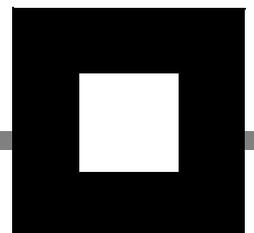




*Cognos*  
*Impromptu<sup>(R)</sup>*

**Discovering PowerPrompts**



## Product Information

This document applies to Impromptu<sup>(R)</sup> Version 7.1 and may also apply to subsequent releases. To check for newer versions of this document, visit the Cognos support Web site (<http://support.cognos.com>).

## Copyright

Copyright (C) 2003 Cognos Incorporated

While every attempt has been made to ensure that the information in this document is accurate and complete, some typographical errors or technical inaccuracies may exist. Cognos does not accept responsibility for any kind of loss resulting from the use of information contained in this document.

This document shows the publication date. The information contained in this document is subject to change without notice. Any improvements or changes to either the product or the document will be documented in subsequent editions.

U.S. Government Restricted Rights. The software and accompanying materials are provided with Restricted Rights. Use, duplication, or disclosure by the Government is subject to the restrictions in subparagraph (C)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013, or subparagraphs (C) (1) and (2) of the Commercial Computer Software - Restricted Rights at 48CFR52.227-19, as applicable. The Contractor is Cognos Corporation, 67 South Bedford Street, Burlington, MA 01803-5164.

This software/documentation contains proprietary information of Cognos Incorporated. All rights are reserved. Reverse engineering of this software is prohibited. No part of this software/documentation may be copied, photocopied, reproduced, stored in a retrieval system, transmitted in any form or by any means, or translated into another language without the prior written consent of Cognos Incorporated.

Cognos, the Cognos logo, Axiant, COGNOSuite, Cognos Upfront, Cognos DecisionStream, Impromptu, NoticeCast, PowerCube, PowerHouse, PowerPlay, Scenario and 4Thought are trademarks or registered trademarks of Cognos Incorporated in the United States and/or other countries. All other names are trademarks or registered trademarks of their respective companies.

Information about Cognos Products and Accessibility can be found at [www.Cognos.com](http://www.Cognos.com)

---

# Table of Contents

---

<b>Welcome</b>	<b>5</b>
<b>Chapter 1: Introduction to PowerPrompts</b>	<b>7</b>
What are PowerPrompts?	7
One Report for Many Report Consumers	8
Before You Start	8
Database Connections	9
Tutorial Files	9
Other Software Requirements to Run the Application	9
Sample PowerPrompts Application	9
<b>Chapter 2: Design the Pages of the PowerPrompts Application</b>	<b>11</b>
Start PowerPrompts	11
Set the Formatting of the Application	11
Choose an HTML Editor	11
Create Your New Application	12
Run the Application	13
Format the Entry, Exit, and Error Pages	13
Add Pages to the Application	18
Set the Links Between the Pages of the Application	19
Delete a Link	19
Add Page Links by Drawing	19
Add Page Links Using the Dialog Box	20
Complete the Links	20
What's Next	21
<b>Chapter 3: Define Link Conditions and SQL Dynamos</b>	<b>23</b>
Define the Links for Your Application	23
Add HTML Controls to the PickFilter Page	23
Define the Conditions for the Links	24
Create the Dynamos for Your Application	25
Create a Dynamo to Select a Country	25
Create a Dynamo to Select a Product Line	27
Create the JavaScript Code to Select a Product Type	29
Run the Application	30
Create an HTML Table to Select a Time Period	30
Use PowerPrompts to Remove Columns from a Report	32
Run the Application	33
What's Next	33
<b>Chapter 4: Complete the Application</b>	<b>35</b>
Create the Script to Run Against the Report	35
Script Editor Dialog Box	35
Select a Country	35
Select a Year	36
Select a Month	36
Select Product Line	36
Select Product Type	36
Remove Columns	37

Test the Application	37
View the Generated Script	37
Run the Application with a Report	38
Notify the Impromptu Web Reports Administrator	38
<b>Index</b>	<b>39</b>

---

# Welcome

---

## What Is in This Document

This tutorial is intended for advanced Impromptu report authors who want to use the same reports for different report consumers. It provides report authors with a step-by-step tutorial for creating a PowerPrompts application.

This tutorial contains information how to

- create and link HTML pages
- define SQL queries
- create scripts to run against Impromptu reports
- test your PowerPrompts application

## What You Must Know to Use This Tutorial Effectively

To create a PowerPrompts application, you should be familiar with

- the Impromptu catalogs for which you are going to create a PowerPrompts application
- HTML
- JavaScript
- SQL
- Database connections
- Web servers

However, enough information is included for you to complete the tutorial even if you are not familiar with the items listed above.

## Other Information

Our documentation includes user guides, tutorial guides, reference books, and other pieces to meet the needs of our varied audience.

All information is available in online help. Online help is available from the Help button in a Web browser, or the Help menu and Help button in Windows products.

The information in each online help system is available in online book format (PDF). However, the information from a given help system may be divided into more than one online book. Use online books when you want a printed version of a document or when you want to search the whole document. You can print selected pages, a section, or the whole book. Cognos grants you a non-exclusive, non-transferable license to use, copy, and reproduce the copyright materials, in printed or electronic format, solely for the purpose of providing internal training on, operating, and maintaining the Cognos software.

In Windows products, online books are available from the Windows Start menu (Cognos) and from the product Help menu (Books for Printing). In a Web browser, online books may be available from the Welcome section of the help system, or from within the Cognos Web portal (Upfront). All online books are available on the Cognos documentation CD. You can also read the product readme files and the installation guides directly from the Cognos product CDs.

Only the installation guides are available as printed documents.

The following documents contain related information, and may be referred to in this document.

Topic	Location
The other PowerPrompts documentation piece is the PowerPrompts help.	<i>PowerPrompts Reference</i> Windows help, online book (PDF)

Welcome

## Questions or Comments?

For the fastest response to questions about using PowerPrompts, contact customer support.

For information about customer support locations and programs, see Cognos on the Web on the Help menu or visit the Cognos Support Web site (<http://support.cognos.com>).

---

# Chapter 1: Introduction to PowerPrompts

---

A PowerPrompts application is a series of HTML pages that lets report consumers customize a report. This tutorial creates a PowerPrompts application to be used with the samples provided with Impromptu.

## What are PowerPrompts?

A PowerPrompts application is a series of HTML pages that allow report consumers to select the information they want to see in a report. PowerPrompts Developer Studio is the development tool that you use to design these applications for Impromptu Web Reports.

With a PowerPrompts application applied to a personal report, report consumers can customize the report from the HTML pages of the application. For example, by using a PowerPrompts application, you can let report consumers add

- columns to a report
- column formatting
- a report template
- conditional formatting
- filters

You can also use a PowerPrompts application to reduce the number of reports you must create and maintain. Rather than creating a number of different filtered reports, you can let report consumers do the filtering by using a PowerPrompts application.

When report consumers access a report with an associated PowerPrompts application, they are guided through a series of prompts, where they select what they want to see in the report. After the selections are made, a script applies a series of changes to the report, and a PDF report is created, based on the report consumers' selections.

You create the scripts and dynamos, and define the navigational path for the HTML pages in the PowerPrompts Developer Studio. You use an HTML editor to create the HTML pages that the report consumers see.

You can use dynamos to format the results of SQL queries against a data source. For example, you can create a dynamo that retrieves a list of countries from a data source. The list appears when the report consumers view the HTML page that contains the dynamo. In the HTML tags, the dynamo is called by the following server-side JavaScript code:

```
<%=App.RunDynamo("Dynamo Name")%>. For more information, see "Create the Dynamos for Your Application" \(p. 25\).
```

PowerPrompts is available with both the Administrator and User versions of Impromptu.

### PowerPrompts Process Overview

To create a PowerPrompts application, you must complete the following tasks:

1. Create a series of HTML pages that will prompt your report consumers to define the contents of a report.
2. Create the instructions to navigate through the HTML pages.
3. Save the instructions as a PowerPrompts application file (.xxm).
4. Test the PowerPrompts application and correct any errors.
5. Inform the Report Administrator that the PowerPrompts application is ready for Impromptu Web Reports users.

When the report consumers run the report, they will be presented with HTML pages that contain prompts for them to use to customize their reports.

In this tutorial, you will go through each step required to create the prompts, including creating the HTML controls, the SQL queries required to create the dynamos, and the scripts that will be run against the report, based on the information specified by the report consumers.

## One Report for Many Report Consumers

Using this tutorial, you can create a PowerPrompts application that is designed to be run against a Sales Rep Performance report (pwp\_tutorial\_js.imr). The web pages of the application contain HTML controls and dynamos that allow you to select a specific time period, and a product line and type, or remove a specific column so that your report consumers can customize the report. PowerPrompts applications consist of an application file (.xml) and a series of HTML pages (.htm) that are used in the application.

