REPORT

Task 5: Add the GenerateReport Function to the Codeunit

High Level Steps

1. Add the GenerateReport function to the codeunit.

Detailed Steps

1. Add the GenerateReport function to the codeunit.
 - a. In the **C/AL Globals** window, click the **Functions** tab.
 - b. On the first empty line, enter GenerateReport.
 - c. Place the pointer on the line that contains the GenerateReport function and then click the **Locals** button to open the **C/AL Locals** window.
 - d. On the **Parameters** tab, select the first empty line.

- e. In the **Name** field, enter DocType.
- f. In the **DataType** field, select Integer.
- g. Select the next line.
- h. In the **Name** field, enter DocNo.
- i. In the **DataType** field, select Code.
- j. In the **Length** field, enter 20.
- k. Click the **Return Value** tab.
- l. In the **Return Type** field, select Text.
- m. In the **Length** field, enter 250.
- n. Close the **C/AL Locals** window.
- o. Close the **C/AL Globals** window to return to the C/AL Editor.

Task 6: Add Code to the GenerateReport Function

High Level Steps

1. Add code to the GenerateReport function.
2. Create the temp folder.

Detailed Steps

1. Add code to the GenerateReport function.
 - a. In the C/AL Editor, enter the following code in the GenerateReport() function trigger.

Code Example

```
// Determine the PDF File Name  
  
FileName := 'C:\temp\';  
  
FileName := FileName + FORMAT(CREATEGUID);  
  
FileName := DELCHR(FileName, '=', '{}');  
  
FileName := FileName + '.pdf';  
  
// Print the Document  
  
IF DocType = 110 THEN BEGIN  
  
    SalesShptHdr.SETRANGE("No.", DocNo);  
  
    IF SalesShptHdr.FINDFIRST THEN  
  
        REPORT.SAVEASPDF(208, FileName, SalesShptHdr);  
  
END;
```

```

IF DocType = 112 THEN BEGIN

    SalesInvHdr.SETRANGE("No.", DocNo);

    IF SalesInvHdr.FINDFIRST THEN

        REPORT.SAVEASPDF(206, FileName, SalesInvHdr);

    END;

    IF DocType = 114 THEN BEGIN

        SalesCMHdr.SETRANGE("No.", DocNo);

        IF SalesCMHdr.FINDFIRST THEN

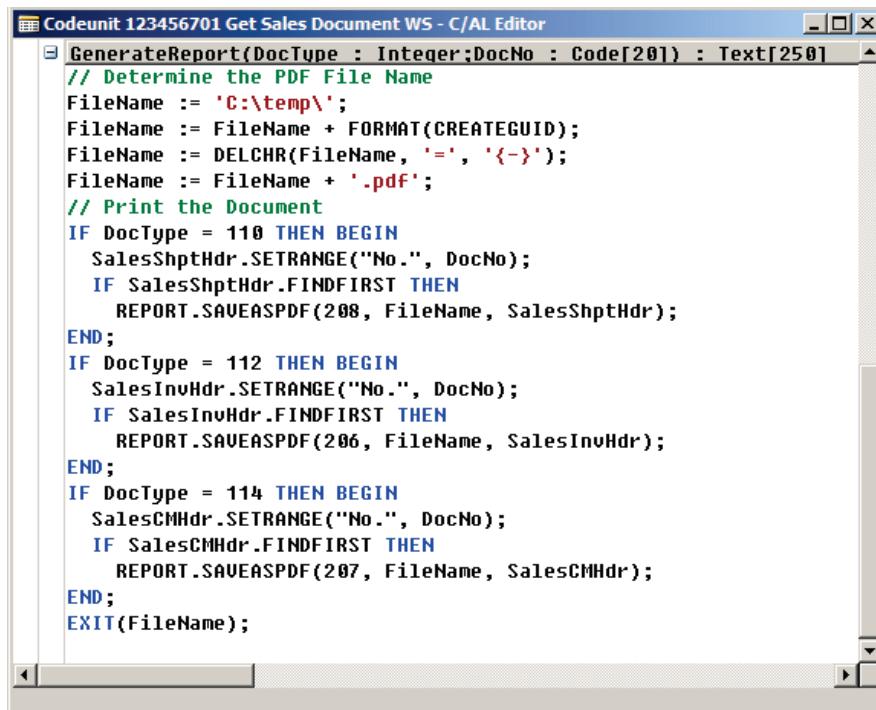
            REPORT.SAVEASPDF(207, FileName, SalesCMHdr);

        END;

    EXIT(FileName);

