

# QueryVision: Report Autosizing Guide

Version 2016.1

## Table of Contents

1.	WHAT'S NEW .....	1
1.1	Qvt Version 2016.1 .....	1
2.	QUICK START .....	1
2.1	Autosizing .....	1
2.2	Adjusting SharePoint page layout for IBM Cognos reports .....	7
3.	DOCUMENT GUIDE .....	10
4.	INTRODUCTION – OPTIMIZING IBM COGNOS/SHAREPOINT INTEGRATION .....	12
5.	PROBLEM OVERVIEW .....	15
6.	SOLUTION OVERVIEW .....	16
6.1	Suggested Solution Strategy .....	16
6.2	Autosizing - enable Cognos reports for dynamic sizing in SharePoint .....	17
6.3	SharePoint Page Layout Modification Tools .....	18
7.	AUTOSIZING .....	19
7.1	Installation and Configuration .....	19
7.2	SharePoint, Cognos and Browser support .....	20
7.3	Report Viewer Web Part Properties & Autosizing .....	20
7.4	Enabling Reports for Autosizing .....	22
7.5	Specifying embedded resizing options .....	23
7.5.1	qv_addHeight - List Frame Footer/Navigation Controls "Icing" .....	26
7.5.2	qv_rowsPerPage – Specify rows per page for SharePoint .....	26
7.6	Drill up/down Behavior .....	27
7.7	Autosizing reports with multiple page definitions (including prompt pages) .....	27
7.8	Behavior of Reports with prompt pages .....	27
7.9	Drill-Through Behavior .....	28
8.	SHAREPOINT PAGE LAYOUT MODIFICATION TOOLS .....	28
8.1	SharePoint specific JavaScript page layout manipulation considerations .....	29
8.2	Example of specific techniques used .....	30
8.3	Why the Script Editor Web Part for JavaScript? .....	30
9.	APPENDIX .....	31
9.1	Troubleshooting .....	31
9.2	Select the appropriate SharePoint page layout .....	31
9.2.1	SharePoint Web Part Page templates .....	31
9.3	Leverage SharePoint Web Part Page, Web Part Zone and Web Part built-in dynamic resizing capabilities .....	32
9.3.1	Understanding and configuring web part zone and web part sizing .....	32

9.3.2	Dynamic techniques .....	34
9.4	Leveraging IBM Cognos Report height and width configuration capabilities .....	34
9.5	Creating a SharePoint page with a web part template .....	37
9.6	Pitfalls with SharePoint's "Focus on content" .....	39
9.7	Guide for Cognos Report Heights in SharePoint.....	40
9.8	Detailed Feature Specifications .....	43
9.9	Advanced Autosizing - Multi-Tabbed Dashboard via Nested Reports .....	44
9.9.1	Introduction .....	44
9.9.2	Container Report Configuration .....	44

# 1. WHAT'S NEW

## 1.1 Qvt Version 2016.1

- **IBM Cognos Analytics 11.X support** - With IBM Cognos Analytics 11.X the relative location of .js files accessed from embedded JavaScript in Reports has changed see section **7.4 Enabling Reports for Autosizing**
- **SharePoint 2016 support**

# 2. QUICK START

Recommended: See the combination of Autosizing and SharePoint Page Layout Modification Tools in the video [Introducing QueryVision Autosizing](#) on the [QueryVision YouTube channel](#)

*Note: HTML/JavaScript Code snippets and sample reports can be found here. It is HIGHLY RECOMMENDED to copy/paste from the files included in the QueryVision SharePoint Web Parts installation folder*

*Snippets provided in the text from this PDF document are for illustration purposes only.*

We have split Quick Start into two sections – applying dynamic Autosizing and Adjusting SharePoint page layout for IBM Cognos reports.

## 2.1 Autosizing

*Note: It is HIGHLY RECOMMENDED to copy/paste from the files included in the QueryVision SharePoint Web Parts installation folder:*

Autosizing Samples.zip

*The following are links latest versions of this guide [QueryVision Report Viewer Autosizing Guide](#) and the [Autosizing Samples](#) on the QueryVision web site ([queryvision.com](http://queryvision.com))*

QueryVision's Autosizing capabilities require embedding references to JavaScript (.js) files and small snippets of JavaScript code into a Report Studio Report's Specifications "using HTML Items" for interactively running the report as HTML in SharePoint. This can support:

- Fixed size – specify the height of the report in pixels
- Dynamic size – enable the report to resize to its actual height, along with a minimum and a maximum height

Configuring for Autosizing requires the following:

- **Provide .js libraries on your Cognos Server** - find the `<installLocation>\ibm\cognos\c10_64\webcontent` folder add a `lqueryvision` sub folder and include the following files from the installed QueryVision\QueryVision SharePoint Web Parts\Configuration folder:
  - ◇ `iframeResizer.contentWindow.min.js`
  - ◇ `qvtResizer.contentWindow.min.js`
- **Include JavaScript in an HTML item in your Report specification** - As a quick demonstration select or create a Cognos Report Studio report with a single page definition – a simple list report best illustrates auto-sizing. In the report header (e.g. beside the Report Title) insert an "HTML item" and include the following:

```
<script type="text/javascript">
    var QvtReportStyleProperties =
    '{"report":{"runAsHtml":{"dynamicSize":{"config":{"sizeMode":"dynamicSize",
    "qvt_addHeight":"25px"}}}}}}';
</script>

<script language="javascript" type="text/javascript"
src="../../queryvision/qvtResizer.contentWindow.min.js"></script>

<script language="javascript" type="text/javascript"
src="../../queryvision/iframeResizer.contentWindow.min.js"></script>
```

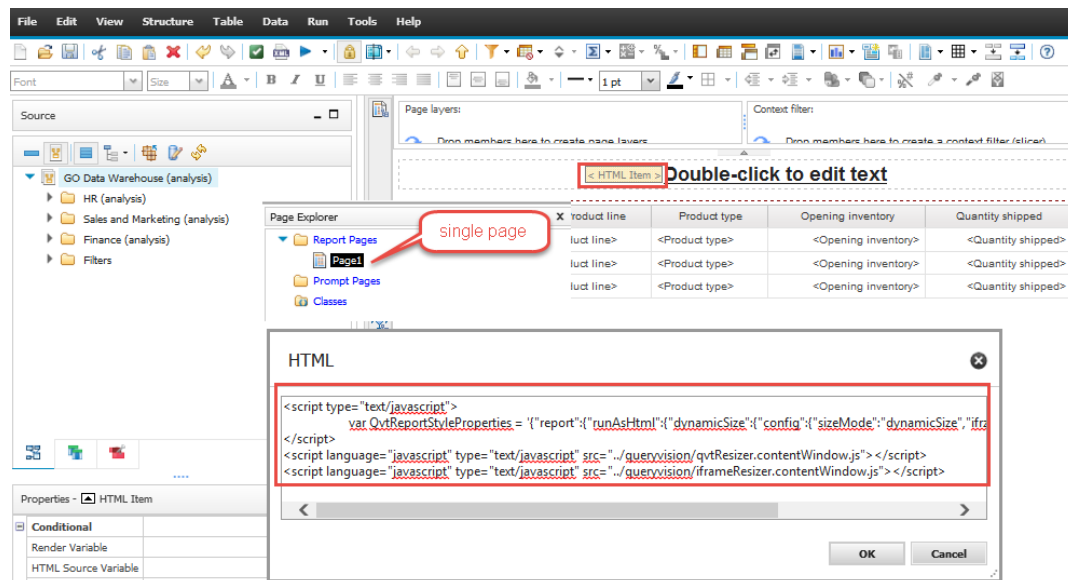
### Notes:

- ◇ the '*var QvtReportStyleProperties*' script specifies the Autosizing options and the other scripts load the necessary resizing JavaScript components.
- ◇ For **IBM Cognos Analytics 11.X** and later, change the relative location of the .js files from *../* to *../../*

For example:

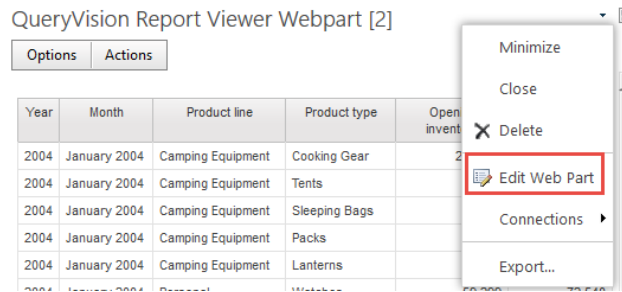
```
<script language="javascript" type="text/javascript"
src="../../queryvision/qvtResizer.contentWindow.min.js"></script>
```

Shown below is an example Cognos Report using Dynamic Autosizing using the IBM supplied sample Go Data Warehouse (analysis) Report Package and database. Note that it has a single report page definition. Also note where the HTML item is inserted in the report (in the Report header to ensure it is rendered on each page)

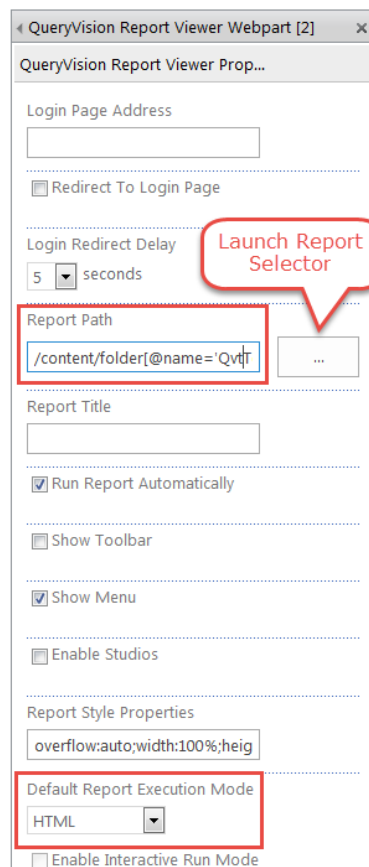


To embed this report in SharePoint requires using a QueryVision Report Viewer web part. To select the report and configure for Autosizing, do the following:

- First ensure you are logged into a Cognos Server on SharePoint via QueryVision Authentication to Cognos.
- Create or use an existing web part page, add a QueryVision Report Viewer to a web part zone, then select to "Edit web part":



- Make the following changes:
  - ◇ Set the Default Report Execution Mode to “HTML” and add either copy/paste a Cognos SearchPath to the Report Path property or click on the button beside the Report Path and navigate to and select the report.



Scroll down to the bottom of the configuration settings and select OK or Apply, then OK to apply the settings. The report will now run with the Autosize settings.

SharePoint

NEW

OneDrive

Sites

Neil J. Thomson

SHARE

FOLLOW

QueryVision Content Manager Tree

Public Folders

QATests

AnalysisKudoProductByYearRevenue

BarOverlaidCharts

DefaultNestedReport\_noAutosize

DrillableReportRatifiers

DrillableReportRatifiers\_23\_1

LongCrossTab\_2\_1

LongWideCrossTab

NestTest\_2\_28FixedSwap2\_10m\_8

NestTest\_3\_buttons

NestTest\_3\_noreports

ProductInventoryList-multipage

ProductInventoryList-multipage-3rows

ProductInventoryList-multipage\_23\_1\_FixedID

ProductInventoryList-multipage\_23\_1\_1

ProductInventoryList-multipage\_23\_1\_noautosize

ProductInventoryList-multipage\_23\_1\_singleItem

ProductLineItems\_sube\_23\_1

ProductLineTarget\_sube\_23\_1

ProductLineTarget\_maniprompt\_23\_1

ProductLineTarget\_maniprompt\_23\_2

PromptPagePromptReport

DrillThrough

ObjTypeTests

Output Tests

PleaseWait

ReportLayout

WaitForFilters

Samples

Samples\_DrillThrough

Samples\_PowerPlay

w28/zen2008 - Cg 10.2.2 RTH

My Folders

QueryVision Report Viewer Webpart

Options Actions

Year	Month	Product line	Product type	Opening inventory	Quantity shipped
2010	January 2010	Camping Equipment	Cooking Gear	236,107	234,034
2010	January 2010	Camping Equipment	Tents	51,253	53,982
2010	January 2010	Camping Equipment	Sleeping Bags	54,218	53,738
2010	January 2010	Camping Equipment	Packs	47,380	46,851
2010	January 2010	Camping Equipment	Lanterns	82,036	86,096
2010	January 2010	Personal Accessories	Watches	56,299	72,540
2010	January 2010	Personal Accessories	Eyewear	162,737	205,101
2010	January 2010	Personal Accessories	Knives	140,107	139,747
2010	January 2010	Personal Accessories	Shoulders	18,204	18,978
2010	January 2010	Personal Accessories	Navigation	33,879	36,034
2010	January 2010	Outdoor Protection	Insect Repellents	251,413	244,385
2010	January 2010	Outdoor Protection	Sunscreen	197,188	199,805
2010	January 2010	Outdoor Protection	First Aid	35,128	35,487
2010	January 2010	Golf Equipment	Irons	6,562	6,639
2010	January 2010	Golf Equipment	Woods	4,805	4,542
2010	January 2010	Golf Equipment	Putters	25,247	25,267
2010	January 2010	Golf Equipment	Golf Accessories	45,533	47,460
2010	February 2010	Camping Equipment	Cooking Gear	585,040	239,527
2010	February 2010	Camping Equipment	Tents	136,819	59,299
2010	February 2010	Camping Equipment	Sleeping Bags	111,647	54,492
2010	February 2010	Camping Equipment	Packs	83,374	46,105
2010	February 2010	Camping Equipment	Lanterns	196,317	89,382
2010	February 2010	Personal Accessories	Watches	50,145	70,492
2010	February 2010	Personal Accessories	Eyewear	27,307	300,382
2010	February 2010	Personal Accessories	Knives	404,335	134,085
2010	February 2010	Personal Accessories	Shoulders	36,116	18,420
2010	February 2010	Personal Accessories	Navigation	87,795	42,277
2010	February 2010	Outdoor Protection	Insect Repellents	431,141	248,008
2010	February 2010	Outdoor Protection	Sunscreen	470,778	197,070

10/14/10

6:40 AM - 10:00 AM

PH: John W. Boudreau

6:40 AM

10/14/10

Top

Page up

Page down

Bottom

The above report uses the default configuration a fixed height of 700px. Note the vertical scroll bar on the report

Report Autosizing Guide V1.1

Page 4

SharePoint

Home | Search | Site | Help | Tools | Settings

QueryVision Content Manager Tree

QueryVision Report Viewer Webpart

Options | Actions

Year	Month	Productline	Product type	Opening inventory	Quantity shipped
2010	January 2010	Camping Cookware	Cooking Gear	225,137	224,834
2010	January 2010	Camping Cookware	Fields	57,133	53,862
2010	January 2010	Camping Cookware	Camping Stages	3,424	3,213
2010	January 2010	Camping Cookware	Tables	47,136	45,831
2010	January 2010	Camping Cookware	Lumbers	62,238	60,858
2010	January 2010	Personal Accessories	Visitors	1,646	1,646
2010	January 2010	Personal Accessories	Accessories	162,127	289,191
2010	January 2010	Personal Accessories	Visitors	162,127	126,747
2010	January 2010	Personal Accessories	Accessories	15,234	13,878
2010	January 2010	Personal Accessories	Navigation	3,616	3,616
2010	January 2010	Outdoor Protection	Inserts/Replacements	22,141	244,245
2010	January 2010	Outdoor Protection	Compasses	187,136	189,825
2010	January 2010	Outdoor Protection	Field Kits	25,136	25,457
2010	January 2010	Multi-Environment	Tools	3,642	3,642
2010	January 2010	Multi-Environment	Visitors	6,025	4,542
2010	January 2010	Gift Equipment	Tables	22,147	22,247
2010	January 2010	Gift Equipment	Gift Accessories	43,232	47,432
2010	February 2010	Camping Cookware	Cooking Gear	363,146	359,827
2010	February 2010	Camping Cookware	Fields	123,116	59,249
2010	February 2010	Camping Cookware	Camping Stages	11,147	54,452
2010	February 2010	Camping Cookware	Tables	62,214	45,828
2010	February 2010	Camping Cookware	Lumbers	76,311	89,822
2010	February 2010	Personal Accessories	Visitors	2,145	70,492
2010	February 2010	Personal Accessories	Accessories	27,227	282,232
2010	February 2010	Personal Accessories	Visitors	42,228	124,888
2010	February 2010	Personal Accessories	Accessories	16,176	16,422
2010	February 2010	Personal Accessories	Navigation	6,146	42,217
2010	February 2010	Outdoor Protection	Inserts/Replacements	42,141	240,828
2010	February 2010	Outdoor Protection	Compasses	47,216	187,273
2010	February 2010	Outdoor Protection	Field Kits	3,144	30,248
2010	February 2010	Multi-Environment	Tools	15,222	6,414
2010	February 2010	Gift Equipment	Visitors	4,276	4,588
2010	February 2010	Gift Equipment	Tables	46,244	35,422
2010	February 2010	Gift Accessories	Gift Accessories	5,642	40,214
2010	March 2010	Camping Cookware	Cooking Gear	363,126	359,492
2010	March 2010	Camping Cookware	Fields	123,113	55,708
2010	March 2010	Camping Cookware	Camping Stages	12,136	48,118
2010	March 2010	Camping Cookware	Tables	62,117	42,548
2010	March 2010	Camping Cookware	Lumbers	23,127	80,822
2010	March 2010	Personal Accessories	Visitors	72,127	110,518
2010	March 2010	Personal Accessories	Accessories	23,146	276,118
2010	March 2010	Personal Accessories	Visitors	21,126	129,622
2010	March 2010	Personal Accessories	Accessories	15,646	70,249
2010	March 2010	Personal Accessories	Navigation	75,146	25,117
2010	March 2010	Outdoor Protection	Inserts/Replacements	42,144	245,788
2010	March 2010	Outdoor Protection	Compasses	44,621	174,522
2010	March 2010	Outdoor Protection	Field Kits	6,121	24,622
2010	March 2010	Gift Equipment	Tools	15,586	5,622
2010	March 2010	Gift Equipment	Visitors	3,232	4,171
2010	March 2010	Multi-Environment	Tables	60,246	37,868

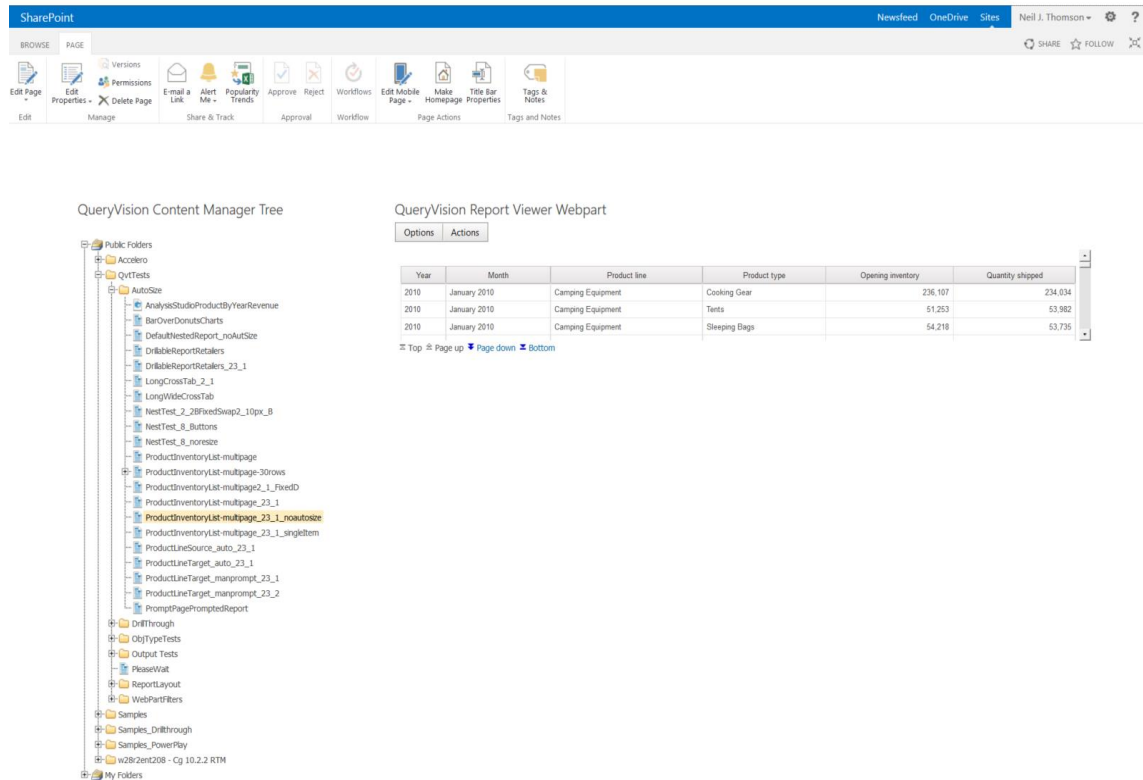
10 Oct 10 1

Top Page up Page down Bottom

1:42:05 PM

The above report has had autosizing applied. Note that the report is taller on the page and lack of a vertical scroll bar on the report.

