Sitecore OMS 1

Report Designer Cookbook

A Guide to Working with Sitecore OMS Reports

Table of Contents

Chapter 1

Introduction

This cookbook is not aimed at one specific end-user. Instead it contains step by step instructions that can be used by anyone working with OMS reports from a marketer to a .NET developer.

The examples included here use the Nicam Web site running on Sitecore OMS and a SQL Server database. The Nicam site contains a large amount demo data which makes it easier to test several different report scenarios without the need to run them in a live environment.

Note - The instructions in this guide include SQL queries based on SQL Server syntax which may not be compatible with other database management systems such as Oracle.

- **Chapter 1 — Introduction**
  This chapter is a description of the content, aims, and the intended audience of this cookbook.

- **Chapter 2 — Report Design Tools**
  This chapter provides an overview of the tools that you need to create and edit reports.

- **Chapter 3 — Editing Reports**
  This chapter covers the basics, from changing font styles to adding charts and images to reports.

- **Chapter 4— The Analytics Database**
  This chapter provides an overview of the Analytics database outlining some useful table groupings to use when designing reports.

- **Chapter 5 — Creating Reports**
  This chapter explains how to create a report from scratch.

- **Chapter 6 — Creating Report Filters**
  This chapter explains how to create a report filter from scratch and how to add it to a report.

- **Appendix**
  This section contains some useful additional information.

Chapter 2

Report Design Tools

This chapter gives you an overview of the two client options available for creating and editing reports in Sitecore OMS.

- Stimulsoft Web Reports Designer

- Stimulsoft Windows Reports Designer

2.1 Stimulsoft Web Reports Designer

Official name: *Stimulsoft Reports Designer.Web 2009.2*

The Web reports designer is the standard report designer available in the Sitecore OMS. In **Sitecore Analytics**, click the **Design** button at the top of any report to access this client.
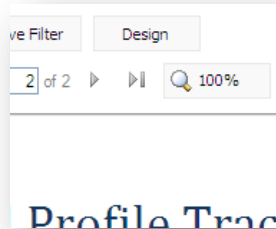
In the Web reports designer, you can edit the format of existing reports or create new reports by making and editing a duplicate copy. Also, you can use the Web reports designer to add images, hyperlinks, and charts to your reports.

The Web reports designer has some limitations. For example, you cannot use keyboard shortcuts and you cannot easily create SQL queries using this tool. However, it is possible to achieve the same end results as with the Windows Reports Designer.

The Web reports designer user interface displayed in Internet Explorer:
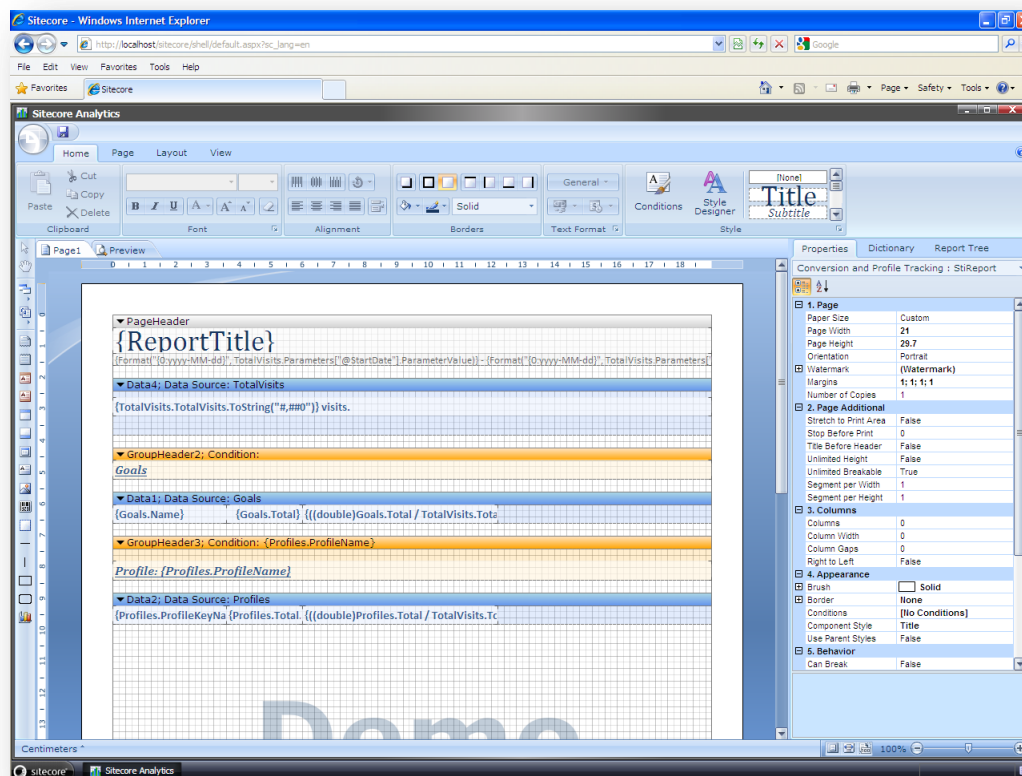
2.2　　　　Stimulsoft Windows Reports Designer

Official name: *Stimulsoft Reports.Net 2009.2*

The Stimulsoft Windows Reports Designer is available for free trial download from the Stimulsoft Web site. It is a richer and more powerful reports designer suitable for developers or report analysts. It gives more precise control over the report design process. Ultimately, you can achieve the same end results using the Web reports designer but if you want to develop a significant number of reports for your Web site, we would recommend that you purchase this tool.

Advantages of using the Windows Reports Designer:

- Create new reports from scratch

- Use wizards to quickly create new reports

- Reports load quickly (the client does not run in a browser window)

- Use a number of keyboard shortcuts (including Ctrl Z)

- Preview functionality

- Create complex SQL queries using the Query Builder

- Import and export formatting styles

The Windows Reports Designer client user interface:
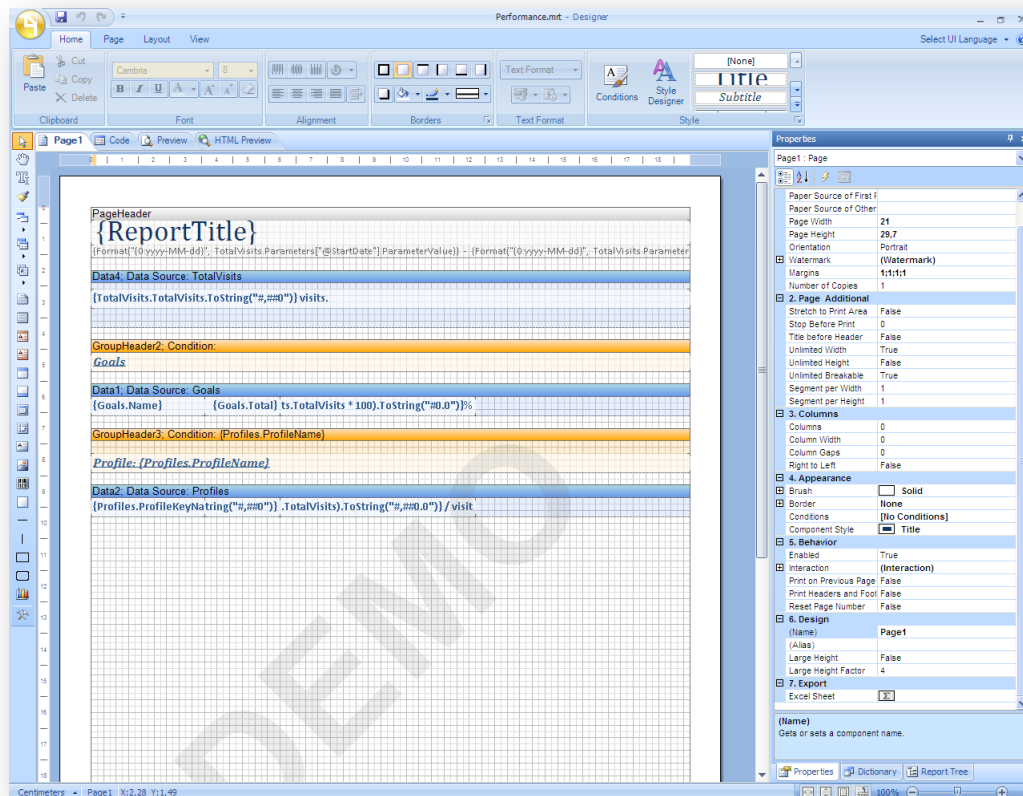
Chapter 3

Editing Reports

This chapter introduces basic report designer concepts such as editing text and adding charts and images to existing reports:

- OMS Reports

- Formatting Text

- Adding Charts to Reports

- Adding Images to Reports

- Adding Links to Reports

3.1   OMS Reports

In Sitecore OMS, a report consists of two components:

- Sitecore report definition item
- Stimulsoft `.mrt` report file

3.1.1   Sitecore Report Definition Item

A Sitecore report definition item is the same as any other content item stored in the Sitecore content tree. To access report definition items, open the Content Editor and navigate to the following location:

`/sitecore/system/Settings/Analytics/Reports/Reports`

Each report item contains the following fields:

- File Name
- Filters
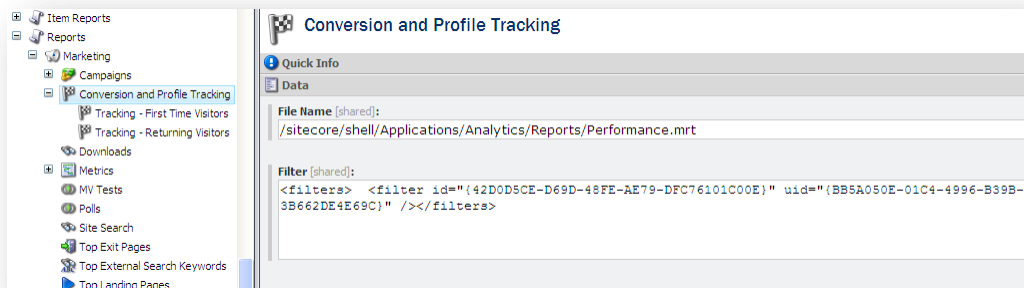
File Name

This field contains a path to the report.mrt file. For example:

`/sitecore/shell/Applications/Analytics/Reports/Performance.mrt`

If you create a report, you must also create a link between the Sitecore definition item and the Stimulsoft mrt file. Enter the correct path to the .mrt file in this field.

Filters

If a report contains filters the Filter field displays metadata that is added automatically when you run the report filter.

You can leave this field blank.

3.1.2   Stimulsoft .mrt File

When you save a report, all the configuration data for the report is saved to an XML file with the .mrt extension. This data in this file includes the following:

- Connection string
- SQL query
- Layout and presentation
- Formatting

The *report.mrt* file is a Stimulsoft file type and is stored in the following location in your Web site file system:

```
wwwroot\<sitename>\WebSite\sitecore\shell\Applications\Analytics\Reports\
```

When you edit a report in either the Web or Windows report designer the changes you make are saved directly to the .mrt file. If a report contains multiple filters, then several reports could be based on the same report .mrt file, so it is easy to overwrite an existing report.

Note

If you create multiple new reports create a new folder for your reports to avoid confusion with the standard Sitecore reports.

3.1.3 Opening a Report in the Web Reports Designer

You can open and edit a report in the report viewer using the built-in Web reports designer. Before making any changes to existing reports, you must duplicate the Sitecore report definition item and the report .mrt file first. To do this you need the appropriate permissions to access to the Sitecore content tree and file system.

IMPORTANT

Some report files, such as `Latest Sessions.mrt` are used by multiple reports, so changes to one report file can affect a number of reports. Duplicating a report and saving it with a different name helps to avoid this problem.

For more detailed steps on how to duplicate a report, see Chapter 5, *Creating Reports*.

To open a report in the Stimulsoft Web reports designer client:

1. In the Sitecore Analytics content tree, click on the report you want to edit.
2. Click **Design** at the top of the report.
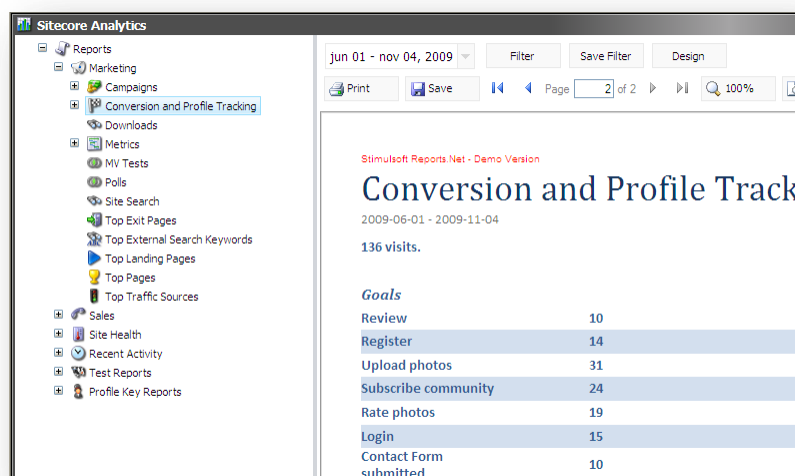
The *Conversion and Profile Tracking* report in the Stimulsoft Web reports designer.

3.1.4   Opening a Report in the Windows Reports Designer

The Windows reports designer allows you to open and edit a report directly from the `.mrt` report file and allows you to create new reports from scratch.
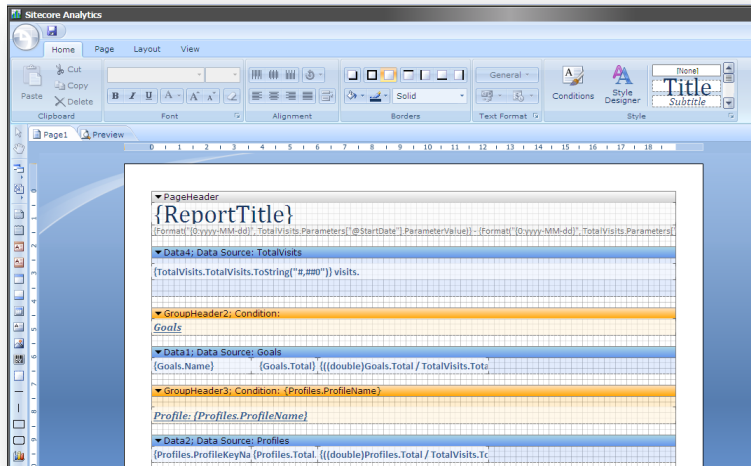
To open a report in the Windows reports designer client:

3. Click the Start menu on your computer and then open the Windows reports designer from the Stimulsoft group in All Programs.

4. Click the Stimulsoft menu, then click **Open Report** and browse to the `.mrt` file you want to edit.
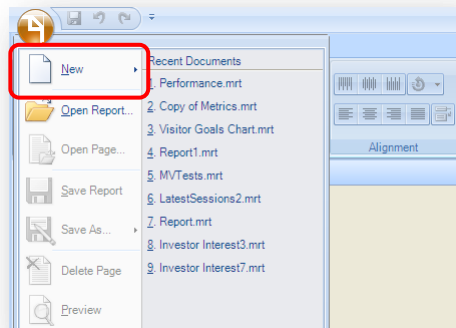
5. Open. `Performance.mrt` to edit the *Conversion and Profile Tracking* report
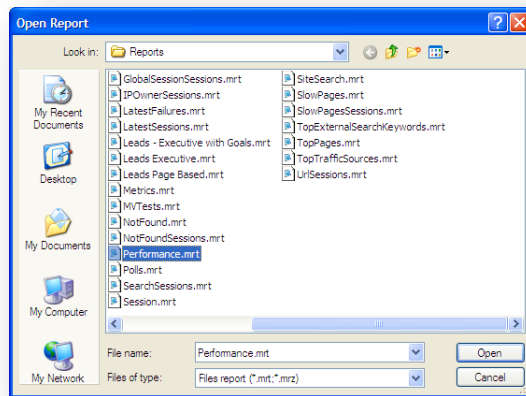
`Performance.mrt` in the Windows reports designer.
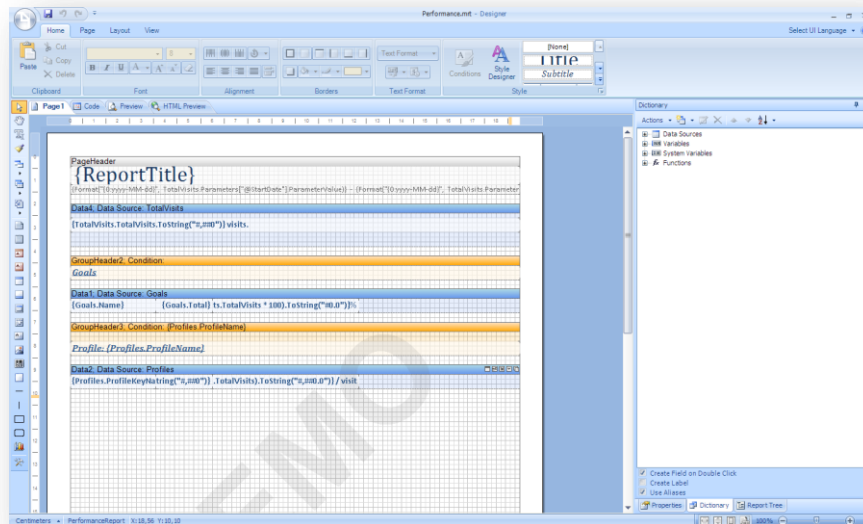
All report the `.mrt` files are stored in the following location:

```
C:\Inetpub\wwwroot\<Web site
name>\WebSite\sitecore\shell\Applications\Analytics\Reports
```

3.2 Formatting Text

This section explains how to format text in an existing Sitecore OMS report. Using the Web reports designer, you can easily change:

- Text colors
- Font style
- Font-size
- Bold
- Underline
- Italic

3.2.1 Formatting Text Using the Ribbon

In Sitecore Analytics, open the *Conversion and Profile Tracking* report in the Web reports designer. Make a duplicate of the report first and then make the following changes:

1. In *GroupHeader2,* click the text box *Goals* and then in the **Font** group, click Text Color.
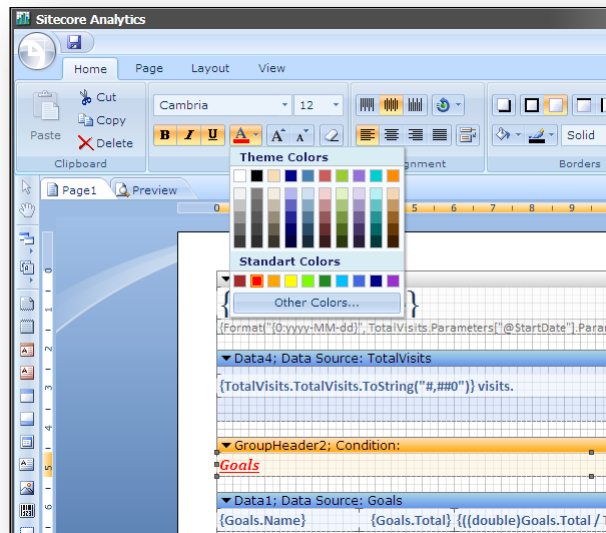
2. Select a different text color, such as red.

3. In the Stimulsoft menu, click **Save Report** to save your changes.

4. Click **Exit** to close the report designer.

5. In **Sitecore Analytics** preview your report.

You can use the Stimulsoft ribbon in exactly the same ways as you would use the Microsoft Word ribbon to change text size and to apply other styles such as bold, underline and italic.

3.2.2 Formatting Text Using Properties

To format text using the **Properties** panel:

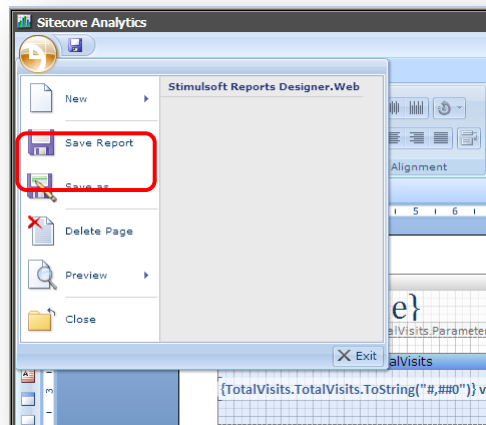1. In *GroupHeader2,* click the text box *Goals* and then in the **Properties** panel in **Text Brush**, **Color** field select *Red*.
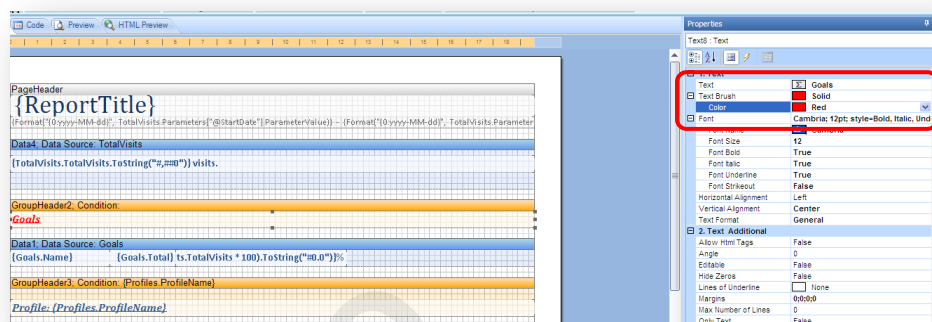
2. To save your changes, click **Save Report**.

3. Click **Exit** to close the report designer.

4. In **Sitecore Analytics** preview your changes.

The **Properties** panel gives you more control than the ribbon over formatting text. For example, you can enter hexadecimal values directly into the **Text Brush**, **Color** property. You can also set additional properties to make more advanced changes. For example you can allow HTML tags to be

used to format text by setting **Allow Html Tags** property to *True*.
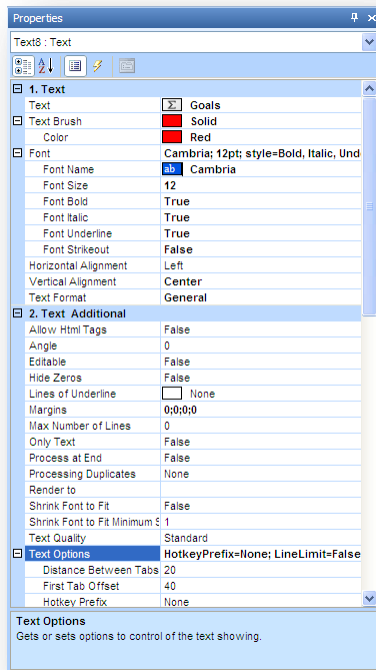
3.2.3 Styles

You can also control the format of your text, headings, and charts using *Styles*.