```

- b. Click File and then click Save to compile the codeunit. The function trigger will resemble this:



The screenshot shows the Microsoft Dynamics C/AL Editor window titled "Codeunit 123456701 Get Sales Document WS - C/AL Editor". The code editor displays the following C/AL code:

```

Codeunit 123456701 Get Sales Document WS - C/AL Editor
GenerateReport(DocType : Integer;DocNo : Code[20]) : Text[250]
// Determine the PDF File Name
FileName := 'C:\temp\';
FileName := FileName + FORMAT(CREATEGUID());
FileName := DELCHR(FileName, '=', '{-}');
FileName := FileName + '.pdf';
// Print the Document
IF DocType = 110 THEN BEGIN
    SalesShptHdr.SETRANGE("No.", DocNo);
    IF SalesShptHdr.FINDFIRST THEN
        REPORT.SAVEASPDF(208, FileName, SalesShptHdr);
END;
IF DocType = 112 THEN BEGIN
    SalesInvHdr.SETRANGE("No.", DocNo);
    IF SalesInvHdr.FINDFIRST THEN
        REPORT.SAVEASPDF(206, FileName, SalesInvHdr);
END;
IF DocType = 114 THEN BEGIN
    SalesCMHdr.SETRANGE("No.", DocNo);
    IF SalesCMHdr.FINDFIRST THEN
        REPORT.SAVEASPDF(207, FileName, SalesCMHdr);
END;
EXIT(FileName);

```

FIGURE 5.20: CODE EXAMPLE WINDOW

- c. Close the C/AL Editor.
d. Close Object Designer.

2. Create the temp folder.
 - a. On the C:\ drive create a new folder called temp.

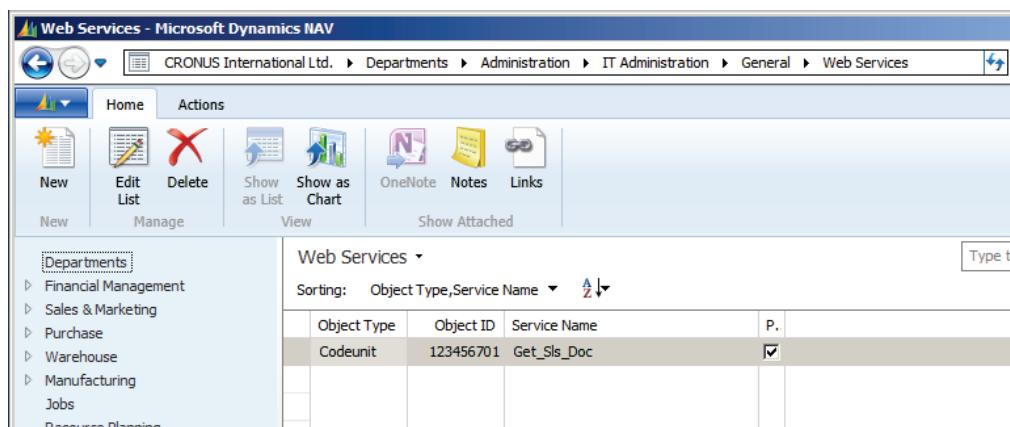
Task 7: Expose the Codeunit as a Web Service

High Level Steps

1. Expose the Codeunit as a web service.

Detailed Steps

1. Expose the Codeunit as a web service.
 - a. In the RoleTailored client, browse to: **Departments > Administration > IT Administration > General**.
 - b. Select **WebServices** in the **Lists** category.
 - c. In the **Web Services** window, click the **New** button.
 - d. In the **Object Type** field, select Codeunit.
 - e. In the **Object ID** field, enter 123456701.
 - f. In the **Service Name** field, enter Get_Sls_Doc.
 - g. Put a check mark in the **Published** field, and then click the **OK** button.
 - h. The codeunit is now exposed as a web service.