You create the JavaScript code that will modify the report after the report customers have made their selections. For example, when they specify a time period or product type, or remove a column.

### The Problem

You are a report administrator for the Great Outdoors Company, and you must create a report detailing the performance of sales representatives for use in annual performance reviews. You know that each of your report consumers will want different information displayed in the report. However, you want to create a number of different reports. You would like the report consumers to select the information they want to see. Rather than allowing report consumers to filter values and columns with regular Impromptu prompts, you want to create a more interactive report that will meet the needs of a number of different report consumers.

### The Solution

You decide to use a PowerPrompts application so that your report consumers can customize the report. You also decide to use a PowerPrompts application because you want to be able to use it on other reports that you have already created, or will create later.

For this PowerPrompts application, you decide to create a page so that report consumers can select

- the sales representatives in a country
- to filter by either time period or product type and line
- the time periods for the sales figures
- the product line of the products sold
- the product type of the products sold
- columns to remove from the report

## Before You Start

Before you begin this tutorial, you must install Impromptu, including the Impromptu and Architect samples. These samples contain the required database and Impromptu report. You must also have a Web server, a Web browser, and a PDF viewer installed on your computer.

You must do this tutorial on a computer running a Windows operating system. PowerPrompts applications can be deployed to UNIX servers, but can only be created on a computer running Windows.

We recommend that you disable Access Manager (as a database information source for PowerPrompts) while completing this tutorial. The steps necessary to integrate Access Manager with PowerPrompts are not covered in this tutorial.

### Steps to Enable Access Manager

1. Open PowerPrompts Developer Studio.
2. From the Tools menu, click Options.
3. In the General tab, clear the Enable Cognos Common Logon check box and click OK.

## Database Connections

PowerPrompts can connect to logical databases as defined in the Cognos.ini file. You will connect to the GOSales database in this tutorial; its logical database name is GOS. For more information about accessing Cognos logical databases, see the *Database Definition Manager* help in Impromptu.

## Tutorial Files

The following files are included with this tutorial.

- `pwp_tutorial_js.imr`  
An Impromptu report that has been formatted for this tutorial. Use this file to test your PowerPrompts application and located in the *installation\_location\samples\Impromptu\additional reports* folder.
- `Header.js`  
A JavaScript file that contains the standard header HTML for all the application's HTML pages. This file is located in the *installation\_location\samples\powerprompts\tutorial* folder.
- `pwp_html.txt`  
A text file that contains all the text that you must enter on each page in the application. You can paste the text from this file to the pages in your application. This file is located in the *installation\_location\samples\powerprompts* folder.
- `pwp_scripts.txt`  
A text file that contains the scripts and the dynamo SQL used in this tutorial. As with the HTML file, you can paste the appropriate text into the Script Editor, Dynamo SQL, and Dynamo Definition dialog boxes. This file is located in the *installation\_location\samples\powerprompts* folder.

## Other Software Requirements to Run the Application

To run your PowerPrompts application, you must also have the following software installed on your computer:

- a Web server
- a Web browser
- Adobe Acrobat Reader

## Sample PowerPrompts Application

The Impromptu sample files include a sample PowerPrompts application. These files are located in the *installation\_location\samples\powerprompts\tutorial* folder where you installed Impromptu.

Sample PowerPrompts application	Description
tutorial.xml	Use to filter report contents by country, then by time or by product line and product type. This application includes a final prompt to remove columns from the report. This is the same application you create by following this tutorial.

### Notes

If PowerPrompts is installed to a location other than `C:\Program Files\Cognos\cern`, the `tutorial.xml` will fail to run the report. You must specify the location of the `pwp_tutorial_js.imr` report.

**Steps**

1. From the File menu select Properties.
2. On the General tab in the Report to Run box, either manually change the location of the pwp\_tutorial\_js.imr report or use the Browse button to navigate to the location of the file.

---

# Chapter 2: Design the Pages of the PowerPrompts Application

---

A PowerPrompts application lets report consumers select the information they want to view in a report, from a series of HTML pages. The HTML pages are designed in an HTML editor, and the path that the report consumers are guided through is defined in the PowerPrompts Developer Studio.

## Start PowerPrompts

PowerPrompts is installed with both the Administrator and User versions of Impromptu. All the program files are installed in the same location as Impromptu, and a Start Menu shortcut is added.

For more information about installing Impromptu, see the installation guide.

### Steps

- Start PowerPrompts.  
The PowerPrompts Developer Studio opens, a blank workspace appears. The workspace is where you add pages and define links between the pages in your PowerPrompts application.

## Set the Formatting of the Application

PowerPrompts pages are formatted using regular HTML tags. A page can contain any of the formatting that you can create using HTML.

To format your HTML pages, you can set an HTML template to apply to each page. Alternatively, you can add the contents of a JavaScript file (.js) to each page, using the #include statement. In this tutorial, you will use the Header.js file. This file is in the *installation\_location*\samples\powerprompts\tutorial folder where you installed Impromptu.

### Notes

- If you do not specify a template, PowerPrompts uses a default that has a button that lets you navigate to the Exit page in the application. This means that you can run an application immediately after it has been created. For more information about running the application, see ["Run the Application" \(p. 13\)](#).

## Choose an HTML Editor

You can use any HTML editor to format the pages of your application. For example, you can use an HTML editor, such as Microsoft FrontPage, the HTML editor included with your Web browser, such as Netscape Composer, or a text editor, such as Notepad. Before you edit your PowerPrompts pages, we recommend you specify an HTML editor. If you don't specify one, a default editor is used.

### Notes

- A text file containing the HTML text is included so that you can copy and paste, rather than type the text. We recommend that you use an editor that allows for easy copying and pasting, such as Notepad. Some HTML editors will substitute characters in the HTML tags for named entities. For example, the < and > symbols in the HTML tags you copy might be substituted with &lt; and &gt;, depending on the editor you are using.

### Try This

1. From the Tools menu, click Options.
2. Click the File Locations tab.
3. In the HTML Editor box, specify the HTML text editor that you want to use.
4. In the Default Application Folder box, specify a location where you want to save your PowerPrompts applications. The folder must already exist.  
Your applications are saved to subfolders within the default application folder.
5. Click OK.

## Create Your New Application

Each PowerPrompts application is created with an

- Entry page, which is the first page that report consumers will see.
- Exit page, which initiates the scripts based on the report consumers' selections. All navigational paths through the PowerPrompts application must end with the Exit page, and no links can leave the Exit page.
- Error page, which is displayed if an error occurs when a PowerPrompts application is running. No links can be made to or from the Error page.

The path between the Entry and Exit pages can contain as many pages as you require. The pages can have direct, linear links between the Entry and Exit pages, looped links that allow report consumers to specify a number of options, and conditional links which follow a branch of the navigational path. The application that you create by using this tutorial contains both direct and conditional links.

Your PowerPrompts application contains nine HTML pages. Each page has the same formatting to maintain consistency. Six of the pages contain your prompts. The other three will be the Entry, Exit, and Error pages. The Entry, Exit, and Error pages are automatically created as HTML files in the folder specified in the New Application dialog box. The files are named EntryPage.htm, ExitPage.htm, and ErrorPage.htm.

### Try This

1. From the File menu, click New.  
The New Application dialog box appears.
2. In the Application Name box, type SalesRep.  
The Location box automatically specifies a subfolder in the default application folder. The subfolder is created with the same name as your PowerPrompts application. The subfolder will be created when you save the application.  
The PowerPrompts application file (\*.xmx) and the HTML files (\*.htm) used in the application are stored in this location.  
Leave the Report to Run box blank for now. You will select a report later. For more information, see ["Run the Application with a Report" \(p. 38\)](#).
3. Click OK.  
A new PowerPrompts application is created that contains an Entry, Exit, and Error page. You can insert the other pages that you add to the application anywhere in the workspace. You can move them at any time in the process, and the links will remain intact. They are also created automatically as HTML files in the same folder, using the template that you have specified.

4. From the File menu, click Save to save your application.  
The Verify Application dialog box appears indicating that you did not specify a report with which to test the application.  
Each time you save an application, the PowerPrompts Developer Studio verifies the application and shows you any errors and warnings.
5. In the Verify Application dialog box, click Save to close the dialog box and save the application.

### Tips

- You can also click Verify from the File menu to check for errors in your application.

## Run the Application

As you create your PowerPrompts application, you can run it to test for errors.

### Try This

1. Turn on your Web server.  
You can run the application only if you have a Web server installed on your computer.
2. From the Tools menu, click Run.  
If the Verify Application dialog box opens, click Save to save the application, and close the dialog box.  
The application's Entry page appears in your Web browser.