**Setting height:100% doesn't do what you would hois not an alternative** - The following is what you get when 'height:100%' is included in the Report Style Properties string (vs. using QueryVision Autosizing or a fixed height of 700px). What you're seeing is a minimal default height setting (e.g. 200 px or less).



QueryVision Content Manager Tree

- Public Folders
  - Accelero
  - QvtTests
    - AutoSize
    - AnalysisStudioProductByYearRevenue
    - BarOverDontCharts
    - DefaultWebReport\_nosize
    - DrillableReportBakers
    - DrillableReportBakers\_23\_1
    - LongCrossTab\_2\_1
    - LongWideCrossTab
    - NestTest\_2\_2FixedSwap2\_10px\_8
    - NestTest\_8\_Buttons
    - NestTest\_8\_nosize
    - ProductInventoryList-multipage
    - ProductInventoryList-multipage-30rows
    - ProductInventoryList-multipage\_23\_1\_FixedID
    - ProductInventoryList-multipage\_23\_1
    - ProductInventoryList-multipage\_23\_1\_nosize
    - ProductInventoryList-multipage\_23\_1\_singleItem
    - ProductLineSource\_auto\_23\_1
    - ProductLineTarget\_auto\_23\_1
    - ProductLineTarget\_manprompt\_23\_1
    - ProductLineTarget\_manprompt\_23\_2
    - PromptPagePromptedReport
  - Drillthrough
  - CityTypeTests
  - Output Tests
  - PleaseVisit
  - ReportLayout
  - WebPartFilters
- Samples
  - Samples\_Drillthrough
  - Samples\_PowerPlay
  - v28/2en208 - Cg 10.2.2 RTM
- My Folders

QueryVision Report Viewer Webpart

Options Actions

Year	Month	Product line	Product type	Opening inventory	Quantity shipped
2010	January 2010	Camping Equipment	Cooking Gear	236,107	234,034
2010	January 2010	Camping Equipment	Tents	51,253	53,982
2010	January 2010	Camping Equipment	Sleeping Bags	54,218	53,735

Top Page up Page down Bottom

**Note:** In general setting HTML height in percent is only effective on within the web page it is included. If the page is embedded in an iframe, particularly in the case (such as SharePoint host page/Cognos iframe scenario) where the host page and iframe page are in different domains, then communication on height does not take place, so 100% loses its context.



## 2.2 Adjusting SharePoint page layout for IBM Cognos reports

This Quick Start shows an example of modifying a SharePoint Web page for IBM Cognos reports via JavaScript. It assumes the following:

- The SharePoint site is based on the “Team Site” template and uses the default master page
- The content page was generated from a SharePoint Web Part Page template

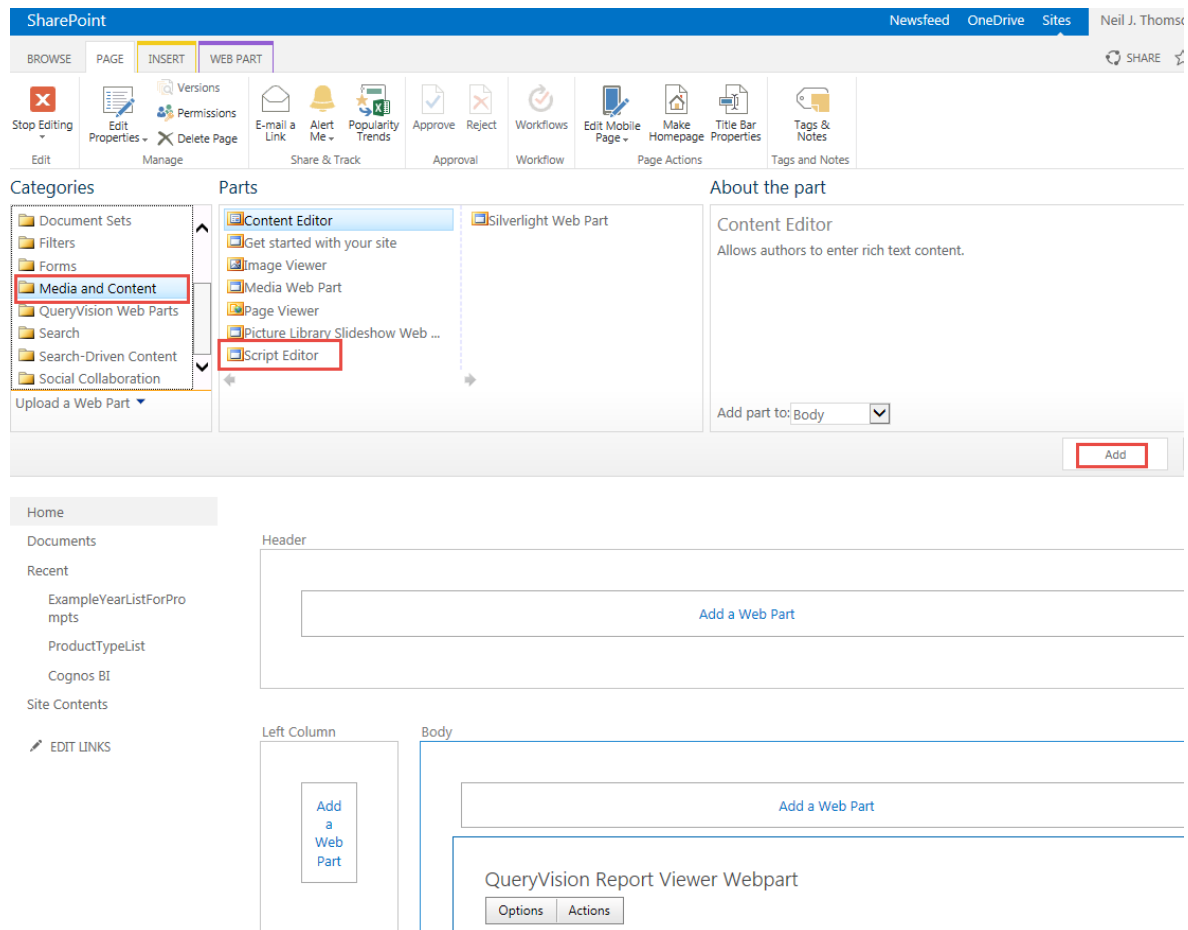
The JavaScript, HTML and CSS provides the following:

- Show and hide of the SharePoint master page side navigation panel (sideNavBox)
  - ◇ Provides the user with an HTML button to toggle show/hide for the side nav panel
  - ◇ Initially hide the side nav panel
  - ◇ Changes the main borders/margins for the web parts and web part zones to 0 pixels

Steps:

This assumes that a QueryVision Report Viewer web has been added to the page and configured to run a report. Autosizing is not required. Including a QueryVision Content Manager Navigation tree Web Part is optional.

- Select to edit the page and add a Script Editor Web part to either the header web part zone or in the “Add a Web Part” area immediately above the QueryVision Report Viewer



- The Script Editor web part should then appear in the selected web part zone with the label "Script Editor". Click on the "Edit Snippet"
- Compose the following using any text or programming editor using the *SharePoint web page layout mods snippets.htm* file as a starting point. This can be done by removing HTML (the other button definitions) and JavaScript (initialization and calls to other show/hide functions) as required:

```
// this adds a single button to show/hide the side nav box

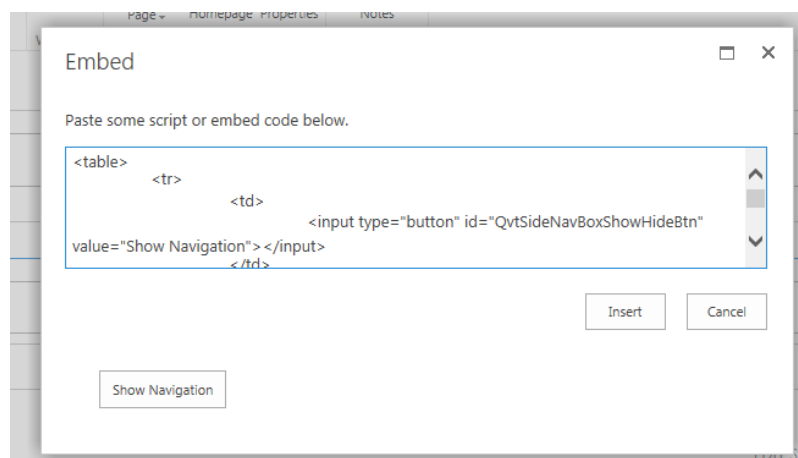
<table>
> <tr>
> > <td>
> > <input type="button" id="QvtSideNavBoxShowHideBtn" value="Show Navigation"></input>
> > </td>
> </tr>
> </table>

<style type="text/css">
> .qvtS4TitlerowHide{ display: none !important;}
> .ms-webpartPage-root {
> .border-spacing: 0px !important;
> }
> .ms-webpartzone-cell {
> .margin: 0px !important;
> }
> </style>

<script type="text/javascript" src="/wpresources/Queryvision.Webparts/js/qvtShrPntWidthUtils.js"></script>

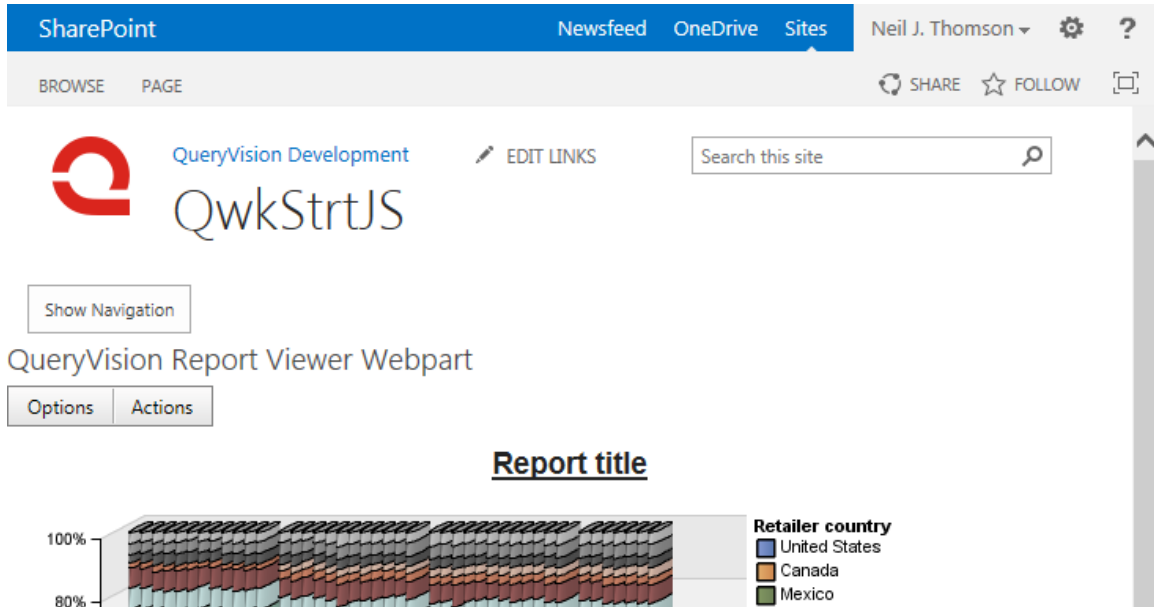
<script language="javascript">
> .spBodyOnLoadFunctionNames.push("qvtSideNavBoxShowHide");
>
> function qvtSideNavBoxShowHide()
> {
> qvtShrPntWidthUtils.sideNavBoxShowHide({
> .showHideBtnID: "QvtSideNavBoxShowHideBtn", // optional, default: no button
> .initialHide: true, // suggested. default: false
> .logging: true // optional, default: false
> .cbMargins: // optional default: '0px 0px 0px 0px'
> });
> };
> </script>
```

- You should now see the following in the Script Editor, which includes rendering the "Show Navigation" button

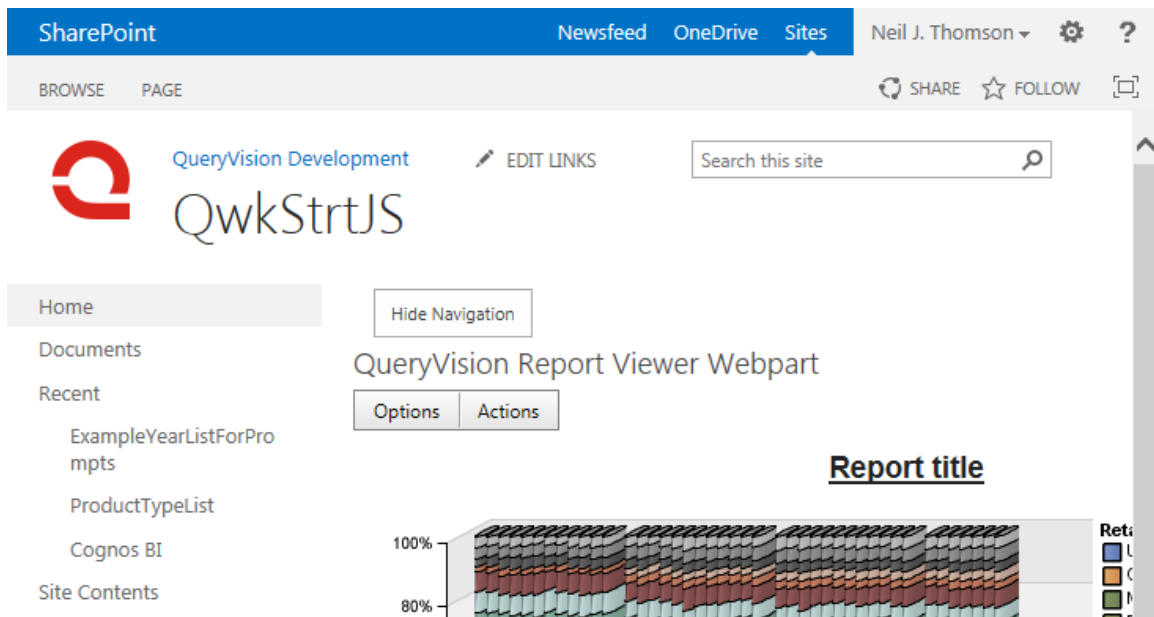


- Click on Stop editing. You should now see the following:

- ◇ The whitespace due to margins and borders have been reduced/removed (set to "0"/zero)
- ◇ The master page side nav box is hidden
- ◇ The Show Navigation button is visible



- Clicking the Show Navigation button should show the side nav panel and change the button label to "Hide Navigation".



### 3. DOCUMENT GUIDE

To help you find the information you are looking for here is a quick description of each section of this document:

- **Quick start** – step by step: Enable and display an IBM Cognos report with Autosizing. Modify the SharePoint page with JavaScript to show and hide the master page side navigation panel
- **Introduction** – Why you should strongly consider using QueryVision's Autosizing and SharePoint Page layout tools
- **Problem Overview** – why embedding IBM Cognos content in SharePoint pages can be difficult
- **Solution Overview** – High level description the QueryVision solution and a recommended strategy on optimizing height and width for SharePoint pages for IBM Cognos content
- **Autosizing**
  - ◇ **Installation and Configuration** –installation and configuration for Cognos Servers.SharePoint installation of Autosizing related components is via the web part installation process.
  - ◇ **SharePoint, Cognos and Browser support** – SharePoint 2013 vs. Cognos Versions vs. Browser brand and version compatibility and limitations
  - ◇ **Report Viewer Web Part Properties & Autosizing** – configuring QueryVision's Report Viewer web part for Autosizing, including recommended settings for web part width and height settings
  - ◇ **Specifying Resizing Options in a Cognos Report** – Autosizing requires embedding JavaScript into Cognos reports via Report Studio "html items". The available Autosizing options and how to format them in a Cognos report
  - ◇ **Drill up/down behavior** – Autosizing support for drill up/down behavior
  - ◇ **Autosizing reports with multiple page definitions** – how to enable Autosizing for reports with one or more prompt pages and/or report page definitions
  - ◇ **Drill-through behavior** – how to enable both source and target reports for Autosizing
- **SharePoint Page Layout Modification Tools**
  - ◇ **SharePoint specific JavaScript page layout manipulation considerations** – some of the key idiosyncrasies of SharePoint generated web pages
  - ◇ **Example of specific techniques used** – some examples of how QueryVision's solution deals with SharePoint specific issues
  - ◇ **Why the Script Editor Web Part?** – why you should not use the Content Editor or other Web Parts.
- **Appendices**
  - ◇ **Troubleshooting**
  - ◇ **Select the appropriate SharePoint page layout**
  - ◇ **Leverage SharePoint built in dynamic sizing capabilities**
  - ◇ **Creating a SharePoint page from a Web Part Page template**
  - ◇ **Pitfalls of Share Points "Focus on content"**

- ◇ **Guide for Cognos Report Heights**
- ◇ **Detailed Feature Specifications** – details on what Autosizing can and cannot do
- ◇ **Advanced Autosizing** – Multi-tabbed Dashboard via Nested Reports – how to use Cognos reports as HTML containers for presenting multiple reports using html controls (tabs, buttons, etc.) for report selection, plus, how to configure this technique for Autosizing

## 4. INTRODUCTION – OPTIMIZING IBM COGNOS/SHAREPOINT INTEGRATION

QueryVision's commitment has always been to make embedding of IBM Cognos content into SharePoint - simpler, faster and easier.

With QueryVision's Autosizing solution suite, we take that a step further, introducing dynamic resizing of IBM Cognos Reports and application page layout modification techniques that dramatically reduce the time and cost to provide seamless integration in SharePoint

The following were the key issues that were identified:

- Customers were increasingly using IBM Cognos content that would dynamically change the height required within the SharePoint page due to user actions, including dynamically changing reports, drill down and drill through. This was causing vertical scroll bars on many reports, which could not be addressed by out-of-the-box SharePoint or Cognos techniques.
- Many IBM Cognos reports were wider than the SharePoint page layout could support with standard configuration options - resulting in horizontal scroll bars.
- This was resulting in expensive solutions – including writing custom Cognos Reports for use in SharePoint or modifying SharePoint pages, Master pages and templates.