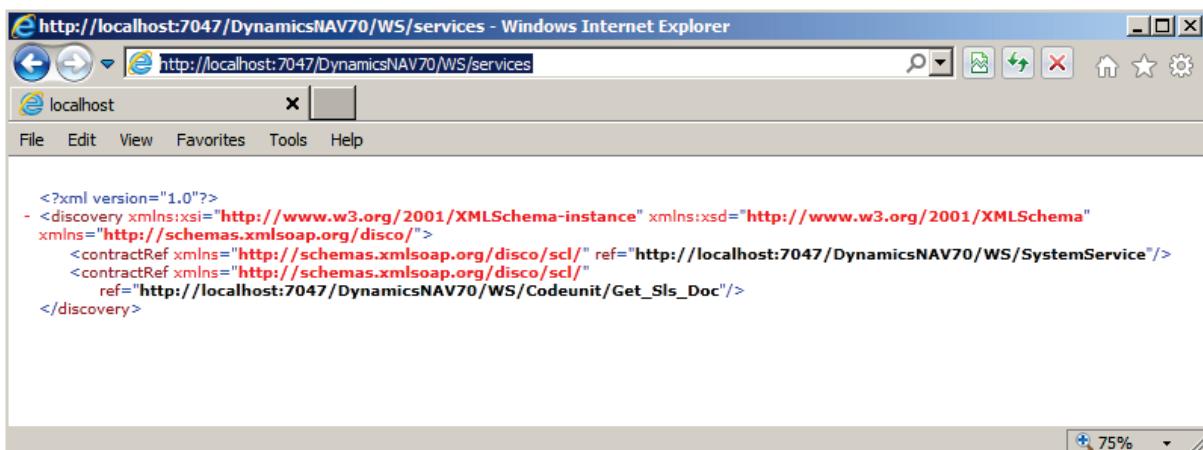
Task 8: Test the Availability of the New Web Service

High Level Steps

1. Test the availability of the new web service.

Detailed Steps

1. Test the availability of the new web service.
 - a. Open Internet Explorer.
 - b. In the browser bar, enter the following URL:
<http://localhost:7047/DynamicsNAV70/WS/services>



The newly published web service appears in the list that uses the service name that is specified in the **Web Services** window.

Task 9: Create a New Website

High Level Steps

1. Create a new website.

Detailed Steps

1. Create a new website.
 - a. Select **Start > All Programs > Microsoft Visual Studio 2010 > Microsoft Visual Studio 2010**.
 - b. Select **File, New, Web Site**.
 - c. In the **New Web Site** window, select the ASP.NET Web Site template.
 - d. In the Location field, enter the following path:
C:\Users\Administrator\Documents\Visual Studio 2010\WebSites\Lab52

Reporting in Microsoft Dynamics® NAV 2013

- e. In the **Language** field, select Visual Basic.
- f. Click the **OK** button to create the new website.
- g. Remove the code in the last asp:content tag.

The result should resemble this:

The screenshot shows the Microsoft Visual Studio interface with the title bar "Lab52 - Microsoft Visual Studio (Administrator)". The menu bar includes File, Edit, View, Website, Build, Debug, Team, Data, Format, Tools, Test, Window, Help. The toolbar has various icons for file operations. The status bar shows "Debug Any CPU". The main window displays the "Default.aspx*" page. The top tab bar says "Client Objects & Events" with "(No Events)". The code editor contains the following ASPX markup:

```
<%@ Page Title="Home Page" Language="VB" MasterPageFile="~/Site.Master" AutoEventWireup="false"
CodeFile="Default.aspx.vb" Inherits="_Default" %>

<asp:Content ID="HeaderContent" runat="server" ContentPlaceHolderID="HeadContent">
</asp:Content>
<asp:Content ID="BodyContent" runat="server" ContentPlaceHolderID="MainContent">
<p>
</p>
</asp:Content>
```

The **ASP Editor** window in the middle displays a tab for the default page of the website. This is called `default.aspx`. Notice the **Design**, **Split** and **Source** buttons at the bottom of the ASP Editor window.

- a. Click the **Design** button to switch to the design view.