## Format the Entry, Exit, and Error Pages

The Entry, Exit and Error page are created with default HTML tags. You add the Header.js file to all three pages to create the standard header. Also, you must add specific HTML or JavaScript code for the scripts to produce the correct results.

### Notes

- Instead of typing the HTML tags, you can copy it from the pwp\_html.txt file that is in the *installation\_location*\samples\powerprompts folder.

### Format the Entry Page

You can format the pages in your application by using any HTML controls that you want. For example, you can use tables, colors, and backgrounds. However, to make it a useful starting point for your report consumers, you must add some text. Also, you add the Header.js file to create the standard header.

### Try This

1. Right-click the Entry page, and click Edit HTML.  
The page opens in your HTML editor and contains the following HTML tags:

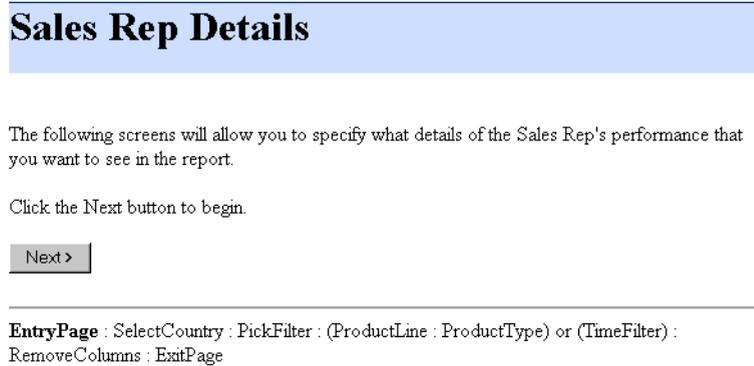
```
<html>
<head><Title><%=App.CurrentPage%></Title>
</head>
<body>
<h1><%= App.CurrentPage %></h1>

<form method="POST">
<input type="submit" value=" Next &gt; " >
</form>

</body>
</html>
```

The `<input type="submit" value=" Next &gt; ">` tag creates a Next button. PowerPrompts uses this button to move report consumers to the next page of the application. The action of this button is controlled by the PowerPrompts server and you do not need to provide additional instructions.

2. Replace the `<%= App.CurrentPage %>` text between the opening and closing Title tags (`<title>...</title>`) with:  
`PowerPrompts Tutorial`
3. Remove the text that follows the opening Body tag (`<body>`):  
`<h1><%= App.CurrentPage %></h1>`
4. After the closing Head tag (`</head>`) and before the opening Body tag (`<body>`), insert the following text:  
`<%#include "Header.js"%>`  
This code adds the standard header.
5. After the opening Body tag (`<body>`), insert the following text:  
`<br><p>The following screens will allow you to specify what details of the Sales Rep's performance that you want to see in the report.</p><p>Click the Next button to begin.</p>`  
This code introduces the PowerPrompts application.
6. After the closing Form tag (`</form>`), insert the following text:  
`<hr><b>EntryPage</b> : SelectCountry : PickFilter : (ProductLine : ProductType) or (TimeFilter) : RemoveColumns : ExitPage<br>`
7. Save your changes and close the file.
8. Copy the Header.js file from `installation_location\Samples\PowerPrompts` to the location where you saved the SalesRep application.
9. Run the application.  
When viewed in a Web browser, the Entry page will look like this.



For more information about running an application, see ["Run the Application" \(p. 13\)](#).

10. From the File menu, click Save to save the PowerPrompts application.  
The Verify Application dialog box appears indicating that you did not specify a report with which to test the application. Ignore this message for now. Click Save to save the application and close the dialog box.

## Format the Exit Page

You add JavaScript code to create two tables that show the user's selections and the Header.js file to create the standard header.

**Try This**

1. Right-click the Exit page, and click Edit HTML.

The page opens in your HTML editor and contains the following HTML tags:

```
<html>
<head><Title><%=App.CurrentPage%></Title>
</head>
<body>
<h1><%= App.CurrentPage %></h1>

<form action="<%= App.FinalURL %>" method="POST">
<input type="submit" value="Run Report">
</form><form action="<%= App.BackURL %>" method="POST">
<input type="submit" value="Cancel">
</form>

</body>
</html>
```

This code produces a run report button that runs the scripts against the report and a cancel button that takes users back to Upfront. These scripts will be generated from the report consumers' selections in the PowerPrompts application.

2. Replace the `<%= App.CurrentPage %>` text between the opening and closing Title tags (`<title>...</title>`) with:

```
PowerPrompts Tutorial
```

3. Remove the text that follows the opening Body tag (`<body>`):

```
<h1><%= App.CurrentPage %></h1>
```

4. After the closing Head tag (`</head>`), insert the following text:

```
<%=include "Header.js"%>
```

This code adds the standard header.

5. Copy the following text from the `pwp_html.txt` file and insert the after the opening `Body` tag (`<body>`). The file is in the `installation_location\Samples\PowerPrompts` folder.

```

<br><b>Table of Variables Selected</b>
<table>
<table border=2 bgcolor=#80FFFF>
<tr width="15%">
<th>Variable Name</th>
<th>Value</th>
</tr>
<%
    if (App.Variables("CountryCd"))
    {
        var rs = new Recordset;
        rs.Open("SELECT Country FROM Country WHERE SalesCountryCode=" +
App.Variables("CountryCd"), "GOS");
        if (!rs.EOF)
        {
            Response.Write("<tr><td>Country</td><td>" + rs.Fields("Country") +
"</td></tr>");
        }
    }
    if (App.Variables("FilterType"))
    {
        Response.Write("<tr><td>Filter Type</td><td>" +
App.Variables("FilterType") + "</td></tr>");
    }
    if (App.Variables("ProdLine"))
    {
        var rs = new Recordset;
        rs.Open("SELECT ProductLine FROM ProductLine WHERE ProductLineCode=" +
App.Variables("ProdLine"), "GOS");
        if (!rs.EOF)
        {
            Response.Write("<tr><td>Product Line</td><td>" +
rs.Fields("ProductLine") + "</td></tr>");
        }
    }
    if (App.Variables("ProdType"))
    {
        var rs = new Recordset;
        rs.Open("SELECT ProductType FROM ProductType WHERE ProductTypeCode=" +
App.Variables("ProdType"), "GOS");
        if (!rs.EOF)
        {
            Response.Write("<tr><td>Product Type</td><td>" +
rs.Fields("ProductType") + "</td></tr>");
        }
    }
    if (App.Variables("OrderYear"))
    {
        Response.Write("<tr><td>Order Year</td><td>" + App.Variables("Orderyear")
+ "</td></tr>");
    }
    if (App.Variables("OrderMonth"))
    {
        switch(App.Variables("OrderMonth").toString())
        {
            case "1" :
                Response.Write("<tr><td>Order Month</td><td>" + "January" + "</td></tr>");
                break;
            case "2" :
                Response.Write("<tr><td>Order Month</td><td>" + "February" + "</td></tr>");
                break;
            case "3" :
                Response.Write("<tr><td>Order Month</td><td>" + "March" + "</td></tr>");
                break;
            case "4" :
                Response.Write("<tr><td>Order Month</td><td>" + "April" + "</td></tr>");
                break;
            case "5" :
                Response.Write("<tr><td>Order Month</td><td>" + "May" + "</td></tr>");
                break;
            case "6" :
                Response.Write("<tr><td>Order Month</td><td>" + "June" + "</td></tr>");
                break;
        }
    }
}

```

```

        case "7" :
        Response.Write("<tr><td>Order Month</td><td>" + "July" + "</td></tr>");
        break;
        case "8" :
        Response.Write("<tr><td>Order Month</td><td>" + "August" + "</td></tr>");
        break;
        case "9" :
        Response.Write("<tr><td>Order Month</td><td>" + "September" + "</td></tr>");
        break;
        case "10" :
        Response.Write("<tr><td>Order Month</td><td>" + "October" + "</td></tr>");
        break;
        case "11" :
        Response.Write("<tr><td>Order Month</td><td>" + "November" + "</td></tr>");
        break;
        case "12" :
        Response.Write("<tr><td>Order Month</td><td>" + "December" + "</td></tr>");
        break;
    }
    if (App.Variables("OrderMonth")== "-1")
    {
        Response.Write("<tr><td>Order Month</td><td>" + "All Months" + "</td></tr>");
    }
}
%>
</table>
<br>
<b>Table of Columns Selected</b>
<table>
<table border=2 bgcolor=#80FFFF>
<tr width="15%">
<th>Column Name</th>
</tr>
<%
        if (App.Variables("FormatLastname"))
        {
            Response.Write("<tr><td>" + "Lastname" + "</td></tr>");
        }
        if (App.Variables("FormatFirstname"))
        {
            Response.Write("<tr><td>" + "Firstname" + "</td></tr>");
        }
        if (App.Variables("FormatOrderdate"))
        {
            Response.Write("<tr><td>" + "Orderdate" + "</td></tr>");
        }
        if (App.Variables("FormatVendorname"))
        {
            Response.Write("<tr><td>" + "Vendorname" + "</td></tr>");
        }
    }
%>
</table>

```

This code produces a summary of user choices on the Exit page.