The requirement was to:

- Slash the development time
- Avoid custom Cognos Reports for SharePoint
- Avoid changing SharePoint Master Pages and templates or use of SharePoint Designer
- Leverage the existing skills of Cognos and SharePoint application developers

The solution includes a suite of components and techniques which can be mixed and matched as required

- Autosizing - Enable Cognos reports for dynamic and static sizing options in SharePoint
- SharePoint Page Layout Modification Tools - Adjust space on the SharePoint page for Cognos Reports

For an overview and demonstration see the video on the [QueryVision YouTube channel](#)

## Autosizing Samples

Here are two examples of IBM Cognos/SharePoint integration – before and after applying Autosizing :

### Single wide & tall report

SharePoint

Newsfeed OneDrive Sites Neil J. Thomson

BROWSE PAGE

QueryVision Development EDIT LINKS

WithoutAutozize

Home

Documents

Recent

ExampleYearListForPro

mpis

ProductTypeList

Cognos BI

Site Contents

EDIT LINKS

QueryVision Report Viewer Webpart

Options Actions

**Product Line (Source) - without Autosizing**

Product line	Product type	Product	Product number	Product key	Introduction date	Product brand code	Product brand key	Product brand	Product size	Campaign	Revenue	Gross margin	Gross profit	Unit cost	Unit price	Product cost	Quantity	Planned revenue
Campagna Equipment	Cooking Gear	TrailChef Canteen	2110	30002	15-Feb-01 12:00:00 AM	701	701	TrailChef	2 liters	Regular sale				7.22	12.88	5,171,962.32		9,162,592.24
		TrailChef Canteen	2110	30002	15-Feb-01 12:00:00 AM	701	701	TrailChef	2 liters	TrailChef Campaign	2,452,413.29	87.6962	1,016,470.83	5.56	13.13	1,435,042.46	256,867	3,360,361.99
		TrailChef Cook Set	5110	30005	15-Feb-01 12:00:00 AM	701	701	TrailChef	6-piece	Regular sale				34.97	54.93	27,283,504.00		42,850,386
		TrailChef Cook Set	5110	30005	15-Feb-01 12:00:00 AM	701	701	TrailChef	6-piece	Rising Star Campaign	0	0	-1,021,880.21	31.31	54.93	1,021,880.21	33,580	1,844,549.4
		TrailChef Cup	4110	30004	15-Feb-01 12:00:00 AM	701	701	TrailChef	330 ml	Regular sale				0.85	3.66	1,096,302.91		4,599,949.74
		TrailChef Cup	4110	30004	15-Feb-01 12:00:00 AM	701	701	TrailChef	330 ml	TrailChef Campaign	1,161,564.8	99.4032	712,778.4	0.85	3.66	448,786.40	527,884	1,932,421.44
		TrailChef Deluxe Cook Set	6110	30006	5-Mar-03 12:00:00 AM	701	701	TrailChef	6-piece	Regular sale				81.03	126.72	35,122,863.85		55,410,867.6
		TrailChef Deluxe Cook Set	6110	30006	5-Mar-03 12:00:00 AM	701	701	TrailChef	6-piece	Rising Star Campaign	0	0	-404,921.99	55.69	126.72	404,921.99	7,271	943,164.12
		TrailChef Double Flame	8110	30008	5-Mar-03 12:00:00 AM	701	701	TrailChef	Double-burner	Regular sale				75.00	151.77	16,454,175.00		33,295,686.53
		TrailChef Double Flame	8110	30008	5-Mar-03 12:00:00 AM	701	701	TrailChef	Double-burner	Rising Star Campaign	1,642,035.11	60.6087	564,510.11	75.00	151.77	1,077,525.00	14,367	2,180,479.59
		TrailChef Double Flame	8110	30008	5-Mar-03 12:00:00 AM	701	701	TrailChef	Double-burner	TrailChef Campaign	1,253,950.72	14.705	368,725.72	75.00	151.77	885,225.00	11,803	1,791,341.31
		TrailChef Kettle	9110	30009	5-Mar-03 12:00:00 AM	701	701	TrailChef	1.5 liters	Regular sale				5.28	13.22	7,746,232.06		19,423,101.62
		TrailChef Kettle	9110	30009	5-Mar-03 12:00:00 AM	701	701	TrailChef	1.5 liters	TrailChef Campaign	6,661,000.97	111.0066	3,253,808.67	4.23	13.22	3,027,262.10	867,729	11,471,377.36
		TrailChef Kitchen Kit	3110	30003	15-Feb-01 12:00:00 AM	701	701	TrailChef	15-piece	Regular sale				15.93	23.80	12,042,761.40		17,962,324

Top Page up Page down Bottom

### Before Autosizing

SharePoint

Newsfeed OneDrive Sites Neil J. Thomson

BROWSE PAGE

QueryVision Report Viewer Webpart

Options Actions

**Product Line (Source) - with Autosizing & rowsPerPage**

Product line	Product type	Product	Product number	Product key	Introduction date	Product brand code	Product brand key	Product brand	Product size	Campaign	Revenue	Gross margin	Gross profit	Unit cost	Unit price	Product cost	Quantity	Planned revenue
Campagna Equipment	Cooking Gear	TrailChef Canteen	2110	30002	15-Feb-01 12:00:00 AM	701	701	TrailChef	2 liters	Regular sale				7.22	12.88	5,171,962.32		9,162,592.24
		TrailChef Canteen	2110	30002	15-Feb-01 12:00:00 AM	701	701	TrailChef	2 liters	TrailChef Campaign	2,452,413.29	87.6962	1,016,470.83	5.56	13.13	1,435,042.46	256,867	3,360,361.99
		TrailChef Cook Set	5110	30005	15-Feb-01 12:00:00 AM	701	701	TrailChef	6-piece	Regular sale				34.97	54.93	27,283,504.00		42,850,386
		TrailChef Cook Set	5110	30005	15-Feb-01 12:00:00 AM	701	701	TrailChef	6-piece	Rising Star Campaign	0	0	-1,021,880.21	31.31	54.93	1,021,880.21	33,580	1,844,549.4
		TrailChef Cup	4110	30004	15-Feb-01 12:00:00 AM	701	701	TrailChef	330 ml	Regular sale				0.85	3.66	1,096,302.91		4,599,949.74
		TrailChef Cup	4110	30004	15-Feb-01 12:00:00 AM	701	701	TrailChef	330 ml	TrailChef Campaign	1,161,564.8	99.4032	712,778.4	0.85	3.66	448,786.40	527,884	1,932,421.44
		TrailChef Deluxe Cook Set	6110	30006	5-Mar-03 12:00:00 AM	701	701	TrailChef	6-piece	Regular sale				81.03	126.72	35,122,863.85		55,410,867.6
		TrailChef Deluxe Cook Set	6110	30006	5-Mar-03 12:00:00 AM	701	701	TrailChef	6-piece	Rising Star Campaign	0	0	-404,921.99	55.69	126.72	404,921.99	7,271	943,164.12
		TrailChef Double Flame	8110	30008	5-Mar-03 12:00:00 AM	701	701	TrailChef	Double-burner	Regular sale				75.00	151.77	16,454,175.00		33,295,686.53
		TrailChef Double Flame	8110	30008	5-Mar-03 12:00:00 AM	701	701	TrailChef	Double-burner	Rising Star Campaign	1,642,035.11	60.6087	564,510.11	75.00	151.77	1,077,525.00	14,367	2,180,479.59
		TrailChef Double Flame	8110	30008	5-Mar-03 12:00:00 AM	701	701	TrailChef	Double-burner	TrailChef Campaign	1,253,950.72	14.705	368,725.72	75.00	151.77	885,225.00	11,803	1,791,341.31
		TrailChef Kettle	9110	30009	5-Mar-03 12:00:00 AM	701	701	TrailChef	1.5 liters	Regular sale				5.28	13.22	7,746,232.06		19,423,101.62
		TrailChef Kettle	9110	30009	5-Mar-03 12:00:00 AM	701	701	TrailChef	1.5 liters	TrailChef Campaign	6,661,000.97	111.0066	3,253,808.67	4.23	13.22	3,027,262.10	867,729	11,471,377.36
		TrailChef Kitchen Kit	3110	30003	15-Feb-01 12:00:00 AM	701	701	TrailChef	15-piece	Regular sale				15.93	23.80	12,042,761.40		17,962,324

30-Nov-15

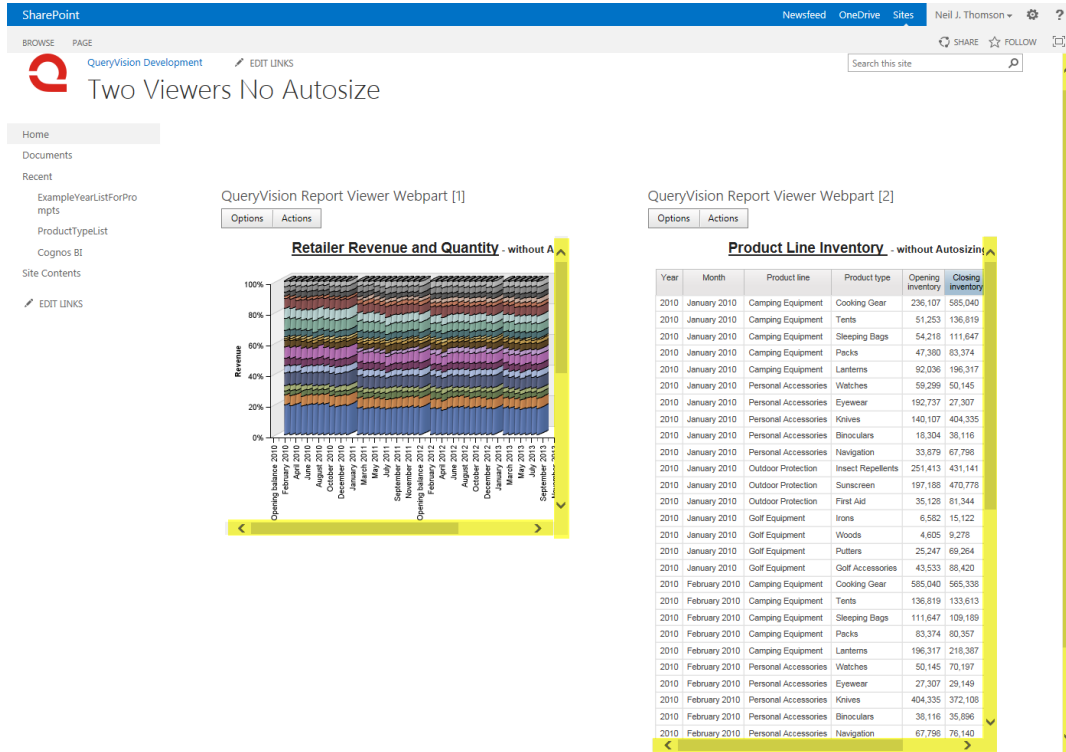
Top Page up Page down Bottom

1

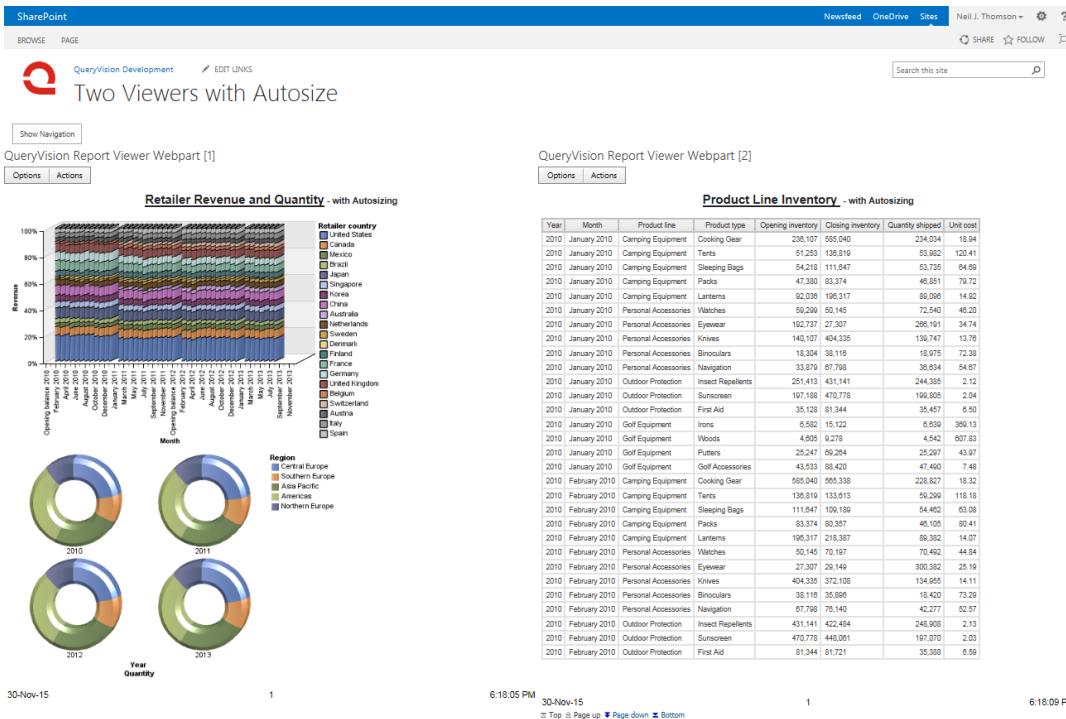
6:20:15 PM

### After Autosizing

## Dual Reports



Before Autosizing



After Autosizing



## 5. PROBLEM OVERVIEW

### ***IBM Cognos/SharePoint integration sizing issues***

Cognos reports that are embedded in SharePoint are frequently wider and taller than the SharePoint pages were designed for, particularly if the Cognos Reports and the SharePoint application pages were not explicitly designed for each other.

The tools in Report Studio to control the width and height are limited. Column, row, list and cross tab width and height can be configured to be fixed or dynamic. Using these settings, width can set to expand and contract in a “responsive” manner, but that is still insufficient in many cases. Height can only be controlled with rows per page, which is not configurable out-of-the-box from SharePoint.

SharePoint width can be configured to 100% for web parts and web part zones, but out-of-the-box, the height can only be set to fixed values. As Cognos height can vary based on the report, the data/filtering, drill-down/up or drill-through this frequently results in scroll bars for the report within the SharePoint page.

In many cases, the only way to get sufficient width in SharePoint for a wide Cognos report is to either change the Master or content pages or use JavaScript to dynamically hide some or all of SharePoint page elements above or beside an embedded Cognos report. However, SharePoint pages generated by Master Pages and some content template pages are notoriously difficult to manipulate reliably with JavaScript/CSS.

And finally, embedding Cognos reports in SharePoint involves the use of iframes, which lacks the mechanism to allow space on the parent/containing SharePoint page to be adjusted for the space required by the child page/Cognos content viewer without additional technology.

## 6. SOLUTION OVERVIEW

At a high level, Autosizing provides the following tools:

- Autosizing Tools

Individual Cognos Report Studio reports can be configured with embedded JavaScript to modify height and width behaviour when rendered in SharePoint:

- Number of rows per page

- Real-time dynamic or static sizing

Changes to Report specifications have been designed and tested to be transparent when running/viewing within Cognos Connection or use by other 3<sup>rd</sup> party applications.

- SharePoint Page Layout Modification Tools

We provide an open source JavaScript library and application techniques to use within individual SharePoint pages to optimize the space available for embedded Cognos content.

### 6.1 Suggested Solution Strategy

Before we start expanding on using these tools and techniques, we'd like to recommend some strategies:

- Think dynamic height/fixed width

SharePoint and Cognos content both tend to be in table/column format, where width is relatively fixed, but height can vary. For example it is easier to work with (and a best practice) a page that scrolls vertically for additional content vs. horizontally.

So when laying out a SharePoint page to accept Cognos content, design it to accommodate the width of the widest SharePoint and Cognos report that will appear beside each other, while allowing for height to adjust dynamically.

- Avoid SharePoint content to the left or right of a Report

If the width of Cognos reports that could be displayed is not known (e.g. users can interactively select from a list of reports which may be constantly added to or updated), then avoid placing other content beside a Cognos Report

- Hide Report selection when not in use

If using either QueryVision's Content Manager Navigation Web part or a custom SharePoint list (or other control) to select reports shown on the page, put the selection control to the left (or right) of the report viewer and use the SharePoint Page Layout Modification Tools to show/hide the selection control web part/web part zone when the user is not actively selecting a report.

- Hide the Master Page header for tall reports

Desktop and laptop screen layouts 16 to 9 (landscape) ratio, so for portrait Cognos content, we suggest using the SharePoint Page Layout Modification Tools to permanently or selectively hide the Master Page header or other content above the report (e.g. other web parts or web part zone content)

- Dedicate SharePoint pages for viewing large reports

Where SharePoint pages can't be rearranged for Cognos reports, provide a link - or a link & a small summary Cognos report/chart – to launch a SharePoint page dedicated to Cognos reports.

This provides a page that maximizes the space for Cognos content through hiding parts of the page, but allows the user to show the report selector or site navigation panel as required

A suggested layout for the page would be:

- ◇ Use a SharePoint Web Part page template with side by side web part zones.
- ◇ Place a report selector web part – either QueryVision Content Manager Navigation web part or a SharePoint list of reports – in the left web part zone
- ◇ Place a QueryVision Report Viewer web part in the web part zone to the right of the report selector web part. Make sure this web part zone has width = 100%.
- ◇ Add script to the page with the SharePoint Page Layout Modification Tools:
  - ◆ Hide the Master Page header and side nav bar/panel/box (initial hide = true), with a button to allow the side nav panel to be shown/hidden.
  - ◆ Add a button to show/hide the web part zone with the report selector. If there is a default report, then set the initial hide = true, otherwise set to false.

## 6.2 Autosizing - enable Cognos reports for dynamic sizing in SharePoint

### *Background*

As JavaScript iframe support allows for a parent page (e.g. SharePoint) and a child/iframe page (e.g. Cognos report viewer) to access each other's page elements, conceptually it is possible to add JavaScript to have the child tell the parent what height and width the child's iframe should be set to.

However, current browsers will block this communication ("same origin policy") as the SharePoint and Cognos servers are in different domains.

Neither Cognos reports nor native JavaScript provides a way to determine the actual rendered height of the Cognos page. This is complicated by the fact the Cognos HTML Report rendering also does some dynamic adjustment of where to place such items as the page footer and page navigation controls (start/previous/next/end) for multipage reports (e.g. list and cross tab reports).

### *Solution*

QueryVision has adapted and extended Dynamic sizing native JavaScript libraries to allow the QueryVision Report Viewer web part and Cognos reports to exchange information to provide real-time dynamic height and width adjustments within SharePoint pages.

The major benefit of dynamic sizing is real-time height adjustments, with additional techniques (see *SharePoint Page Layout Modification Tools*, below) included to provide additional height and width optimization.

Dynamic sizing is optional and backwards compatible. No changes are required to existing QueryVision Report Viewer web parts or Cognos reports.

If a IBM Cognos report is not enabled for Autosizing, then the QueryVision Report Viewer "Report Style Properties" parameter "height" value is used (default: 700px).

The solution requires including JavaScript within IBM Cognos reports, which requires authoring in Report Studio to insert HTML elements to contain the scripts. Other Cognos report/content types can be supported by converting them to Report Studio reports.