Task 10: Add Controls to the Website

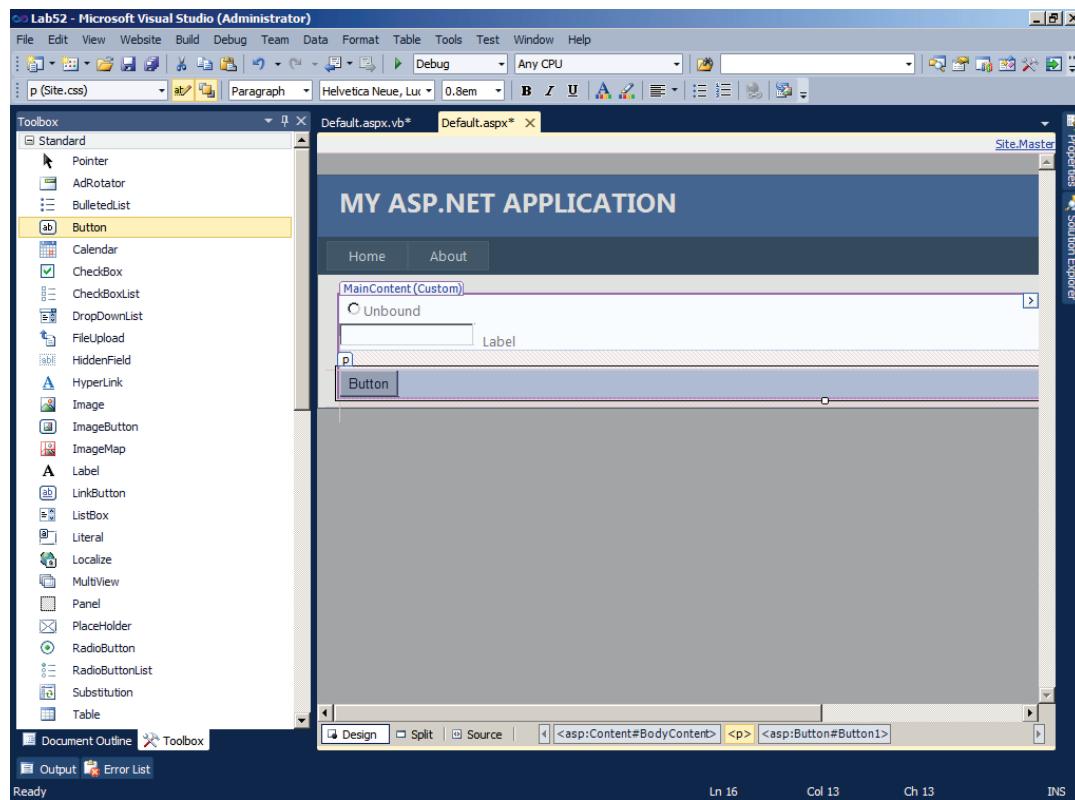
High Level Steps

1. Add controls to the website.

Detailed Steps

1. Add controls to the website.
 - a. In the **Toolbox dialog box**, select the RadioButtonList control.
 - b. Drag it inside the `<p>` element on the `default.aspx` page. The control remains selected on the design surface.
 - c. Press the right arrow to place the pointer next to the control.
 - d. Press **Enter**.
 - e. In the **Toolbox** window, select the text box control.
 - f. Drag it inside the `<p>` element, under the RadioButtonList.
 - g. With the Text box control selected, press the **spacebar** two times to insert two blank spaces after the control.
 - h. In the **Toolbox** window, select the Label control.
 - i. Drag it inside the `<p>` element, right next to the text box.

- j. With the Label control selected, press the **right arrow key** to place the cursor next to the control.
- k. Press **Enter**.
- l. In the **Toolbox** window, select the Button control.
- m. Drag it inside the <p> element, under the text box control. The result will resemble this.



Task 11: Design the Controls on the Website

High Level Steps

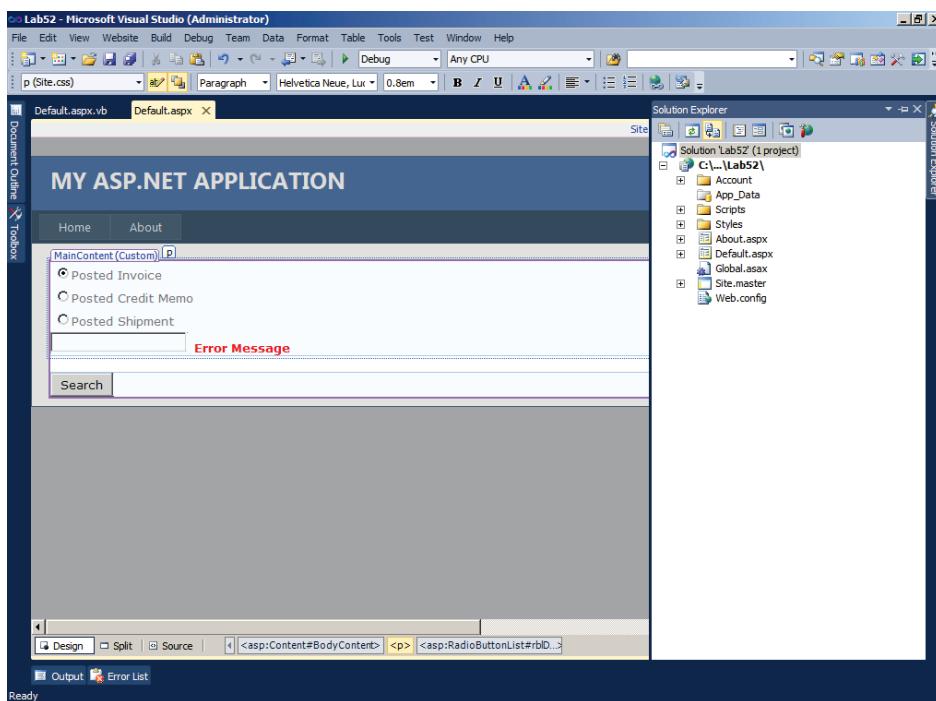
1. Design the controls on the website.