**Note:** When copying from the pwp\_html.txt file, maximize your text editor (for example, Notepad) and turn off word wrapping. This should remove any line returns in string code. You can also remove them manually. Line returns in strings can prevent server-side JavaScript from working properly.

6. After the closing Form tag (</form>), insert the following text:

```

<hr>EntryPage : SelectCountry : PickFilter : (ProductLine : ProductType) or
(TimeFilter) : RemoveColumns : <b>ExitPage</b><br>

```

7. Save your changes, and close the file.

## Format the Error Page

You add the Header.js file to create the standard header.

**Try This**

1. Right-click the Error page, and click Edit HTML.

The page opens in your HTML editor and contains the following HTML tags:

```
<html>
<head><Title><%=App.CurrentPage%></Title>
</head>
<body>
<h1><%= App.CurrentPage %></h1>

<h1>Error</h1>
<p>
<pre><%
    for (var i = 0; i < App.Errors.Count; i++)
    {
        Response.Write(App.Errors(i) + "<hr/>");
    }
%></pre>
</p><p>Check the server event log for more details.</p>

</body>
</html>
```

This code will display an error message to the report consumer if the PowerPrompts server encounters a problem, such as an error in the application. The App.Errors displays a list of errors that occurred while generating the page.

For more information about PowerPrompts code, see the online Help.

2. Replace the `<%= App.CurrentPage %>` text between the opening and closing Title tags (`<title>...</title>`) with:

```
PowerPrompts Tutorial
```

3. After the closing Head tag (`</head>`), insert the following text:

```
<%#include "Header.js"%>
```

This code adds the standard header.

4. Remove the following text after the opening Body tag (`<body>`):

```
<h1><%= App.CurrentPage %></h1>
```

```
<h1>Error</h1>
```

5. Save your changes and close the file.

The Entry, Exit, and Error pages are now complete. Now you will add the remaining pages of your application.

## Add Pages to the Application

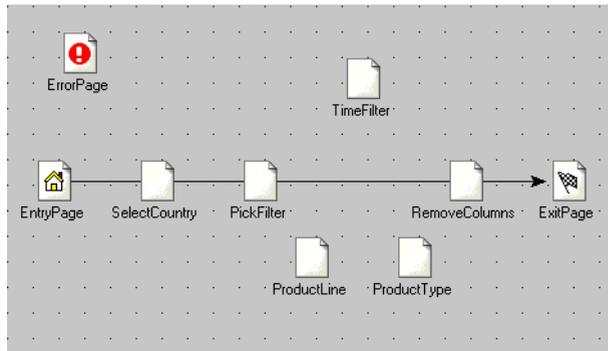
Each page that you add to the application represents an individual HTML page that will provide a prompt for report consumers. You can include any number of prompts on an HTML page, but this tutorial uses one page for each prompt.

**Try This**

1. Click the Insert Page button.
2. Click the workspace where you want the new page inserted.  
The page is inserted, with the name selected.
3. Type `SelectCountry`, and press Enter.
4. Repeat steps 1 to 3 to create the following pages:
  - `PickFilter`
  - `TimeFilter`
  - `ProductLine`
  - `ProductType`
  - `RemoveColumns`

5. Arrange the pages to match the following illustration.

To move a page, click the Select Page button, click the page, and then drag it to a new position in the workspace.



6. Save the PowerPrompts application.

The Verify Application dialog box opens, indicating that you did not specify a report with which to test the application, that several pages cannot be reached, and that some pages have no default links specified. Ignore these messages for now. Click Save to save the application and close the dialog box.

## Set the Links Between the Pages of the Application

The navigational path of the application still links between the Entry page and the Exit page. In the following tasks, you will remove that link. You will also create new links. These links represent the possible navigational paths that report consumers are guided through when they want to view a report that has this PowerPrompts application associated with it.

### Delete a Link

Before you add the links to the pages of the application, remove the link between the Entry and Exit pages.

#### Tips

- If you want to give report consumers the option of viewing the report as it is, include a path from the Entry page to the Exit page. You must define the conditions for this link.

#### Try This

1. Right-click the Entry page, and click Properties.
2. In the Page Properties dialog box, click the Links tab.  
The link from the Entry page to the Exit page is represented by the <DEFAULT> condition, and the Exit page is the linked page.
3. Select <DEFAULT>.
4. Click Delete, then click OK.  
The link from the Entry page to the Exit page no longer appears.



### Add Page Links by Drawing

You can create a link between the pages directly in the workspace. This method lets you quickly create your PowerPrompts applications.

#### Try This

1. Select the Create Link button.
2. Click the Entry page and drag it to the SelectCountry page.  
A new link, represented by an arrow, is drawn between the pages.



3. Create a link from the SelectCountry page to the PickFilter page.

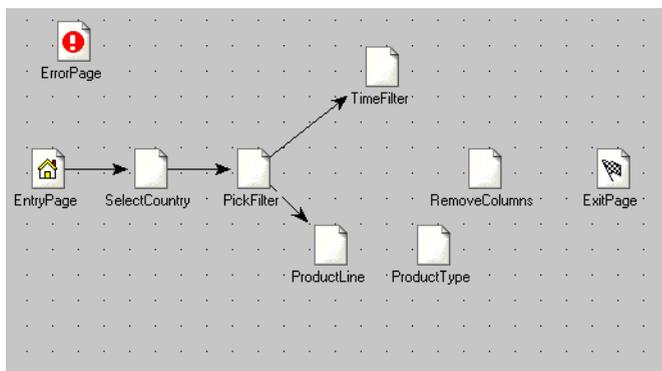
## Add Page Links Using the Dialog Box

You use conditions to give report consumers a choice in the navigational path. In your application, you will give report consumers a choice between filtering the report by time period or by product line and product type. You use the Page Properties dialog box to define the conditions. However, you can also use the Page Properties dialog box to set the links between the pages.

### Try This



1. If it is not already selected, click the Select Page button.
2. In the workspace, right-click the PickFilter page, and click Properties.
3. In the Page Properties dialog box, click the Links tab.  
The Condition and Page lists should be empty. If they are not, select the entry, and click the Delete button.
4. Click the New button.  
A condition named <DEFAULT> is created. You will define this condition in the next chapter.
5. In the Page column, select TimeFilter, and click Apply.  
A new link is added, and the Page Properties dialog box remains open. Move the Page Properties dialog box so that you can see the workspace. The new link is added between the PickFilter and TimeFilter pages.
6. Click New again.
7. In the Page column of the second <DEFAULT> condition, select ProductLine, and click OK.  
The Page Properties dialog box closes, and the new links are added. Your application should look like the following illustration.



## Complete the Links

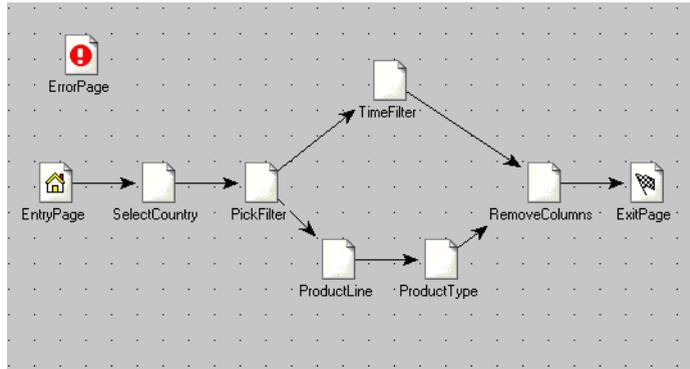
Before you go on to the next step, you must finish adding the navigational links to your application. You can either click the Create Link button and draw the links, or use the Page Properties dialog box.

### Try This

1. Using either of the methods previously described, create the following links:

- TimeFilter to RemoveColumns
- ProductLine to ProductType
- ProductType to RemoveColumns
- RemoveColumns to ExitPage

If you are using the Page Properties dialog box, leave the entries in the Condition column as <DEFAULT>.



2. Save your PowerPrompts application.

The Verify Application dialog box opens, indicating that you did not specify a report with which to test the application, and that a page has two or more links with the same condition. The other errors that were listed have been corrected by linking all the pages. Ignore the remaining messages for now. Click Save to save the application and close the dialog box. By following the path that you created, as displayed in the illustration, report consumers can navigate from the Entry page to the Exit page by following one of two branches. If report consumers choose to filter by time, they cannot filter by product, and conversely, if they choose to filter by product, they cannot filter by time.