Dynamic sizing only applies to running reports in HTML mode. It does not apply to running reports in pdf or other output formats. It does not apply to displaying saved output, including the "default output" option (equivalent to the Cognos Connection "default action" value "view most recent report") which will use a saved output if available or run the report live otherwise.

Adding dynamic sizing JavaScript to Cognos Report specifications does not impact any aspect of using reports in Cognos Connection or with other 3<sup>rd</sup> party applications.

The in-report scripts include referencing the required JavaScript files/libraries, plus a Javascript defined JSON block which specifies the dynamic sizing options. The key options are

- Dynamic height
  - ◇ Min and maximum height (pixels)
  - ◇ Rows per page (where applicable)
  - ◇ Additional height (pixels) for the page footer/page navigation controls for multi-page reports
- Static height
  - ◇ height in pixels
  - ◇ Rows per page (where applicable)

For more information see the following:

<< pointers to specific sections in the document for autosizing>>

## 6.3 SharePoint Page Layout Modification Tools

The annotated screen shot, below, shows a Cognos Report embedded in a typical SharePoint page. The SharePoint Site was generated from the "Team Site" template, and uses the default Team Site master page. This particular page was generated by selecting to create a web part page using one of the built in web part page templates.

The screenshot shows a SharePoint page with a blue header bar containing the 'WithoutAutoize' logo and a search bar. Below the header is a light blue sidebar with navigation links like 'Home', 'Documents', and 'Recent'. The main content area is titled 'Report Viewer + Menu' and displays a table of product data. A green box highlights the report area, labeled 'IBM Cognos Report'. The table has columns for Product line, Product type, Product number, Product key, Introduction date, Product brand code, Product brand key, Product size, Campaign, Revenue, Gross margin, Gross profit, Unit cost, Unit price, Product cost, and Quantity. The table is titled 'Product Line (Source) - without Autosizing'.

The area highlighted in green (IBM Cognos Report) is the maximum space available by default, which is 75% of the width, 58% of the height and 50% overall for the page.

So in addition to dynamic sizing for height, there needs to be reliable ways to change the SharePoint page layout without changing the SharePoint master page or content pages at the HTML source level.

The result is the toolkit of JavaScript, HTML and CSS overrides via use of the SharePoint Script Editor Web Part.

We provide a package of open source JavaScript functions and solution specific sample scripts plus CSS styles and HTML to:

- Show/hide elements of Master pages
- Show/hide adjacent web part zones
- Various strategies and techniques for optimizing white space.

The following shows an example of applying these techniques to the page shown above:

SharePoint

Newsfeed OneDrive Sites

Neil J. Thomson

SHARE FOLLOW

BROWSE PAGE

QueryVision Report Viewer Webpart

Options Actions

Product Line (Source) - with Autosizing & rowsPerPage

Product line	Product type	Product	Product number	Product key	Introduction date	Product brand code	Product brand key	Product brand	Product size	Campaign	Revenue	Gross margin	Gross profit	Unit cost	Unit price	Product cost	Quantity	Planned revenue
Campus Equipment	Cooking Gear	TrailChef Canteen	2110	30002	15-Feb-01 12:00:00 AM	701	701	TrailChef	2 liters	Regular sale				7.22	12.88	5,171,982.32		6,182,580.24
			2110	30002	15-Feb-01 12:00:00 AM	701	701	TrailChef	2 liters	TrailChef Campaign	2,452,413.29	87.8052	1,018,470.83	5.98	13.13	1,435,042.49	258,887	3,399,361.99
		TrailChef Cook Set	5110	30005	15-Feb-01 12:00:00 AM	701	701	TrailChef	6-piece	Regular sale				34.97	54.93	27,283,594.00		42,855,388
			5110	30005	15-Feb-01 12:00:00 AM	701	701	TrailChef	6-piece	Rising Star Campaign	0		-1,021,880.21	31.31	54.93	1,021,880.21	33,580	1,844,549.4
		TrailChef Cup	4110	30004	15-Feb-01 12:00:00 AM	701	701	TrailChef	330 ml	Regular sale				0.85	3.06	1,096,302.91		4,996,948.74
			4110	30004	15-Feb-01 12:00:00 AM	701	701	TrailChef	330 ml	TrailChef Campaign	1,101,584.8	99.4032	712,778.4	0.85	3.06	449,788.40	527,984	1,932,421.44
		TrailChef Deluxe Cook Set	6110	30006	5-Mar-03 12:00:00 AM	701	701	TrailChef	9-piece	Regular sale				81.03	129.72	35,122,883.85		56,410,887.8
			6110	30006	5-Mar-03 12:00:00 AM	701	701	TrailChef	9-piece	Rising Star Campaign	-0		-404,621.99	55.69	129.72	404,621.99	-7,271	943,194.12
		TrailChef Double Flame	8110	30008	5-Mar-03 12:00:00 AM	701	701	TrailChef	Double-burner	Regular sale				75.00	151.77	16,454,175.00		33,295,888.53
			8110	30008	5-Mar-03 12:00:00 AM	701	701	TrailChef	Double-burner	Rising Star Campaign	1,642,035.11	60.8087	594,510.11	75.00	151.77	1,077,525.00	14,367	2,180,476.59
			8110	30008	5-Mar-03 12:00:00 AM	701	701	TrailChef	Double-burner	TrailChef Campaign	1,253,950.72	14.705	368,725.72	75.00	151.77	885,225.00	11,803	1,791,341.31
		TrailChef Kettle	9110	30009	5-Mar-03 12:00:00 AM	701	701	TrailChef	1.5 liters	Regular sale				5.28	13.22	7,748,232.08		19,423,101.82
			9110	30009	5-Mar-03 12:00:00 AM	701	701	TrailChef	1.5 liters	TrailChef Campaign	6,681,090.97	-111.5066	-3,253,808.87	4.23	13.22	3,627,282.10	-867,729	-11,471,377.38
		TrailChef Kitchen Kit	3110	30003	15-Feb-01 12:00:00 AM	701	701	TrailChef	15-piece	Regular sale				15.93	23.80	12,042,781.40		17,992,324

30-Nov-15

1

6:20:15 PM

Top

Page up

Page down

Bottom

For more information see the following:

- Section 8 SharePoint Page Layout Modification Tools

## 7. AUTOSIZING

### 7.1 Installation and Configuration

#### SharePoint Server

Installation of the Autosizing components for SharePoint is included as part of the Web Part installation process.

#### Cognos Server

On each Cognos Server, several JavaScript libraries/.js files need to be copied.

Find the `<installLocation>\ibm\cognos\c10_64\webcontent` folder add a `queryvision` sub folder and include the following files from the installed QueryVision\QueryVision SharePoint Web Parts\Configuration folder:

- ◇ `iframeResizer.min.js`
- ◇ `qvtResizer.min.js`
- ◇ `iframeResizer.contentWindow.min.js`
- ◇ `qvtResizer.contentWindow.min.js`

Not all of these .js files are required for all Autosizing scenarios. [See the in depth section for details](#)

## 7.2 SharePoint, Cognos and Browser support

QueryVision Resizing is supported for SharePoint 2013/2016 for Cognos 10.2.0 (FP2+), 10.2.1 (FP3+), 10.2.2 (RTM+) and IBM Cognos Analytics 11.0 (current to 11.0.3.0).

Cognos Content support:

- SharePoint Page Layout Modification Tools are supported for any Cognos content
- Dynamic and Fixed/Static resizing is supported for Report Studio reports only.

Dynamic and Fixed/Static resizing is supported for Web Browsers/Versions:

- Microsoft Internet Explorer (8\*/9/10/11), Microsoft Edge\* (Windows 10)
- Firefox version 32 and later
- Google Chrome\* version 32 and later

\* Support for IE 8, Chrome and Microsoft Edge have exceptions

- **IE 8**

- ◊ Multiple Report Viewers on a web part page with dynamic resizing on reports in both Report Viewer.

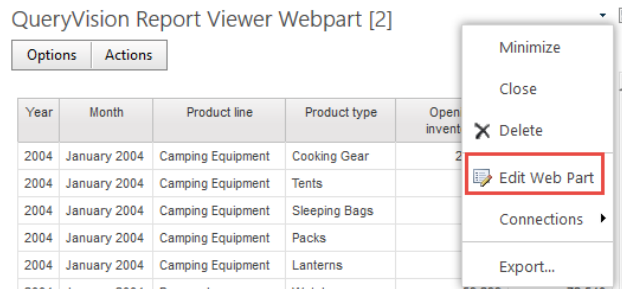
While general resizing will occur, vertical and horizontal scroll bars will appear as a result of the page footer and navigation controls at the bottom of the page.

- ◊ Nested reports where the container report has more than a single iframe will only render one of the iframes.

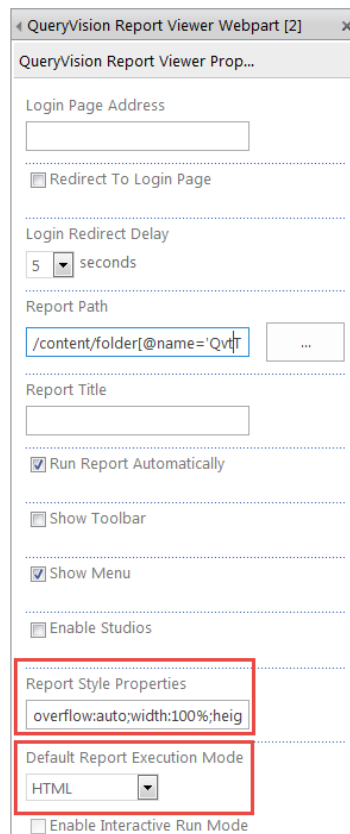
- **Google Chrome** – Cognos (and therefore QueryVision) provides limited support for Chrome – Chrome support is primarily targeted as a report execution/view environment.
  - ◊ **Cognos 10.2.2 FP1 and later** - Client tools are limited to WorkSpace, WorkSpace Advanced, Report Studio and Drill-through Definitions
  - ◊ **Pre Cognos 10.2.2 (e.g. Cognos 10.2.0)** – Client tools are limited to Cognos Workspace and Drill-through definitions.
- **Microsoft Edge** – Cognos does not currently acknowledge or list information on Windows 10 Microsoft Edge support and compatibility. Testing with Microsoft Edge shows the same capabilities and behavior as Google Chrome. Essentially this should only be considered for executing or viewing Cognos Report Studio reports in HTML mode.

## 7.3 Report Viewer Web Part Properties & Autosizing

The Report Viewer Web Part properties can be set by selecting to Edit Web Part from the drop down menu at the right of the web part.



There are two key properties that need to be set to enable Autosizing:



- **Report Style Properties** –allow setting HTML “style” attribute properties for the iframe that contains the Cognos Report.

Any CSS style property type can be included using the style ‘string’ format of

```
<property_name>:<property_value>;<property_name>:<property_value>;...<property_name>:<property_value>;
```

There are 3 mandatory property types: *overflow*, *width* and *height* which must specified or the properties and their default values will be added by the Report Viewer web part. The following are the default values

- ◇ overflow:auto
- ◇ width:100%
- ◇ height:700px



The overflow and width values are recommended, the height value is used as the default height for reports without Autosizing enabled (for setting height during Autosizing, see **Specifying embedded resizing options**, below, for details). It is recommended to specify the default height in pixels (e.g. 1250px) vs. percentage (100%).

The default value of the Report Style Properties reflects the individual default properties/values:

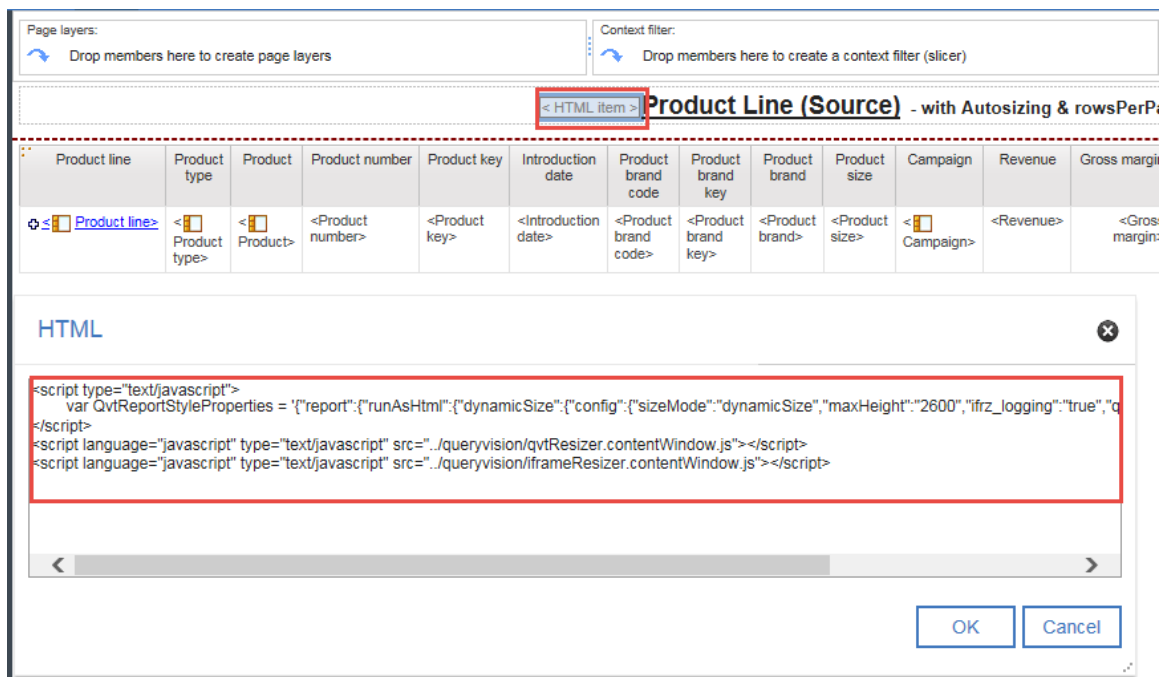
overflow:auto;width:100%;height:700px

- **Default Report Execution Mode** – must be set to HTML for Autosizing (dynamic or fixed/static).

## 7.4 Enabling Reports for Autosizing

**Note:** Only Report Studio reports support Autosizing. For IBM Cognos Analytics 11, reports which are identified as “interactive” (runInAdvancedViewer = true) will have any embedded JavaScript stripped out prior to execution.

To enable Autosizing requires insertion of an HTML item containing JavaScript to specify the embedded resizing options and to include the required JavaScript libraries. Which libraries to include are dependent on whether the current report is being autosized or whether this is a Container Report (see Advanced Autosizing – Multi-Tabbed Dashboard, later in this document)



The screenshot shows the IBM Cognos Report Studio interface. At the top, there are sections for 'Page layers' and 'Context filter'. Below these, a table titled 'Product Line (Source) - with Autosizing & rowsPerP' is displayed. The table has columns for Product line, Product type, Product, Product number, Product key, Introduction date, Product brand code, Product brand key, Product brand, Product size, Campaign, Revenue, and Gross margin. Below the table, an 'HTML' dialog box is open, showing the following JavaScript code:

```
<script type="text/javascript">
  var QvtReportStyleProperties = {"report":{"runAsHtml":{"dynamicSize":{"config":{"sizeMode":"dynamicSize","maxHeight":"2600","ifrz_logging":"true","q
</script>
<script language="javascript" type="text/javascript" src="..\queryvision\qvtResizer.contentWindow.js"></script>
<script language="javascript" type="text/javascript" src="..\queryvision\iframeResizer.contentWindow.js"></script>
```

The dialog box has 'OK' and 'Cancel' buttons at the bottom right.

Note that the HTML item is added to the Title.

The 'var QvtReportStyleProperties = <JSON parameters>' JavaScript and including scripts for the .js libraries must be rendered only once per page definition. The definition to be included should be identical for each page definition.

The following is an example of the full set of scripts in the HTML item

```
<script type="text/javascript">
```

```
  var QvtReportStyleProperties =
  '{"report":{"runAsHtml":{"dynamicSize":{"config":{"sizeMode":"dynamicSize","maxHeight":"2600","ifrz_logging":"true","qv_addHeight":"18px","qv_rowsPerPage":"14"}}}}';
```



```
</script>
<script language="javascript" type="text/javascript"
src="../../queryvision/qvtResizer.contentWindow.min.js"></script>
<script language="javascript" type="text/javascript"
src="../../queryvision/iframeResizer.contentWindow.min.js"></script>
```

### Important Note for IBM Cognos Analytics 11 and later

While the installation folder location for the .js libraries is the same for ICA 11 as for 10.X, the relative folder location of for the embedded script src= statement for ICA 11 reports is slightly different from 10.X

10.X

```
<script language="javascript" type="text/javascript"
src="../../queryvision/iframeResizer.contentWindow.min.js"></script>
```

11.X

```
<script language="javascript" type="text/javascript"
src="../../../queryvision/iframeResizer.contentWindow.min.js"></script>
```

## 7.5 Specifying embedded resizing options

To specify height within a Cognos Report Studio “Report/object” specification requires the use of an HTML item. Example:

```
<script type="text/javascript">
    var QvtReportStyleProperties =
    '{"report":{"runAsHtml":{"dynamicSize":{"config":{"sizeMode":"dynamicSize","qv_addHeight ":"25px"}}}}';
</script>
```

The report size information is packaged in a JavaScript block as a JSON string (orange) assigned to a JavaScript variable with the name QvtReportStyleProperties

This **must be a single line of JavaScript** (as multi-line string support in JavaScript is potentially error prone). It is recommended to build your JSON in multi-line form (to avoid missing brackets, quotes, etc.) and then to convert to a single line with a utility like <http://www.freeformatter.com/json-formatter.html>

### Dynamic Size

The structure must include “dynamicSize” as the JSON structure containing the “config” properties (highlighted below). This example shows all possible JSON properties that can be specified

```
{
  "report": {
    "runAsHtml": {
      "dynamicSize": {
        "config": {
          "sizeMode": "dynamicSize",
          "htmlStyle": "overflow:auto;width:100%;";
          "minHeight": "199px",
          "maxHeight": "1449px",
          "ifrz_autosize": "true",
          "ifrz_logging": "true",
          "ifrz_heightCalcMode": "lowestElement",
          "ifrz_bodyMarginStr": "0 0 0 0",
          "qv_addHeight": "20px"
        }
      }
    }
  }
}
```

The following shows a more typical example where the default values are acceptable for most properties:

```
{
  "report": {
    "runAsHtml": {
      "dynamicSize": {
        "config": {
          "sizeMode": "dynamicSize",
          "qv_addHeight": "20px"
        }
      }
    }
  }
}
```

#### Properties

- **sizeMode** (mandatory, value must be "dynamicSize")
- **htmlStyle (optional)** – can contain any CSS *html style* properties, except *height* (will be removed).  
See the *Special Note: height & htmlStyle*, below
- **minHeight** (optional, default is 200px) – must be in px (pixels) vs. percent.
- **maxHeight** (optional, default is 1320px) - must be in px (pixels) vs. percent.
- **height** – if this JSON property is included in a dynamicSize configuration, it will be ignored
- **ifrz\_autosize** (optional, default is true) - it is not recommended to specify this value without consulting QueryVision.
- **ifrz\_logging** (optional, default is false) – turn on to see log messages on auto sizing in the browser console log

- **ifrz\_heightCalcMode** (optional, default is "lowestElement") - it is not recommended to specify this value without consulting QueryVision.
- **ifrz\_bodyMarginStr** (optional, default is "0 0 0 0" – sets the HTML body margin for the iframe - it is not recommended to specify this value without consulting QueryVision.
- **qv\_addHeight** (optional, default is 0px) – adds height to compensate for Cognos report rendering for page footers and multi-page page navigation - See the section *qv\_addHeight - List Frame Footer/Navigation Controls "Icing"*, below for details
- **qv\_rowsPerPage** (optional, see section 7.5.2 *qv\_rowsPerPage – Specify rows per page for SharePoint* for details and default) – this allows setting of the rows per page for multi-page reports containing list or cross tab report components when rendered in SharePoint pages.