In the Stimulsoft ribbon, **Style** group you can access all your style options.
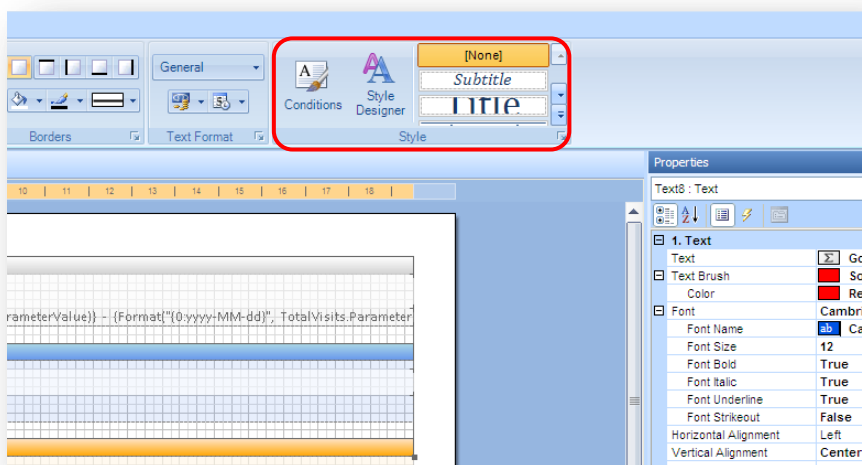
To view all styles or to add a new style, open the **Style Designer** window.
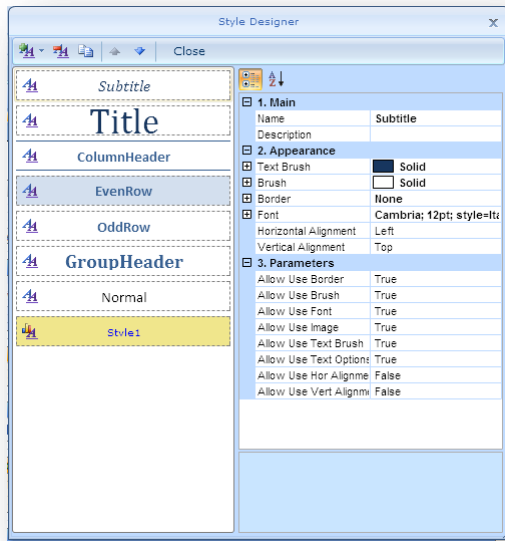
All the styles contained in the Style Designer are stored in a `.sts` file. You can export styles that you want to save or import new styles from a saved `.sts` file.

The **Conditions** window allows you to create conditional formatting conditions based on values or expressions.
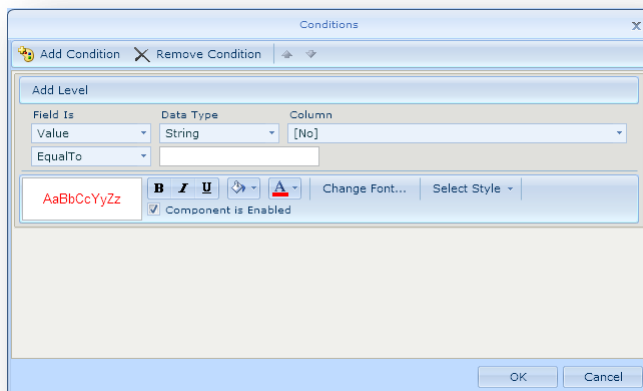
3.2.4   Applying Styles

To apply a different style to a report:

1. Click on a text box in a report that you want to edit.

2. Select a text box and click on one of the styles visible in the right hand side of the **Styles** panel. For example, *GroupHead*.
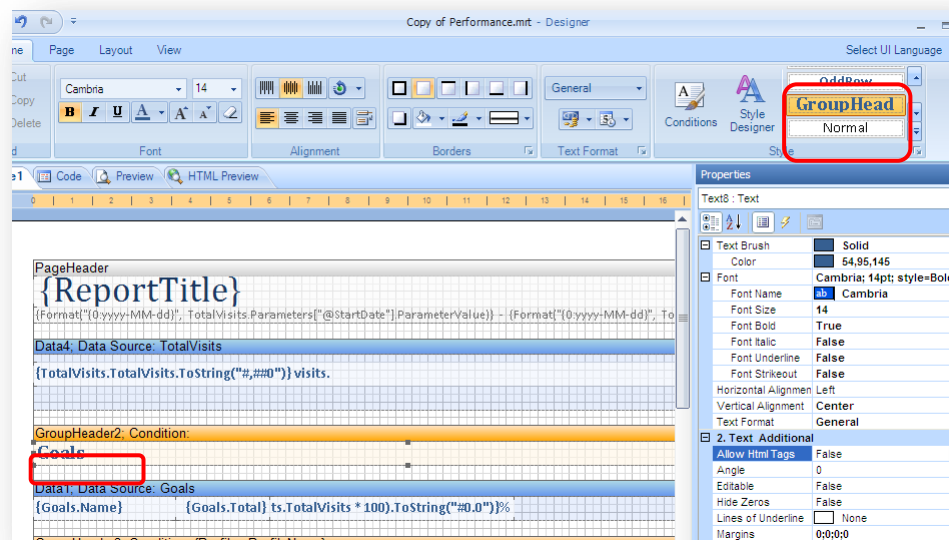
3. Save your changes and preview your report in **Sitecore Analytics**.

3.3 Adding Charts to Reports

By providing a visual overview of data, charts and graphs can make reports more meaningful at a glance. This can be particularly useful to marketing and sales executives who quickly need to identify trends or potential leads.

You can easily add a chart or graph to a report using the Web reports designer.

The following example explains how to add a bar chart to an existing Sitecore report.

Start by opening a report in Sitecore Analytics, such as *Conversion and Profile Tracking* (first make a duplicate of this report to avoid overwriting the original). It is easier to use an existing report so we can re-use the data contained in the report.

You are going to make a simple bar chart that only displays information related to conversions, so first remove any information about profiles that appears on this report. If you wish, you can also change the heading.

3.3.1 Creating a Bar Chart

To create a new bar chart:

1. In **Sitecore Analytics**, select the report *Conversion and Profile Tracking*.

2. Click **Design** at the top of the page, to open the report in the Web reports designer.

3. In the report designer toolbox, click **Chart** and with the pencil tool draw a container for your bar chart.
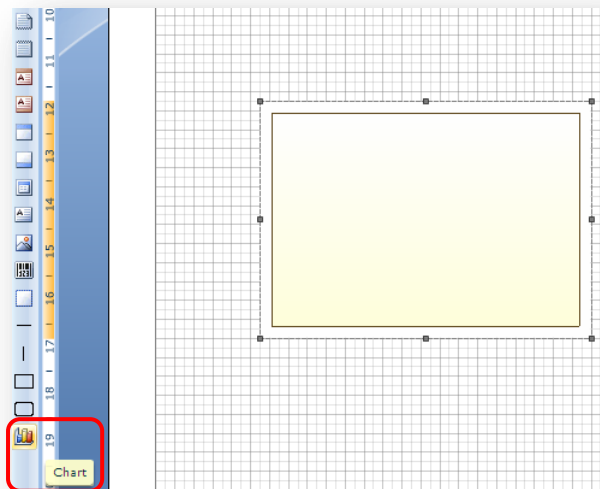
4. Double click the box you created to open the **Series Editor** window.
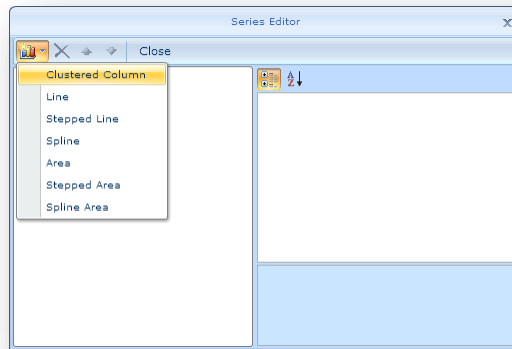
5. Select *Clustered Column* as your chart type.

6. Once you have chosen a chart type, the right hand panel of the **Series Editor** window displays properties that you can set on your chart. Select property **4. Argument**.

7. In the **Argument Data Column** field, enter *Goals.Name* in the pop-up **Text Editor** window. This is how you set the X axis which appears at the base of your chart. *Goals* refers to the database table and *Name* is the column or field in the table that you have selected.
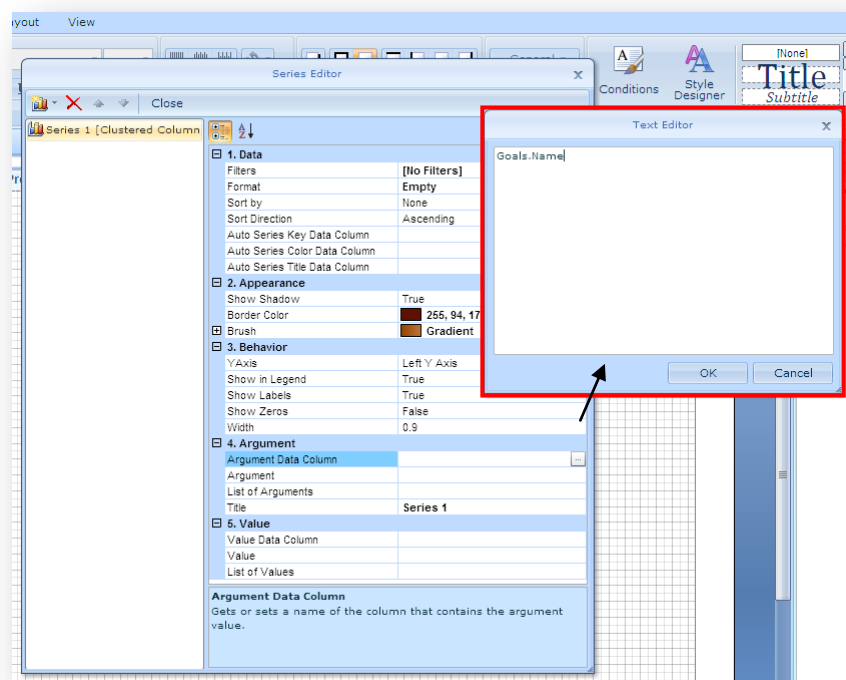
8. In the **Value Data Column** property, enter *Goals.Total*.

9. Select any other field in the **Series Editor** and then click **Close** to save your changes and exit the **Series Editor** window.

10. Save your report, close the report designer and preview the report in **Sitecore Analytics**. Your chart should now look something like this:
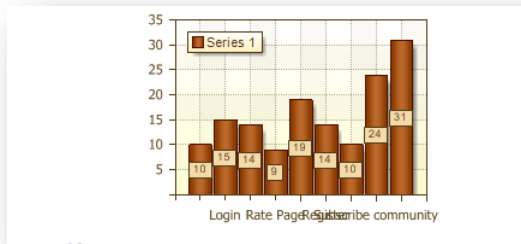
Next you can try to improve the formatting of your chart.

3.3.2 Formatting Charts

In the report designer, using the bar chart that you created, add styles and set properties to add color and other formatting.

You can place a chart on any other component of a report such as a report header, group header, data source or text box but usually charts appear above or below the other components on the report.

Resizing a Chart

1. Click on the chart you created earlier.

2. Make sure your chart is placed directly below the **Goals** data band. Then move the mouse over the chart to see the re-size arrows.
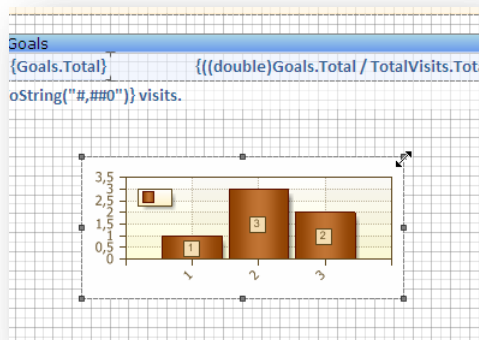
3. Adjust the chart size to fill the full width of the report work area.
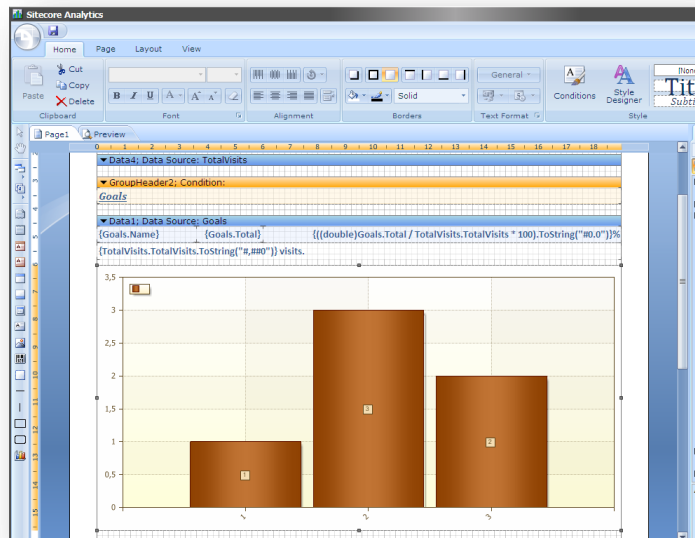
Changing Chart Colors

In the report designer, you can change the color scheme of a chart using pre-defined styles. You can also create your own custom color schemes.

How to change a chart color scheme:

1. In the report designer, select your chart.

2. Click the **Properties** tab on the right-hand side of the page to view the chart properties.

3. In property number **1. Chart**, click **Style,** and then click the drop-down menu to see all the available chart styles.
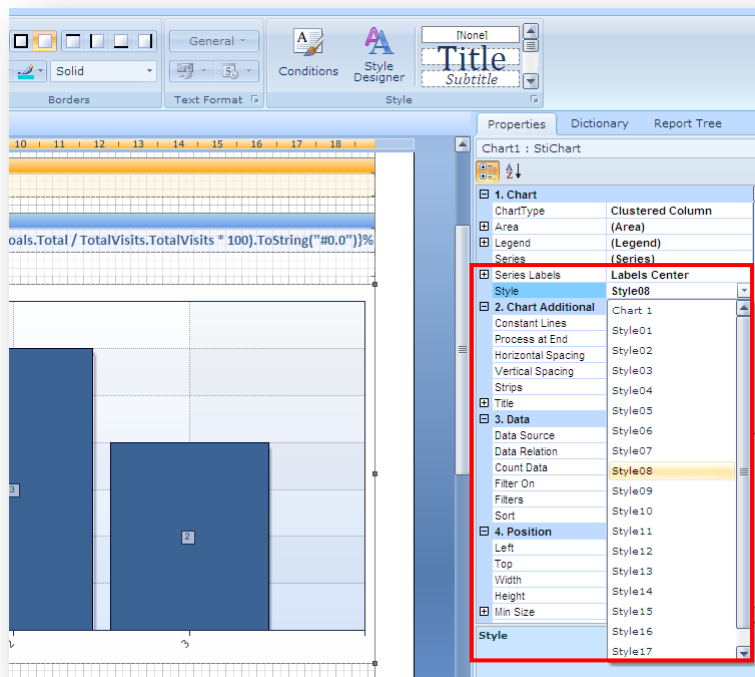
4. Select *Style 08*.

   When you select a style, you see a preview of the selected color scheme displayed in the report designer.

5. Save your changes and close the report designer.

6. Preview your report in **Sitecore Analytics**. Your report should now look like this:
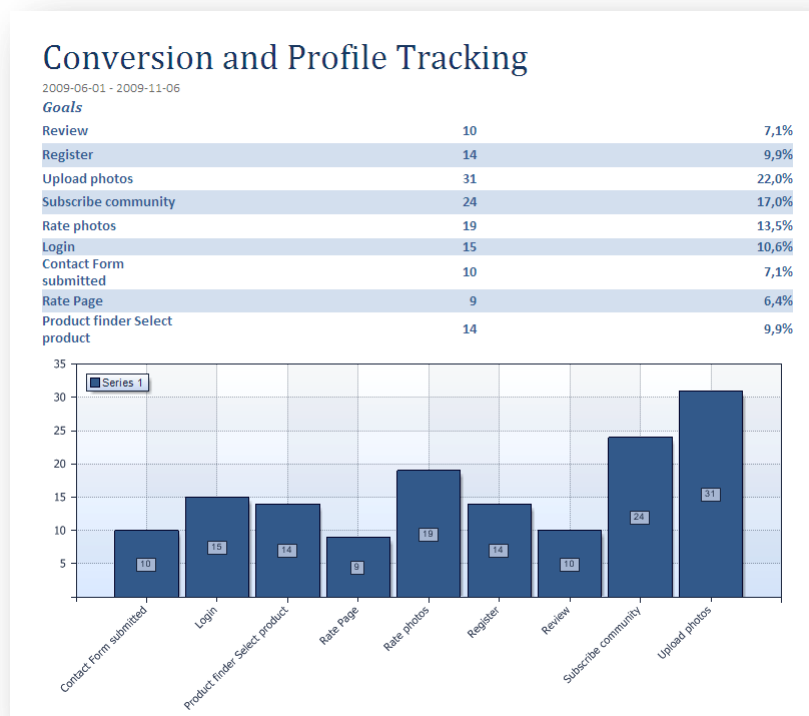
Tip
The Web reports designer takes a few seconds to load each time you open and close it. Keep a second instance of Sitecore Analytics open to make it quicker to preview your changes.

Adding Chart Labels

You can add labels to your charts. For example, add labels to the X and Y axis, a title or add a legend. A legend is like a key that you can refer to if you have multiple columns of different colors

Overview of Chart Labels:

Use the **Properties** panel to add labels to your chart. There are many different properties that you can set on a chart, so start by adding the following:

- Titles to the X and Y axis

- A key or legend

- Labels to appear on each column

To do this, use property **1. Chart**.

The following table gives you an overview of some of the main properties in **Chart**.

| Property (example)	Description	Value

Area
 • X Axis
   ○ Title
     ▪ Text
 • Y Axis
   ○ Title
     ▪ Text
 • Color Each
	Area contains several properties that allow you to control the format of the chart area. This could include the border, the background, or labeling the X and Y axis.
	
	 Color Each lets you decide whether you want all columns to be the same color or different.
		Enter a title to appear on the X axis (below the horizontal line at the base of the chart)
		
		 Enter a title to appear on the Y axis (alongside the vertical line at the side of the chart)
		 True or *False*

Legend • Title	*Legend* refers to the key. This is particularly useful in Pie Charts when the key provides more information about the different colored segments of the chart.	Enter a title to appear in the key. For example, *Goals*
Series Labels • Visible	Set *Series Labels* properties to add labels or to change the formatting of the columns that appear in your chart. For example, you can add a label to each column and choose whether to make them visible or not.	*True* or *False*

To add a chart legend label:

1. In the Web report designer, select your chart.

2. In the **Properties** panel, select **Chart**, and then expand the **Legend** node.

3. In the **Title** field enter *Goals*.
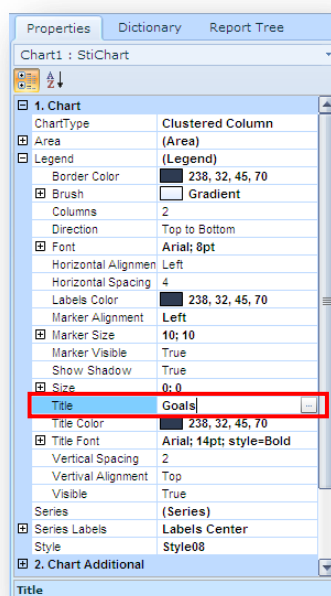
To add labels to the X and Y axis of your chart:

1. In the report designer, select your chart.

2. In the **Properties** panel, select **Chart**, and then expand the following properties:

   o Area

   o X Axis

   o Labels

   o Angle

3. In the **Angle** property, enter *45*.
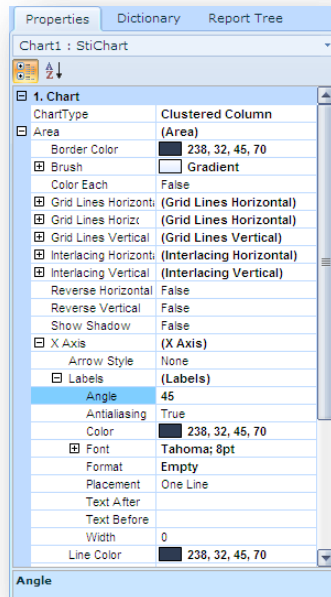
This means that labels displayed on the X Axis will be angled by 45 degrees, so that all text is visible in the chart.

To add a chart title:

1. In the report designer, select your chart.

2. Expand the following properties:

   o Chart Additional

   o Title

   o Text

3. In the **Text** property, enter the name *Conversion Tracking*.

4. Set the **Visible** property to *True*.

5. Chart Additional allows you to set several other properties.

   Set the following additional properties on your chart:

Property	Value
Alignment	Center
Dock	Bottom
Font Size	18
Font Bold	True
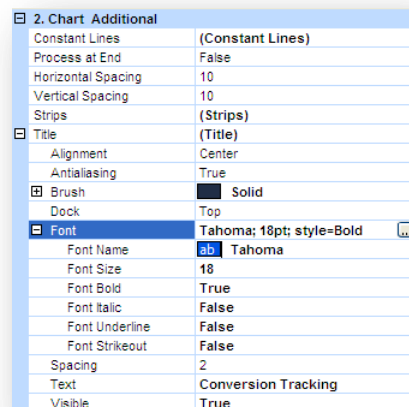
6. Save your chart and preview it in Sitecore Analytics.
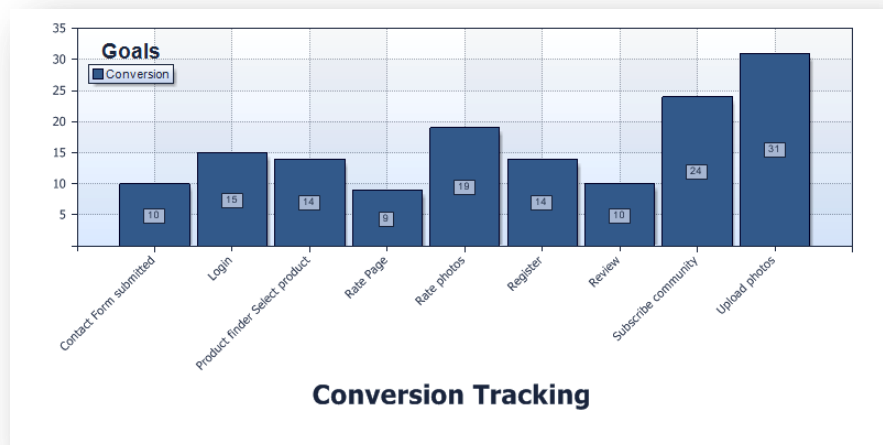
3.3.3 Creating a Pie Chart

Create a report that displays visitor goals as a pie chart.