Detailed Steps

1. Design the controls on the website.
 - a. Select the text box control.
 - b. In the **Properties** window, set the (ID) property to **txtDocNo**.
 - c. Expand the Font property collection.
 - d. Set the Name property to Verdana.
 - e. Set the Size property to Small.
 - f. Select the Label control.
 - g. In the **Properties** window, set the (ID) property to **lblErrorMsg**.

- h. Set the ForeColor property to Red.
- i. Set the Text property to Error Message.
- j. Set the Visible property to False.
- k. Expand the Font property collection.
- l. Set the Bold property to True.
- m. Set the Name property to Verdana.
- n. Set the Size property to Small.
- o. Select the Button control.
- p. In the **Properties** window, set the (ID) property to cmdSearch.
- q. Set the Text property to Search.
- r. Expand the Font property collection.
- s. Set the Name property to Verdana.
- t. Set the Size property to Small.
- u. Select the RadioButtonList control.
- v. In the **Properties** window, set the (ID) property to rblDocType.
- w. Expand the Font property collection.
- x. Set the Name property to Verdana.
- y. Set the Size property to Small.
- z. Click the angle bracket (>) button on the upper-right corner of the RadioButtonList control.
- aa. In the **RadioButtonList Tasks** pane, select Edit Items.
- bb. In the **ListItem Collection Editor** window, click the **Add** button.
- cc. In the **ListItem properties** window, set the Text property to Posted Invoice.
- dd. Change the Value property to 112.
- ee. Set the Selected property to True.
- ff. Click the **Add** button. (The newly added item will appear in the **Members** list.)
- gg. In the **ListItem properties** window, set the Text property to Posted Credit Memo.
- hh. Change the Value property to 114.
- ii. Click the **Add** button. (The newly added item will appear in the **Members** list.)
- jj. In the **ListItem properties** window, set the Text property to Posted Shipment.
- kk. Change the Value property to 110.
- ll. Click the **OK** button to close the **ListItem Collection Editor** window.
- mm. Click File and then click Save to save the file.

The result will resemble this:



Task 12: Add the Web Service Reference

High Level Steps

1. Add the web service reference.

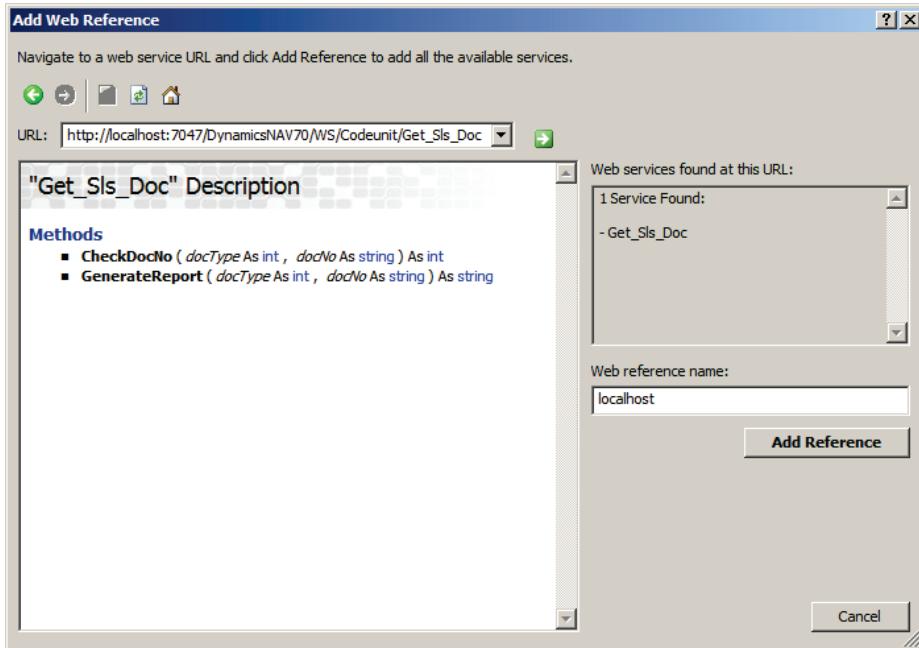
Detailed Steps

1. Add the web service reference.

Because including a web service is not part of the scope for this training, it is recommended to check the online help for Microsoft Dynamics NAV 2013 or the MSDN website at <http://go.microsoft.com/fwlink/?LinkId=267301>.