### Fixed Size

The structure must include "fixedSize" as the JSON structure containing the "config" properties (highlighted below)

```
{
  "report": {
    "runAsHtml": {
      "fixedSize": {
        "config": {
          "sizeMode": "fixedSize",
          "htmlStyle": "overflow:auto;width:100%; height: 700px"
        }
      }
    }
  }
}
```

A more typical example would only specify the *height* property, which will use the default *htmlStyle* value

```
{
  "report": {
    "runAsHtml": {
      "fixedSize": {
        "config": {
          "sizeMode": "fixedSize",
          "height": "700px"
        }
      }
    }
  }
}
```

### Properties

- **sizeMode** (mandatory, value must be "fixedSize")
- **htmlStyle (optional)** — can contain any CSS *html style* properties, except *height* (will be removed).

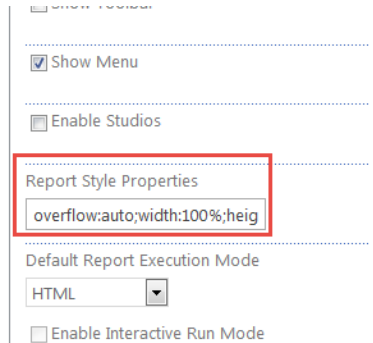
See the *Special Note: height & htmlStyle*, below

- **height (optional)** – states the fixed size height in pixels (e.g. “1320px”)
- **qv\_rowsPerPage** – see definition for Dynamic Size, above

*Special Note: height & htmlStyle properties*

HTML element attributes allow setting of the height via the “height” attribute or as part of the “style” attribute (e.g. style.height=<value> or style= “overflow:auto;height:700px”). Current browsers will use “style.height” in preference to the “height” value.

- The QueryVision Report Viewer web part “**Report Style Properties**” property provides the default value for the “style” attribute for the iframe containing the IBM Cognos content and will accept all CSS style properties, **including** height.



- The **QvtReportStyleProperty** JSON *htmlStyle* and *height* properties can be used to override the QueryVision Report Viewer web part “**Report Style Properties**” value  
It is recommended to use the JSON *htmlStyle* property to specify any CSS style property **except** height. Specify the height value using the JSON *height* property
  - ◇ **htmlStyle** – this will accept any valid CSS style attributes including height, but **the height property will be discarded**
  - ◇ **height** – is used to specify the height in pixels (e.g. 1250px) for fixed/static Autosizing

## 7.5.1 qv\_addHeight - List Frame Footer/Navigation Controls “Icing”

Cognos renders HTML interactive output with some dynamic resizing of its own: the [page] footer and navigation controls (first, prev, next, end) are placed at the lowest limit of the current window/iframe.

The result is that the interaction with dynamic resizing of the iframe within SharePoint may result in that the page is approx. 15 to 20 pixels longer than the dynamic resizing code can detect, which results in a vertical scroll bar which user has to scroll and see the page footer/navigation bar. This is particularly true of long (multipage) list and cross tab reports.

To address this issue, the QvtReportStyleProperties includes the optional parameter qv\_addHeight to add additional height in pixels or percent to the height, which will resolve this issue.

This can also be used for nested “container” reports to avoid vertical scroll bars where nested reports are used.

## 7.5.2 qv\_rowsPerPage – Specify rows per page for SharePoint

Dynamic sizing has the option of setting the rows per page via the qv\_rowsPerPage parameter when rendering IBM Cognos reports within SharePoint according to the formula outlined below.

As long as the rows per page are not explicitly specified in the report specification, this allows overriding the default Cognos Connection rows per page setting.

Rows per page can be specified for Cognos reports with the following priority order

1. In the Report specification - for list or cross-tab report components
2. When a “rows per page” parameter is included via a URL run request (which QueryVision uses)
3. In Cognos Connection as a default run property (default: 20 rows per page)

## 7.6 Drill up/down Behavior

Resizing for drill up/down behavior within a report is fully supported.

This includes an Analysis Studio or PowerPlay Studio report converted to Report Studio report or a Report Studio report designed with drill up/down behavior.

However, due to the behavior unique to Cognos reports, report size can only increase. Actions that decrease the height will render with the default blank color (e.g. white) below report content within the QueryVision Report Viewer web part.

## 7.7 Autosizing reports with multiple page definitions (including prompt pages)

A typical simple Cognos report only has a single page definition, regardless of whether the rendered report spans multiple pages due to result size.

Reports with authored prompt pages have one or more additional (prompt) page definitions in addition to the report page.

Reports can have multiple prompt page and report page definitions to accommodate different layouts and other features.

The embedded JavaScript must be inserted in the page definitions as HTML items as follows:

- The ‘var QvtReportStyleProperties = <JSON parameters>’ JavaScript must be rendered only once per page definition (e.g. in the page header or footer). The definition to be included should be identical for each page definition.  
  
Different definitions, including a mix of dynamic and fixed/static resizing or fixed/static resizing with different heights **cannot** be mixed, but the `qv_addHeight` value can be different for each page definition for dynamic resizing.
- The JavaScript to include the .js libraries should be rendered only once per page definition and ultimately only once per HTML page (e.g. in the page header or footer).

## 7.8 Behavior of Reports with prompt pages

For Report Studio objects/reports, optional and mandatory parameters in the report can be serviced by prompts on the report or on authored prompt pages prior to report execution or be automatically generated when rendered.

### Reports WITHOUT QueryVision Autosizing enabled

Automatic or authored prompt pages and report pages will be rendered according to the size specified in the QueryVision Report Viewer configuration Report Style Properties values (default 700px).

### Reports WITH QueryVision Autosizing enabled

**Authored Prompt Pages** – To ensure correct fixed/static sizing of all page definitions within a report, including authored prompt pages, ensure that the embedded script specifying the QvtReportStyleProperties JavaScript is included, specifying the height in pixels **on each prompt page** (e.g. page header)

#### Automatically generated Prompt Pages

**Fixed/static Autosizing** – the prompt page will be treated as a non-Autosize enabled report page and will use the Report Viewer configuration Report Style Properties value (e.g. default of 700px)

**Dynamic Autosizing** – the prompt page will be treated as a non-Autosize enabled report page and will use the minHeight property specified by the dynamic size version of the QvtReportStyleProperties JavaScript embedded elsewhere in the report (e.g. '200px').

## 7.9 Drill-Through Behavior

Autosizing options for drill-through source and target reports

QueryVision recommends that dynamic Autosizing be enabled in all source and target reports.

The first report presented in a QueryVision Report Viewer Web Part (e.g. source report) determines the Autosizing behavior – whether dynamically auto-sized, fixed/static Autosize enabled or not Autosize enabled. Drill target settings will be ignored with the exception of the qv\_addHeight property.

Prompt page considerations

Prompt pages for target reports must be configured with an Autosize configuration that is compatible with the source report and target report pages.

If the source report provides all the mandatory prompt parameters then the prompt page for a target report will not be presented, so no-Autosizing is required for those prompt pages.

## 8. SHAREPOINT PAGE LAYOUT MODIFICATION TOOLS

The SharePoint Page Layout “toolkit” consists of a JavaScript .js/library of functions which are called from JavaScript embedded along with HTML and CSS in a SharePoint page using a SharePoint 2013 Script Editor Web Part.

*Note:* See the documentation and comments in the files *qvtShrPntWidthUtils.js* and *SharePoint web page layout mods snippets.htm* for details

**Notice:** *qvtShrPntWidthUtils.js and the script samples that are included in SharePoint web page layout mods snippets.htm are available “as is” and is not supported by QueryVision. These files/code can be considered open source without license restrictions and can be extended or modified as required.*

The “toolkit library” is the file *qvtShrPntWidthUtils.js*, which is installed during the QueryVision Web Parts installation in the same folder as the Autosizing .js files. This file is available in .min.js and .js versions.

It uses self-initializing JavaScript namespace to wrapper the functions to avoid interference with other potential .js functions.

The main entry points for the library/.js file are functions for each of:

Initialization functions are defined for each of the page elements to be shown or hidden, some of which will also include changes to related html element borders and margins. The elements

defined include the master page header, the master page side nav bar/box and the web part zone that contains a QueryVision Content Manager Navigation Web Part (if one is defined).

Each initialization function has options defined using a JavaScript option block parameter.

Each show/hide initialization function has:

- The option to automatically hide (initialHide) the specified HTML element(s)
- An optional button (or other HTML control) which will allow the user to show/hide the HTML element(s). If the button element is defined, then a function to handle toggling showing/hiding the element is attached to button/control .onclick event.

The file SharePoint web page layout mods snippets.htm provides a composite example of using all of the functions and options:

- An HTML definition for the optional buttons which will be placed within location within the Script Editor Web Part defined area on the page.
- A CSS style definition section
- Script src definition for the qvtShrPntWidthUtils.js
- Script which defines functions and sets the parameters to call the initialization functions defined in qvtShrPntWidthUtils.js and also registers them to run once the SharePoint page is loaded.

## 8.1 SharePoint specific JavaScript page layout manipulation considerations

In looking to optimize the SharePoint page layout for IBM Cognos reports, you should be aware of the following:

- While the general concepts of how to manipulate (e.g. show and hide) HTML elements on a SharePoint page (e.g. CSS style changes, <html element>.style.display = 'none', ...) using JavaScript apply, this is complicated by:
  - ◇ SharePoint uses its own unique and inconsistent pattern of page composition, including how it identifies HTML elements, use of CSS styles and HTML specified style related properties.
  - ◇ SharePoint itself makes extensive use of JavaScript, so (particularly on Team Sites) there are frequently conflicts when attempting to add custom JavaScript to the page. For example:

Custom JavaScript successfully hides a page element and then SharePoint provided JavaScript unhides it later in the page lifecycle.
  - ◇ Using custom JavaScript in SharePoint related to overall page layout manipulation requires specialized registration and execution consideration.
- Each SharePoint page is a composite of the Master Page defined elements + individual page content.

A Master Page ([Introduction to SharePoint master pages](#)) defines site page layout and elements common to every page on the site, including navigation links, branding/logos, site search controls, possibly menus, etc. A site can have multiple Master pages (but doing so is not trivial).

The page content in this case is a web part page which has multiple web part zones of which only a single QueryVision Report Viewer web has been added.



- ◇ SharePoint 2013 provides a fixed set of web part page templates which use the same CSS stylesheets
- ◇ The default CSS/HTML style for web part zones are defined such that if there are no web parts added to a web part zone, then the web part zone is hidden such that it takes no space on the page.
- ◇ The default CSS specifies whitespace padding (HTML/CSS borders/margins) around all sizes of each web part within a web part zone and additional padding around each web part zone.
- Microsoft has customized the page lifecycle for SharePoint pages – at least where it comes to Javascript .js file loading and loading order and initial JavaScript execution. Hint, it's not when \$(document).ready() is true.

There are various mechanisms you can use to ensure that custom javascript (that manipulates the page “in competition” with MSs SharePoint JavaScript) is executed when the page is ready, but the one QueryVision found reliable was:

```
_spBodyOnLoadFunctionNames.push("<function>");
```

If you are going to add your own custom JavaScript which depends on SharePoint provided .js files then you should investigate SP.SOD.executeOrDelayUntilScriptLoaded(). Some links to investigate:

[Ensuring SharePoint JavaScript API is loaded \(sp.js\)](#)

[Execute JavaScript function after SharePoint is initialized](#)

## 8.2 Example of specific techniques used

The techniques described here are due to specific issues with SharePoint specific HTML or JavaScript issues.

The following are 2 examples:

- To work around conflicts with SharePoint's “Focus on content” feature and it's manipulation by internal JavaScript, the SharePoint master page header (HTML element id “s4-titlerow”) is shown and hidden by adding or removing a CSS style class “qvtS4TitlerowHide”. This class uses “!important” to override any other definitions.
- In order to find the web part zone that contains a QueryVision Content Manager Navigation requires knowing the full, page instance id of the web part “ctl00\_ctl40\_g\_d751faa9\_e46e\_4185\_822a\_105105bc5751\_QVTree” and using a parent search on the element “class” attribute to see if it contains the class “ms-webpart-zone”

For other SharePoint specific techniques see the *qvtShrPntWidthUtils.js* and *SharePoint web page layout mods snippets.htm* file comments and JavaScript code.

## 8.3 Why the Script Editor Web Part for JavaScript?

Using a web part to inject JavaScript to the page allows use of SharePoint web parts ability to be edited directly within the browser without any additional tools (such as SharePoint Designer). When combined with use of built in browser debugging tools (the Firefox/Firebug combination is recommended), this provides a competent environment for this particular task.

At first glance, there are two candidate web parts in SharePoint 2013: the Content Editor Web Part and the Script Editor Web Part. However, the Content Editor Web Part was only intended for HTML and CSS style content and will effectively disallow any JavaScript.

The Script Editor web part will allow HTML, CSS and JavaScript.



An additional advantage of using a web part is that it provides a container for the positioning of HTML controls (such as buttons for hide/show) which can be positioned with a web part page through simply adding the web part to any web part zone and then ordering the web part with others within the zone.

## 9. APPENDIX

### 9.1 Troubleshooting

#### *Menu run action vs. configuration run action*

If Autosizing is not occurring ensure that both the web part configuration option and the action menu bar are both set to HTML. If one or either are set to Default Output, resizing may not occur.

### 9.2 Select the appropriate SharePoint page layout

The first step is to create a SharePoint page with Web Part Zone(s) that provides sufficient space for the IBM Cognos Reports you want to embed.

Here we'll cover:

- Options on creating SharePoint pages with web parts
- Approaches for maximizing the screen real-estate for web parts

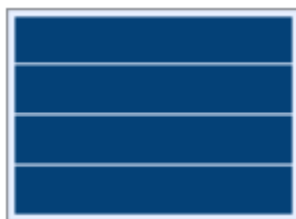
#### 9.2.1 SharePoint Web Part Page templates

SharePoint supports multiple ways of including web parts on a SharePoint application page including the following in order of time and effort.

1. Create a SharePoint web part page by selecting from built-in or custom developed Web Part Page templates.
2. Create a web part page from a template then modify the page (e.g. in SharePoint Designer)
3. Using SharePoint Designer (or other web page editor), create a custom SharePoint page and add one or more "web part zones".

Here we will focus on Web Part Page templates, which are the core for either options 1 or 2.

There are 8 built-in web part templates of which the following is a sample.



Full Page web part zone

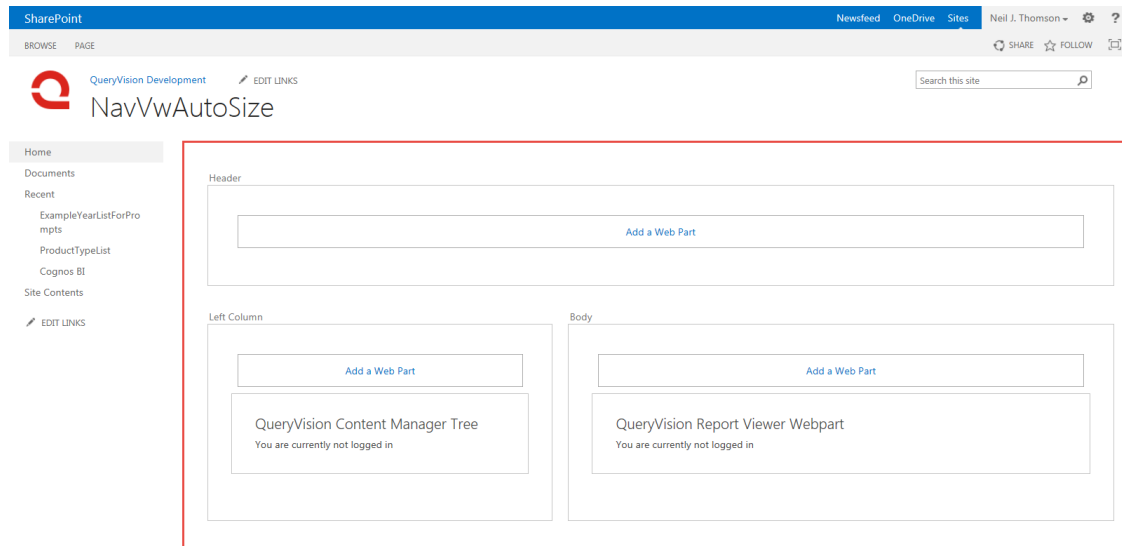


- Title web part zone that spans the page width
- Left column fixed web part zone
- Right column responsive web part zone



- Title web part zone that spans the page width
- Left, Right column responsive web part zones
- Center column responsive web part zone

The following is an example using the middle web part page template (Title, Left, Right web part zones) within a standard SharePoint 2013 Master Page. The section within the red rectangle is the space on the page provided for the web part page template. The rest of the page is defined by the site Master Page



While it is possible to use SharePoint Web Parts on any SharePoint page, the full features (including browser editing) are supported on **SharePoint Web Part Pages**.

A SharePoint web part page consists of one or more **Web Part Zones**, which can be arranged in table like fashion, with the ability to configure both individual zone vertical and horizontal sizing plus how they resize relative to each other and the page itself, including how they react when the browser window is resized.

SharePoint ships with several web part page templates, which can be used to create web part pages.

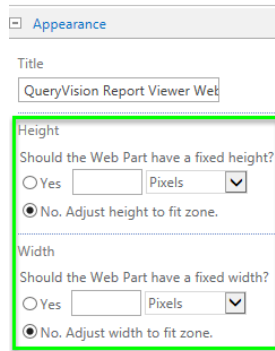
To see how SharePoint Web Part Pages and Web Part Zones work to provide dynamic sizing, let's first build a Web Part Page.

## 9.3 Leverage SharePoint Web Part Page, Web Part Zone and Web Part built-in dynamic resizing capabilities

SharePoint web page layout can be through a combination of both static configuration and dynamic (e.g. JavaScript) techniques.

### 9.3.1 Understanding and configuring web part zone and web part sizing

The actual width and height of individual **web parts** within the zone are governed by the web part's configuration Appearance section's Height and Width properties.



**For web parts with Cognos content, it is recommended to leave the Appearance Height and Width properties as shown above in the “Adjust to fit zone” setting whether QueryVision Autosizing is used or not**, which allows the web part to grow and shrink both horizontally and vertically as the browser window size changes. Effective vertical resizing requires QueryVision’s Autosizing feature.

Web Part Page Layout Templates as shipped with SharePoint 2013 have **their web part zones** configured for height and width via the `iframe` element properties *height* and *width* (which can be examined using browser developer tools or opening and examining the page html with SharePoint Designer).

Web Part zones are configured

- Height of web part zones are all set with the html attribute *height='100%'*. This provides the capability for web parts within the zones grow or shrink with their contents – but this requires communication between the web part content and the web part page, which won’t occur for SharePoint/Cognos integration without QueryVision’s Autosizing feature.
- Width of web part zones are governed by the html attribute *width*:
  - ◊ Fixed width (HTML attribute “width” is not defined for the web part zone) - the width of the zone is defined by the widest web part in the zone.
  - ◊ Dynamic (HTML attribute *width='XX%'*) – enables growing/shrinking with the browser. A typical value is 100% – width is determined by the.