Preparation:

1. In **Sitecore Analytics**, select the report *Conversion and Profile Tracking*.

You will use this report as the basis for your pie chart report. This report already contains a Goals data source that you can re-use.

2. Find the report definition item in the content tree.

```
/sitecore/system/Settings/Analytics/Reports/Reports/Marketing/Conversion and Profile Tracking
```

3. Duplicate this report and rename it *Visitor Goals Chart*.

4. Locate the `.mrt` file associated with this report: `Performance.mrt`.

```
wwwroot\<site name>\WebSite\sitecore\shell\Applications\Analytics\Reports\
```

5. Rename `Performance.mrt` to `VisitorGoalsChart.mrt`.

To create a new pie chart:

1. In **Sitecore Analytics**, refresh the reports content tree.

2. Select the report *Visitor Goals Chart*.

3. Click **Design**, to open the Web report designer.

4. Remove all data bands and headers that refer to profiles. To do this, click on a data band and then in the ribbon, click **Delete**. Your report layout should now only display goals:
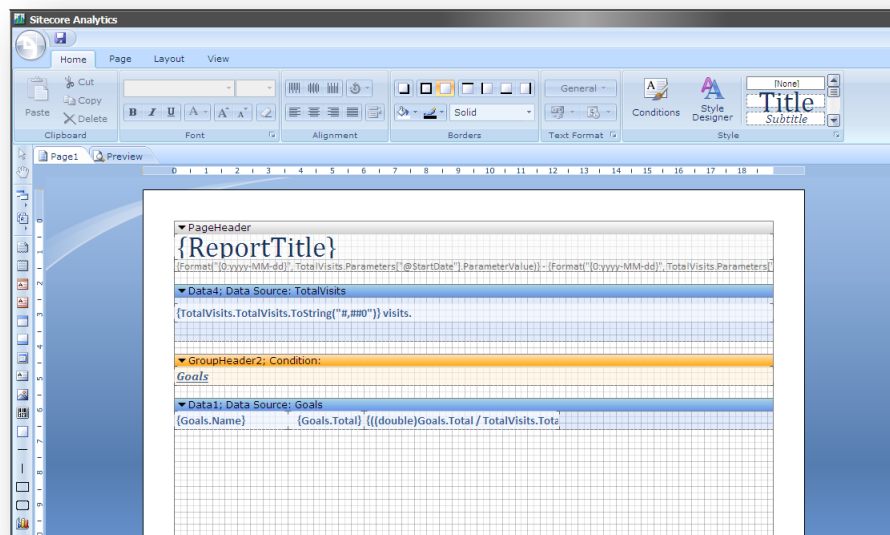
5. In the toolbox, click **Chart**.

6. Using the pencil tool, draw a box for your chart on the report layout.
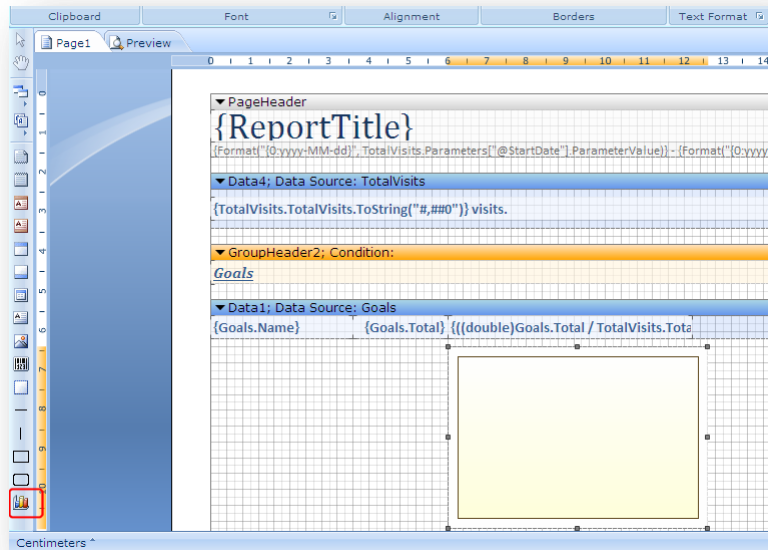
7. Select the empty box and in the **Properties** panel, select *Pie* from the **Chart Type** property.
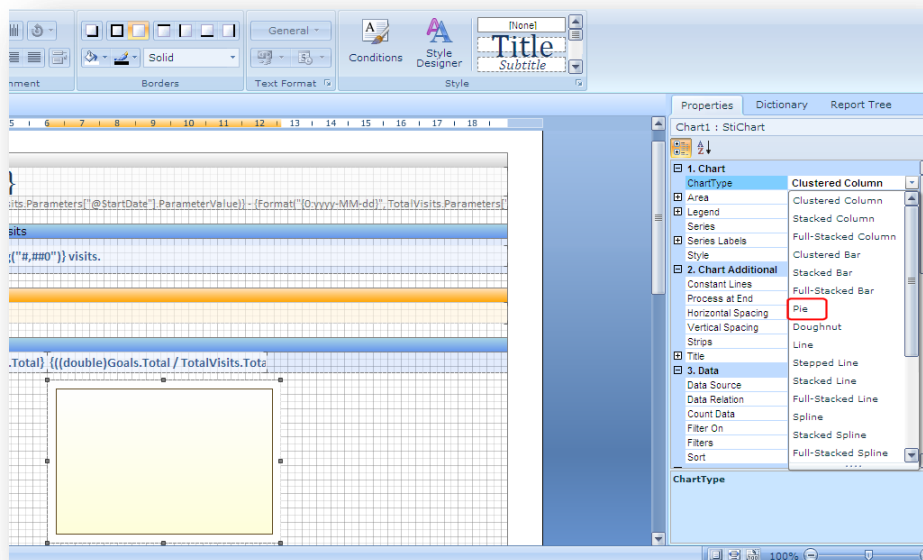
8. Double click the pie chart to open the **Series Editor** window.

9. In the **Series Editor** window, click **Pie** to display the properties of the pie chart.
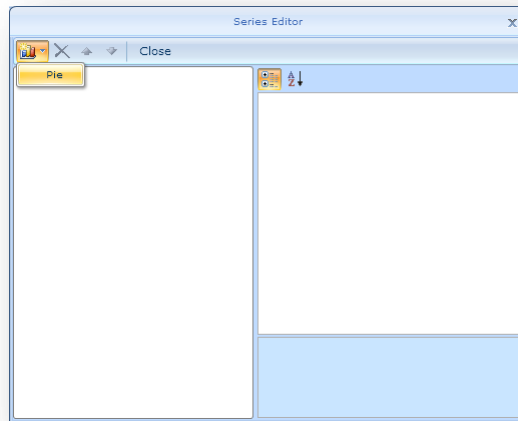
10. In the **Series Editor** window, you can select columns to display in your pie chart.
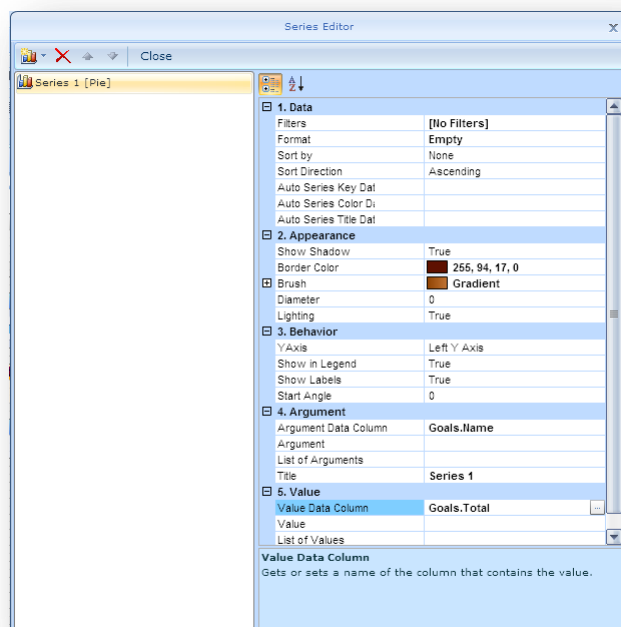
11. In the *Argument Data Column* property, enter *Goals.Name* or type it into the **Text Editor** window.

12. In the *Value Data Column* property, enter *Goals.Total.*

13. To save your changes, select any field in the **Series Editor** window and then click **Close**.

14. Select your pie chart. In the **Properties** panel, click **Chart**, **Style** and then select *Style08* from the drop-down list.
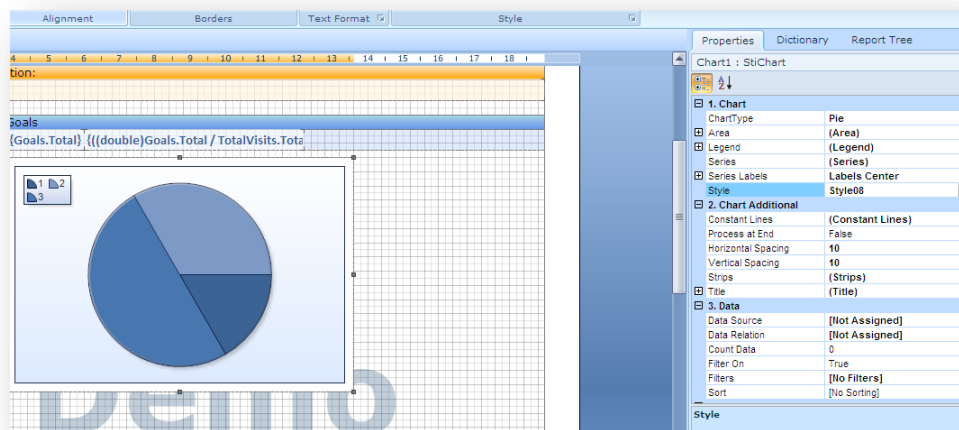
15. To add a legend and to format it, set the following properties in the **Properties** panel:

Property	Value
Legend, Title	Goals achieved
Legend, Columns	1
Legend, Direction	Right to Left
Series Labels	Outside
Series Labels, Legend Value Type	Argument

16. In the Stimulsoft menu, click **Save Report**.

17. In **Sitecore Analytics**, preview your Goals pie chart.

Visitor Goals Chart

2009-06-01 - 2009-11-19

153 visits.

Goals

Goal		
Review	10	6,5%
Register	14	9,2%
Upload photos	31	20,3%
Subscribe community	24	15,7%
Rate photos	19	12,4%
Login	15	9,8%
Contact Form submitted	10	6,5%
Rate Page	9	5,9%
Product finder Select product	14	9,2%

Goals achieved

- Contact Form submitted
- Login
- Product finder Select product
- Rate Page
- Rate photos
- Register
- Review
- Subscribe community
- Upload photos
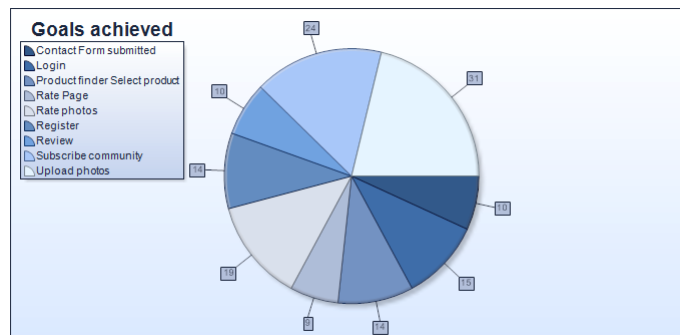
3.4 Adding Images to Reports

In the Web reports designer, if you want to add an image to report, enter a URL link to the image. If you use the Windows reports designer, you can also browse your computer for images.

In this section, add an image to the *Profile Key Report*. This report is based on the Sitecore *Leads - Executive - Prospects* report. The .mrt file for this report is called `Leads Executive.mrt`.

Alternatively, add an image to any of your existing reports but remember to duplicate the report and the .mrt file first.

3.4.1 Linking to an Image in the Media Library

To add an image to a report:

1. Open *Leads - Executive - Prospects* in the report designer.

2. In the tool box, click **Image** and drag a new image placeholder onto your report.

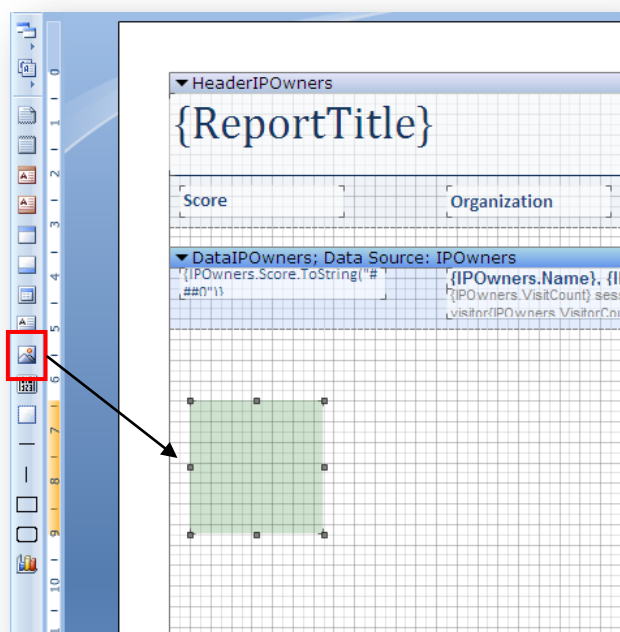3. Double click the box to open the **Image** dialog box.

4. In the **Image** dialog box, you get the following options:

Option	Description
Data Column	Link to a BLOB image stored in a database
Image Data	Enter an image variable, for example {MyImageVariable}
URL	Enter a URL string that links to an image in the Media Library or on an external server.

5. Link to an image stored in the Media Library. Open the Media Library and locate the image *header_logo* in the content tree.

6. Select the image and in the ribbon, **Media** group, click **View**.
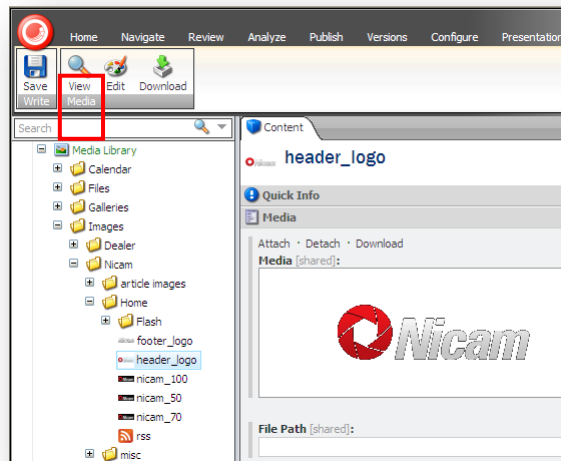
7. Right-click the pop-up window and select properties.
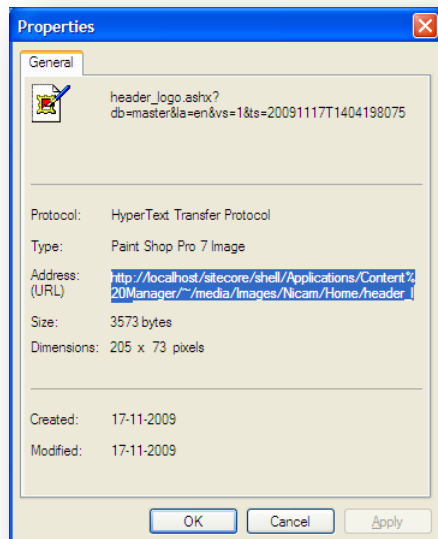
8. In the **Properties** window, in the **Address** field, highlight and copy the image URL.

9. In the report designer **Image** dialog box, paste the URL into the **Image URL** field.
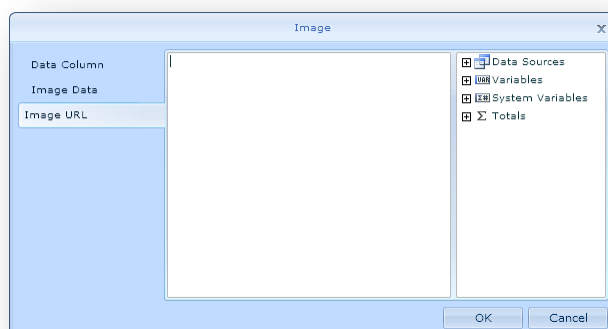
For the image *header_logo*, the URL should look like this:

```
http://localhost/sitecore/shell/Applications/Content%20Manager/~/media/Images/Nicam/Ho
me/header_logo.ashx?db=web&la=en&vs=1&ts=20091109T1305394906
```

10. Move your image to a better position on your report. For example, in the top right corner, next to the report title.

    You can easily add an image box to a report header or any other component of an OMS report.

11. Click **OK** to close the **Image** dialog box.

12. Save your changes and preview your report in **Sitecore Analytics**.

3.4.2 Linking to Images on an External Server

If you have images stored on an external server, you can link to them using the URL field.

To add a link to an image on an external server:

1. Use the toolbox to add an image box to your report.

2. Double click the image box to open the **Image** dialog box.
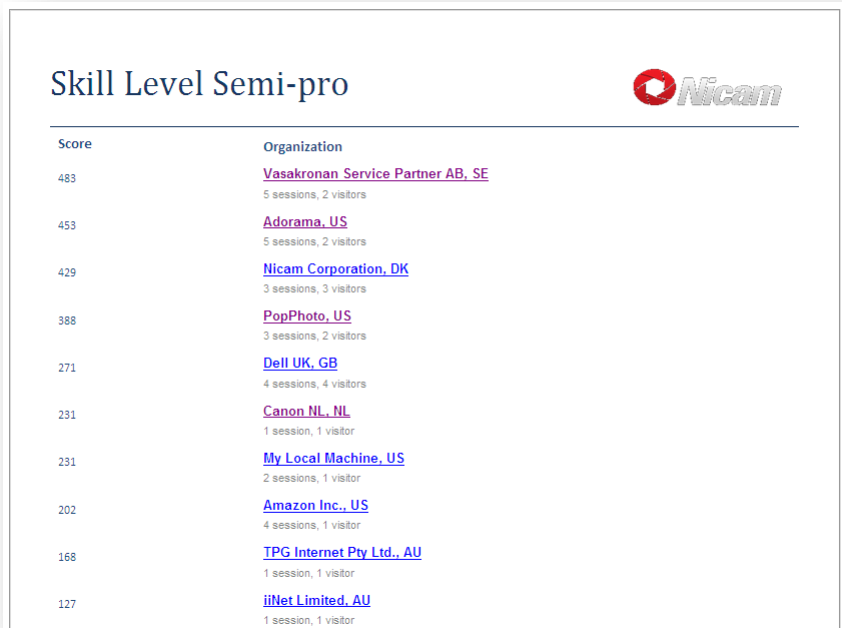
3. In the **URL** field, enter a link to the location of your image. In this example, I have another version of the Nicam logo stored on an external Web site. Most images stored in this way have a full URL string in their properties. Open the image properties and copy the image URL string.

    The URL I copied for my image looks like this:

```
http://lh5.ggpht.com/_0xuwaz1htow/Svghj9admCI/AAAAAAAACog/6pSI0dJ1ps0/nicam_100.jpg
```

4. In the **Image** dialog box, paste the URL into the **Image URL** field.

5. Click **OK** and save your report.

6. Preview your report in **Sitecore** Analytics.

3.4.3 Uploading Images

In this section, upload an image from your local computer. Use the same report and add a second image to the title section. To do this you need to use the Stimulsoft Windows reports designer client.

1. Open the Windows reports designer client.

2. Add another image box to the title header of your report.

---
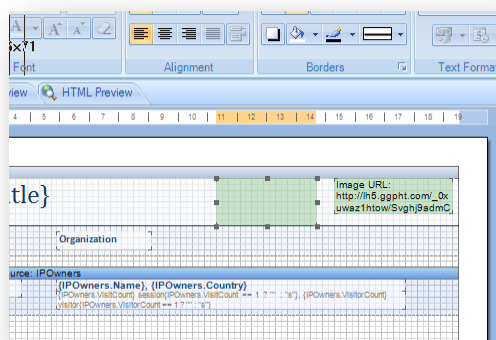
3. Double click on the image box to open the **Image** dialog box.
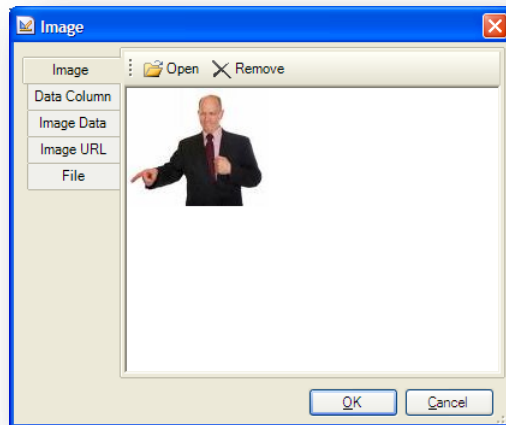
4. In the **Image** field, click **Open** to browse your computer for suitable images. Find an image and then click **OK**.