- a. In the **Solution Explorer** window, right-click the **Lab52 project** and select the **Add Web Reference** button to open the **Add Web Reference** window. In case the Add Web Reference button is not available then in the Solution Explorer window, right-click the Lab52 project and select the Add Service Reference ,click the Advanced button, in the Service Reference Setting window click the Add Web Reference button to open the Add Web Reference window.
- b. In the **URL** field, enter
`http://localhost:7047/DynamicsNAV70/WS/services.`

- c. Click the **Go** button.
- d. In the left list box, **Discovery Page**, click the **View Service** link for the Get_Sls_Doc service.



- e. Click the **Add Reference** button to add a reference to the web service.

Task 13: Add Code to Include the Web Service

High Level Steps

1. Add code to include the webservice.

Detailed Steps

1. Add code to include the webservice.
 - a. Click the **Default.aspx** tab.
 - b. Double-click the **Search** button in Design view.
 - c. On the **Default.aspx.vb** tab, enter the following line of code in the **cmdSearch_Click** procedure.

Code Example

```
Dim service As localhost.Get_Sls_Doc = New localhost.Get_Sls_Doc  
  
Dim pdfFileName As String  
  
Dim DocCheck As Integer  
  
service.UseDefaultCredentials = True
```

```
service.Url =
"http://localhost:7047/DynamicsNAV70/WS/CRONUS%20International%20Ltd./Co
deunit/Get_Sls_Doc"

DocCheck = service.CheckDocNo(rblDocType.SelectedValue, txtDocNo.Text)

If DocCheck = 0 Then

    lblErrorMsg.Text = rblDocType.SelectedItem.Text & " " & txtDocNo.Text & " was
not found."

    lblErrorMsg.Visible = True

    txtDocNo.Text = ""

Else

    lblErrorMsg.Visible = False

    service.UseDefaultCredentials = True

    service.Url =
"http://localhost:7047/DynamicsNAV70/WS/CRONUS%20International%20Ltd./Co
deunit/Get_Sls_Doc"

    Response.ContentType = "application/pdf"

    pdfFileName = service.GenerateReport(rblDocType.SelectedValue,
txtDocNo.Text)

    Response.TransmitFile(pdfFileName)

End If
```

Reporting in Microsoft Dynamics® NAV 2013

The result will resemble this:

The screenshot shows the Microsoft Visual Studio IDE with the code editor open. The title bar indicates two files are open: 'Default.aspx.vb*' and 'Default.aspx*'. The toolbar above the editor has icons for 'cmdSearch' and 'Click'. The code itself is written in VB.NET. It defines a partial class '_Default' that inherits from 'System.Web.UI.Page'. It contains a 'Page_Load' event handler and a 'cmdSearch_Click' event handler. The 'cmdSearch_Click' handler uses a service named 'Get_Sls_Doc' to check if a document exists by its number. If it does not exist, an error message is displayed. If it does exist, a PDF file is generated and transmitted back to the client. The code also includes logic to handle the case where the document number is not found.

```
Partial Class _Default
    Inherits System.Web.UI.Page

    Protected Sub Page_Load(ByVal sender As Object, ByVal e As System.EventArgs) Handles Me.Load
        End Sub

    Protected Sub cmdSearch_Click(ByVal sender As Object, ByVal e As System.EventArgs) Handles cmdSearch.Click
        Dim service As localhost.Get_Sls_Doc = New localhost.Get_Sls_Doc
        Dim pdfFileName As String
        Dim DocCheck As Integer
        service.UseDefaultCredentials = True
        service.Url = "http://localhost:7047/DynamicsNAV/WS/CRONUS International Ltd/Codeunit/Get Sls Doc"
        DocCheck = service.CheckDocNo(rblDocType.SelectedValue, txtDocNo.Text)
        If DocCheck = 0 Then
            lblErrorMsg.Text = rblDocType.SelectedItem.Text & " " & txtDocNo.Text & " was not found."
            lblErrorMsg.Visible = True
            txtDocNo.Text = ""
        Else
            lblErrorMsg.Visible = False
            service.UseDefaultCredentials = True
            service.Url = "http://localhost:7047/DynamicsNAV/WS/CRONUS International Ltd/Codeunit/Get Sls Doc"
            Response.ContentType = "application/pdf"
            pdfFileName = service.GenerateReport(rblDocType.SelectedValue, txtDocNo.Text)
            Response.TransmitFile(pdfFileName)
        End If
    End Sub
End Class
```

- d. Click **File, Save All** to save the modified files.