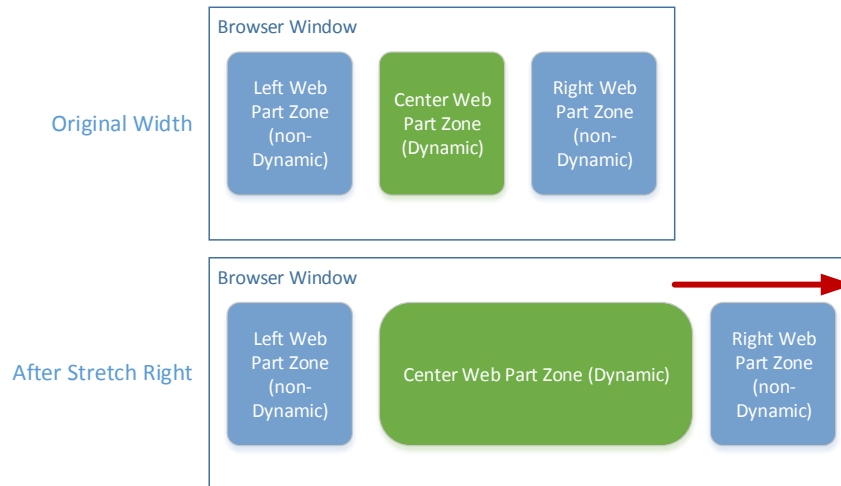
**It is recommended to use web part zones with Dynamic width and height for web parts with Cognos content**

In a web part template which has multiple web part zones horizontally, such as:

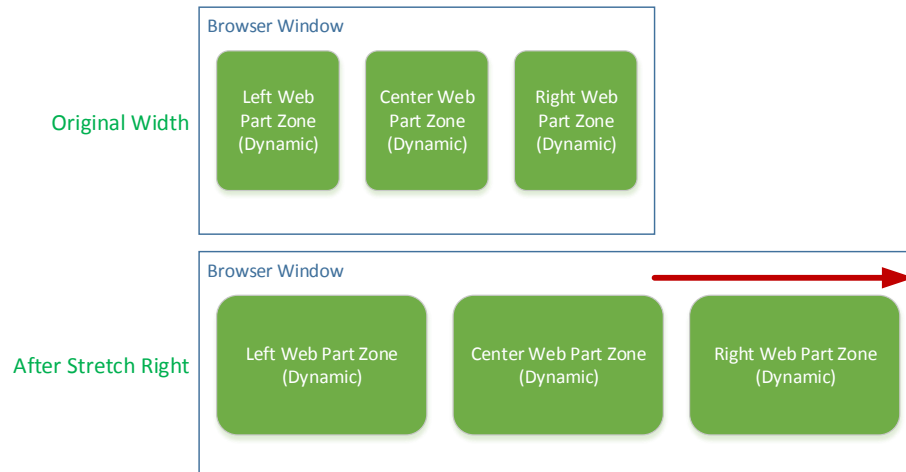


the zones typically have a mix of dynamic and non-dynamic width settings

- ◆ Left and Right web part zones as non-dynamic with the Center part zone with “width:100%” in a side by side configuration. When growing/shrinking the width of the browser window, the widths of Left/Right will not change, the width of Center will expand/shrink to take up the remaining width in the browser window.



- ♦ Mix of dynamic width web part zones (Left, Center, Right) all with "height:100%" in a side by side configuration. Each of Left, Center, Right will grown and shrink, taking 1/3 of the available width.



### 9.3.2 Dynamic techniques

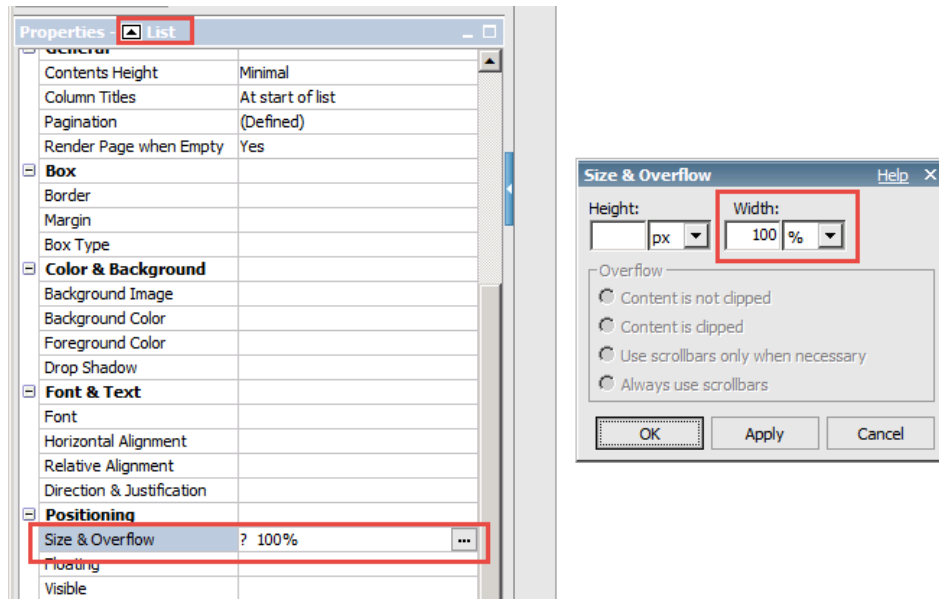
Dynamic techniques to provide more room for web parts without having to resorting to custom Master Pages or SharePoint Designer can be found in the section **8 SharePoint Page Layout Modification Tools**.

## 9.4 Leveraging IBM Cognos Report height and width configuration capabilities

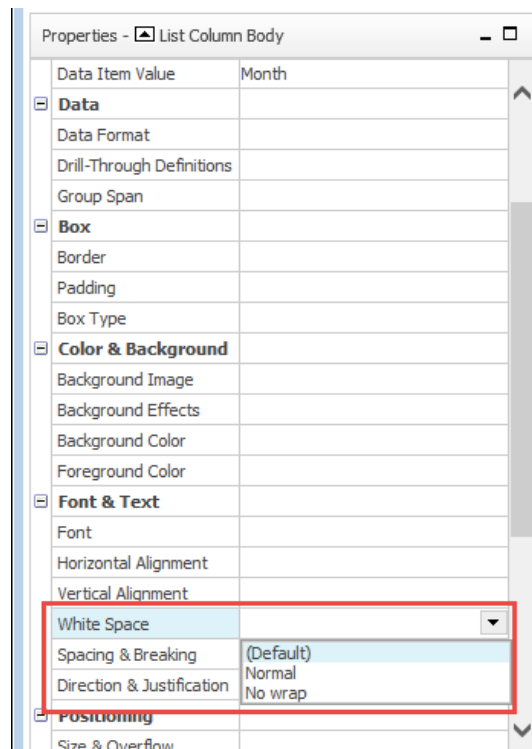
Cognos Report Studio report elements (e.g. list, crosstab ...) can be configured for both height and width to fit within their window/iframe by setting their values to 100%. While QueryVision Autosizing is required for vertical responsive sizing, Cognos reports can adjust column width etc. as the parent window size is changed.

There are two sets of settings – the "Size & Overflow" property for a list or cross tab, plus the "White Space" property for individual columns (list, crosstab) and/or row (crosstab)

The following shows changing the default settings for a List report object width to 100% for the Size and Overflow property:



The following shows setting shows setting the White Space property for a List Column [Body]. Note that “Normal” allows wrapping of column content (text, plus some dates/times), which is also the Default. “No wrap” prevents column content wrapping.



#### *“Responsive” Width - example*

While not fully “Responsive” in the multi-device web page adaptation sense, Cognos reports will adjust column width, content wrapping within the column and report height. The following shows a

Cognos list style report with the list Size & Overflow width=100% where the browser width is gradually being reduced from the right hand side (sequence is from left to right).

Note that the column width reduces and eventually column contents word wrap and as a last resort (right-most example) as the height reaches a defined max # of pixels, a vertical scroll bar is included:

QueryVision Report Viewer Webpart  
Options Actions

**Product Inventory List**  

Year	Month	Product line	Product type	Opening inventory	Quantity shipped
2004	January 2004	Camping Equipment	Cooking Gear	236,107	234,034
2004	January 2004	Camping Equipment	Tents	51,253	53,952
2004	January 2004	Camping Equipment	Sleeping Bags	54,210	53,735
2004	January 2004	Camping Equipment	Packs	47,380	48,851
2004	January 2004	Camping Equipment	Lanterns	60,036	68,096
2004	January 2004	Personal Accessories	Watches	59,299	72,540
2004	January 2004	Personal Accessories	Eyewear	192,737	206,191
2004	January 2004	Personal Accessories	Gloves	142,107	139,747
2004	January 2004	Personal Accessories	Biscuits	15,304	19,975
2004	January 2004	Personal Accessories	Navigation	33,879	36,634
2004	January 2004	Outdoor Protection	Insect Repellents	251,413	244,385
2004	January 2004	Outdoor Protection	Sunscreen	197,166	199,028
2004	January 2004	Outdoor Protection	First Aid	35,125	35,457
2004	January 2004	Golf Equipment	Irons	6,562	6,639
2004	January 2004	Golf Equipment	Woods	4,605	4,542
2004	January 2004	Golf Equipment	Putters	25,247	25,297
2004	January 2004	Golf Equipment	Golf Accessories	43,533	47,493

QueryVision Report Viewer Webpart  
Options Actions

**Product Inventory List**  

Year	Month	Product line	Product type	Opening inventory	Quantity shipped
2004	January 2004	Camping Equipment	Cooking Gear	236,107	234,034
2004	January 2004	Camping Equipment	Tents	51,253	53,952
2004	January 2004	Camping Equipment	Sleeping Bags	54,210	53,735
2004	January 2004	Camping Equipment	Packs	47,380	48,851
2004	January 2004	Camping Equipment	Lanterns	60,036	68,096
2004	January 2004	Personal Accessories	Watches	59,299	72,540
2004	January 2004	Personal Accessories	Eyewear	192,737	206,191
2004	January 2004	Personal Accessories	Gloves	142,107	139,747
2004	January 2004	Personal Accessories	Biscuits	15,304	19,975
2004	January 2004	Personal Accessories	Navigation	33,879	36,634
2004	January 2004	Outdoor Protection	Insect Repellents	251,413	244,385
2004	January 2004	Outdoor Protection	Sunscreen	197,166	199,028
2004	January 2004	Outdoor Protection	First Aid	35,125	35,457
2004	January 2004	Golf Equipment	Irons	6,562	6,639
2004	January 2004	Golf Equipment	Woods	4,605	4,542
2004	January 2004	Golf Equipment	Putters	25,247	25,297

QueryVision Report Viewer Webpart  
Options Actions

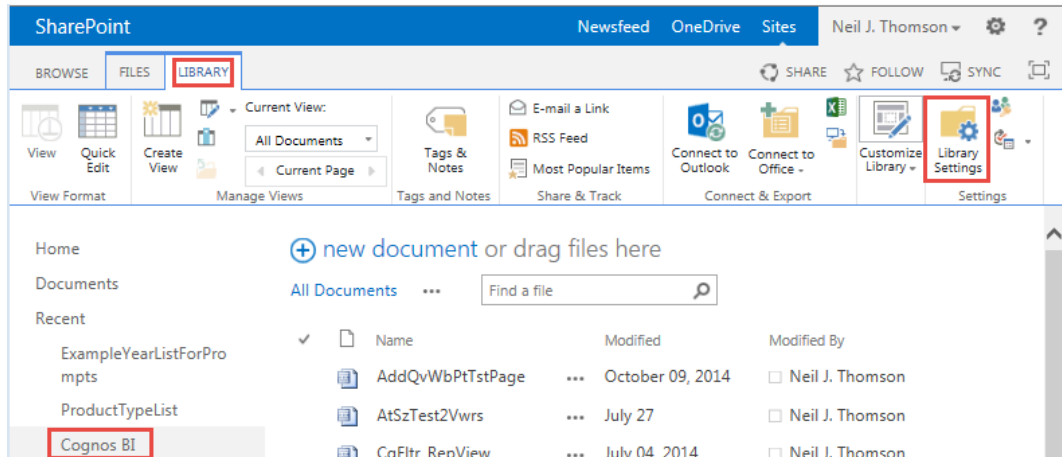
**Product Inventory List**  

Year	Month	Product line	Product type	Opening inventory	Quantity shipped
2004	January 2004	Camping Equipment	Cooking Gear	236,107	234,034
2004	January 2004	Camping Equipment	Tents	51,253	53,952
2004	January 2004	Camping Equipment	Sleeping Bags	54,210	53,735
2004	January 2004	Camping Equipment	Packs	47,380	48,851
2004	January 2004	Camping Equipment	Lanterns	60,036	68,096
2004	January 2004	Personal Accessories	Watches	59,299	72,540
2004	January 2004	Personal Accessories	Eyewear	192,737	206,191
2004	January 2004	Personal Accessories	Gloves	142,107	139,747
2004	January 2004	Personal Accessories	Biscuits	15,304	19,975
2004	January 2004	Personal Accessories	Navigation	33,879	36,634
2004	January 2004	Outdoor Protection	Insect Repellents	251,413	244,385
2004	January 2004	Outdoor Protection	Sunscreen	197,166	199,028
2004	January 2004	Outdoor Protection	First Aid	35,125	35,457
2004	January 2004	Golf Equipment	Irons	6,562	6,639
2004	January 2004	Golf Equipment	Woods	4,605	4,542
2004	January 2004	Golf Equipment	Putters	25,247	25,297

## 9.5 Creating a SharePoint page with a web part template

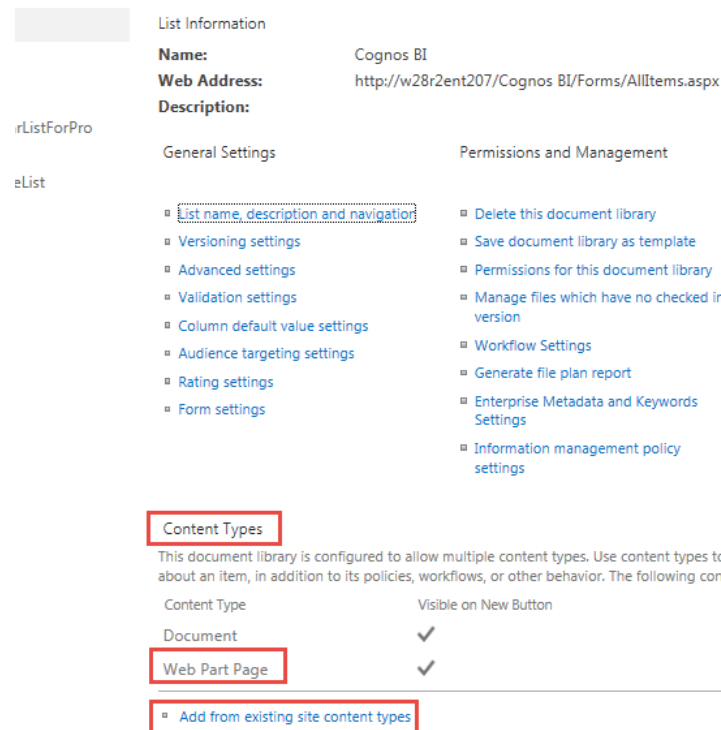
To build a web part page from a template, do the following (SharePoint 2013):

- Ensure that the SharePoint [document] library you want to add the web part page to has had Web Part Page added to the Content Types via that library's "Library Settings" icon in the Library tab

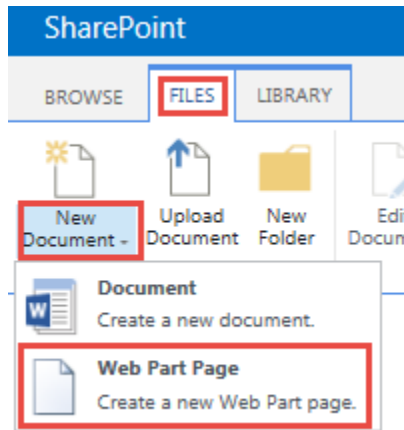


If Web Part Pages are permitted in the library they will be shown in the Content Types table as shown below. If Web Part Pages is not enabled, they can be added from the "Add from existing site content types, also shown below

### Cognos BI - Settings



- In the Library Files tab, select to add a new Web Part Page:



You will now see a pallet of possible Web Part Page Layout Templates showing their Web Part Zones. Note that the Layout Templates (for the templates shipped with SharePoint) are named for the number and position of Web Part Zones:

## Site Contents ▸ New Web Part Page ⓘ

Name

Type a file name for your Web Part Page. The file name appears in headings and links throughout the site.

Name:

.aspx
 

☐ Overwrite if file already exists?

arListForPro

Layout

Select a layout template to arrange Web Parts in zones on the page. Multiple Web Parts can be added to each zone. Specific zones allow Web Parts to be stacked in a horizontal or vertical direction, which is illustrated by differently colored Web Parts. If you do not add a Web Part to a zone, the zone collapses (unless it has a fixed width) and the other zones expand to fill unused space when you browse the Web Part Page.

Choose a Layout Template:

Header, Footer, 3 Columns

Full Page, Vertical

Header, Left Column, Body

Header, Right Column, Body

Header, Footer, 2 Columns, 4 Rows

Header, Footer, 4 Columns, Top Row

Left Column, Header, Footer, Top Row, 3 Columns

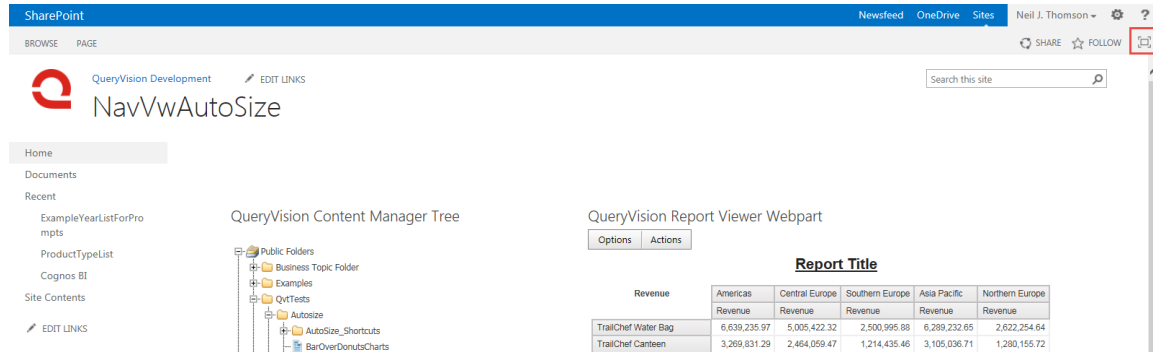
Right Column, Header, Footer, Top Row, 3 Columns

The diagram shows a 3-column layout template. Red arrows point from the text 'Header, Footer, 3 Columns' to the three columns of the template. The template itself is represented by three blue rectangles stacked vertically, with red arrows pointing to each one.