### Tips

- If you want to give report consumers access to more than one filter, create a link from the RemoveColumns page back to the PickFilter page. Later, when you define your conditions, you must include a condition for linking back to the PickFilter page.

## What's Next

Now that the PowerPrompts application is designed and the pages are formatted, you can create the link conditions, dynamos and JavaScript code for the application.



---

# Chapter 3: Define Link Conditions and SQL Dynamos

---

The links between pages in a PowerPrompts application determine the path that report consumers will navigate through. If a page links two or more pages, then you must set conditions. If a page links to a single page, you do not need to edit the default condition.

You can also create dynamic lists from which report consumers can select values for the report. You create these lists by using dynamos that retrieve columns from a data source.

## Define the Links for Your Application

Each link is defined by a condition. The links that you created in the last chapter were created with a default condition.

For example, the SelectCountry page links only to the PickFilter page. In this case, the default condition allows the report consumer to properly navigate to the PickFilter page from the SelectCountry page.

If a page links to two or more pages, you must define conditions for the links to work properly. For example, the PickFilter page links to either the TimeFilter page or the ProductLine page. For this link to work properly, you must edit the conditions. To do so, add the HTML tags to the page so that report consumers can choose a path, and then set the conditions in the Page Properties dialog box.

## Add HTML Controls to the PickFilter Page

When you designed the PowerPrompts application, you wanted to give report consumers the option of selecting whether they wanted to filter by time periods or by product line and product type. This option made your single report more useful for report consumers who were more interested in sales by time periods and for those more interested in sales by product.

To let report consumers select a path, you must add the controls to the PickFilter page of your application.

### Notes

- Instead of entering the HTML tags, you can copy it from the `power_prompts_html.txt` file that is in the `installation_location\samples\powerprompts` folder.

### Try This

1. In the PowerPrompts Developer Studio workspace, right-click the PickFilter page and click Edit HTML.

The page opens in your HTML editor and contains the following HTML tags:

```
<html>
<head><Title><%=App.CurrentPage%></Title>
</head>
<body>
<h1><%= App.CurrentPage %></h1>

<form method="POST">
<input type="submit" value=" Next &gt; " >
</form>

</body>
</html>
```

- Replace the `<%= App.CurrentPage %>` text between the opening and closing Title tags (`<title>...</title>`) with:

```
PowerPrompts Tutorial
```

- Remove the text that follows the opening Body tag (`<body>`):

```
<h1><%= App.CurrentPage %></h1>
```

- After the closing Head tag (`</head>`) and before the opening Body tag (`<body>`), insert the following text:

```
<%#include "Header.js"%>
```

This code adds the standard header.

- After the opening Body tag (`<body>`), insert the following text:

```
<br><p><H2>Restrict by Time Periods or Product Line</H2></p>
<p>Choose one of the following, and click Next. If you do not select one of the
choices, Filter by Product Line will be selected by default.</p>
```

This code produces instructions indicating what the report consumer must do.

- Between the opening Form tag (`<form method="POST">`) and the Input Type tag (`<input type="submit" value="Next">`), insert the following text:

```
<p><input type="radio" name="FilterType" value="ByTime"><b> Filter by Time
Periods</b> (e.g. year or month)</p>
<br>
<p><input type="radio" name="FilterType" value="ByProductLine" checked><b> Filter by
Product Line </b> (e.g. Golf or Mountaineering Equipment)</p>
```

This code produces radio buttons for the report consumer to specify a navigational path.

The checked attribute for the second radio button means that the page will appear with the Filter by Product Line option button already selected.

- After the closing Form tag (`</form>`), insert the following text:

```
<hr>EntryPage : SelectCountry : <b>PickFilter</b> : (ProductLine : ProductType) or
(TimeFilter) : RemoveColumns : ExitPage<br>
```

- Save your changes and close the file.

## Define the Conditions for the Links

Use the Page Properties dialog box to define the conditions for the links in your application.

### Try This

- Right-click the PickFilter page, and click Properties. The Page Properties dialog box appears.
- Click the Links tab. The tab should contain the following list of conditions and pages.

Condition	Page
<DEFAULT>	TimeFilter
<DEFAULT>	ProductLine

- Click the `<DEFAULT>` condition that links to the TimeFilter page. The tab should contain the following list.

Condition	Page
<DEFAULT>	TimeFilter
<DEFAULT>	ProductLine



- Click the Edit button. The Link Condition dialog box appears.



- Click the New button. A FilterType = "ByTime" entry is created from the code you entered on the HTML page. PowerPrompts Developer Studio reads the HTML page and inserts the variables in the Link Condition dialog box.

- In the Value box, ensure that ByTime is selected.

- Ensure that the equal sign (=) is selected in the box that is between the Name and Value boxes.

The entry should look like `FilterType = "ByTime"`, where `FilterType` represents the name of the condition, and `ByTime` equals the value that report consumers will select.

- Click OK.

The Link Conditions dialog box closes, but the Page Properties dialog box remains open. The first condition on the page is now defined. The Page Properties dialog box should contain the following list:

Condition	Page
FilterType = "ByTime"	TimeFilter
<DEFAULT>	ProductLine

Leave the link to the ProductLine page as `<DEFAULT>`.

Each page in a PowerPrompts application must have a default link. In this case, if the report consumers do not specify what they want to filter the report by, ProductLine will be selected by default.

**Note:** The default link should always be at the bottom of the list. The conditions are evaluated in the order in which they appear on the Links tab of the Page Properties dialog box. If the default link is listed first, report consumers will not reach the other links. You can use the Move Up and Move Down arrows to change the order of the links.

- Click OK to close the Page Properties dialog box.
- Save the PowerPrompts application.

The Verify Application dialog box opens indicating that you did not specify a report with which to test the application. You have corrected the error message that said "This page has two or more links with the same condition." Ignore the remaining message. Click Save to save the application and close the dialog box.

## Create the Dynamos for Your Application

You can use dynamos to format the results of SQL queries against a data source. Typically, dynamos are used to generate HTML controls that are placed on a page of a PowerPrompts application. The benefit of using dynamos is that if the data source changes, the dynamos automatically display the updated results, because the query is run when the report consumer views the page.

In your PowerPrompts application, you will use SQL queries to retrieve information from the GOSales data.

You can create dynamos manually, or by using the Dynamo wizard. Alternatively, you can write JavaScript code to do what a dynamo would do. You will do all three in this tutorial.

### Notes

You may receive an error message in a dynamo when trying to add a data source on Windows 2000 or Windows 2000 Professional. If so, search your local drive for the file, `MSSTDFMT.DLL`. Ensure that the file is located in the `\system32` directory on your computer. If you cannot find this file, contact your system administrator.

All of the text for the SQL queries used in the dynamos are included in a text file that was installed with PowerPrompts. The file is in the `installation_location\samples\powerprompts` folder. The file is named `pwps_scripts.txt`.

## Create a Dynamo to Select a Country

For your PowerPrompts application, you created a page that lets report consumers select a country for analyzing the performance of sales representatives. Rather than inserting an entry for each country on the HTML page, you decide to use a dynamo to create a list and use a list box to allow the report consumers to select the country for the report.

The dynamo used to select a country is an SQL query that returns the countries listed in the GOS data source. After the dynamo is defined, you will add the HTML controls for the list box to the SelectCountry page of your application. When the report consumers view that page of the application, they will be able to select a country for their report.

When you create a dynamo using this method, the Dynamo wizard identifies the data source, creates the SQL, and creates the dynamo definition. All you have to do is to create the HTML controls on the page of the PowerPrompts application.

### Try This

- 
1. From the Tools menu, click Dynamo Manager.
  2. In the Dynamo Manager dialog box, click the Dynamo wizard button and click Next.
  3. Type Countries for the dynamo name, and click Next.
  4. Select the Drop Down List for the display style, and click Next.
  5. In the Logical Database box, type GOS and click Next.
  6. Clear the Use This Authentication Information check box and click Next.  
You don't need to provide a user ID and password or set a record retrieval limit for this data source.
  7. From the list of tables, select Country and click Next.  
**Tip:** If the data source contains multiple entries for some values, you can select the Use Only Unique Values check box to ensure that no values are repeated in the list box.
  8. In the HTML Control Name box, type CountryCd.  
This is the value that is selected by the report consumer and passed to the scripts that will be run against the report.
  9. In the HTML Control Value box, ensure that SalesCountryCode is selected.  
This control sets what is used by PowerPrompts. In this data source, SalesCountryCode is a list of numbers that represent the countries; for example, 4 for Canada, 16 for Australia, and 22 for Finland.
  10. In the Control Label box, select Country.  
The country name will be displayed to the report consumer, even though the sales country code, specified in step 10, is used by PowerPrompts.
  11. Click Next and then click Finish.  
The dynamo is added to the Dynamo Manager list box.
  12. Select the Countries dynamo, click the Edit button, and click SQL.  
The Dynamo SQL dialog box appears.
  13. In the SQL box, append order by country to the existing SQL (add this text inside the closing single quotation mark).  
The SQL statement now looks like this:  

```
Dynamo.SQL= 'SELECT "Country"."SalesCountryCode", "Country"."Country" FROM "Country" order by country';
```
  14. Click OK, and then close the Dynamo Manager dialog box.