Task 14: Run the Web Application

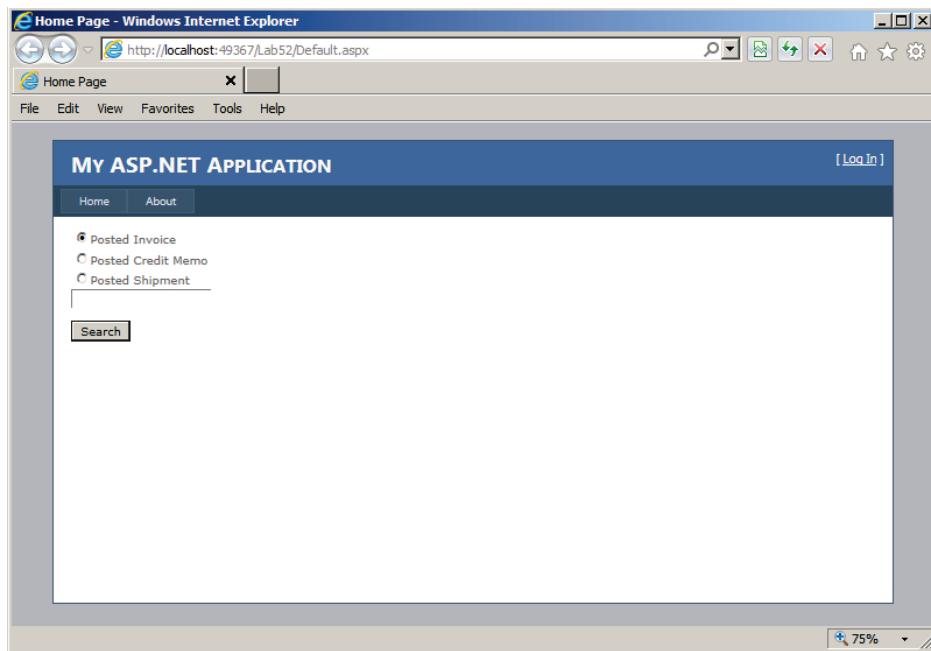
High Level Steps

1. Run the web application.

Detailed Steps

1. Run the web application.
 - a. On the Default.aspx page, press **F5**.
 - b. If the **Debugging Not Enabled** dialog box is displayed, select the **Start without debugging** option and click the **OK** button.

The result will resemble this:



Task 15: Test the Web Application

High Level Steps

1. Test the web application.

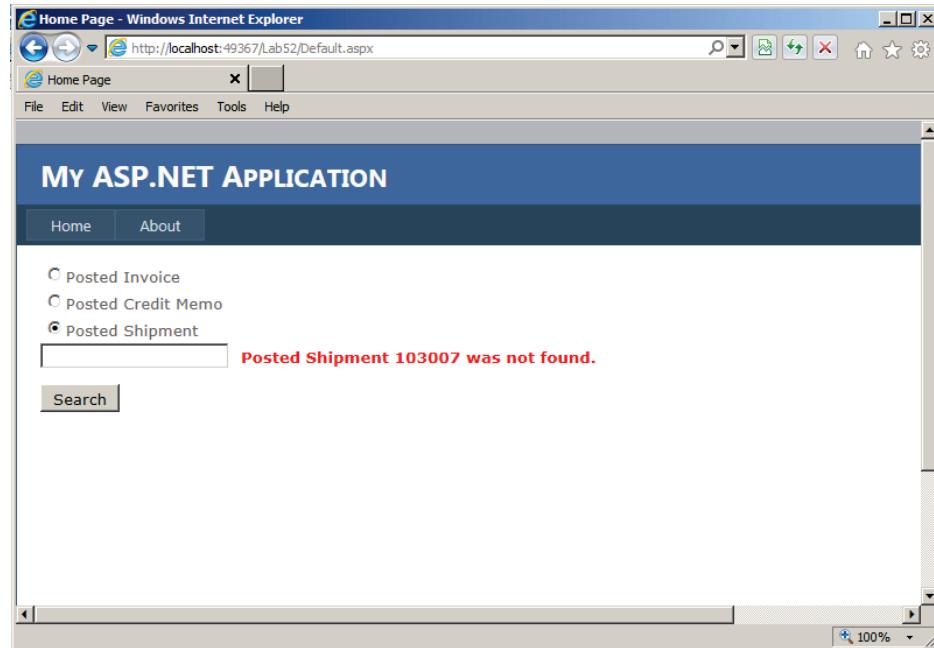
Detailed Steps

1. Test the web application.

To test the application, use the following scenario.