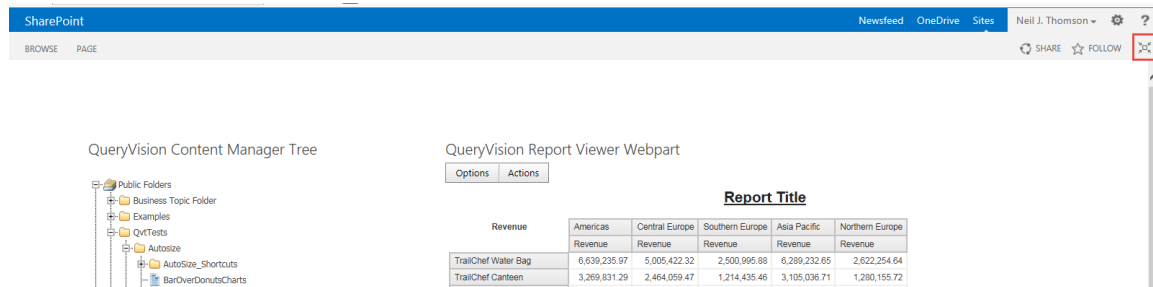


## 9.6 Pitfalls with SharePoint's “Focus on content”

In SharePoint 2013, a feature was added by default to SharePoint pages called “Focus on content”, which is implemented as a button in the top right corner of the page outlined in red as shown below:



Users can click on this button and show/hide the master page content



A bit of digging into this feature uncovers the `SetFullScreenMode(T/F)` which can be called directly invoke this feature there are a number of drawbacks:

- Full screen mode is for the site – if full screen mode is enabled on one page – by the user or through JavaScript, it remains active when navigating to other pages (via a cookie).
- Attempting to enable or disable full screen mode is subject to interference/override by SharePoint JavaScript that controls the feature. For example:
 

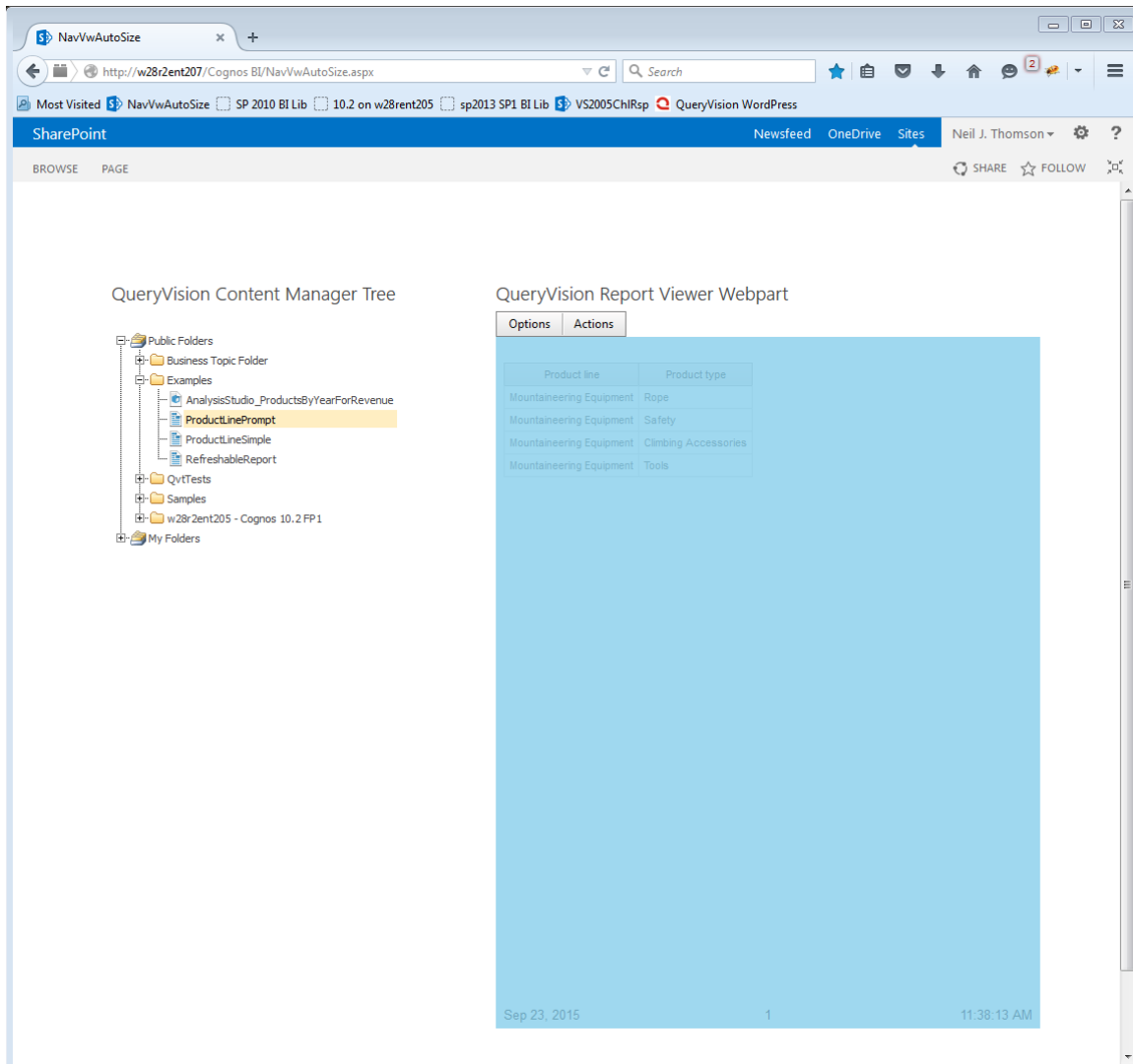
On page initialization enable/disable the feature via custom JavaScript may be overridden later in the initialization event handling by the SharePoint JavaScript.
- Attempting to use custom JavaScript to isolate the full screen mode setting to only affecting the current page was unsuccessful in QueryVision testing. It was not possible to avoid toggling full screen mode on/off during refresh of the current page as detecting refresh vs. true transition to another page was not possible (due to the way Microsoft implemented the feature) without spreading the code to other pages in the site.

For this reason we chose to implement independent show/hide of each of the master page header and master page side nav bar/box via CSS and style overrides.

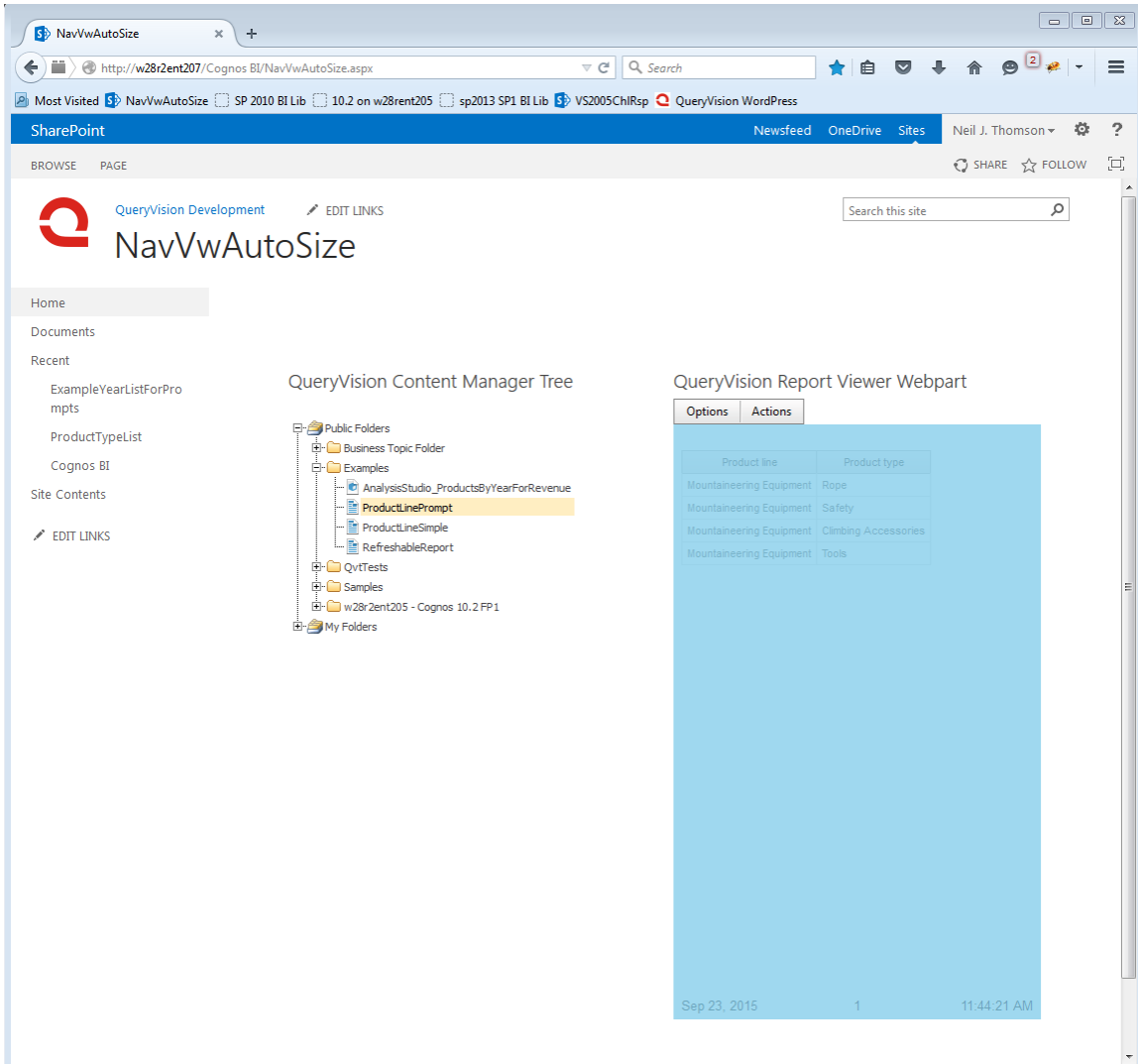
## 9.7 Guide for Cognos Report Heights in SharePoint

### *Guidelines to heights and common SharePoint page layouts*

Where a QueryVision Report Viewer is on a SharePoint Page with only the simple SharePoint header/menu bar (see below), then on a 1920x1080 screen, 700px is the maximum height for a list report “page” to be fully visible (blue shaded area):



Using a standard web part page, with a standard SharePoint 2013 side bar to the left and a title and search area on the page, the maximum is 600px (example below):



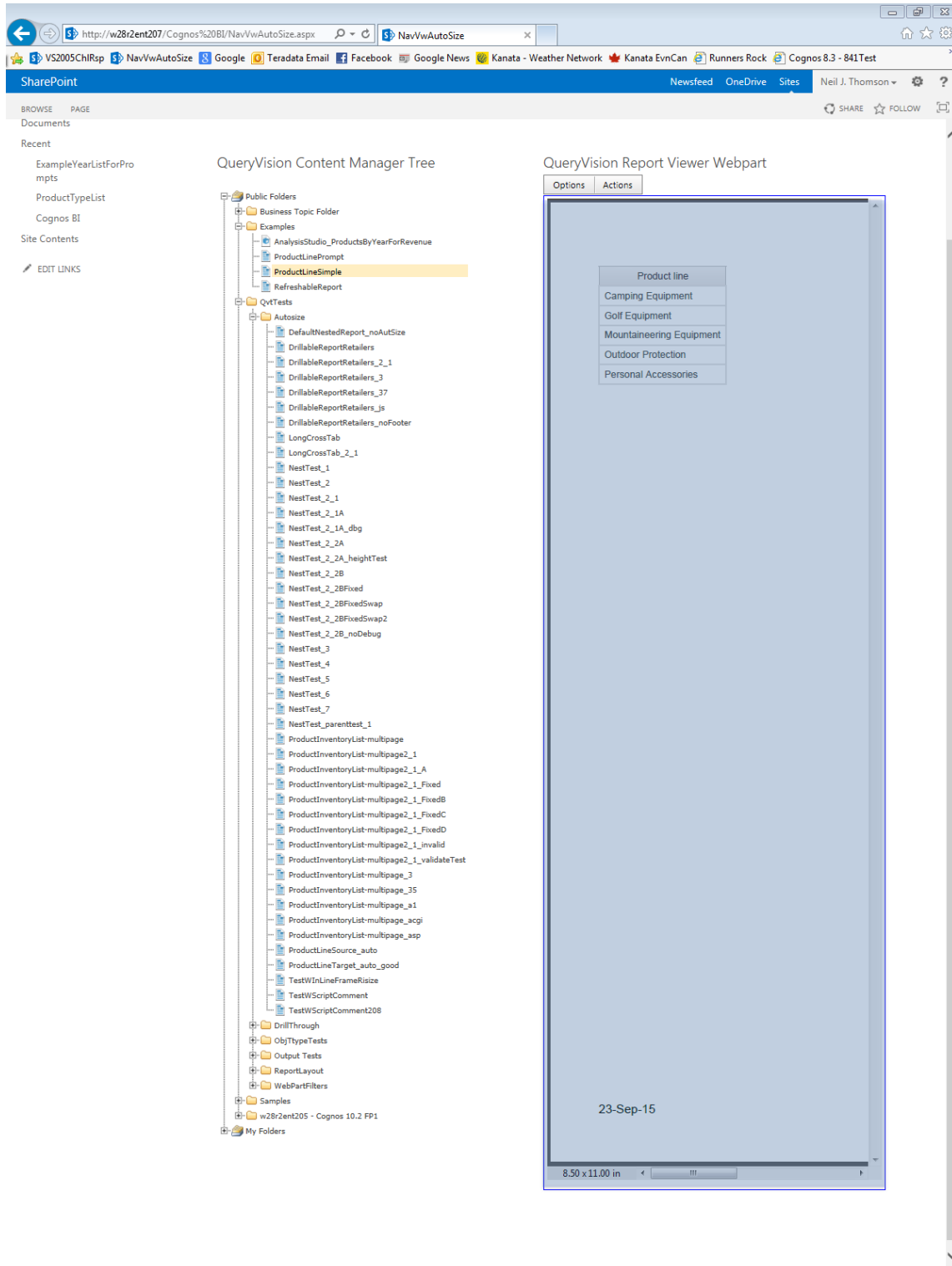
The screenshot shows a SharePoint 2013 web page titled "NavVwAutoSize". The page layout includes a top navigation bar with "SharePoint", "Newsfeed", "OneDrive", and "Sites". Below this is a secondary navigation bar with "BROWSE", "PAGE", "SHARE", "FOLLOW", and a search box. The main content area is divided into three sections:

- Left Sidebar:** Contains a "Home" button, "Documents", "Recent" (with links like "ExampleYearListForPro mpts", "ProductTypeList", "Cognos BI", "Site Contents"), and "EDIT LINKS".
- QueryVision Content Manager Tree:** A tree view showing a hierarchy of folders and files. The "ProductLinePrompt" folder is highlighted. The tree structure includes:
  - Public Folders
    - Business Topic Folder
    - Examples
      - AnalysisStudio\_ProductsByYearForRevenue
      - ProductLinePrompt (highlighted)
      - ProductLineSimple
      - RefreshableReport
    - QvtTests
    - Samples
    - w28r2ent205 - Cognos 10.2 FP 1
  - My Folders
- QueryVision Report Viewer Webpart:** A table with two columns: "Product line" and "Product type". The table contains four rows of data:
 

Product line	Product type
Mountaineering Equipment	Rope
Mountaineering Equipment	Safety
Mountaineering Equipment	Climbing Accessories
Mountaineering Equipment	Tools

 Below the table, the date "Sep 23, 2015", the number "1", and the time "11:44:21 AM" are displayed.

If rendering a Report as PDF using an 8.5x11 inch PDF page or in HTML report with the height set for an 8.5x11 page (List/Crosstab report with 40 rows per page or less), then use 1320px as the maximum height. If the page setup is for A4 then the maximum height will be 1404px.



The screenshot shows a SharePoint web browser interface. The top navigation bar includes links for Newsfeed, OneDrive, and Sites. The main content area is divided into two sections:

- QueryVision Content Manager Tree:** A tree view showing the hierarchy of reports and folders. The 'ProductLineSimple' report is highlighted under the 'Examples' folder.
- QueryVision Report Viewer Webpart:** A webpart displaying a report titled 'Product line'. The report content is a list of product categories: Camping Equipment, Golf Equipment, Mountaineering Equipment, Outdoor Protection, and Personal Accessories.

The bottom status bar indicates the page size is 8.50 x 11.00 in and the date is 23-Sep-15.

## 9.8 Detailed Feature Specifications

### Mobile Support

Neither designed nor tested for mobile devices (e.g. iPad), but may work.

### Report dynamic width supported

If the Cognos report specification has dynamic width enabled (e.g. list frame width:100%) then this continues to be supported and will cause Autosizing as required (e.g. shrinking the width at some point may cause the report height to increase, for which Autosizing will recognize the height change and will increase the height

### Multiple Autosize enabled reports on the same page

If there are multiple QueryVision Report Viewers on the same SharePoint page, then each will size/Autosize independently based on their web part zone, web part and Autosizing configuration

### Report studio reports only

The required auto-sizing report html item can only be added in Report Studio reports. For other Cognos report “types” such as Analysis Studio, Query Studio or PowerPlay Studio, that can be opened in Report Studio and saved to (and run as) a Report Studio report can also be auto enabled.

Cognos Workspace/Business Insight dashboards, Active Reports, etc. are not supported for auto-sizing.

Cognos Report Studio page elements which have the Autosize script configured HTML element that are used in a Cognos Workspace/Business Insight dashboards will not Autosize within the dashboard.

### Use Report Studio only

The Business Insight Advanced/Workspace Advanced editors do not support inserting HTML elements

### Grow only

When viewing an auto-sized Cognos Report, for operations such as drill down/drill up, the page will get taller if required, but will not shrink without refreshing the page

**Saved outputs (html, pdf, csv, xml ...) will not Autosize**, including for Reports that have been Autosize enabled.

### Only Dynamic or Fixed Size within a single report or drill-through combination

Where a report has multiple page definitions (e.g. one or more report pages plus one or more prompt pages), the Autosizing options have to be identical with the exception of the qv\_addHeight property for dynamic resizing)

- ◇ Fixed reports – option specification must be the same height
- ◇ Dynamic reports – all pages must be dynamic size enabled
- ◇ Drill-through – if a source report drills through to a target report within SharePoint, then both must be either fixed or dynamic (recommend dynamic)

### Only Default Report Execution Mode = HTML supports auto-sizing

Using the Report Viewer in “Default Output” will not auto-size reports if an existing saved output exists (html/pdf) as Autosizing does not work for saved outputs. The report will Autosize in “Default Output” mode if there are no saved outputs as the report will run/execute if no outputs are available.

## 9.9 Advanced Autosizing - Multi-Tabbed Dashboard via Nested Reports

### 9.9.1 Introduction

There are multiple ways you can create BI dashboards from multiple Cognos Reports or Report elements (widgets), but one that some customers have used is to use nested Cognos reports along with simple HTML buttons to emulate tabs to allow selection from a range of reports within a single Cognos Report.

A potentially key advantage of this approach is to leverage Cognos Framework Manager session information to filter what reports/buttons to offer the user based on their IBM Cognos user or group membership.

Support Autosizing for this scenario requires an adaptation of Autosizing.

First we will look at how to construct nested reports, then we'll cover how to enable them for QueryVision Autosizing within SharePoint.