### Create the HTML Controls for the Countries Dynamo

With the dynamo created and defined, you must add the controls for the dynamo to the SelectCountry page.

**Try This**

1. Close the Dynamo Manager dialog box, or move it to the side so that you can see the application in the PowerPrompts Developer Studio workspace.
2. Right-click the SelectCountry page, and click Edit HTML.

The page opens in your HTML editor and contains the following HTML tags:

```
<html>
<head><Title><%=App.CurrentPage%></Title>
</head>
<body>
<h1><%= App.CurrentPage %></h1>

<form method="POST">
<input type="submit" value=" Next &gt; " >
</form>

</body>
</html>
```

3. Replace the `<%= App.CurrentPage %>` text between the opening and closing Title tags (`<title>...</title>`) with:

```
PowerPrompts Tutorial
```

4. Remove the text that follows the opening Body tag (`<body>`):

```
<h1><%= App.CurrentPage %></h1>
```

5. After the closing Head tag (`</head>`) and before the opening Body tag (`<body>`), insert the following text:

```
<%=include "Header.js"%>
```

This code adds the standard header.

6. After the opening Body tag (`<body>`), insert the following text:

```
<br><p><H2>Select a Country</H2></p>
<p>Choose the country for the Sales Reps you want displayed in the report, and click
Next.</p>
```

This code produces two lines of text indicating what the report consumer must do.

7. Between the opening Form tag (`<form method="POST">`) and the Input Type tag (`<input type="submit" value="Next">`), insert the following code:

```
<p><%=App.RunDynamo("Countries")%>
```

The `<%=App.RunDynamo("Countries")%>` code runs the specified dynamo.

8. After the closing Form tag (`</form>`), insert the following text:

```
<hr>EntryPage : <b>SelectCountry</b> : PickFilter : (ProductLine : ProductType) or
(TimeFilter) : RemoveColumns : ExitPage<br>
```

9. Save your changes and close the file.

**Run the Application**

You can run your PowerPrompts application now to test your dynamo. You should be able to choose a country from the SelectCountry page and choose a filter type from the PickFilter page. The remaining pages have yet to be completed.

**Try This**

1. Start your Web server.  
You can run the application only if you have a Web server installed on your computer.
2. From the Tools menu, click Run.  
If the Verify Application dialog box opens, click Save to save the application and close the dialog box.  
The application's Entry page appears in your Web browser. Follow the instructions you created in the application, and specify the information that you want to see in the report.

**Create a Dynamo to Select a Product Line**

The second dynamo in your PowerPrompts application will allow your report consumers to select a product line for the report.

The ProductLines dynamo shows the product lines available in the data source. After the dynamo is defined, you will add the HTML controls for the results of the dynamo to be shown in a list box on the ProductLine page of your PowerPrompts application.

To see what steps the Dynamo wizard automated when you created the Countries dynamo, create this dynamo manually.

### Try This

1. From the Tools menu, click Dynamo Manager.
2. In the Dynamo Manager dialog box, click the New button.  
A new dynamo is created.
3. Type ProductLines in the Name column for the new dynamo, and press Enter.
4. With the ProductLines dynamo selected, click the Edit button, and then click Data Source.
5. Click Use This Logical Database.
6. From the drop-down list, select GOS, and click OK.
7. Clear the Use This Authentication Information check box and click OK.  
You don't need to provide a user ID and password or set a record retrieval limit for this data source.
8. Click the Edit button, and then click SQL.
9. In the SQL pane of the Dynamo SQL dialog box, type the following code and click OK:

```
Dynamo.SQL = 'Select ProductLineCode, ProductLine from ProductLine order by ProductLine';
```

**Tip:** You can also drag table and column names from the Metadata pane to the SQL pane of the Dynamo SQL dialog box, or select the table and click Insert to add it to the SQL pane.

10. Click OK to close the Dynamo SQL dialog box.
11. Click the Edit button, and then click Definition.
12. In the Dynamo Definition dialog box, type the following code and click OK:

```
var sOutput = '<p><select name="ProdLine">';

while (!rs.EOF)
{
    sOutput = sOutput + "<option value='" + rs.Fields("ProductLineCode") + "'>" +
rs.Fields("ProductLine") + "</option>";
    rs.MoveNext();
}

sOutput += '</select></p>';
return sOutput;
```

### Tips

- In this dynamo, the value used in the script and the value shown to the report consumer, ProductLine, which is indicated by rs.Fields("ProductLine"), are the same.
- You can also add a description of the dynamo. Right-click the dynamo, and click Description. In the Dynamo Description dialog box, type a description of the dynamo and click OK.

To view a dynamo definition, select the dynamo in the Dynamo Manager, click the Edit button, and click Definition.

## Create the HTML Controls for the ProductLines Dynamo

You must add the controls for the ProductLines dynamo to the ProductLine page.

**Try This**

1. Close the Dynamo Manager dialog box, or move it to the side so that you can see the application in the PowerPrompts Developer Studio workspace.
2. Right-click the ProductLine page, and click Edit HTML.

The page opens in your HTML editor and contains the following HTML tags:

```
<html>
<head><Title><%=App.CurrentPage%></Title>
</head>
<body>
<h1><%= App.CurrentPage %></h1>

<form method="POST">
<input type="submit" value=" Next &gt; " >
</form>

</body>
</html>
```

3. Replace the `<%= App.CurrentPage %>` text between the opening and closing Title tags (`<title>...</title>`) with:

```
PowerPrompts Tutorial
```

4. Remove the text that follows the opening Body tag (`<body>`):

```
<h1><%= App.CurrentPage %></h1>
```

5. After the closing Head tag (`</head>`) and before the opening Body tag (`<body>`), insert the following text:

```
<%=#include "Header.js"%>
```

This code adds the standard header.

6. After the opening Body tag (`<body>`), insert the following text:

```
<br><p><H2>Restrict the Report by Product Line</H2></p>
<p>Choose a product line for your report, and click Next.</p>
```

This code produces two lines of text indicating what the report consumer must do.

7. Between the opening Form tag (`<form method="POST">`) and the Input Type tag (`<input type="submit" value="Next">`), insert the following text:

```
<%Response.Write(App.RunDynamo("ProductLines"))%>
```

8. After the closing Form tag (`</form>`), insert the following text:

```
<hr>EntryPage : SelectCountry : PickFilter : (<b>ProductLine</b> : ProductType) or
(TimeFilter) : RemoveColumns : ExitPage<br>
```

9. Save your changes and close the file.

**Create the JavaScript Code to Select a Product Type**

Instead of creating a dynamo that lets your report consumers select a product type for the report, you can write JavaScript code do to this. This code displays a list of product types depending on the product line selected by the report consumer on the previous page of the PowerPrompts application. While it is easier to add JavaScript code to a page than create a dynamo manually, JavaScript code embedded on pages cannot be reused like a dynamo can.

**Try This**

1. Right-click the ProductType page, and click Edit HTML.

The page opens in your HTML editor and contains the following HTML tags:

```
<html>
<head><Title><%=App.CurrentPage%></Title>
</head>
<body>
<h1><%= App.CurrentPage %></h1>

<form method="POST">
<input type="submit" value=" Next &gt; " >
</form>

</body>
</html>
```

2. Replace the `<%= App.CurrentPage %>` text between the opening and closing Title tags (`<title>...</title>`) with:

```
PowerPrompts Tutorial
```

3. Remove the text that follows the opening Body tag (`<body>`):

```
<h1><%= App.CurrentPage %></h1>
```

4. After the closing Head tag (`</head>`) and before the opening Body tag (`<body>`), insert the following text:

```
<%#include "Header.js"%>
```

This code adds the standard header.

5. After the opening Body tag (`<body>`), insert the following text:

```
<br><p><H2>Restrict the Report by Product Type</H2></p>
<p>Choose a product type for your report, and click Next.</p>
```

This code produces two lines of text indicating what the report consumer must do.