- a. Select the **Posted Shipment** option.
- b. In the text box, enter 103007.
- c. Click the **Search** button.

Reporting in Microsoft Dynamics® NAV 2013



- d. Select the **Posted Invoice** option.
- e. In the text box, enter 103007.
- f. Click the **Search** button.

Designstudio Gmunden
Fr. Birgitte Vestphael
Seepromenade 1b
AT-4810 Gmunden
Austria

Sales - Invoice
Page 1

CRONUS International Ltd.
5 The Ring
Westminster
W2 8HG London

Bill-to Customer No.	43687129	Phone No.	0666-666-6666
VAT Registration No.	ATU89759098	E-Mail	
Invoice No.	103007	Home Page	
Order No.	101011	VAT Req. No.	GB77777777
Posting Date	01/11/14	Giro No.	888-9999
Due Date	02/10/14	Bank	World Wide Bank
Document Date	January 10, 2014	Account No.	99-99-888
Payment Terms	1 Month/2% 8 days	Salesperson	John Roberts
Shipment Method	Ex Warehouse		
Prices Including VAT	No		

No.	Description	Posted Shipment Date	Quantity	Unit of Measure	Unit Price	Discount %	VAT	Amount
1			1 / 1				116.42%	

Module Review

Module Review and Takeaways

This module described how to run a report from the Role Tailored Client from a Page, Role Center and Departments section. It also explained how to create hyperlinks from and to reports and pages and how to run a report from the SharePoint Site. There's a description on how to run a report standalone in Visual Studio for debugging purposes. The labs in this module enforced knowledge on how to add a report to the application and how to consume a report as a web service.

Test Your Knowledge

Test your knowledge with the following questions.

1. Which functionalities are available from the ReportViewer Control Toolbar?

- () Find
- () Save Page Setup
- () Browse
- () Sort

2. Which property determines whether a menu or menu group will be included as a page in the **Departments** page of the RoleTailored Client?

- () Department Menu
- () Department Page
- () Department Level
- () Department Category

3. Which department categories are available in the **Create Item** window?

- () Lists, Tasks, Reports and Analysis, Documents, Departments, Home
- () Lists, Tasks, Reports and Analysis, Documents, History, Setup
- () Lists, Tasks, Reports and Analysis, Documents, History, Administration
- () Lists, Tasks, Reports and Analysis, Documents, Departments, Periodic Activities

Reporting in Microsoft Dynamics® NAV 2013

4. What is the RoleTailored client's equivalent for the **navigation** pane?

- () The Role Center Pages
- () The Activities Pages
- () The Action Pane
- () The Departments Page

5. What is true about MenuSuites for the RoleTailored client?

- () They are stored as separate MenuSuite objects.
- () They are stored inside the MenuSuite object for the Classic client.
- () They compromise the Departments page.
- () They compromise the navigation pane.

Test Your Knowledge Solutions

Module Review and Takeaways

1. Which functionalities are available from the ReportViewer Control Toolbar?

- () Find
- () Save Page Setup
- () Browse
- () Sort

2. Which property determines whether a menu or menu group will be included as a page in the **Departments** page of the RoleTailored Client?

- () Department Menu
- () Department Page
- () Department Level
- () Department Category

3. Which department categories are available in the **Create Item** window?

- () Lists, Tasks, Reports and Analysis, Documents, Departments, Home
- () Lists, Tasks, Reports and Analysis, Documents, History, Setup
- () Lists, Tasks, Reports and Analysis, Documents, History, Administration
- () Lists, Tasks, Reports and Analysis, Documents, Departments, Periodic Activities

4. What is the RoleTailored client's equivalent for the **navigation** pane?

- () The Role Center Pages
- () The Activities Pages
- () The Action Pane
- () The Departments Page

5. What is true about MenuSuites for the RoleTailored client?

- (✓) They are stored as separate MenuSuite objects.
- () They are stored inside the MenuSuite object for the Classic client.
- () They compromise the Departments page.
- () They compromise the navigation pane.