5. Preview your report in Sitecore Analytics.
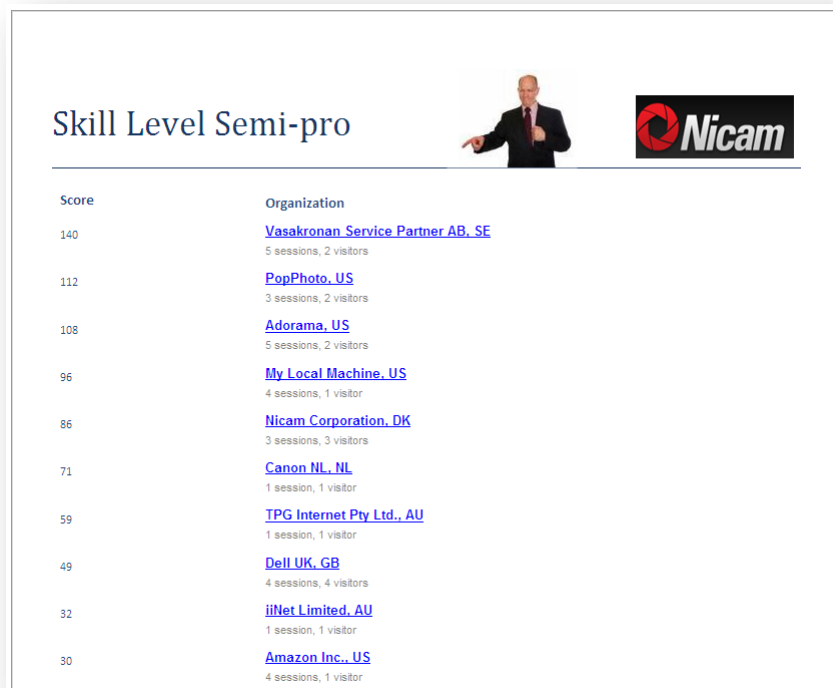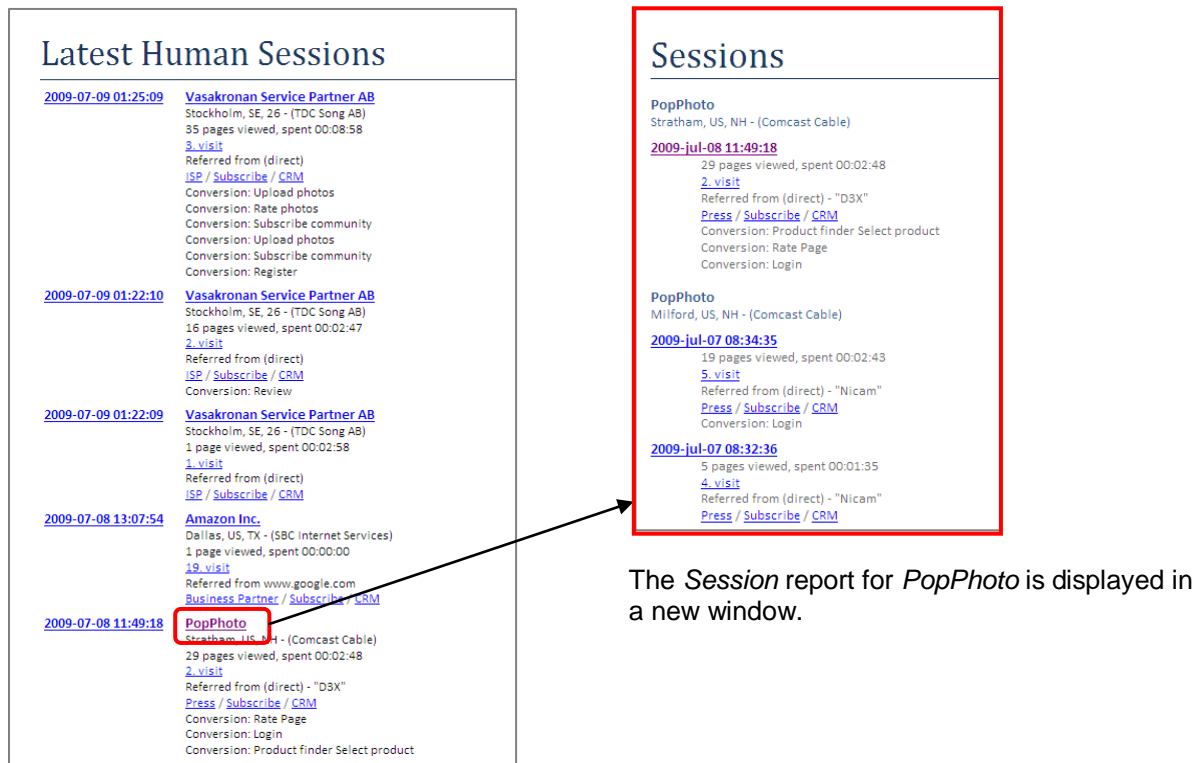
Note
The **Image** dialog box in the Windows reports designer has more options than the Web reports designer. For example, you can also use the **File** tab to link to a locally stored image. You can specify a variable name for your image in the **Image Data** field. This variable or expression can then be used by developers working with images.

3.5 Adding Links to Reports

In Sitecore OMS, there are three types of hyperlinks you can use:

- Link to another page on the same Web site

- Link to an external Web site

- Link to another report in Sitecore OMS

For example, the *Latest Human Sessions* report contains a link to the *Session* report from the visitor organization name.

The *Session* report for *PopPhoto* is displayed in a new window.

The *Latest Human Sessions* report.

3.5.1 How to Link to a Web site

To demonstrate how to create a hyperlink, I have created a simple report based on the *IPOwner* table. This report lists organizations that have visited the Nicam Web site, with visitor information,

such as country, visitors, the number of visits, and the number of pages visited.

Note

You do not need to create this report if you already have a suitable report to use. You can select any report in Sitecore Analytics and follow the same steps, but if you do this, remember to duplicate the report first.

To create an internal link from the Nicam logo to the home page of the Nicam Web site:

1. Open your chosen report in the report designer.

2. Click on the text or image you want to make into a hyperlink. In this example, click on the Nicam logo.
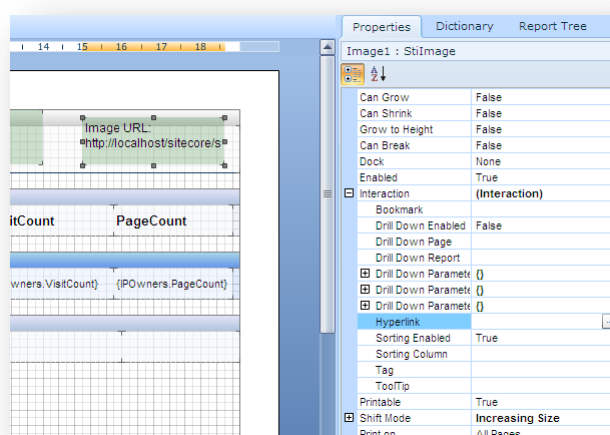
3. In the **Properties** panel, expand the following properties:

   o Behavior

   o Interaction

   o Hyperlink

4. Click the box next to the **Hyperlink** field and open the **Text Editor**.
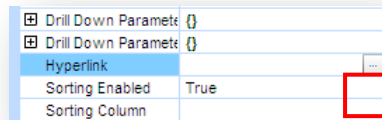
5. In the **Text Editor** window, enter a URL string that links to the Nicam home page.
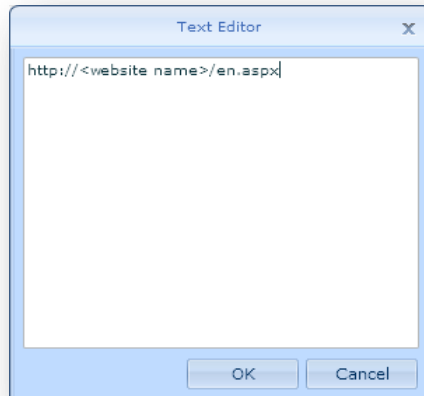

```
http://<website name>/en.aspx
```

6. Click **OK** to close the **Text Editor** window.
7. Save you report and in **Sitecore Analytics** preview your report and check your link.

How to Link to an External Web site

Use the same report and create a link to the www.sitecore.net Web site from the Sitecore logo.

Follow the same steps and enter the full address of the Web site that you want to link to. For example, enter the address of www.sitecore.net in the **Text Editor** window:

```
http://www.sitecore.net/
```

3.5.2 How to Create a Link to another Report

Creating a link to another report in Sitecore OMS requires a little more thought. Again, use the same report. This time create a link from the organization name that appears in the body of the report to the *Session* report.

Linking to the session report is useful because it gives you more information about the activities of each visitor, such as pages viewed, conversions, and profile scores.

To make a link to the *Session* report:

1. Open the Visitor Information report or another report of your choice.

2. In the report designer, select the text box that contains the organization name.
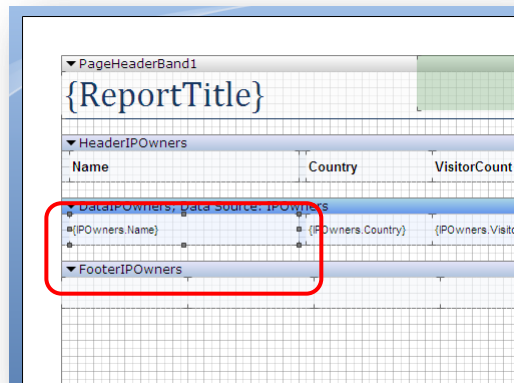
3. In the **Properties** panel, select the **Hyperlink** property and open the **Text Editor**.

4. In the **Text Editor** window, enter the following URL string to create a link to the *Session* report:

```
/sitecore/shell/applications/analytics/default.aspx?r=B33272F6-E2C1-43DF-802E-
969A955BC6D6&p=1&ipowner={IpOwners.IPOwnerIdObject}
```

The format of this URL string is quite different from a normal hyperlink. The following table explains each part of the URL.

String	Description
/sitecore/shell/applications/analytics/default.aspx?	Sitecore page that executes the report runner
r=B33272F6-E2C1-43DF-802E-969A955BC6D6	*Session* report Item ID
&p=1&ipowner	Opens a pop-up window
={IpOwners.IPOwnerIdObject}	Links the visitor session to the organization name displayed in the report.

`IpOwners.IPOwnerIdObject` refers to the data sources configured in your report. This part of the URL can be edited depending on what data sources you have configured.

The item ID will be different depending on what report you want to create a link to.

5. Click **OK** to close the **Text Editor** window.

6. Save your report and in **Sitecore Analytics** run your report and test your link.

Visitor Information

Name	Country	VisitorCount	VisitCount	PageCount
My Local Machine	US	10	61	177
Vasakronan Service Partner AB	SE	2	8	146
Amazon Inc.	US	1	20	141
PopPhoto	US	3	8	134
Dell UK	GB	6	10	119
Nicam Corporation	DK	4	6	99
Adorama	US	5	12	90
Canon NL	NL	1	3	72
Google	VN	39	39	39
Celcom Internet Service Provider	MY	1	1	29
Nikon AU	AU	2	2	26
Ebay	US	4	6	25
TeleMost Ltd	UA	2	7	25
iiNet Limited	AU	1	1	24

Chapter 4

The Analytics Database

This chapter introduces the OMS Analytics database. All OMS reports are based on tables from this database.

- Understanding the Analytics Database

- Useful Analytics Table Groupings

4.1   Understanding the Analytics Database

When you create a new report consider the following:
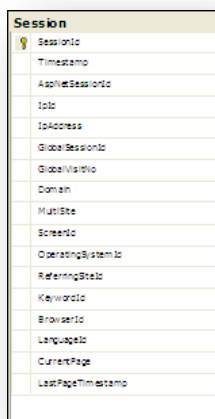
- What information do I want to display in the report?

- What tables do I need to use?

- Is there a report that I can adapt that already uses these tables?

- How do I want to display this information my report layout?

4.1.1   Main Table Groupings

To understand the Analytics database, start by getting an overview of the important table groupings. The following pages in this chapter provide diagrams of each table grouping with a brief explanation of the purpose of each table.

4.1.2   The Session Table

The Session table is the most important table in the Analytics database. It is related to all other tables (apart from Status) and can be thought of as your point of access to all the information you want to retrieve from the database.

All Web site visitors create a new session every time they visit the Web site and this is recorded in the Session table. Most reports need to access this table first before accessing information from the other tables.

If you study any of the existing reports in Sitecore Analytics, you can see that the Session table appears in the majority of SQL queries. The following SQL query is taken from the *Campaign Summary* report, *Profiles* data source.
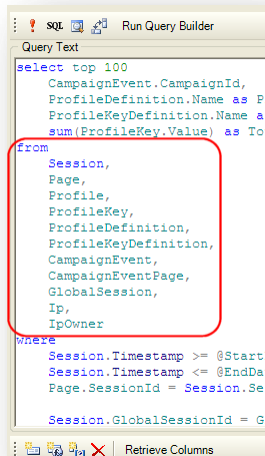
This query starts by accessing the Session table and then uses a number of other tables to retrieve the data it needs.

4.1.3   SQL Query Language

The data displayed in a report is retrieved from a database using the SQL query language. When you create a report, you also need to create or adapt a SQL query.

You can either write SQL queries by hand or use a tool to help you.

In the Windows reports designer, the Query Builder makes it easier to write SQL queries and other tools such as SQL Server Management Studio can help to automate this process.

As a starting point, we would recommend that you study the standard reports in Sitecore OMS to help you to understand the structure of each query. Each report consists of one or more data sources and each data source contains a SQL query. Examine the query first to understand which tables are being used and which columns are being displayed.

Although you can create reports with a limited knowledge of the SQL query language, particularly if you use a tool like the Query Builder, an understanding of SQL and the table structure of the Analytics database will make your task quicker and easier.

Note
If you intend to upgrade from a SQL Server database to an Oracle database, be aware that the SQL query syntax used in Oracle is slightly different from SQL Server. Also, Transact SQL (TSQL) is not compatible with an Oracle database.

The following section outlines some useful table groupings that may help you to find the tables and columns that you need to include in your report.

4.2   Useful Analytics Table Groupings

1 = Visitor Information

2 = Profile Information

3 = MV Testing

4 = Campaigns

5 = Page Events

6 = System Information
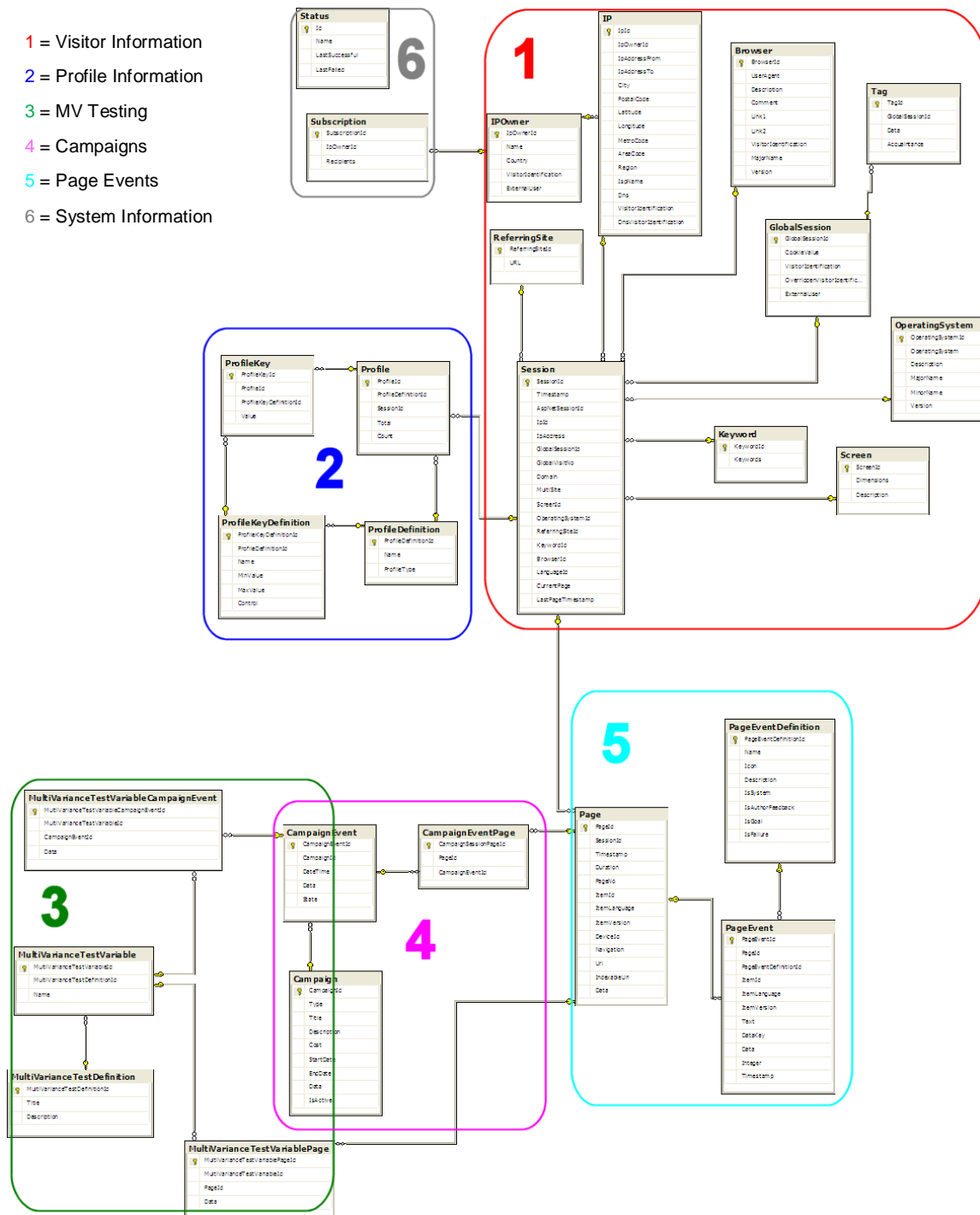
4.2.1   Visitor Information

This group of tables captures information about your site visitors.
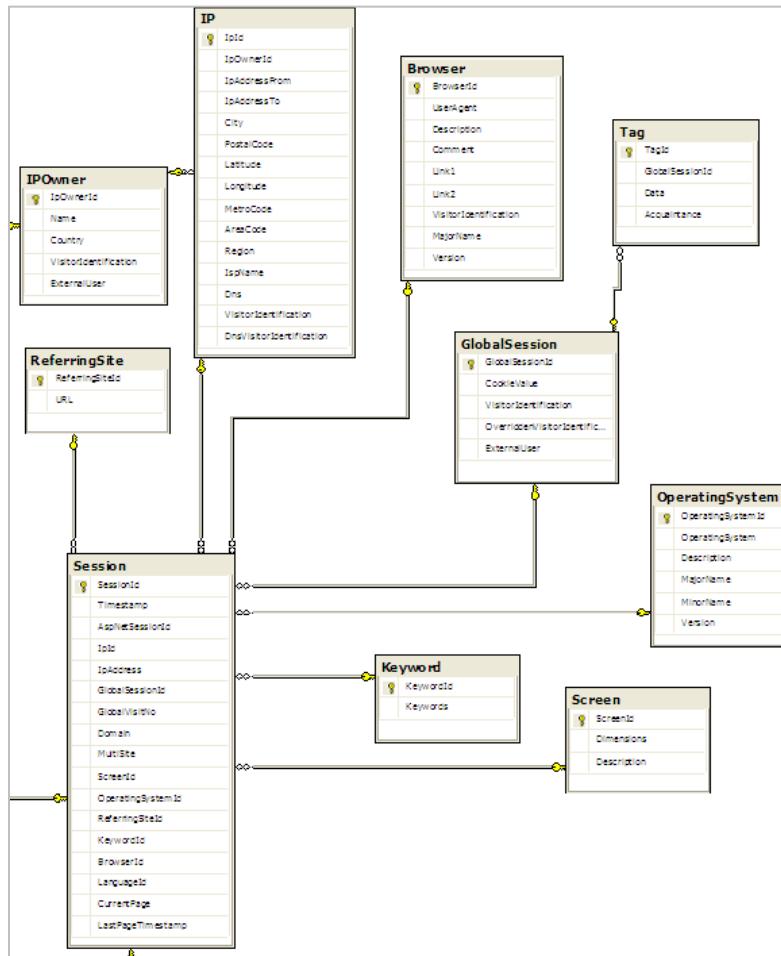
Table	Description
Session	Primary Analytics table. All visitors and site visits are recorded in this table.
IP	Visitor IP information obtained from IP lookup service. For example, IPAddress, city, post code, latitude or longitude.
IPOwner	Name of the company or organization associated with IP address
Browser	Type of browser used
Referring Site	Web site where the visitor came from
Global Session	One global session is recorded for each site visitor. Global Session binds each session to a global session to identify returning visitors. Detected by using a cookie.
Operating System	Type of operating system used
Keyword	Search engine search terms used
Tag	Additional information – This field is often used in Web forms
Screen	Screen size

4.2.2 Profile Information

This group of tables contains definitions and settings for profiles and profile keys. They also record the scores achieved by your Web site visitors.

Table	Description
Profile	Records the sum total of all profile keys for a single profile group during one visitor session. For example, for the *Skill Level* profile group.
ProfileDefinition	Where all the configuration data for your profiles is stored. In the Content Editor, when you add a new profile to the Marketing Center, the name of the profile and the method of calculation are stored in this table. For example, *Name = Skill Level* and *Profile Type = Sum*.
ProfileKey	Records the score for individual profile keys for a single visitor session. For example, the score for the *Semi-pro* profile key.
ProfileKeyDefinition	Where all the configuration data for your profile keys is stored. In the Content Editor, when you add a new profile key to the Marketing Center, the name of the profile key and the range of values allowed are stored in this table. For example, *Name = Semi-pro, MinValue = 0 and MaxValue = 10*

4.2.3 MV Testing

This group of tables stores all data related to multivariate tests. There are two ways of triggering multivariate tests; from pages or campaigns.
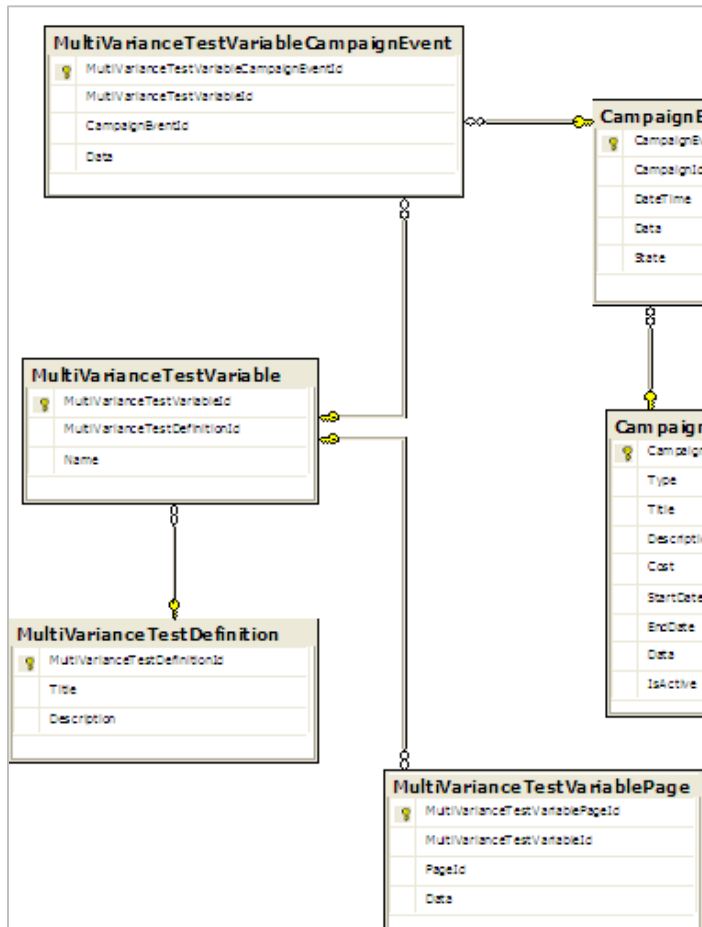
Table	Description
MultiVarianceTestDefinition	This table contains the definition item for your MV test. Every time you create a new MV test in the Marketing Center, a new entry is added to this table.
MultiVarianceTestVariable	This table contains the individual multivariate test variables for your MV test.
MultiVarianceTestVariablePage	For example, a new button on a page. When the test is triggered the MV test variables are bound to the page on which the test was executed (where the button is displayed).
MultiVarianceTestVariableCampaignEvent	When a MV test is triggered from a campaign. For example, a newsletter campaign could contain an MV test. So when the test is triggered, the MV test variables are bound to the campaign event.