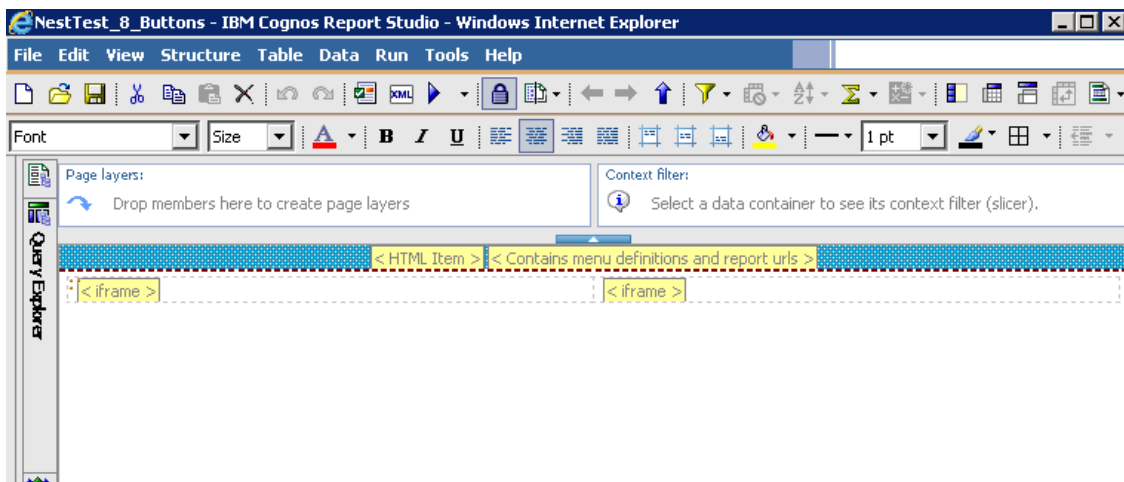
### 9.9.2 Container Report Configuration

The core of the nested report solution is to have a top level "container" report that is primarily a container for custom HTML which can leverage FM based user/group filtering for conditional layout/rendering of buttons and/or reports offer to users:

- One or more iframes to present the "BI content" Cognos reports
- HTML controls (in the examples we'll show – using buttons) to allow selection of reports to appear in the iframes
- Setting the iframe src to the 'Default action URL' which can be found by selecting a report and properties in Cognos

#### 9.9.2.1 STRUCTURING THE PAGE

The following is an example of a Container report page shown in Report Studio:



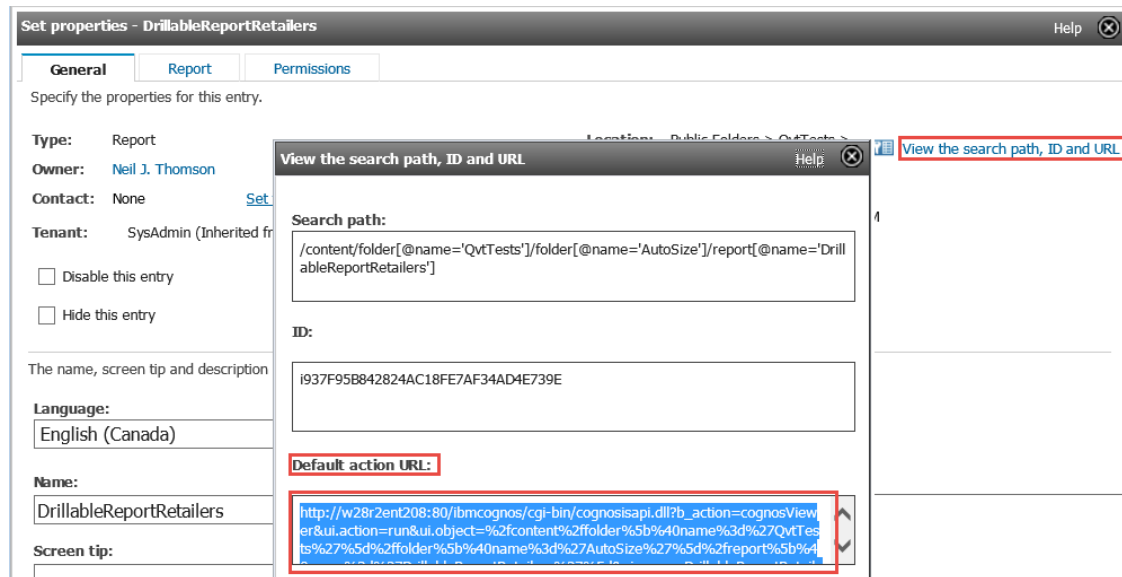
Create a report with a blank body along with a page header which is where the button/tab controls will be presented.

Then add an HTML item to the header

If only presenting a single nested report at a time then add a single HTML item to the body of the report. If presenting 2 or more, it is recommended to add a table and use a table cell for each nested report. Add an HTML item to each table cell into which the iframe HTML will be added.

The next step is to obtain the URL's used to run the "BI Content" reports – which are any Standard Cognos report that can run in a report viewer. While the URL can be hand crafted, the default URL to run a report (which can then be modified as required) can be obtained from the following location in a Report's properties.

Click on "view the search path, ID and URL" then copy the contents of the Default action URL text box.



The following is a typical example:

[http://w28r2ent208:80/ibmcognos/cgi-bin/cognosisapi.dll?b\\_action=cognosViewer&ui.action=run&ui.object=%2fcontent%2ffolder%5b%40name%3d%27QvtTests%27%5d%2ffolder%5b%40name%3d%27AutoSize%27%5d%2freport%5b%40name%3d%27DrillableReportRetailers%27%5d&ui.name=DrillableReportRetailers&run.outputFormat=&run.prompt=true](http://w28r2ent208:80/ibmcognos/cgi-bin/cognosisapi.dll?b_action=cognosViewer&ui.action=run&ui.object=%2fcontent%2ffolder%5b%40name%3d%27QvtTests%27%5d%2ffolder%5b%40name%3d%27AutoSize%27%5d%2freport%5b%40name%3d%27DrillableReportRetailers%27%5d&ui.name=DrillableReportRetailers&run.outputFormat=&run.prompt=true)

Notes:

- *run.outputFormat=* property is not specified, which defaults to "run as HTML".
- The Cognos header and toolbar that displays by default in the report viewer can be hidden with the following additions the "Default action url":
  - ◇ Hide the header: *&cv.header=false*
  - ◇ Hide the toolbar: *&cv.toolbar=false*

Adding this to the URL shown above results in the following

[http://w28r2ent208:80/ibmcognos/cgi-bin/cognosisapi.dll?b\\_action=cognosViewer&ui.action=run&ui.object=%2fcontent%2ffolder%5b%40name%3d%27QvtTests%27%5d%2ffolder%5b%40name%3d%27AutoSize%27%5d%2freport%5b%40name%3d%27DrillableReportRetailers%27%5d&ui.name=DrillableReportRetailers&run.outputFormat=&run.prompt=true&cv.header=false&cv.toolbar=false](http://w28r2ent208:80/ibmcognos/cgi-bin/cognosisapi.dll?b_action=cognosViewer&ui.action=run&ui.object=%2fcontent%2ffolder%5b%40name%3d%27QvtTests%27%5d%2ffolder%5b%40name%3d%27AutoSize%27%5d%2freport%5b%40name%3d%27DrillableReportRetailers%27%5d&ui.name=DrillableReportRetailers&run.outputFormat=&run.prompt=true&cv.header=false&cv.toolbar=false)

*Iframes*

Now the iframe html element can be crafted. The pragmatic approach to using an iframe is to provide it with a default report for initial viewing with the ability to change the iframe contents by the button/tab controls. This results in the following example iframe html element:

```
<iframe name="content" id="content" src="http://w28r2ent208:80/ibmcognos/cgi-bin/cognosisapi.dll?b_action=cognosViewer&ui.action=run&ui.object=%2fcontent%2ffolder%5b%40name%3d%27QvtTests%27%5d%2ffolder%5b%40name%3d%27AutoSize%27%5d%2freport%5b%40name%3d%27DrillableReportRetailers_23_1%27%5d&ui.name=DrillableReportRetailers_23_1&run.outputFormat=&run.prompt=true" frameborder="0" scrolling="no" width="100%" height="100%" ></iframe>
```

Some key aspects to note here:

- Name and ID, which will be needed for Autosizing and to have an HTML button/tab controls
- The **src=** element which has the “Default action URL” as copied from the Report Properties
- The setting of iframe html items such as frameborder, scrolling, width and height

### *Tabs/Buttons to change the iframe content*

Now the controls and which reports will substitute for each iframe need to be defined. While the full HTML, including CSS style elements can be seen in the example report and snippets, we'll concentrate on the key HTML here:

```
<script>

function putContent(n)
{
    var aLinks = document.getElementsByTagName("a");

    for (i=0;i<aLinks.length;i++)
    {
        if(aLinks[i].className == "on")
        {
            aLinks[i].className = "off";
        }

        document.getElementById("link"+n).className = "on";
    }

    switch(n)
    {
        case 1:
            document.getElementById("content").src = "http://w28r2ent208:80/ibmcognos/cgi-bin/cognosisapi.dll?b_action=cognosViewer&ui.action=run&ui.object=%2fcontent%2ffolder%5b%40name%3d%27QvtTests%27%5d%2ffolder%5b%40name%3d%27AutoSize%27%5d%2freport%5b%40name%3d%27DefaultNestedReport_noAutSize%27%5d&ui.name=DefaultNestedReport_noAutSize&run.outputFormat=&run.prompt=true";

            break;

        case 2:
            document.getElementById("content").src = "http://w28r2ent208:80/ibmcognos/cgi-bin/cognosisapi.dll?b_action=cognosViewer&ui.action=run&ui.object=%2fcontent%2ffolder%5b%40name%3d%27QvtTests%27%5d%2ffolder%5b%40name%3d%27AutoSize%27%5d%2freport%5b%40name%3d%27ProductInventoryList-multipage_23_1%27%5d&ui.name=ProductInventoryList-multipage_23_1&run.outputFormat=&run.prompt=true";

            break;
    }
}

</script>

<div id="demo-container">
    <ul id="simple-menu">
        <li><a href="javascript:putContent(1);void(0);" title="DefaultNestedReport_noAutSize" id="link1">
```



```

class="on">DefaultNestedReport_noAutSize</a></li>
<li><a href="javascript:putContent(2);void(0);" title="DrillableReportRetailers" id="link2"
class="off">DrillableReportRetailers</a></li>
</ul>
</div>

```

The above HTML item content defines two buttons/links via the <div id="demo-container"> which correspond to the case 1 and case2 statements in the putContent(n) function.

The case statement gets the iframe by id and replaces the src attribute with the URL to the desired report. Note that there can be multiple buttons each of which can update individual iframes defined in the container report depending via specifying the iframe id.

The following is a screen shot of a Container Report with two iframes (green) with a report in each iframe and two buttons (red) to change the content of each iframe.

QueryVision Report Viewer Webpart

Options Actions

ProductInventoryList ProductLineSource

### Product Inventory List

Year	Month	Product line	Product type	Opening inventory	Quantity shipped
2004	January 2004	Camping Equipment	Cooking Gear	236,107	234,034
2004	January 2004	Camping Equipment	Tents	51,253	53,982
2004	January 2004	Camping Equipment	Sleeping Bags	54,218	53,735
2004	January 2004	Camping Equipment	Packs	47,380	46,851
2004	January 2004	Camping Equipment	Lanterns	92,036	89,096
2004	January 2004	Personal Accessories	Watches	59,299	72,540
2004	January 2004	Personal Accessories	Eyewear	192,737	266,191
2004	January 2004	Personal Accessories	Knives	140,107	139,747
2004	January 2004	Personal Accessories	Binoculars	18,304	18,975
2004	January 2004	Personal Accessories	Navigation	33,879	36,634
2004	January 2004	Outdoor Protection	Insect Repellents	251,413	244,385
2004	January 2004	Outdoor Protection	Sunscreen	197,188	199,805
2004	January 2004	Outdoor Protection	First Aid	35,128	35,457
2004	January 2004	Golf Equipment	Irons	6,582	6,639
2004	January 2004	Golf Equipment	Woods	4,605	4,542
2004	January 2004	Golf Equipment	Putters	25,247	25,297
2004	January 2004	Golf Equipment	Golf Accessories	43,533	47,490
2004	February 2004	Camping Equipment	Cooking Gear	585,040	228,827
2004	February 2004	Camping Equipment	Tents	136,819	59,299
2004	February 2004	Camping Equipment	Sleeping Bags	111,647	54,462
2004	February 2004	Camping Equipment	Packs	83,374	46,105
2004	February 2004	Camping Equipment	Lanterns	196,317	89,382
2004	February 2004	Personal Accessories	Watches	50,145	70,492
2004	February 2004	Personal Accessories	Eyewear	27,307	300,382
2004	February 2004	Personal Accessories	Knives	404,335	134,955
2004	February 2004	Personal Accessories	Binoculars	38,116	18,420
2004	February 2004	Personal Accessories	Navigation	67,798	42,277
2004	February 2004	Outdoor Protection	Insect Repellents	431,141	248,908
2004	February 2004	Outdoor Protection	Sunscreen	470,778	197,070
2004	February 2004	Outdoor Protection	First Aid	81,344	35,388

### DefaultReportForNestedTests

Product brand type
TrailChef
Star
Blue Steel
Hibemator
Firefly
Extreme
Canyon Mule
EverGlow
Husky
Granite
Mountain Man
Polar
Edge
Seeker
Glacier
BugShield
Sun
Course Pro
Hallstorm
Relief

11-Oct-15 1 2:12:33 PM

Top Page up Page down Bottom

## 9.9.2.2 AUTOSIZING CONSIDERATIONS

So far we have talked about nested reports as they would be used in Cognos Connection. However, to get them to Autosize in SharePoint with QueryVision's Report Viewer (or any other techniques) you will need to add JavaScript elements to both the Container report and the nested "BI Content" Reports.

Enabling individual Cognos reports for Autosizing in SharePoint for use as nested "BI Content" Reports does not change.

However, the Container report requires some special treatment. It needs to be upgraded with the following components.

1. Include the following JavaScript libraries:

iframeResizer.min.js

qvtResizer.min.js

iframeResizer.contentWindow.min.js

qvtResizer.contentWindow.min.js

2. Include the QvtReportStyleProperties to specify the resizing configuration for the container report itself
3. Include JavaScript elements for the BI content reports to be presented in the iframes. This can be either for dynamic or static resizing.

Points 1 and 2 can be covered by an HTML item in the Container report page header:

```
<script type="text/javascript">
  var QvtReportStyleProperties = '{"report":{"runAsHtml":{"dynamicSize":{"config":{"sizeMode":"dynamicSize","qv_addHeight":"10px"}}}}';
</script>
<script type="text/javascript" src="../../queryvision/qvtResizer.contentWindow.min.js"></script>
<script type="text/javascript" src="../../queryvision/iframeResizer.contentWindow.min.js"></script>
<script type="text/javascript" src="../../queryvision/qvtResizer.min.js"></script>
<script type="text/javascript" src="../../queryvision/iframeResizer.min.js"></script>
```

Note that the QvtReportStyleProperties are configured for dynamic sizing and adding an additional 10px to the height. Your requirements may vary, but we suggest starting with a value of '10px'

The following needs to be added for each case where the iframe.src is being set, including the default src specified iframe html element definition and each time the src is changed by the button/tab control.

For **dynamic** resizing, the following needs to be added:

```
<script type="text/javascript">
  qvtResizer.setResizeOptions('leftColumnIframe',true,200,1440,'0 0 0',true,'lowestElement',1000);
  if (qvtResizerContentWindow.get_qvtsParent())
  {
    qvtResizer.doResize('content');
  }
</script>
```

Parameters for both the setResizeOptions() and doResize() methods need to be configured.

The parameters are defined as follows:

- setResizeOptions(string <iframe id>, Boolean <loggingEnabled>, integer <minHeight>, integer <maxHeight>, string <bodyMargin>, boolean <autosize>, string <heightCalcMode>, integer <retryTime>)
- doResize(string <iframe id>)

The parameters for the qvtResizer.setResizeOptions set the **dynamic** Autosizing properties for the nested BI content report, and it is highly recommended that the parameter values match those embedded in the BI content report for its JavaScript embedded options

var QvtReportStyleProperties = <JSON parameters>

The parameters and how they matchup between the embedded values and setResizerOptions are defined as follows - Note that all parameters **must** have value.

setResizerOptions Parameter name	QvtReportStyleProperties parameter name	Example value	Comment
iframe id	N/A	'leftColumnIframe'	string - Must be specified with single quotes
logging	ifrz_logging	false	boolean enables logging to the browser console for debugging
min height	minHeight	200	integer
max height	maxHeight	1400	integer
body Margin	ifrz_bodyMarginStr	'0 0 0 0'	string - must be specified with single quotes
autosize	ifrz_autosize	true	boolean - Only use false under advice from QueryVision
auto height calculation method	ifrz_heightCalcMode	'lowestElement'	string - Must be specified with single quotes  Only use other values under advice from QueryVision.
retry time	N/A	0	integer  Parameter currently not used.

If a nested BI content report is Autosize enabled for static height (e.g. height:700px) or is not Autosize enabled, then the height must be specified in pixels as follows:

```
<script type="text/javascript">
    qvtResizer.setIframeHeight('rightColumnIframe', '700px');
</script>
```

The parameters are as setIframeHeight(string <iframe id>, string <height in pixels>).

Note:

- Both are strings that must be enclosed in single quotes (as per the example, above).
- Only 'px' (pixels) is supported, setting '100%' is not effective.

For both the dynamic or static Autosizing within a container report, it is recommended to insert such script where the iframe is specified or where the iframe src is updated (e.g. on a button push).

Using the example outlined above the following specifies a dynamic resizing for a report where the iframe html element is specified:

```
<iframe name="content" id="leftColumnIframe" src="http://W28R2ENT205:80/ibmcognos/cgi-bin/cognos.cgi?b_action=cognosViewer&ui.action=run&ui.object=%2fcontent%2ffolder%5b%40name%3d%27QvtTest%27%5d%2ffolder%5b%40name%3d%27Autosize%27%5d%2freport%5b%40name%3d%27DrillableReportRetailers_2
```

```
3_1%27%5d&ui.name=DrillableReportRetailers_23_1&run.outputFormat=&run.prompt=true&cv.header=false&cv.toolbar=false" frameborder="0" scrolling="no" width="100%" height="100%" ></iframe>
<script type="text/javascript">
    qvtResizer.setResizeOptions('leftColumnIframe',true,200,1440,'0 0 0 0',true,'lowestElement',1000);
    if (qvtResizerContentWindow.get_qvtIsParent())
    {
        qvtResizer.doResize('content');
    }
</script>
```

The following shows examples of static resizing (case 1:) and dynamic sizing (case 2:) when the iframe.src values are updated by the button/tab html control logic.

```
switch(n)
{
    case 1:
        document.getElementById("content").src = "http://W28R2ENT205:80/ibmcognos/cgi-bin/cognos.cgi?b_action=cognosViewer&ui.action=run&ui.object=%2fcontent%2ffolder%5b%40name%3d%27QvtTest%27%5d%2ffolder%5b%40name%3d%27Autosize%27%5d%2freport%5b%40name%3d%27DefaultNestedReport_noAutSize%27%5d&ui.name=DefaultNestedReport_noAutSize&run.outputFormat=&run.prompt=true&cv.header=false&cv.toolbar=false";
        qvtResizer.setIframeHeight('content', '700px');
        break;

    case 2:
        document.getElementById("content").src = "http://W28R2ENT205:80/ibmcognos/cgi-bin/cognos.cgi?b_action=cognosViewer&ui.action=run&ui.object=%2fcontent%2ffolder%5b%40name%3d%27QvtTest%27%5d%2ffolder%5b%40name%3d%27Autosize%27%5d%2freport%5b%40name%3d%27ProductInventoryList-multipage_23_1%27%5d&ui.name=ProductInventoryList-multipage_23_1&run.outputFormat=&run.prompt=true&cv.header=false&cv.toolbar=false";
        qvtResizer.setResizeOptions('content',true,200,1440,'0 0 0 0',true,'lowestElement',1000);
        if (qvtResizerContentWindow.get_qvtIsParent())
        {
            qvtResizer.doResize('content');
        }
        break;
}
```