6. Between the opening Form tag (`<form method="POST">`) and the Input Type tag (`<input type="submit" value="Next">`), insert the following text:

```
<p><select name="ProdType">
<%
var rs = new Recordset;
rs.Open("Select ProductTypeCode, ProductType from ProductType where ProductLineCode
= " + App.Variables("ProdLine"), "GOS");
while(!rs.EOF)
{
    Response.Write("<option value='" + rs.Fields("ProductTypeCode") + "'" +
rs.Fields("ProductType") + "</option>");
    rs.MoveNext();
}
rs.Close();
%>
</select></p>
```

The ProdType value is the value that will be inserted in the scripts that are run against the report.

**Note:** When copying from the `pwp_html.txt` file, maximize your text editor (for example, Notepad) and turn off word wrapping. This should remove any line returns in string code. You can also remove them manually. Line returns in strings can prevent server-side JavaScript from working properly.

7. After the closing Form tag (`</form>`), insert the following text:

```
<hr>EntryPage : SelectCountry : PickFilter : (ProductLine : <b>ProductType</b>) or
(TimeFilter) : RemoveColumns : ExitPage<br>
```

8. Save your changes and close the file.

## Run the Application

You can run your PowerPrompts application now to test the two dynamos and the JavaScript code. From the SelectCountry page, choose a country and then, from the PickFilter page, choose filter by product line. You should be able to select a product line and a product type.

### Try This

1. Start your Web server.  
You can run the application only if you have a Web server installed on your computer.
2. From the Tools menu, click Run.  
If the Verify Application dialog box appears, click Save to save the application and close the dialog box.

## Create an HTML Table to Select a Time Period

Dynamos offer a dynamic list of options from which the report consumer can select, but you can also use static lists from which the report consumer can select a number of options. For example, you can create a drop-down list that report consumers can use to select different time periods for the report.

In this tutorial, you will create two drop-down lists containing years and months that the report consumer can use to select the years and months they want displayed in the report. These variables will be used in the scripts that you will create in the next chapter.

## Notes

- Instead of entering the HTML tags, you can copy it from the `pwp_html.txt` file that is in the `installation_location\samples\powerprompts` folder.

## Try This

1. In the PowerPrompts Developer Studio workspace, right-click the TimeFilter page, and click Edit HTML.

The page opens in your HTML editor and contains the following HTML tags:

```
<html>
<head><Title><%=App.CurrentPage%></Title>
</head>
<body>
<h1><%= App.CurrentPage %></h1>

<form method="POST">
<input type="submit" value=" Next &gt; " >
</form>

</body>
</html>
```

2. Replace the `<%= App.CurrentPage %>` text between the opening and closing Title tags (`<title>...</title>`) with:

```
PowerPrompts Tutorial
```

3. Remove the text that follows the opening Body tag (`<body>`):

```
<h1><%= App.CurrentPage %></h1>
```

4. After the closing Head tag (`</head>`) and before the opening Body tag (`<body>`), insert the following text:

```
<%=include "Header.js"%>
```

This code adds the standard header.

5. After the opening Body tag (`<body>`), insert the following text:

```
<br><p><H2>Restrict the Report by Time Period</H2></p>
<p>Select the time periods you want displayed in your report, and click Next.</p>
```

This code produces two lines of text indicating what the report consumer must do.

6. Between the opening Form tag (`<form method="POST">`) and the Input Type tag (`<input type="submit" value="Next">`), insert the following text:

```
Choose Year:<br>
<select name="OrderYear">
<option>1999</option>
<option>2000</option>
<option>2001</option>
</select>
<br>
Choose Month:<br>
<select name="OrderMonth">
<option value = "-1">[all]</option>
<option value = "1">January</option>
<option value = "2">February</option>
<option value = "3">March</option>
<option value = "4">April</option>
<option value = "5">May</option>
<option value = "6">June</option>
<option value = "7">July</option>
<option value = "8">August</option>
<option value = "9">September</option>
<option value = "10">October</option>
<option value = "11">November</option>
<option value = "12">December</option>
</select>
<br><br>
```

This code produces a drop-down list for the years, and another for the months. Report consumers can select all months by selecting the [all] option.

7. After the closing Form tag (</form>), insert the following text:

```
<hr>EntryPage : SelectCountry : PickFilter : (ProductLine : ProductType) or
(<b>TimeFilter</b>) : RemoveColumns : ExitPage<br>
```

8. Save your changes and close the file.

## Use PowerPrompts to Remove Columns from a Report

You can use a PowerPrompts application to allow report consumers to remove columns from a report. You will add a table to a report, and a check box for each column that can be removed. The input name is different for each column that you remove. You will create a prompt that allows the report consumers to remove up to four columns from the report.

You will create the scripts required to remove columns in the next chapter of this tutorial. For now, you will add only the HTML tags to the page.

### Try This

1. In the PowerPrompts Developer Studio workspace, right-click the RemoveColumns page, and click Edit HTML.

The page opens in your HTML editor and contains the following HTML tags:

```
<html>
<head><Title><%=App.CurrentPage%></Title>
</head>
<body>
<h1><%= App.CurrentPage %></h1>

<form method="POST">
<input type="submit" value=" Next &gt; " >
</form>

</body>
</html>
```

2. Replace the <%= App.CurrentPage %> text between the opening and closing Title tags (<title>...</title>) with:

```
PowerPrompts Tutorial
```

3. Remove the text that follows the opening Body tag (<body>):

```
<h1><%= App.CurrentPage %></h1>
```

4. After the closing Head tag (</head>) and before the opening Body tag (<body>), insert the following text:

```
<%#include "Header.js"%>
```

This code adds the standard header.

5. After the opening Body tag (<body>), insert the following text:

```
<br><p><H2>Remove Columns</H2></p>
<p>Clear the check boxes for the columns you want to remove from the report, and
click Next.</p>
```

This code produces two lines of text indicating what the report consumer must do.

6. Between the opening Form tag (<form method="POST">) and the Input Type tag (<input type="submit" value="Next">), insert the following text:

```
<table border="0" cellspacing="8">
<tr>
<td><input type="checkbox" checked name="FormatLastname" value="1"><font size="2"
face="Arial">Lastname</font></td>
<td><font size="2" face="Arial"><input type="checkbox" checked
name="FormatFirstname" value="1">Firstname</font></td>
</tr>
<tr>
<td><font size="2" face="Arial"><input type="checkbox" checked
name="FormatOrderdate" value="1">Orderdate</font></td>
<td><font size="2" face="Arial"><input type="checkbox" checked
name="FormatVendorname" value="1">Vendorname</font></td>
</tr>
</table>
```

7. After the closing Form tag (</form>), insert the following text:

```
<hr>EntryPage : SelectCountry : PickFilter : (ProductLine : ProductType) or  
(TimeFilter) : <b>RemoveColumns</b> : ExitPage<br>
```

8. Save your changes and close the file.

## Run the Application

You can run your PowerPrompts application now to test the dynamos, the navigational path, the HTML table, and the page to add columns.

### Try This

1. Start your Web server.  
You can run the application only if you have a Web server installed on your computer.
2. From the Tools menu, click Run.  
If the Verify Application dialog box opens, click Save to save the application and close the dialog box.  
The application's Entry page appears in your Web browser. Follow the instructions you created in the application, and specify the information that you want to see in the report.

## What's Next

In this chapter, you set the conditions for page navigation and created dynamos or adding JavaScript code so that report consumers can select a country, a product line, and a product type. You also created an HTML table to allow consumers to select time periods for the report, and the HTML controls for them to remove columns from the report.

Now you create the scripts that will be run against the report to display the report consumers' selections. The next chapter of this tutorial explains how to create the scripts for applying the report consumers selections, test the PowerPrompts application, and notify the Impromptu Web Report administrators.



---

# Chapter 4: Complete the Application

---

Report consumers select the information they want to see in a report from the pages in the PowerPrompts application. These selections are applied to the report by using a script.

After the pages are designed and the script is created, you can test the application before making it available for Impromptu Web Reports administrators. Administrators can then attach the PowerPrompts application to reports that report consumers can access.

## Create the Script to Run Against the Report

A script is the instructions that implement the selections made by the report consumers. A script is written in JavaScript code and uses objects, properties, and methods that are described in the online help.

### Script Editor Dialog Box

Use the Script Editor dialog box edit the script in your PowerPrompts application.

If there are script errors, they will only be shown in the Web browser after the application is run with a report. For more information, see ["Run the Application with a Report" \(p. 38\)](#).

For more information about creating a script, see the PowerPrompts online help.

#### Note

- Instead of typing the script code, you can copy it from the `pwp_scripts.txt` file that is in the `installation_location\samples\powerprompts` folder. Remember to maximize your text editor (for example, Notepad) and turn off word wrapping to avoid line returns in string code.

### Select a Country

This part of the script filters the report by the country selected on the SelectCountry page of the application.

#### Try This

1. Open the Script Editor dialog box.
2. Insert the following text:

```
if (App.Variables("CountryCd"))
{
    GetQuery().AndFilterBy("[\\Country\\Salescountrycode] = " +
App.Variables("CountryCd"));
}
```

The `App.Variables` method returns the value the report consumer sets for `CountryCd`. The `GetQuery()` report method returns the main query of the report. The `AndFilterBy()` query method is a detail filter that filters by the country that the report consumer selected on the `SelectCountry` page. The two parts are separated by a period.

#### Notes

- JavaScript is case sensitive; for example, `var` is not the same as `VAR`.
- You can also insert the methods by right-clicking in the Script Editor dialog box, clicking `Insert Impromptu Report Method`, and clicking the method. You must edit the values manually.
- You can copy and paste certain locations, such as `[\\Admin.\\Country\\Country Cd]` in step 2, from the `Folders` dialog box of `Impromptu Administrator`.

## Select a Year

This part of the script filters the report by the years that the report consumers select on the TimeFilter page of the application.

This part of the script is run only if the report consumer selects a year on the TimeFilter page.

### Try This

- In the Script Editor dialog box, insert the following text:

```
if (App.Variables("OrderYear"))
{
    GetQuery().AndFilterBy("year([\Orders\Orderdate]) in (" +
App.Variables("OrderYear") + ")");
}
```

The GetQuery() report method returns the main query of the report. The AndFilterBy() query method is a detail filter that filters by the years that the report consumer selected on the TimeFilter page.

## Select a Month

This part of the script filters the report by the month that the report consumers selects on the TimeFilter page of the application. If the report consumer selects all months, then the report is not filtered.

### Try This

- In the Script Editor dialog box, insert the following text:

```
if (App.Variables("OrderMonth") && App.Variables("OrderMonth") != "-1")
{
    GetQuery().AndFilterBy("month([\Orders\Orderdate]) in (" +
App.Variables("OrderMonth") + ")");
}
```

The && is the AND logical operator. The report is filtered if a value has been set for OrderMonth, and that value isn't equal to -1 (which happens when [all] months are selected).

## Select Product Line

This part of the script filters the report by the product line that the report consumers selects on the ProductLine page of the application.

### Try This

- In the Script Editor dialog box, insert the following text:

```
if (App.Variables("ProdLine"))
{
    GetQuery().AndFilterBy("[\Products\Productlinecode] = " +
App.Variables("ProdLine"));
}
```

## Select Product Type

This part of the script filters the report by the product type that the report consumers selects on the ProductType page of the application.

### Try This

- In the Script Editor dialog box, insert the following text:

```
if (App.Variables("ProdType"))
{
    GetQuery().AndFilterBy("[\Products\Producttypecode] = " +
App.Variables("ProdType"));
}
```

## Remove Columns

This part of the script filters the report by the columns that the report consumers removes on the RemoveColumns page of the application.

### Try This

- In the Script Editor dialog box, insert the following text:

```
if (App.Variables("FormatLastname") != "1")
{
    GetQuery().RemoveColumn("Lastname")
}

if (App.Variables("FormatFirstname") != "1")
{
    GetQuery().RemoveColumn("Firstname")
}

if (App.Variables("FormatOrderdate") != "1")
{
    GetQuery().RemoveColumn("Orderdate")
}

if (App.Variables("FormatVendorname") != "1")
{
    GetQuery().RemoveColumn("Vendorname")
}
```

The RemoveColumn report method removes the named column from the main query of the report.

After you complete the script, all you have left to do is to test your PowerPrompts application and then notify the Impromptu Web Reports administrators.

## Test the Application

You can test the application by viewing the script that would be applied to the report, or by running the application with a report.

For both methods, your default Web browser opens at the Entry page of the application. You can then proceed through the pages and select what you want to see in the report. If you run the application without a report, the script that would be applied to the report is shown. If you run the application with a report, the modified report appears in PDF format in your browser.

By testing the application, you can verify the navigational path through the application, and verify whether the script you have created produces a report that contains the selections that the report consumer requested.

You can test your PowerPrompts application at any stage while you are creating the application. To check the application, you must have the following software installed on your computer:

- a Web server
- a Web browser
- Adobe Acrobat Reader

## View the Generated Script

You can generate the script that would be applied to the report, by running the application without a report. This process provides a way to test your PowerPrompts application, and to debug your script.

### Try This

1. Start your Web server.  
You can run the application only if you have a Web server installed on your computer.
2. From the Tools menu, click Run.

3. Select the information you want to see in the report.  
A script, with the values you have selected, appears in your Web browser.

## Run the Application with a Report

You can run a PowerPrompts application with different reports provided that the script you created applies to the report. A sample report is included with which to test the application you created from this tutorial.

### Try This

1. Turn on your Web server.
2. From the File menu, click Properties.
3. In the General tab, click Browse.
4. From the *installation\_location*\samples\Impromptu\additional reports folder, select *pwp\_tutorial\_js.imr* and click OK.
5. From the File menu, click Save.
6. From the Tools menu, click Run.

The application's Entry page appears in your Web browser. Follow the instructions you created in the application, and specify the information that you want to see in the report. If you want to compare your application to our version, open the completed sample application in *installation\_location*\samples\powerprompts\tutorial.

### Tips

- If you do not want to be prompted to save changes before running your PowerPrompts application, from the Tools menu, click Options, and select Save Changes.

## Notify the Impromptu Web Reports Administrator

When you have completed and tested your PowerPrompts application, notify the Impromptu Web Reports administrator. When report consumers access a report with a PowerPrompts application associated with it, the application runs in the report consumers' Web browser, and they are prompted for information about what they want to see in the report.

For more information about what the report administrator needs to know after you have completed the PowerPrompts application, see the Upfront *Cognos Web Portal User Guide*.

---

# Index

---

## A

- adding
  - pages, 18
- applications
  - creating, 12
  - entry pages, 12
  - error pages, 12
  - exit pages, 12
  - running, 13, 27, 30, 33, 37
  - samples, 9
  - testing, 37
  - verifying, 37

## C

- conditions
  - HTML code for links, 23
- copyright, 2
- creating
  - applications, 12
  - code to select a product line, 29
  - drop downs to select time periods, 30
  - dynamos to select a country, 25
  - dynamos to select a product type, 27
  - script, 35, 36, 37
- customizing reports, 7

## D

- database connections
  - logical databases, 9
  - ODBC data sources, 9
- defining
  - links, 23
- deleting links, 19
- document
  - version, 2
- drawing links, 19
- Dynamo wizard, 25
- dynamos
  - creating, 25, 27
  - creating the HTML for selecting a country, 26
  - creating the HTML for selecting a product line, 28
  - using a wizard, 25

## E

- entry pages, 12
- error pages, 12
- exit pages, 12

## F

- files included with tutorial, 9

## H

- HTML
  - controls for selecting a country, 26
  - controls for selecting a product line, 28
  - for conditional links, 23
  - selecting time periods, 30

## I

- introduction, 7

## L

- links
  - adding, 19, 20
  - completing, 20
  - creating, 19, 20
  - defining, 23
  - deleting, 19
  - drawing, 19
- logical databases, 9

## N

- navigation
  - defining links, 23

## O

- ODBC data sources, 9
- other software required, 9

## P

- pages
  - adding, 18
  - navigating, 23
- process overview, 7
- product
  - version, 2

## R

- removing
  - columns from a report, 32
- reports, 38
  - removing columns, 32
- running
  - sample application, 38
- running applications, 9, 13, 27, 30, 33, 37

## S

- sample application
  - running, 38
- samples, 9
- script, 35
  - creating, 35, 36, 37

## Index

### script (*cont'd*)

- generating, [37](#)
- removing report columns, [37](#)
- selecting a country, [35](#)
- selecting months, [36](#)
- selecting product line, [36](#)
- selecting product type, [36](#)
- selecting years, [36](#)
- viewing, [37](#)

### Script Editor, [35](#)

#### selecting

- a country using a dynamo, [25](#)
  - a product line using code, [29](#)
  - a product type using a dynamo, [27](#)
  - months, [30](#)
  - years, [30](#)
- software requirements, [9](#)
- starting PowerPrompts, [11](#)

## T

### testing

- applications, [37](#)

### tutorial files

- header.js, [9](#)
  - pwp\_html.txt, [9](#)
  - pwp\_scripts.txt, [9](#)
  - pwp\_tutorial\_js.imr, [9](#)
- tutorial overview, [8](#)

## V

- verifying applications, [37](#)

### version

- product, [2](#)

## W

### wizards

- for creating dynamos, [25](#)