4.2.4 Campaigns

When you create a campaign all your campaign and sub campaign configuration data is stored in these three tables. Campaign events also link to the Page and MV Testing group of tables.
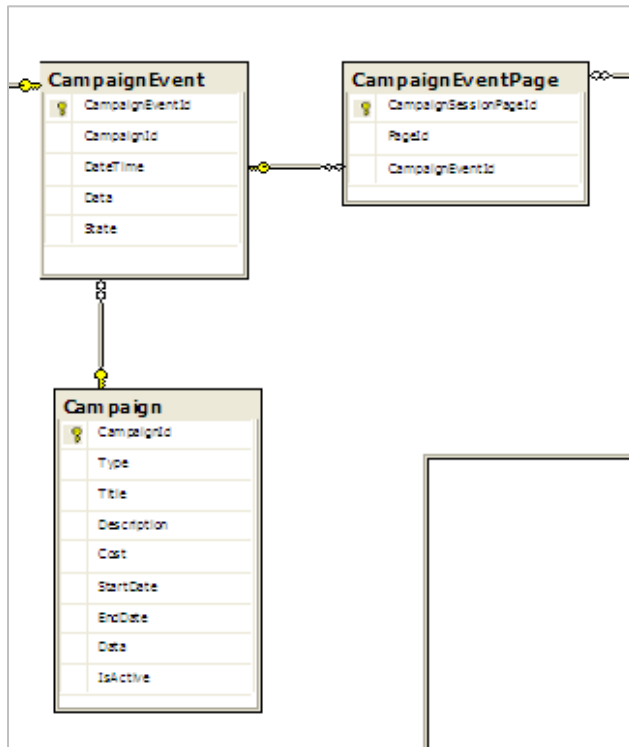
Table	Description
Campaign	Defines the master campaign, for example *Google AddWords*.
CampaignEvent	Defines sub campaigns, for example *.NET CMS*
CampaignEventPage	Links to the Page table. When a campaign is triggered, you can see which page the visitor was viewing when the campaign was triggered.

4.2.5   Page Events

Every visitor session is linked to the Page table. For example, fields from the Page table are used in creating a session trail for each site visitor.
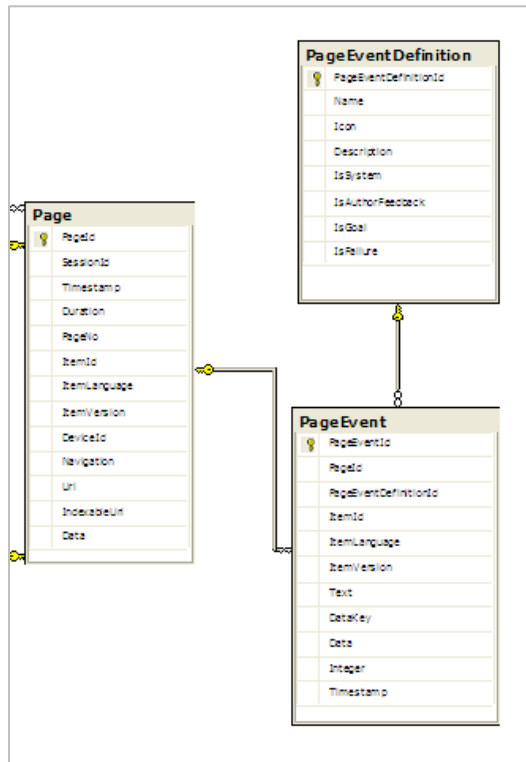
Table	Description
Page	Records information on pages viewed by each site visitor and links to the Session table.
PageEvent	For example, *Page Visited*, *Page Not Found, Campaign*
PageEventDefinition	Contains all the configuration information for page events that you define in the Content Editor.

4.2.6 System Information

System tables are for internal Sitecore use only. You won't need to include any of these tables in your reports.

Status
- 🔑 Id
- Name
- LastSuccessful
- LastFailed

Subscription
- 🔑 SubscriptionId
- IpOwnerId
- Recipients
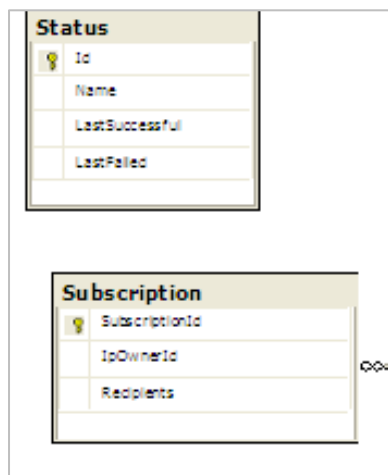
Table	Description
Status	Used to log errors
Subscription	Used when visitors subscribe to a report

Chapter 5

Creating Reports

This chapter introduces the component parts of a report and explains how to create a report without overwriting any standard Sitecore OMS reports. It also provides tips on creating SQL queries and step by step instructions on how to design the layout of your report.

This chapter contains the following sections:

- Creating and Saving a Report

- Connecting to the Analytics Database

- Creating Data Sources

- Configuring Data Sources

- Designing a Report Layout

5.1 Creating and Saving a Report

In Sitecore OMS, a report consists of two components:

- Sitecore report definition item

- Stimulsoft .mrt report file

You can only create reports from scratch using the Windows reports designer.

If you want to create or edit a report to use with the Web reports designer, you must first duplicate an existing report.

We do not recommend that you open and edit reports in Sitecore Analytics without making a duplicate first. Some report files, such as Latest Sessions.mrt are used by multiple reports, so changes to a single report file can affect more than one report.

A benefit of using this approach is that you can re-use the existing report functionality and strip out the functionality you do not need.

To illustrate this, we will use the *Leads - Executive - Humans* report as the basis for a report that will only display information about visitor profile keys. We will edit this report so that it only displays information about profiles and remove all content related to goals.

In this section, you will learn how to create and save a report called *Profile Key Report* to use with the Web reports designer. To do this, you must complete the following steps:

- Duplicate a report definition item

- Duplicate a report .mrt file:

- Link your report definition item to your new report

- Open your duplicate report in the Web reports designer

To duplicate a report definition item:

1. Open the Content Editor, content tree and navigate to the following location:

   /sitecore/system/Settings/Analytics/Reports/Reports/Leads - Executive - Humans
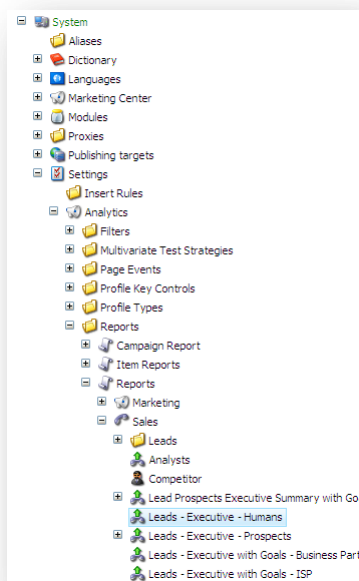
2. Right click the report *Leads - Executive - Humans.* Click **Duplicate**.

3. Accept the default name for the duplicate report and click **OK**.

4. Create a new folder for your report definition item. Select the **Reports** node and in the ribbon, **Developer** tab click **Folder**. Enter the name *MyReports* for the new folder.
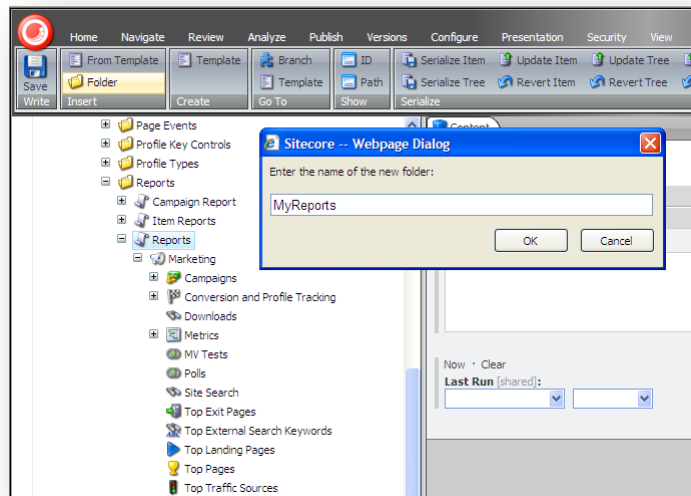
5. Drag and drop your duplicate report into the *MyReports* folder.

6. Re-name your report definition item *Profile Key Report*

7. In the ribbon, click **Save**.

To duplicate a report `.mrt` file:

1. Navigate to the folder in your Web site that contains all your report `.mrt` files:

   wwwroot\<site name>\WebSite\sitecore\shell\Applications\Analytics\Reports\

2. Copy `Leads Executive.mrt`.
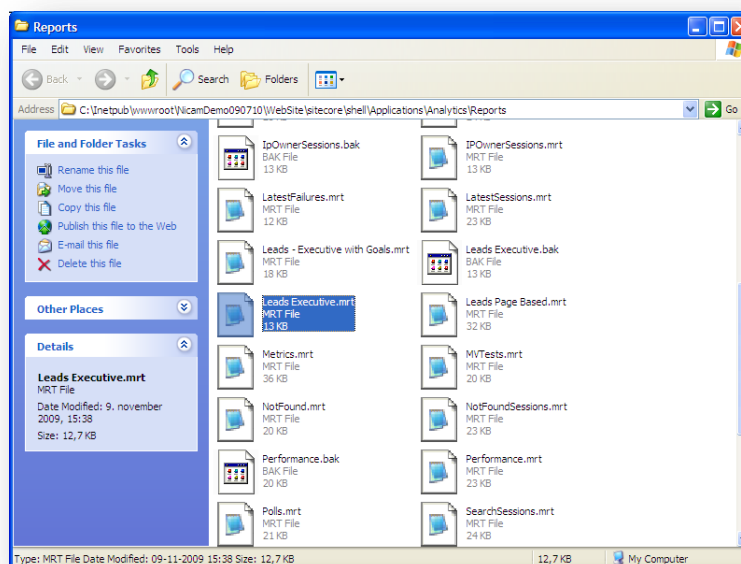
3. Create a new folder called *MyReports* to store all your own custom reports.

4. Paste `Leads Executive.mrt` into the *MyReports* folder.

   By creating a new folder, you ensure that all your new custom reports are not confused with the standard OMS reports.

5. Re-name the `Leads Executive.mrt` file `ProfileKeyReport.mrt`.

Link your report definition item to your new report:

1. In the **Content Editor** content tree, select *Profile Key Report*.

2. In the **File Name** field, edit the path to your report `.mrt` file. Enter the following path:

   `/sitecore/shell/Applications/Analytics/Reports/MyReports/ProfileKeyReport.mrt`
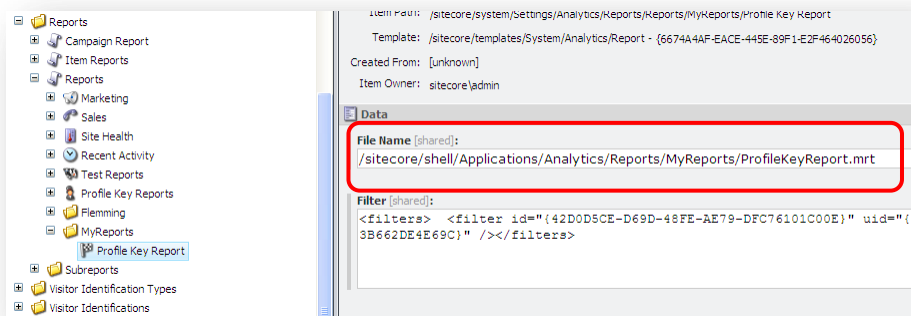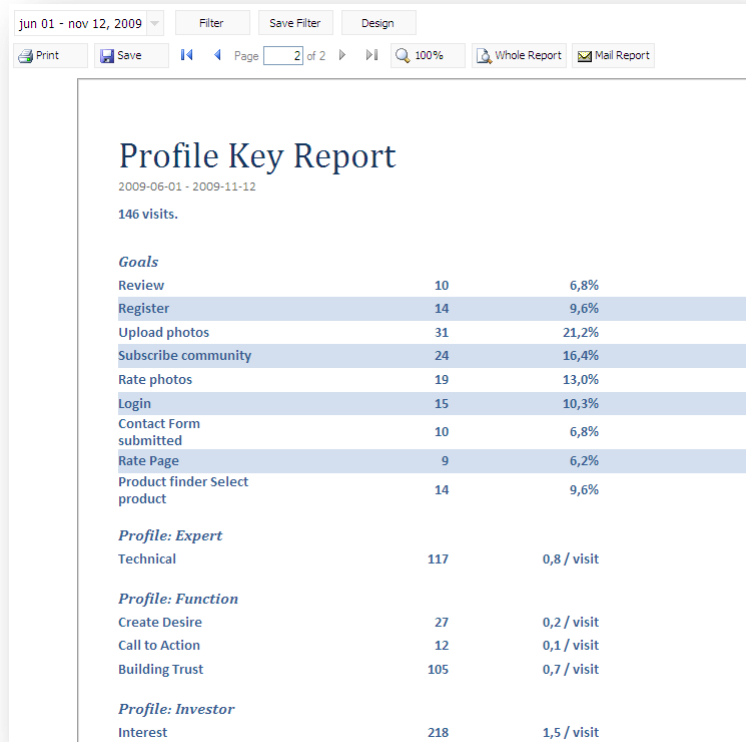
This is the link between the Sitecore report definition item and the Stimulsoft `.mrt` report file.

3. In the Sitecore ribbon, click **Save**.

To open your new report in the Web reports designer:

1. Open **Sitecore Analytics**. If you do not see your report, open and close the **Reports** root node to refresh the tree.

2. Select the *Profile Key Report*.

3. Your duplicate report should look like this.

If you see an error message check that:

- o The path in the report definition item **File Name** field is correct.

- o The `.mrt` file name in this path matches exactly the `.mrt` file name in your *MyReports* folder.

4. Click **Design** to open the report in the Web reports designer.
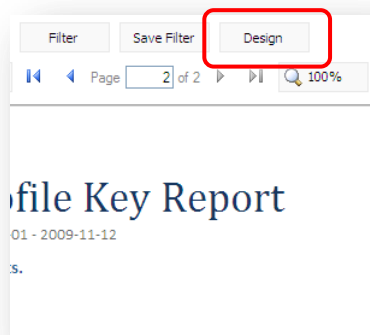
5.2   Connecting to the Analytics Database

If you are using the Web reports designer, you are automatically connected to your Analytics database. In the Windows reports designer, you must add the correct connection string.

To view a connection string in the Web reports designer:

1. In **Sitecore Analytics**, open a report.

2. In the **Dictionary** panel, click **Data Connection**.

3. Click **Edit** to view the connection string.
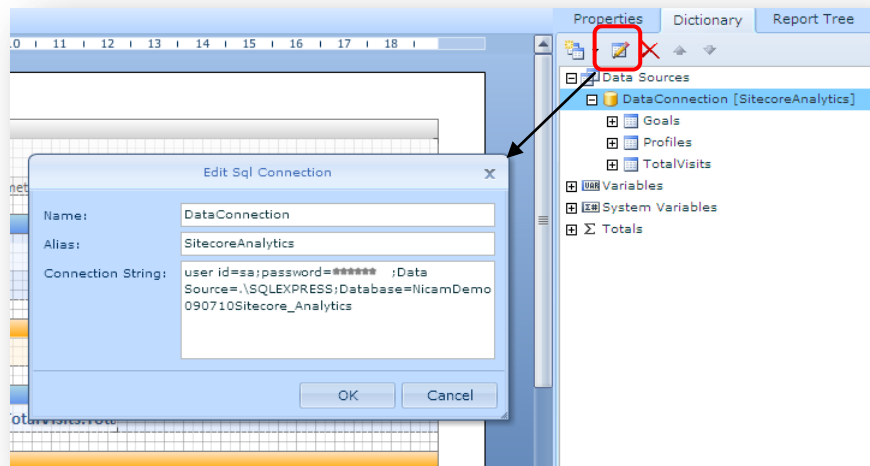
To add a connection string in the Windows reports designer:

1. Open the Windows reports designer (from the Start menu on your computer).

2. In the **Designer**, in the **Dictionary** panel, click **New Item**, **New Connection**.

3. Select **Sql Connection** as the type of connection.

4. Click **OK**.

   In the **New Sql Connection** dialog box, you can enter a connection string that includes your SQL database name, username, and password. If you don't know this information, look at your existing Sitecore connection string or open SQL Server Management Studio to find out.

   This is the format you need for the connection string:

```
user id=<username>;password=<password>
;Data Source=<name of datasource>
;Database=<name of database>
```

   Example:

```
user id=sa;password=******
;Data Source=.\SQLEXPRESS
;Database=Nicam_Sitecore_Analytics
```

5. Click **Test** to test the connection.
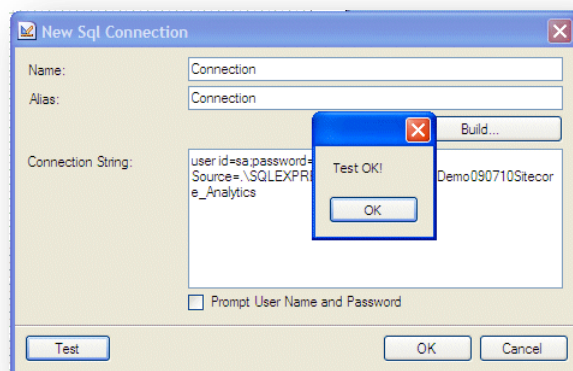
In the next section you learn how to create a data source.

5.3 Creating Data Sources

The next step after you have created a report and connected to the analytics database, is to configure a data source.

Data sources use SQL queries to retrieve columns from the Analytics database and then display them in the Dictionary window. You can then decide how your want to display this data in your report layout.

You can create your own names for data sources and can also use aliases in your SQL query to make the names of columns more meaningful.

If you examine any of the standard reports in Sitecore OMS, you can see that each report contains one or more data sources. These are visible in the report designer, Dictionary panel. The following example is a report related to campaigns and conversion tracking
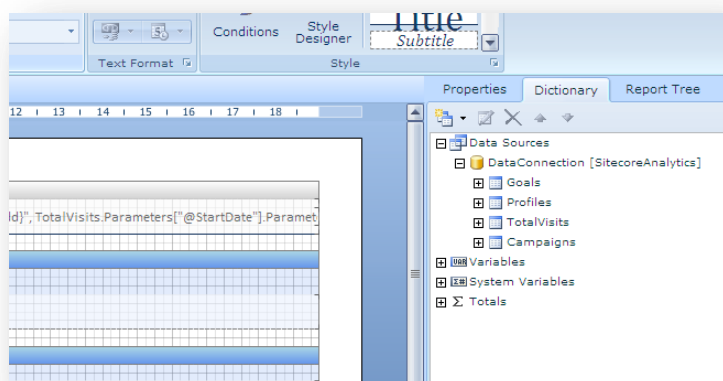
In this section, you configure a new Data Source called *IpOwners* to use in the *Profile Key Report* you created earlier.

In the Web reports designer, *IpOwners* is already visible but if you are using the Windows reports designer you need to follow these steps:

1. In the report designer, **Dictionary** panel, click **Data Connection** and click **New Data Source**.

2. Select **DataConnection [SQL Connection]** and click **OK**.

3. In the **Select Data** window, click **Query New** to open the **New Data Source** window.

4. Enter a *Name* and an *Alias* for the Data Source (leave the field *Name in Source* unchanged). Give your Data Source the name *IpOwners*.

5. Click **OK** and the new Data Source is added to the Data Dictionary below your Data Connection.
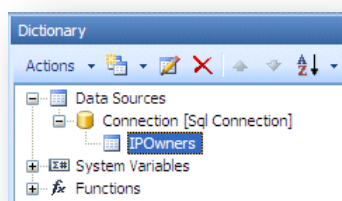
In the next section, you will learn how to create a SQL query to configure your new data source.

5.4   Configuring Data Sources

All reports contain data sources that use SQL queries to extract data from the Analytics database.

To view a SQL query in the Web reports designer:

1. In the **Dictionary** panel, expand the Data Connection node and select a data source, for example, *IpOwners*.

2. Click **Edit**.
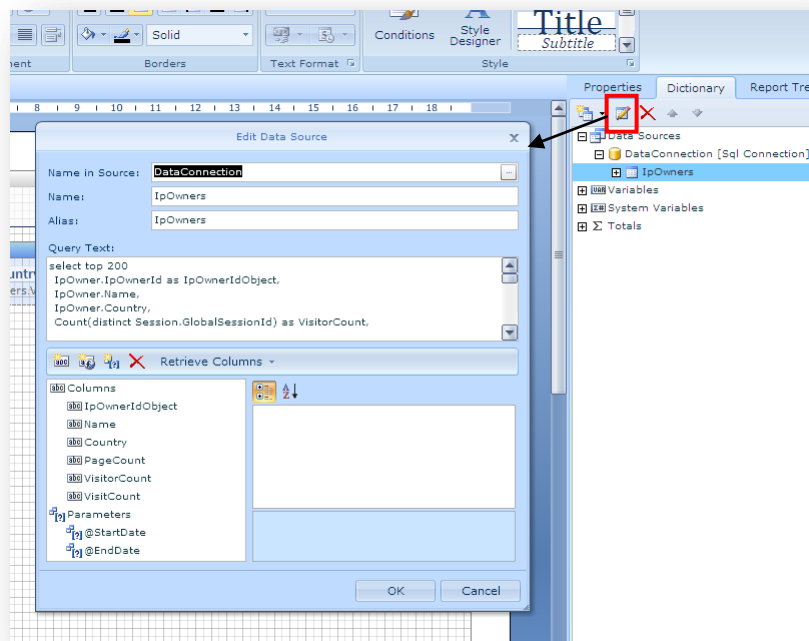
In the **Edit Data Source** window you can see the SQL query.

Note

In the Web reports designer, it is possible to edit the text of your SQL query but SQL editing functionality is limited. We recommend you construct your SQL queries in another tool, then copy and paste your query into the **Edit Data Source** window.

To view a SQL query in the Windows reports designer:

1. Follow the same steps but right click *IpOwners* and then click **Edit**.
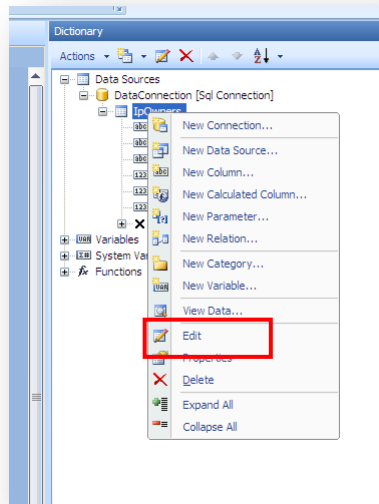
2. In the **Edit Data Source** window, you can view and edit your SQL query.
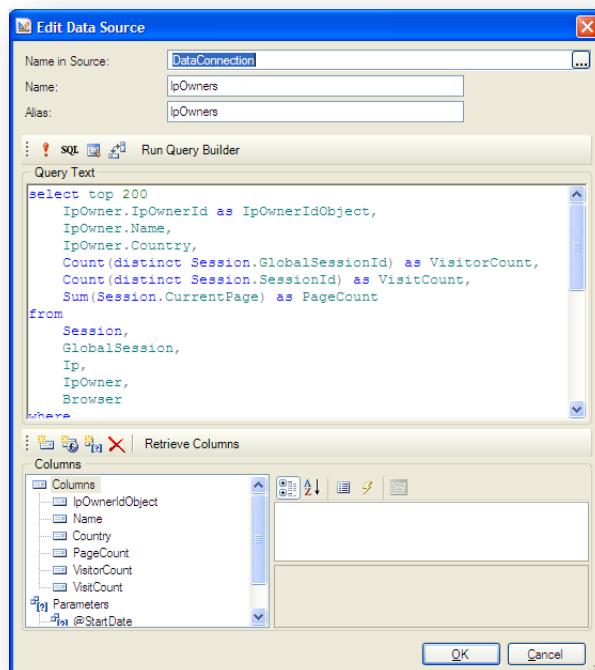
3. Click **Run Query Builder** to edit your query.
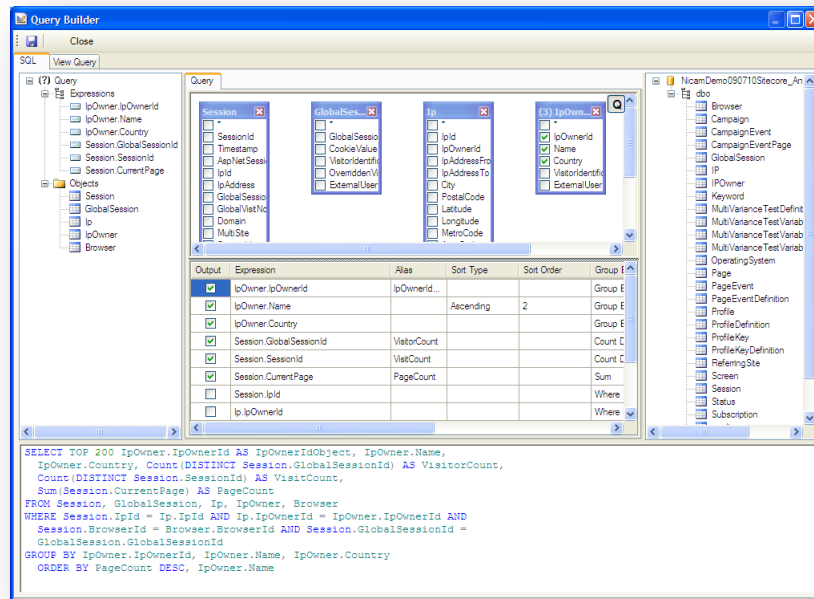
In the Query Builder you can edit your query or construct new queries from scratch. This is a powerful tool for building SQL statements.

In the right hand panel you can see all the tables in the Analytics database. If you double click on any table, it is added to your query. The SQL code is automatically added to the query.

If you only have access to the Web reports designer, you can use another tool to construct your queries. For example, SQL Server Management Studio has the same functionality.

5.5 Creating a SQL Query

To retrieve the correct data from the Analytics database you must create a SQL query in the *IpOwners* data source.

The query must retrieve profile key information for the Semi-pro profile key and fulfill the following criteria:

- List companies/organizations and their profile key scores

- Display the highest score at the top of the report.

- For each organization, the report must display the number of visitors and the number of visits.

- Show the name of the selected profile key.

You can copy and paste the finished SQL query included here into your *IpOwners* data source or follow the step by step instructions on how to create this query included in the Appendix section.

The finished SQL query:

```
SELECT TOP 20 IpOwner.IPOwnerId AS IPOwnerIdObject, IpOwner.Name, IpOwner.Country,
  Count(DISTINCT Session.GlobalSessionId) AS VisitorCount,
  Count(DISTINCT Session.SessionId) AS VisitCount,
  Sum(DISTINCT ProfileKey.Value) AS Score, ProfileKeyDefinition.Name
FROM Session INNER JOIN
  Profile ON Session.SessionId = Profile.SessionId INNER JOIN
  GlobalSession ON GlobalSession.GlobalSessionId = Session.GlobalSessionId
  INNER JOIN
  Ip ON Ip.IpId = Session.IpId INNER JOIN
  IpOwner ON IpOwner.IpOwnerId = Ip.IpOwnerId INNER JOIN
  Browser ON Browser.BrowserId = Session.BrowserId INNER JOIN
  ProfileKey ON ProfileKey.ProfileId = Profile.ProfileId INNER JOIN
  ProfileKeyDefinition ON ProfileKeyDefinition.ProfileKeyDefinitionId =
  ProfileKey.ProfileKeyDefinitionId

WHERE ProfileKeyDefinition.Name = 'Semi-pro'

GROUP BY IpOwner.IPOwnerId, IpOwner.Name, IpOwner.Country,
  ProfileKeyDefinition.Name

 ORDER BY Score DESC, IpOwner.Name
```

To add this query to the IpOwners data source:

1. In the Web reports designer, copy and paste the query into the **Query Text** panel of the **Edit Data Source** window.

2. Click **Retrieve Columns**.

3. Click **OK** to save your changes and close the window.

You should now see the columns you retrieved listed after the *IpOwners* data source.
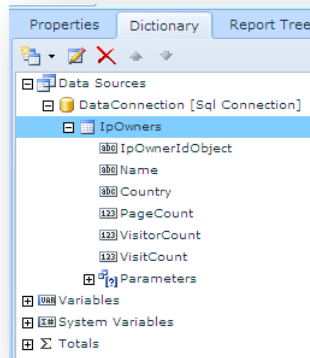
5.6 Designing a Report Layout

When you have configured a data source, you can use the Web reports designer to create a layout for your *Profile Key Report*.

To create a report layout complete the following steps:

- Create an initial report layout

- Format column headings

- Add a link to the Session trail from each organization name

- Group organization and country together

- Group number of sessions and number of visitors below organization name

- Add a report header and format text

5.6.1 Create an Initial Report Layout

If you have re-used the *Leads - Executive – Humans* report, first remove any text boxes that appear on the existing data band. To do this, select a text box and in the ribbon and click Delete. You can keep the report title header and add new columns to the data band.

1. To add content to your report, drag and drop the columns you need from the Dictionary panel to the data band.

    Add the following columns:

    o Score

    o Name

    o Country

    o Visitor Count

    o Visit Count

2. First select **Score**, then drag and drop this column onto the report data band.
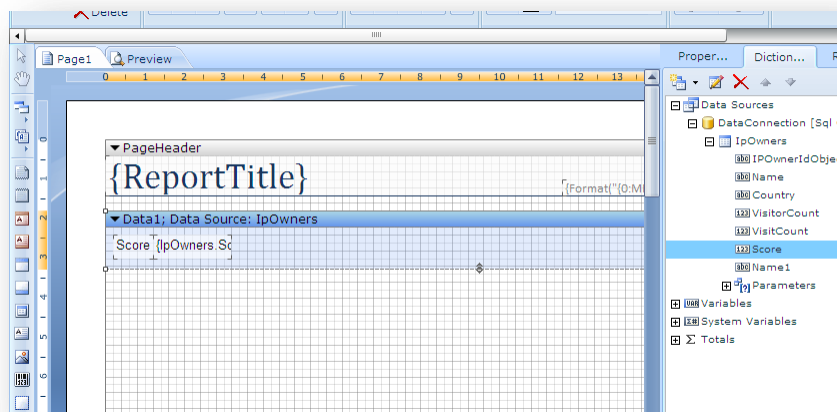
3. Notice that you have added two text boxes. **Score** and **IpOwners**. Re-size using the arrows that appear when you move the mouse over the text boxes.

4. In the report designer toolbox, click **Header** and drag a new header onto the report. Insert it between the Page Header and the Data band.
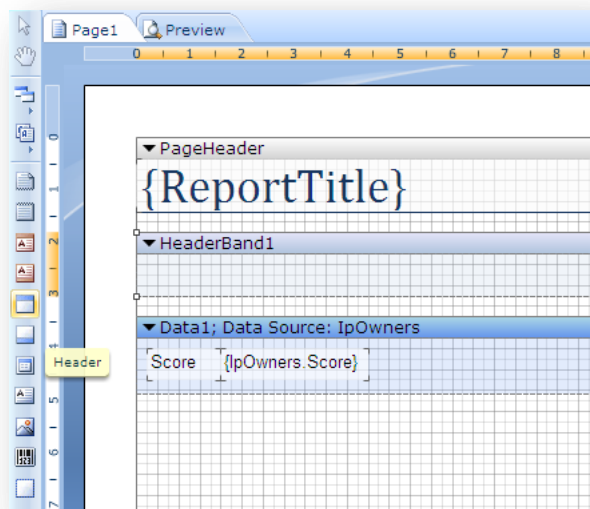
5. Move the **Score** text box into the header band, this is the column heading. **IpOwners** stays in the Data band and outputs the data for this column.

6. Add **Name**, **Country**, **Visitor Count** and **Visit Count** in the same way until your layout looks something like this:
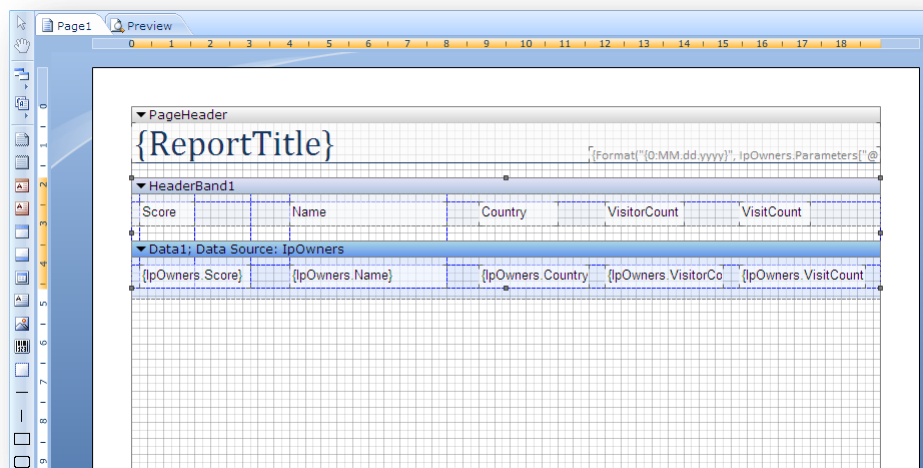
7. In the Stimulsoft **File** menu, Click **Save Report**.

8. Preview your report in Sitecore Analytics to check that it displays the information that you need.

You can see all the top scoring organizations listed for the Nicam site profile key *Semi-pro*. A top clause in the SQL query limits the results to the top 20.

5.6.2   Formatting Column Headings

To format all your column headings as bold:

1. Press the SHIFT key and then click each of the header text boxes.
2. When you have selected all column headings, in the report designer ribbon click **Bold**.
3. Click **Save Report**.

5.6.3   Add a Link to the Session Report

Linking to the session report gives you more information about the activities of each visitor listed in your report, such as pages viewed (session trail), conversions, and profile scores.

1. In the Data band select the *{IpOwners.Name}* text box and click the **Properties** panel.

---

2. In the **Properties** panel, expand the **5. Behavior** node, click **Interaction** and then click **Hyperlink**.
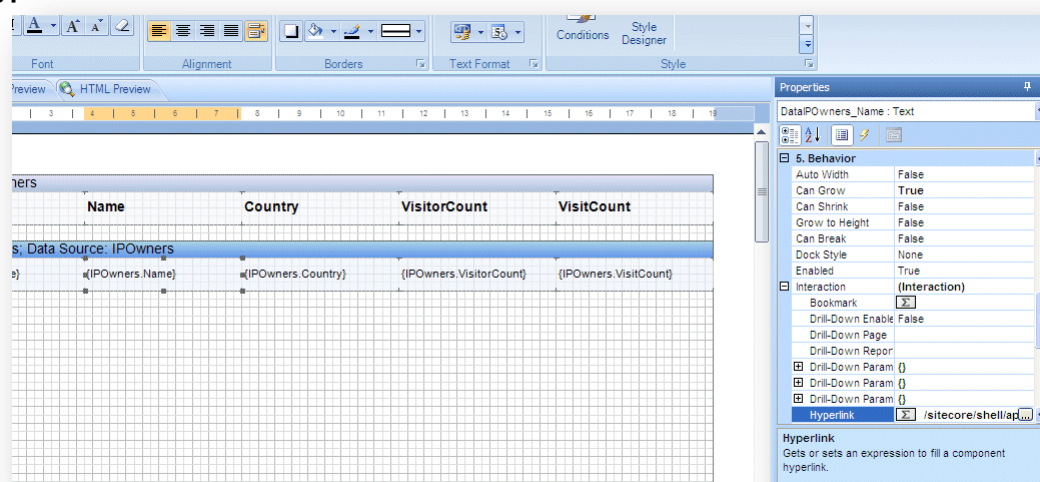
3. Click the box next to the **Hyperlink** field to open the Text Editor.
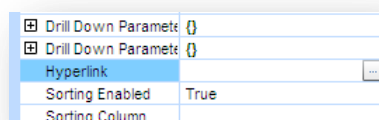
4. In the **Text Editor** window, enter the following string to create a link to the Session report.

    ```
    /sitecore/shell/applications/analytics/default.aspx?r=B33272F6-E2C1-43DF-802E-
    969A955BC6D6&p=1&ipowner={IPOwners.IPOwnerIdObject}
    ```

    The link is taken from the *LatestSessions.mrt* report. The only text that has been changed is highlighted. This text refers to the alias used in my Data Source.

    IpOwner.IPOwnerId AS IPOwnerIdObject

    This means that when you click on an organization name in the report you link to the session report for that specific organization.

5. Save your changes and preview the *Profile Key Report* again.

    Now for each organization listed, you can view the session trail.

5.6.4   Group Organization and Country

To make visitor organization and country appear together as a hyperlink:

1. Double click the *{IpOwners.Name}* text box and click the **Expression** tab in the Text Editor.

2. Enter the following expression:

    ```
    {IPOwners.Name}, {IPOwners.Country}
    ```

This expression will display the organization name and country next to each other as a hyperlink.

PopPhoto, US
3 sessions, 2 visitors

You can now remove the *Country* text boxes from the Header and the Data band.

Group Sessions and Visitors

To group the number of sessions and visitors below each organization name:

1. Select the data band, and then in the toolbox select text box and then drag and drop the text box from the toolbox.
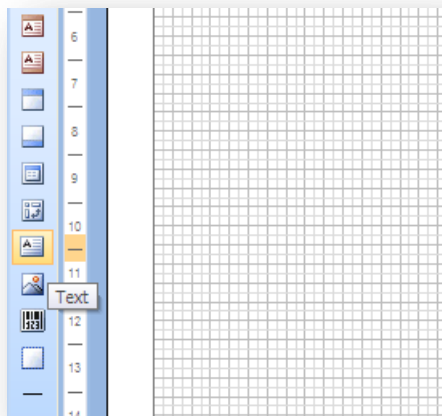
Place the text box directly below the *{IPOwners.Name}* text box.
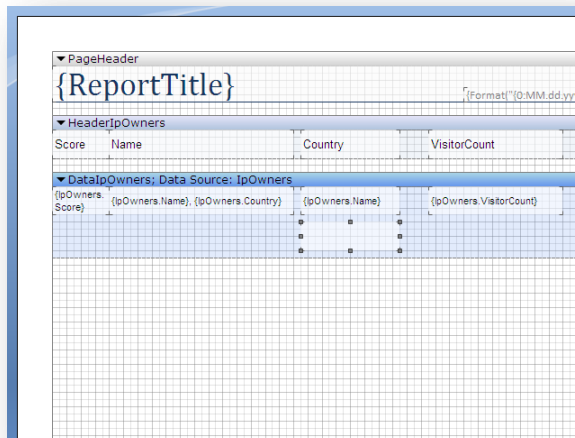
2. Double click the new text box. Copy and paste the following expression into the **Expression Editor** window.

```
{IPOwners.VisitCount} session{IPOwners.VisitCount == 1 ? "" : "s"},
{IPOwners.VisitorCount} visitor{IPOwners.VisitorCount == 1 ? "" :
"s"}
```

This expression displays **Visitor Count** and **Visit Count** together below the organization name. It also makes the word *session* and the word *visitor* either singular or plural depending on the amount of sessions or visitors. This expression is constructed using syntax very similar to C#.

PopPhoto, US
3 sessions, 2 visitors

3. You can now remove all the *Visit Count and Visitor Count* text boxes from the Header and the Data band, as they are no longer needed.

Add a Report Header

Your report already has a report header, follow these steps if you need to create a new header:

1. In the **Data Dictionary**, click **New Item**, and then click **New Variable**.
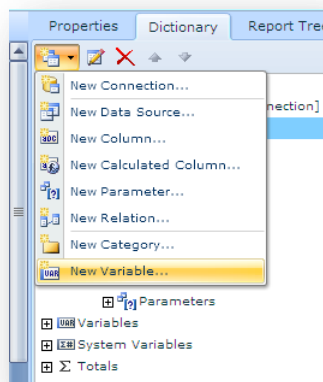
2. In the **New Variable** window, in the **Name** and **Alias** fields enter **ReportTitle**.

3. In the **Type** field, select *String* and click **OK**.

4. Add a text box to the report header band.

5. Double click the text box to open the Expression Editor.

6. In the **Expression Editor** window, drag and drop the *ReportTitle* variable you created into the **Expression** tab.

7. Click **OK**.
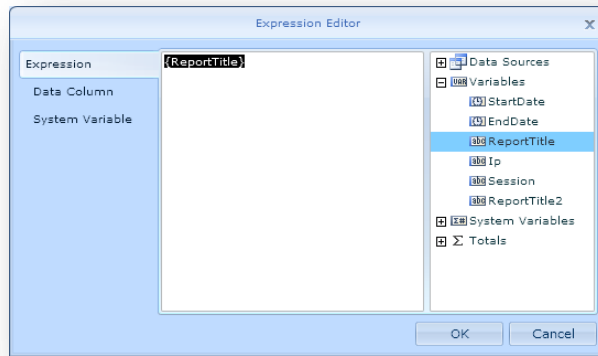
Left align the text box and your report title should look something like this.
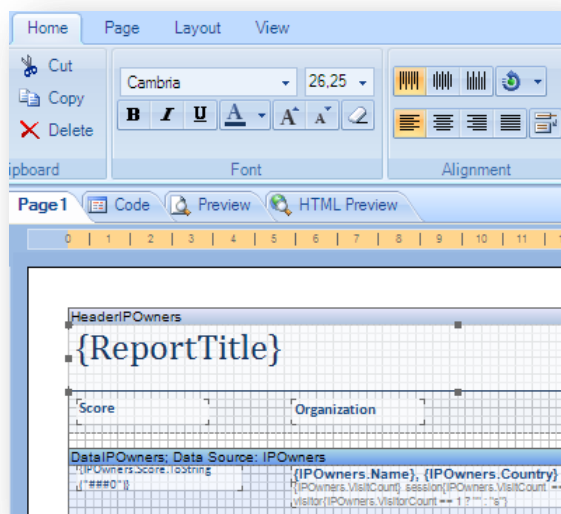
Note

To format text in the same way as other Sitecore reports, you must also import the same Styles from another Sitecore report.

If you are using the Windows reports designer, you can use the style designer to save styles locally or use it to import other saved styles.
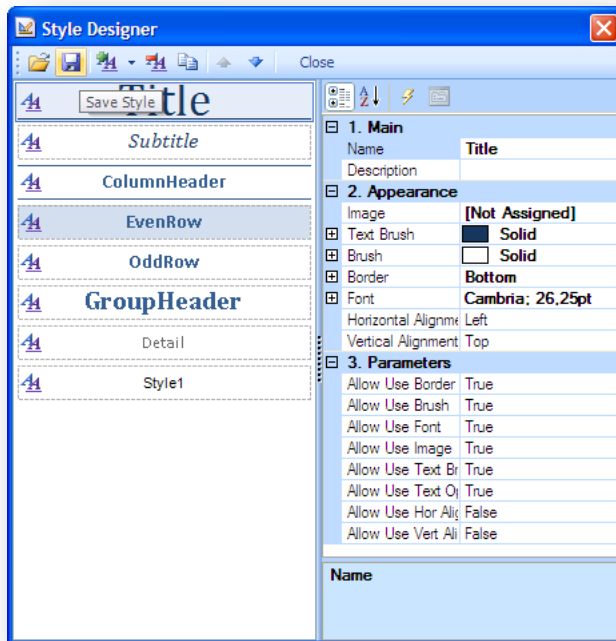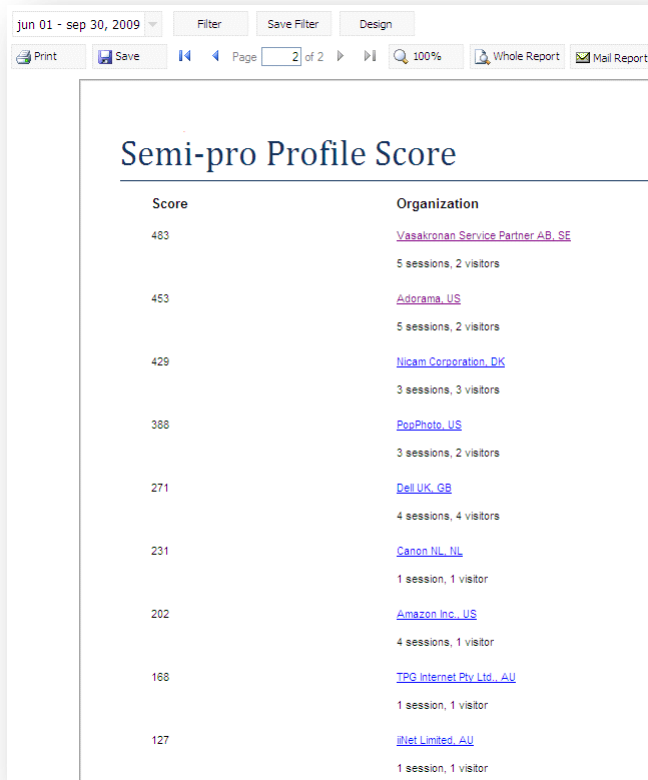
The completed report layout should look something like this.
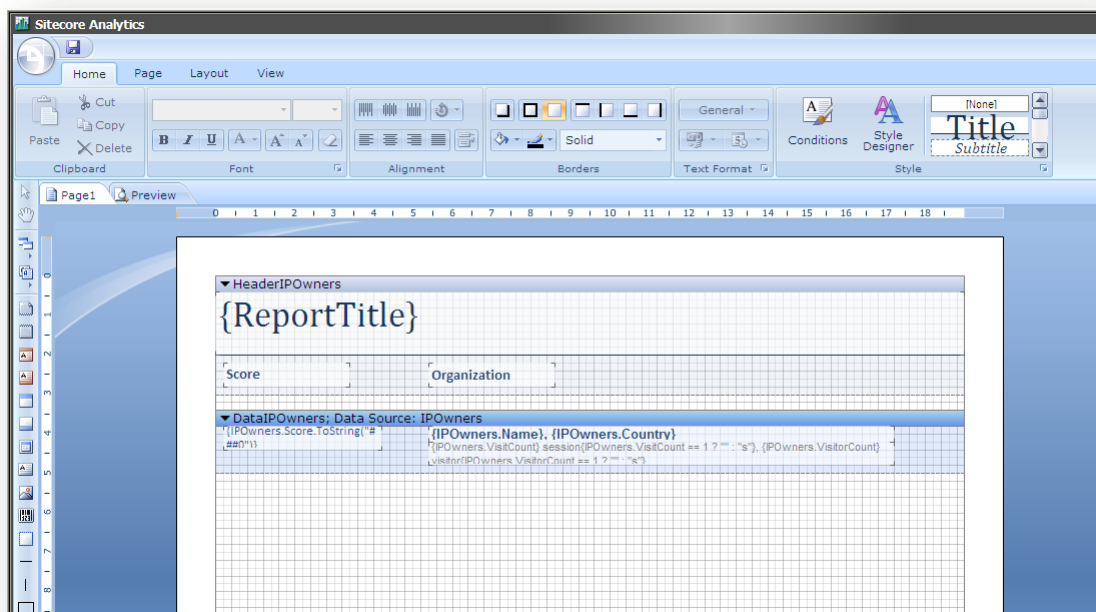
Preview your Report

Save your changes and preview the report again to check everything looks OK. When you have completed these steps your report should look something like this:

| jun 01 - sep 30, 2009 | | Filter | Save Filter | Design |

| Print | Save | |◀ ◀ Page | 2 of 2 ▶ ▶| | 🔍 100% | 📄 Whole Report | ✉ Mail Report |

Semi-pro Profile Score

Score	Organization
483	Vasakronan Service Partner AB, SE
	5 sessions, 2 visitors
453	Adorama, US
	5 sessions, 2 visitors
429	Nicam Corporation, DK
	3 sessions, 3 visitors
388	PopPhoto, US
	3 sessions, 2 visitors
271	Dell UK, GB
	4 sessions, 4 visitors
231	Canon NL, NL
	1 session, 1 visitor
202	Amazon Inc., US
	4 sessions, 1 visitor
168	TPG Internet Pty Ltd., AU
	1 session, 1 visitor
127	iiNet Limited, AU
	1 session, 1 visitor

Note
To change the name that appears at the top of your report, in the Content Editor, edit the name of the report definition item.

Chapter 6

Creating Report Filters

In this chapter, you will learn how to create a Sitecore OMS filter by extending the Profile Key Report that we created in the last chapter.

You will create a new filter that allows you to choose any profile key from the Nicam site.

This chapter contains the following sections:

- Prerequisites and Preparation

- Creating a Filter Definition Item

- Creating a Filter Class in Visual Studio

- Adding a Filter Placeholder to a SQL Query

- Configuring a Filter Definition Item

- Testing a Filter

6.1 Prerequisites and Preparation

This section contains a list changes you need to make to the *ProfileKeyReport* before you can create a new profile key report filter.

6.1.1 Prerequisites

When you create a new filter, you place the logic behind the filter in a C# class file. Therefore, to create new filters it is necessary to have some knowledge of C# and Visual Studio.

You also need to install the following:

- Sitecore OMS

- Stimulsoft Report Designer (*Stimulsoft Reports.Net 2009.2*)

- Visual Studio 2005 or 2008

6.1.2 Preparation

In the last chapter, you created a report to display top scores and organizations for the *Semi-pro* profile key. You used the following SQL query that was hard coded into the report:

```sql
WHERE ProfileKeyDefinition.Name = 'Semi-pro'
```

In this chapter, you create a filter that allows end-users to select any profile key value from the Nicam site without the requirement to change the underlying SQL query.

To do this, you first need to create an unfiltered report that displays all profile keys.

Start by duplicating the *ProfileKeyReport.mrt* file and rename it to *ProfileKeyFilter.mrt*.

Changes to the SQL Query

1. Open `ProfileKeyFilter.mrt` in the Web reports designer.

2. In the **Dictionary** panel, click *IpOwners* and then click **Edit** to open the SQL query in the **Edit Data Source** window.

3. Copy and paste the following SQL statement into the **Edit Data Source** window replacing the existing query:

```sql
SELECT IpOwner.IPOwnerId AS IPOwnerIdObject, IpOwner.Name,
IpOwner.Country,
  Count(DISTINCT Session.GlobalSessionId) AS VisitorCount,
  Count(DISTINCT Session.SessionId) AS VisitCount,
  Sum(DISTINCT ProfileKey.Value) AS Score, ProfileKeyDefinition.Name,
  ProfileKeyDefinition.Name AS Prof
FROM Session INNER JOIN
  Profile ON Session.SessionId = Profile.SessionId INNER JOIN
  GlobalSession ON GlobalSession.GlobalSessionId =
Session.GlobalSessionId
  INNER JOIN
  Ip ON Ip.IpId = Session.IpId INNER JOIN
  IpOwner ON IpOwner.IpOwnerId = Ip.IpOwnerId INNER JOIN
  Browser ON Browser.BrowserId = Session.BrowserId INNER JOIN
  ProfileKey ON ProfileKey.ProfileId = Profile.ProfileId INNER JOIN
  ProfileKeyDefinition ON ProfileKeyDefinition.ProfileKeyDefinitionId
=
    ProfileKey.ProfileKeyDefinitionId
WHERE Session.IpId = Ip.IpId AND Ip.IpOwnerId = IpOwner.IpOwnerId AND
  Session.BrowserId = Browser.BrowserId AND Session.GlobalSessionId =
  GlobalSession.GlobalSessionId
GROUP BY IpOwner.IPOwnerId, IpOwner.Name, IpOwner.Country,
  ProfileKeyDefinition.Name
  ORDER BY Prof, Score DESC, IpOwner.Name
```

4. Add a column to your data source called *Prof*, so you can display the name of the currently selected profile in your report.

5. In the **Edit Data Source** window, click **New Column**.
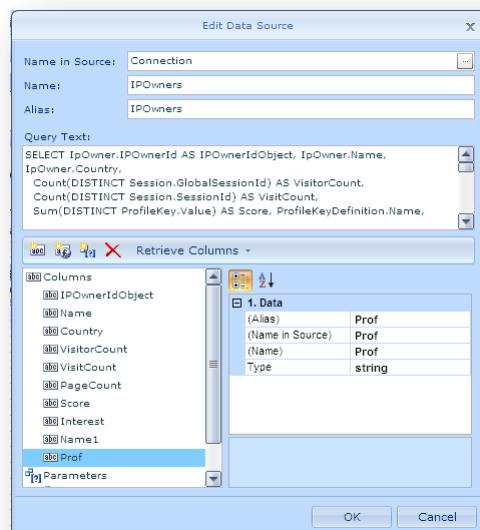
6. In the **Data** panel, enter *Prof* in the **Alias**, **Name in Source** and **Name** fields.

7. In the **Type** field, select **string**.

8. In the Stimulsoft menu, click **Save Report**.

Explanation of Changes to the SQL Query:

SQL code removed	Explanation
`Top 20`	The TOP clause limits the number of records returned to the top of the result set (based on the sorting order). Using a TOP clause helps avoid reports which can have a negative effect on performance.

SQL code added	Explanation
`ProfileKeyDefinition.Name AS Prof`	*Prof* alias added to display the name of the selected profile key as an extra column on the report.
`ORDER BY Prof, Score DESC, IPOwner.Name`	Add *Prof* to the ORDER clause
`WHERE Session.IpId = Ip.IpId AND Ip.IpOwnerId = IpOwner.IpOwnerId AND Session.BrowserId = Browser.BrowserId AND Session.GlobalSessionId = GlobalSession.GlobalSessionId`	WHERE clause includes table joins.

Changes to the Report Layout

1. Drag and drop the column **Prof** from IpOwners onto your report.
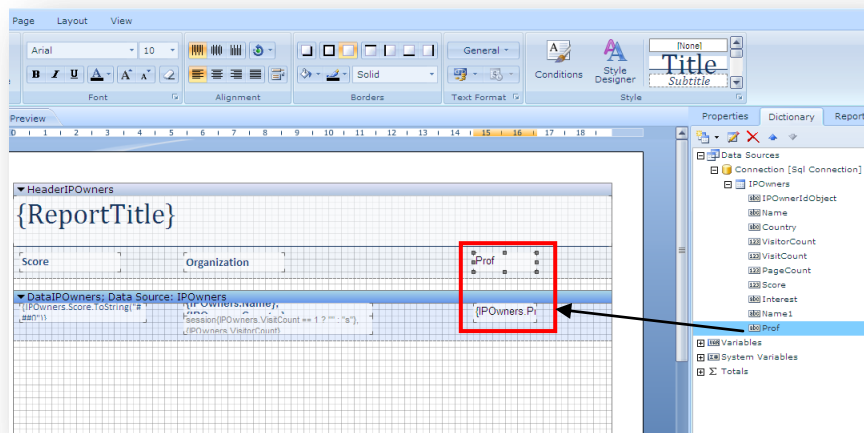
2. Drag the *Prof* text box to the header band.

3. Format this text box using the **OddRow** style.

4. Double click the *Prof* text box and in the **Expression Editor** window, enter the following expression:
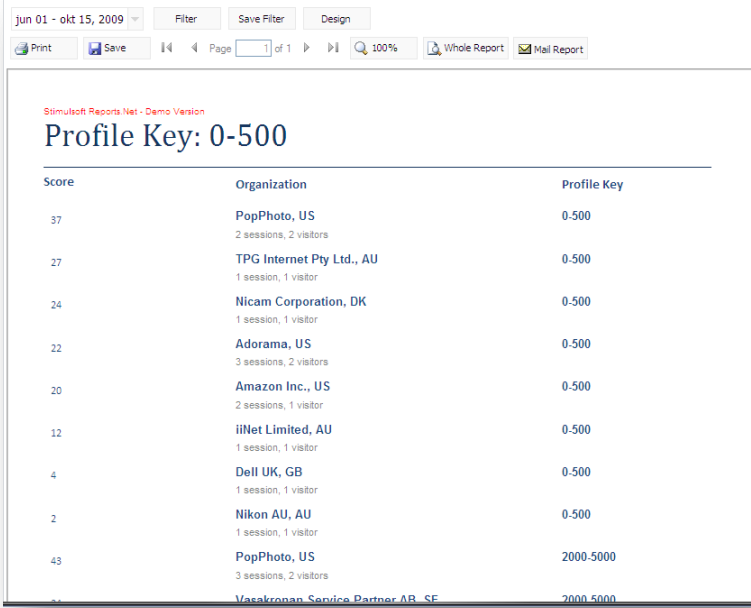
   `{(IPOwners.Score) > 0 ? "Profile Key" : ""}`

   This expression displays the profile name if the database contains a score for the selected profile key. If the score = zero, no data is displayed.

> **Note**
> Write C# expressions in the Expression Editor to format data that you retrieve from the Analytics database. You can also use conditional formatting expressions in the ribbon, **Conditions** window.

5. Add a similar expression to the other column headings for *Score* and *Organization*.

6. In your report title, enter the following expression in the **Expression Editor** to display the selected profile key name:

```
Profile Key: {IPOwners.Prof}
```

7. Finally, format the report title using the **Title** style.

8. When you have made all these changes your report should look something like this:

You cannot see which profile key you have selected if you do not make these changes.

Now follow these steps to create a new filter:

1. Create a filter criteria definition item.

2. Create your filter implementation in Visual Studio.

3. Add a placeholder to your report SQL query.

4. Configure the filter definition item.

5. Test your filter.

6.2 Creating a Filter Definition Item

A Sitecore report filter consists of the following components:

- Sitecore filter definition item

  (Location - Criteria folder in the content tree)

- C# class .cs file

  (Location - Sitecore bin folder in your Web site file directory)

Once you have created a Sitecore filter definition file, add a reference to the C# class in the **Type** field and add a placeholder for the filter in your SQL query

To create a filter definition item:

1. Open the **Content Editor** and in the content tree navigate to the following location: `/sitecore/system/Settings/Analytics/Filters/Criteria`

2. In the ribbon, click **Insert from Template**. Select the **Filter** template: `/System/Analytics/Filter`
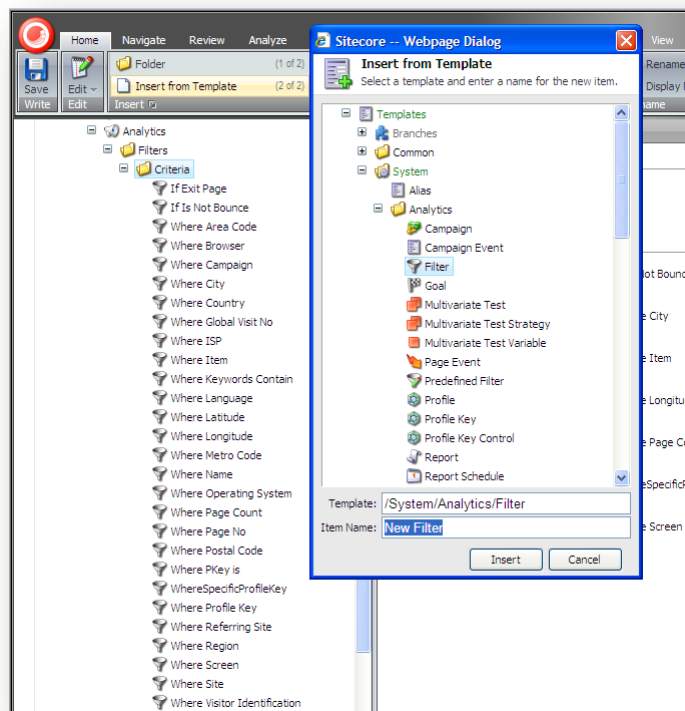
3. Name your filter *WhereSpecificProfileKey*.

4. In the **Header** field, enter:

   where the profile key is specific value

5. In the **Text** field, enter: `where the profile key is [profilekeyid,ProfileKey,,specific value]`

6. The **Type** field contains a reference to your Visual Studio filter implementation. Leave this field blank for the moment.
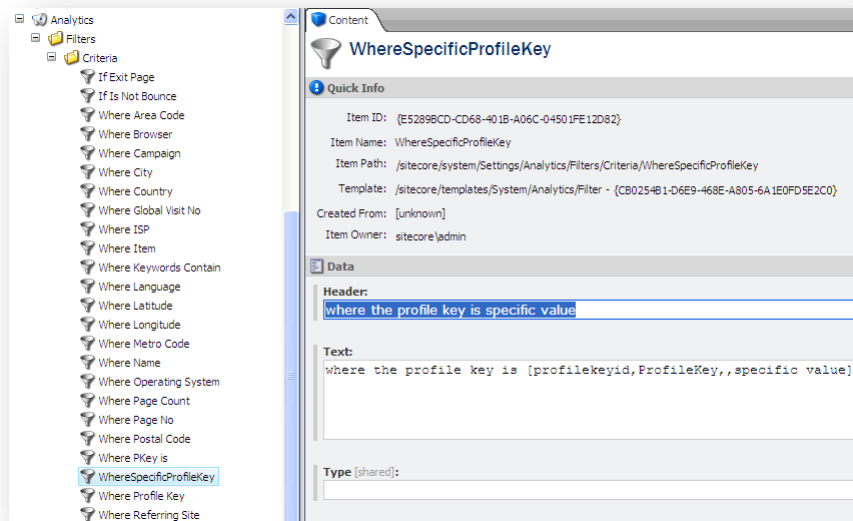
7. Save your changes.

6.3   Creating a Filter Class in Visual Studio

To create a new report filter, you must use Visual Studio to create a new C# class.

6.3.1   Create a new C# Class file

1. In Visual Studio, create a new C# project.

2. Create a new C# class (.cs file) to add to your project.

   Make sure that you add the following references to your class:

   o Sitecore.Kernel

   o Sitecore.Analytics

3. Name your class *WhereSpecificProfileKey*.

   Ensure that your new class inherits from the *Operator Filter Base* class.

   ```
   public class WhereSpecificProfileKey : OperatorFilterBase
   ```

   The Operator filter base class contains the functionality you need to create different filter conditions. Filter conditions are like rules that make it easy to create SQL statements to select data from the Analytics database. For example, Sitecore Analytics contains a pre-configured filter called *Where Profile Key* that uses the following operators:

   • Select a profile key – From all the keys in the Nicam site.

   • Select a comparison – For example, *is equal to* or *is greater than*

   • Enter a value – For example, when score is greater than *20*

   To keep things simple, your filter will only contain the first condition; the option to choose a profile key from all profile keys in the Nicam site.

   The following pop-up window appears when you select this condition in your completed filter:
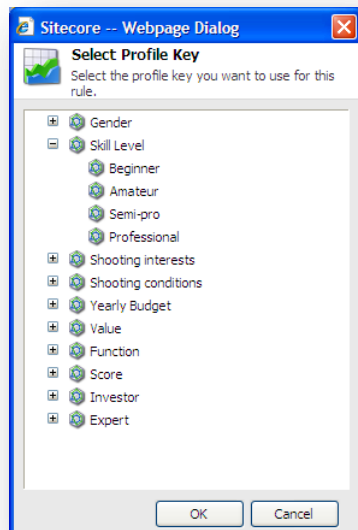
4. In your new class, use the following two C# methods to write your filter:

   • ApplyFilter

   • IsApplicable

5. Compile your code and ensure that it is added to your Sitecore bin folder:

```
C:\Inetpub\wwwroot\<site name>\WebSite\bin
```

6.3.2   Explaining the Code

You can either copy and paste the code included in this section into your C# class file or use the *ApplyFilter* and *IsApplicable* methods to write your own filter logic.

Include the following namespaces at the top of your filter class:

```csharp
using System;
using System.Data;
using System.Configuration;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Web.UI.HtmlControls;
using Sitecore.Analytics.Data.Filters.Filters;
using Sitecore.Analytics.Data.Filters;
using Sitecore.Diagnostics;
```

Declare the following variable:

```csharp
private string profileKeyId;
```

This variable is used in the *get* property of your class.

Use the following two methods:

ApplyFilter

The *ApplyFilter* method allows you to select a profile key from the Nicam site. It then inserts a SQL WHERE clause into the SQL query on your report which retrieves the chosen profile.

```csharp
public override void ApplyFilter(SqlCommand sqlCommand)
        {
            Assert.ArgumentNotNull(sqlCommand, "sqlCommand");
                sqlCommand.AddWhereClause("MyProfileKey",
        "ProfileKeyDefinition.ProfileKeyDefinitionId = '" + this.ProfileKeyId + "' ",
        base.Except);
        }
```

IsApplicable

This method tells your class exactly where in your SQL query to insert the WHERE clause by using a placeholder called *MyProfileKey*.

The only text you need to change in these methods is the placeholder name. You can either use the same name as I have or replace it with a placeholder name of your own. Both methods must have the same placeholder name.

```csharp
public override bool IsApplicable(SqlCommand sqlCommand)
        {
            Assert.ArgumentNotNull(sqlCommand, "sqlCommand");
            return (sqlCommand.HasPlaceholder("MyProfileKey") );
        }
```

Properties

Create the following properties:

```
// Properties
        public string ProfileKeyId
        {
            get
            {
                return (this.profileKeyId ?? string.Empty);
            }
            set
            {
                Assert.ArgumentNotNull(value, "value");
                this.profileKeyId = value;
            }
```

6.4   Adding a Filter Placeholder to a SQL Query

To add a filter to a report, you must add a variable to the SQL query in your report `.mrt` file that has the same name as the placeholder in your C# class. If you open any Sitecore report that contains filters, you can see all filter variables appear below the WHERE clause.
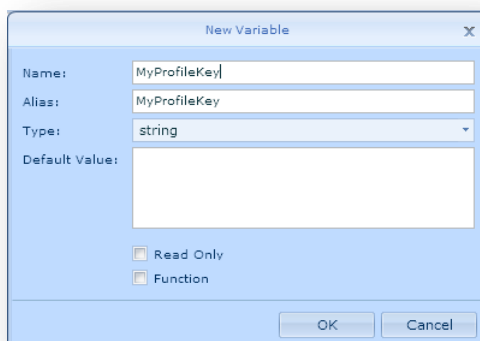

Tip

To add an existing Sitecore filter to a report, simply copy the appropriate filter placeholder into the WHERE clause of the report.

For example, if you add the `{IPOwner}` placeholder variable to your report you get all the filters related to visitor country.

To add your new filter to the *Profile Key Filter* report:

1. In the Web report designer, click **Open Report** and locate the `ProfileKeyFilter.mrt` file.

    `C:\Inetpub\wwwroot\<your site>\WebSite\sitecore\shell\Applications\Analytics\Reports`

2. First create a variable for your placeholder. In the **Dictionary** panel, click **New Item**, **New Variable**.

    In the report designer, all filters placeholders must have a corresponding variable.

3. In the **Name** field, enter *MyProfileKey* and enter the same value in the **Alias** field.

4. In the **Type** field select **string** and then click **OK**.

5. Add your variable to the SQL query. In the **Dictionary** panel, click the *IPOwners* data source and then click **Edit**.

    This enables you to make changes directly to your SQL query.

6. Add your placeholder `{MyProfileKey}` to the `WHERE` clause of your SQL Query.
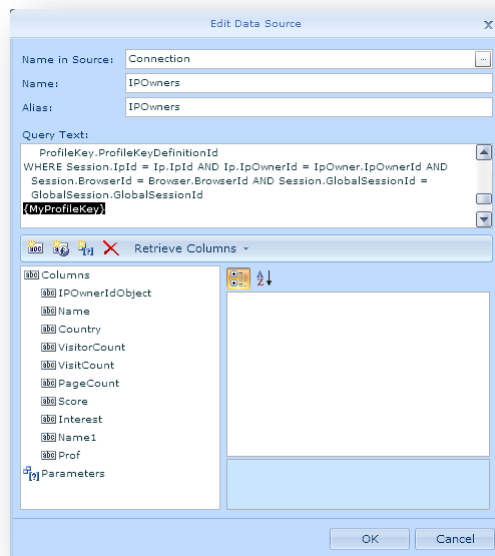

7. Click **OK** and save your changes in the report designer.

6.5 Configuring a Filter Definition Item

When you have created your filter class and added a placeholder variable to your SQL query, then configure your Sitecore filter definition item to point to your C# assembly.

1. In the Content Editor, **Criteria** folder, select your filter definition item.

   `/sitecore/system/Settings/Analytics/Filters/Criteria/WhereSpecificProfileKey`
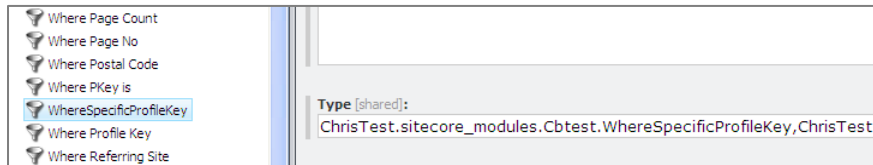
2. In the **Type** field, enter the correct path to your C# class and assembly.

   `ChrisTest.sitecore_modules.Cbtest.WhereSpecificProfileKey,ChrisTest`

   namespace   class name   assembly

3. Save your changes.

6.6 Testing a Filter

View your new report and test your filter in Sitecore Analytics.

1. In **Sitecore Analytics**, to refresh your reports, expand and close the **Reports** node.

2. Click on your new report: *ProfileKeyFilter*

3. You should see an unfiltered report that displays all the profile keys in the Nicam site.

4. Click **Filter**.

    You should see your new filter displayed in the **Filter Editor** window.
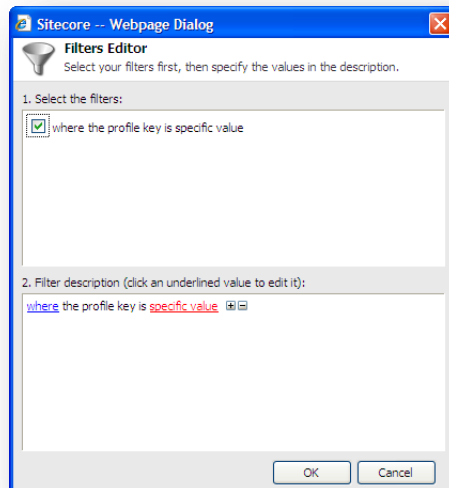
5. Select the filter and then click *specific value*.

6. The **Select Profile Key** dialog box, displays all the profiles and profile keys. Select a profile key, such as *Semi-pro*.
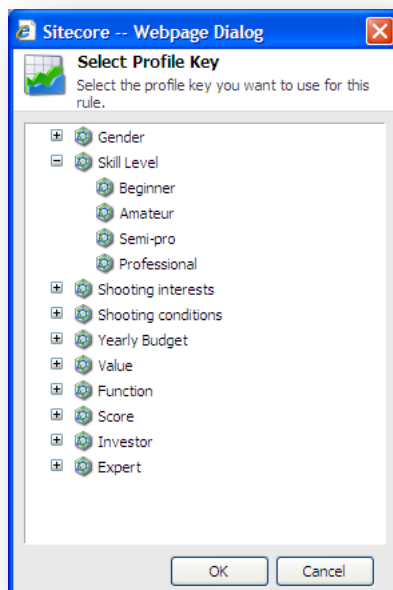
7. Click **OK** to close the **Filter Editor** window.

You should see the following report filtered by the *Semi-pro* profile key:

You can now use this report filter to show the top scores and organizations for any given profile key in the Nicam site.

Troubleshooting

If you only see a blank report in Sitecore Analytics, check the following and run your report again:

Date Range

In Sitecore Analytics, check that you have selected a wide enough date range.

1. Click the control next to the report date to open the calendar pop-up
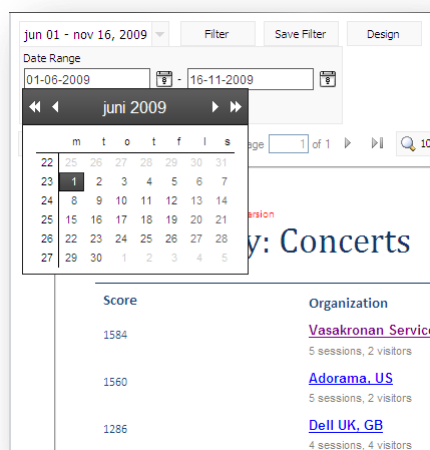
2. Select a date at least two months in the past to ensure your report includes enough data.

Print if Empty Property

1. In the report designer, select the header band.

2. In the **Properties** panel, expand **Behavior**, then select **Print if Empty**.
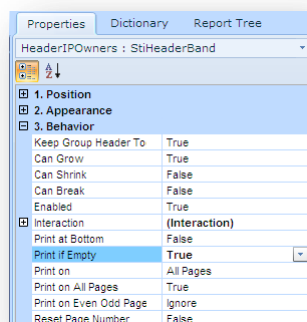
3. Set the **Print if Empty** property to *True*.

4. Save your changes and run your report again.

Chapter 7

Appendix

This section contains the following supplementary information:

- Stimulsoft Report Designer Client Comparison

- Creating a SQL Query for a Profile Key Report

7.1 Stimulsoft Report Designer Client Comparison

End user action	Web reports designer		Windows reports designer	
Create new reports	✗	Duplicate and adapt an existing report Workaround - duplicate a report, rename it and also rename the .mrt file. Then open it from Sitecore Analytics.	✓	Click New Report and save .mrt file.
Create new reports using a wizard	✗	Wizards are available but you cannot save a new report created in this way. You must use the workaround.	✓	Wizard allows you to create and save a new report
Save As	✗	You can only save the report you are working on.	✓	You can save a new report as an .mrt file in your Sitecore reports folder. You must also create a new Sitecore report definition item before you can view the report in Sitecore Analytics.
Save	✓	Click the save icon in the designer or use the File menu and click Save Report to save an existing or duplicated report.	✓	Click the save icon in the designer or use the File menu. You can also use the shortcut Ctrl S
Add a data source	✓	Option to create new data sources but requires a query to be written	✓	Yes, right click Data Connection and select New Data Source. Select check boxes to select tables form the Analytics database. Add more tables or create more complex queries using the Query Builder.
Undo	✗	No undo shortcut	✓	Use the Ctrl Z shortcut
Delete	✓	Use the Delete button in the designer. No delete shortcut	✓	Use the Delete button in the designer or use the keyboard Delete button
Format Text Text color, style, bold, italic	✓	Use the ribbon or properties panel	✓	Use the ribbon or properties panel
Format Text in Variables	✓	Select the text box that contains the variable. Change format using the ribbon or the properties panel.	✓	Select the text box that contains the variable. Change format using the ribbon or the properties panel.
Add styles to the Style Designer	✓	Click Add Style	✓	Click Add Style

End user action	Web reports designer	Windows reports designer
Import and export styles	✗ No functionality in the Web report designer, Style Designer to do this.	✓ Save Style and Open Style buttons included in the Windows report designer Style Designer
Conditional Formatting	✓ In the ribbon, click Conditions to add a value based condition or an expression based condition.	✓ In the ribbon, click Conditions to add a value based condition or an expression based condition.
Add Image	✓ Yes, but you can only link to images using a URL. You can point to any image on the Web or a Sitecore image stored as a BLOB. For example, an image from the Media Library.	✓ Browse for an image file or Link to an image using a URL – this includes images in the Media Library Add Watermark images
Add Chart	✓ Create a chart by dragging a chart from the toolbox. No wizards available in Web reports designer.	✓ Use a chart wizard or create a chart from scratch
Configure Chart	✓ Select data from the tables in the report data source.	✓ Select data from the tables in the report data source.
Change Chart Colors	✓ Yes but limited Change chart colors using the properties panel Chart Style.	✓ Yes but limited Change chart colors using the properties panel Chart Style Also use Wizards to choose a color scheme.
Add Custom Chart Colors	✓ Add a custom color scheme to the Style Designer and it is added to the available styles.	✓ Add a custom color scheme to the Style Designer and it is added to the available styles.
Add Hyperlink	✓ Properties, Behavior, Interaction, Hyperlink	✓ Properties, Behavior, Interaction, Hyperlink
Drag and drop tables into a report	✓ Yes but limited Drag and drop one column at a time	✓ Drag and drop all columns in a data source moving all columns at once.
Combine multiple tables in a data source	✓ Create a table relation but no right click options and more limited dialog boxes	✓ Right click and create a table relation
Add New Variable	✓ In the Data Dictionary, click New Item, New Variable	✓ In the Data Dictionary, click New Item, New Variable
Add New Parameter	✓ In the Data Dictionary, click New Item, New Parameter	✓ In the Data Dictionary, click New Item, New Parameter
Use SQL statements	✓ Yes but limited Create SQL queries manually. Click Edit Data Source and type a SQL query directly into a query text editor or copy and paste from another application.	✓ Create SQL queries automatically. Click Edit Data Source. Enter a query directly into the query text editor window or select tables and construct queries from scratch using the Query Builder.

7.2 Creating a SQL Query for a Profile Key Report

It is relatively easy to create complex SQL queries in the Windows Reports designer without being an expert in the SQL query language. To start creating your SQL query right click the *IpOwners* data source, click Edit, and then click *Run Query Builder*.

To create your SQL query, you need to do the following:

- Select tables and columns.

- Add aliases.

- Add groupings.

- Add a SQL Count function.

- Add a SQL SUM function.

- Add a WHERE condition to select profile key.

- Add sorting.

- Make some manual changes to your SQL query.

- Review the Final SQL statement.

- Test the SQL query.

- Save the SQL query.

Step 1: Select Tables

In the Query Builder, you can drag and drop tables straight into the query window. Choose fields (columns) by selecting the appropriate check box next to the column name.

Select the following tables and related columns:

Table	Columns
Session	GlobalSessionId, SessionId
Global Session	
IP	
IPOwner	IPOwnerId, Name, Country
Profile	
ProfileKey	Value
ProfileKeyDefinition	Name

When you have selected these tables and columns the following SQL query is automatically generated and displayed in the lower panel of the Query Builder:

```
SELECT Session.SessionId, Session.GlobalSessionId, IPOwner.IpOwnerId,
  IPOwner.Name, IPOwner.Country, Profile.Total, ProfileKeyDefinition.Name
FROM Session INNER JOIN
  GlobalSession ON GlobalSession.GlobalSessionId = Session.GlobalSessionId
  INNER JOIN IP ON IP.IpId = Session.IpId INNER JOIN
  IPOwner ON IPOwner.IpOwnerId = IP.IpOwnerId INNER JOIN
  Profile ON Session.SessionId = Profile.SessionId INNER JOIN
  ProfileKey ON Profile.ProfileId = ProfileKey.ProfileId INNER JOIN
  ProfileKeyDefinition ON ProfileKeyDefinition.ProfileKeyDefinitionId =
  ProfileKey.ProfileKeyDefinitionId
```

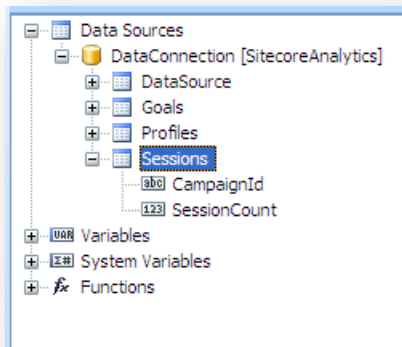The SELECT part of the statement lists the tables and columns that we want include in our query.
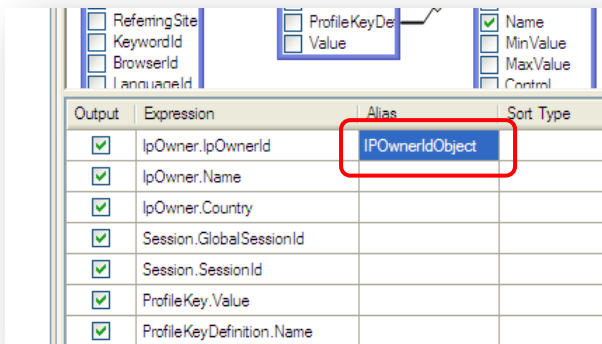
Use the following syntax `<table>.<column>`. For example, to select the *ProfileKey* table and the *Value* column enter *Profile.Value.*

The `FROM` part of the statement refers to the joins between tables. All tables are linked by their primary keys. An INNER JOIN is created automatically. This means that values are taken from columns in each table that is included in the join.

Step 2: Add Aliases

An alias is a substitute name for a table or table relation.

For example, in the *Sessions* Data Source below, *SessionCount* is an alias used to replace the following SQL statement: *CampaignEvent.CampaignId, Count(*).*

In SQL you create an alias for a table using the AS syntax:

`<Table>.<column> AS <alias name>`

Example

`CampaignEvent.CampaignId, Count(*) AS SessionCount`

In the **Query Builder**, use the **Alias** column to add new aliases to your SQL query.

Create aliases for the following table/column relationships:

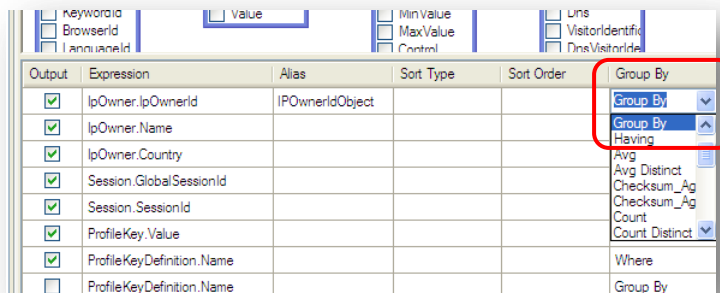| Table Relationship	Alias Name
IPOwner.IPOwnerId	IPOwnerIdObject
Session.GlobalSessionId	VisitorCount
Session.SessionId	VisitCount

Table Relationship	Alias Name
ProfileKey.Value	Score

Step 3: Add Groupings

Define the sort order of the columns that you have selected by using GROUP AS. All columns that appear in the SELECT column list at the beginning of your SQL query must also appear in the GROUP AS clause.

In the Query Builder select Group By in the drop-down menu for the following:

- IPOwner.IpOwnerId
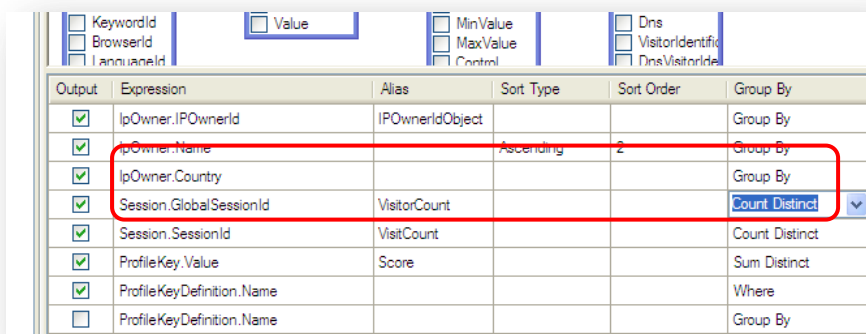- IPOwner.Name
- IPOwner.Country

In the SQL query, the following SQL code is created:

```
GROUP BY IPOwner.IpOwnerId, IPOwner.Name, IPOwner.Country
```

Step 4: Add a SQL COUNT Function

Add COUNT and DISTINCT to:

- Session.GlobalSessionId (Visitor Count)- the total number of visitors recorded
- Session.SessionId (Visit Count)- the total number of sessions recorded

COUNT means that the number of global and individual sessions is counted to give a total for each. DISTINCT means that no duplicate values are displayed in the results.

Step 5: Add a SQL SUM Function

Add a SUM DISTINCT function to ProfileKey.Value (Score) using the **Group By** drop-down menu.

Sum calculates a total of all values for a single profile key, to give you a total profile key score for each organization listed.

Add SUM and DISTINCT to:

- ProfileKey.Value

Add SUM DISTINCT by using the same Group By drop-down menu.

Step 6: Add WHERE Condition to Select a Profile Key

Add the following WHERE condition: WHERE ProfileKeyDefinition.Name = *'Semi-pro'*

This condition acts like a filter so that only Scores for the *'Semi-pro'* profile key are displayed in the query results.

Step 7: Add SORTING

Add SORTING to *IPOwner.Name* (Ascending) and *Profile.Total* (Descending). Select Sort Order 1 for Score and 2 for *IPOwner.Name*.

This means that the results of your query are sorted first by Score and then by Organization.

Step 8: Manual Additions to your SQL Query

SQL Top

We only want to select the top 20 records that are returned. To do this we can use the SQL Top clause. This means we will only see the top 20 results rather than all the records that are generated this query. Enter Top 20 manually at the beginning of your SQL statement.

Group By

Earlier, you added a WHERE condition to: *ProfileKeyDefinition.Name*.

Now add *ProfileKeyDefinition.Name* to your Group BY clause.

You need to add this because all columns that appear in the SELECT column list at the beginning of your SQL query must also appear in the GROUP AS clause. Add this manually to the GROUP BY clause.

Step 9: Review the Final SQL Statement

The Query Builder window and SQL query should now look like this:

The finished SQL query:

```sql
SELECT TOP 20 IpOwner.IPOwnerId AS IPOwnerIdObject, IpOwner.Name, IpOwner.Country,
  Count(DISTINCT Session.GlobalSessionId) AS VisitorCount,
  Count(DISTINCT Session.SessionId) AS VisitCount,
  Sum(DISTINCT ProfileKey.Value) AS Score, ProfileKeyDefinition.Name
FROM Session INNER JOIN
  Profile ON Session.SessionId = Profile.SessionId INNER JOIN
  GlobalSession ON GlobalSession.GlobalSessionId = Session.GlobalSessionId
  INNER JOIN
  Ip ON Ip.IpId = Session.IpId INNER JOIN
  IpOwner ON IpOwner.IpOwnerId = Ip.IpOwnerId INNER JOIN
  Browser ON Browser.BrowserId = Session.BrowserId INNER JOIN
  ProfileKey ON ProfileKey.ProfileId = Profile.ProfileId INNER JOIN
  ProfileKeyDefinition ON ProfileKeyDefinition.ProfileKeyDefinitionId =
  ProfileKey.ProfileKeyDefinitionId

WHERE ProfileKeyDefinition.Name = 'Semi-pro'

GROUP BY IpOwner.IPOwnerId, IpOwner.Name, IpOwner.Country,
  ProfileKeyDefinition.Name

  ORDER BY Score DESC, IpOwner.Name
```
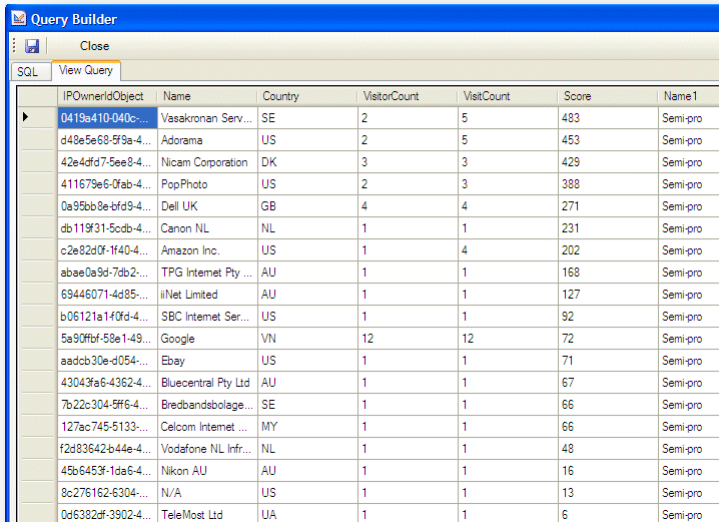
Step 10: Test the SQL Query

Click *View Query* to see the columns that you selected are returned and that there are no errors in your SQL code.

If you get an error, check your SQL query again.

Step 11: Save your Query

Click the Save icon to save your query.

In the **New Data Source** window, click **Retrieve Columns** and click **OK** to save your query.

In the **Select Data** window, click **OK** to return to the Report Designer.

Expand the Data Connection node in the Data Dictionary to view the columns in the *IpOwners* Data Source that you just created: