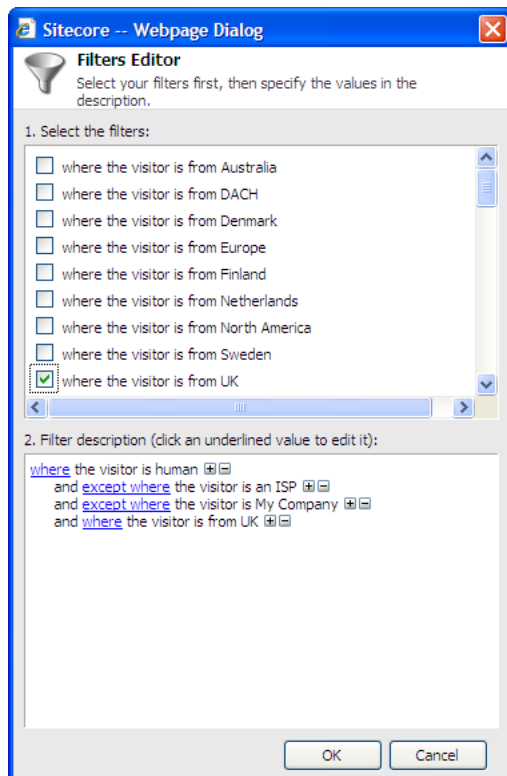


Creating a Report Filter in Sitecore OMS

Introduction

Welcome to my second blog on creating Sitecore OMS reports.

If you select any report in Sitecore Analytics, you can see a number of different pre-configured filters displayed in the Filters Editor window. You can select or clear any of these using the appropriate check box to make your reports more specific and meaningful. For example, in the *Latest Company Sessions* report you can filter by visitor country or only show human visits to exclude visits by robots from your reports.

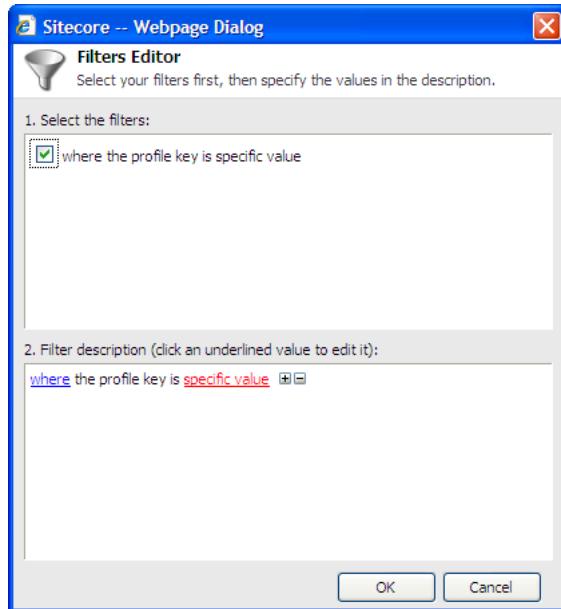


The aim of this blog is to show you how to create a new filter from scratch and provide you with all the steps that you need to achieve this goal.

You will create a new filter that enables you to filter a report by profile key, then you will associate the filter with a report (similar to the one created in my previous post) which then displays the top scores and organizations for the chosen key. A filter like this can provide marketers with a useful insight into the browsing behavior of visitors to their Web site, enabling them to see at a glance which organizations could be potential customers or important leads to follow up.

Before we start, we need to make some small changes to the profile key report used in my previous blog. This is necessary so that we can begin with an unfiltered report that displays all profile keys and scores in the same report.

When you have completed all the steps in this blog you will see the following filter displayed in the Filters Editor window.



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

Preparation

In my first blog, I created a report for a single profile key displaying top scores and organizations for the *Semi-pro* profile key. The SQL query that specified which profile key to select from the Analytics database was hard coded into the report:

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

Here is a link to my first blog:

<http://cbushnell.wordpress.com/2009/10/06/creating-an-oms-profile-score-report/>

In my second blog, I will 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 in one long scrolling report. Start by making a duplicate of the *ProfileKeyReport.mrt* file and rename it to *ProfileKeyFilter.mrt*.

Open *ProfileKeyFilter.mrt* in the Report Designer. In the Dictionary panel, right click *IpOwners* and then click *Edit* to open the SQL query.

Changes to the SQL Query

Make the following changes to the SQL query:

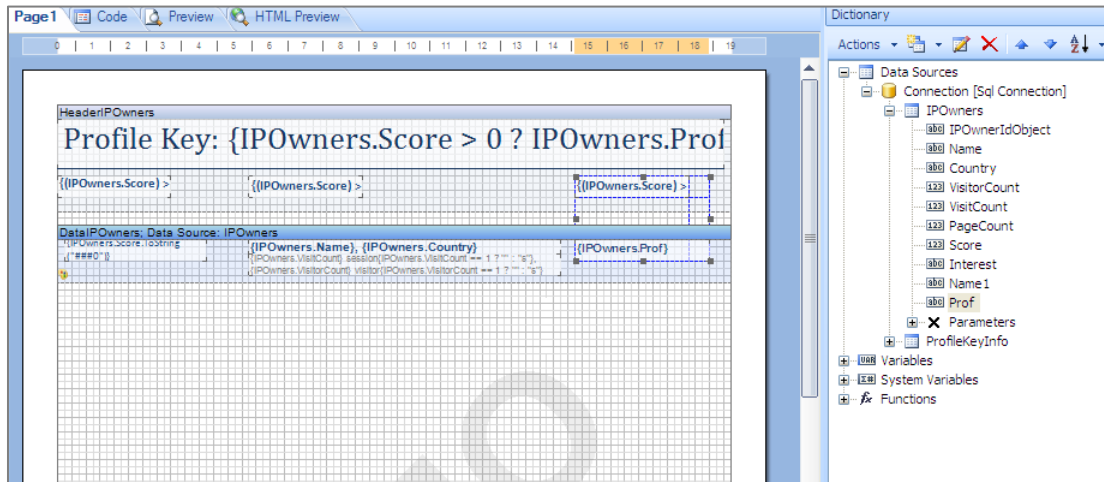
- Remove `Top 20` from the beginning of the SQL statement
- Add the following alias: `ProfileKeyDefinition.Name AS Prof`
- Change the Order clause: `ORDER BY Prof, Score DESC, IPOwner.Name`
- Add the following where clause:
`WHERE Session.IpId = Ip.IpId AND Ip.IpOwnerId = IpOwner.IpOwnerId AND
Session.BrowserId = Browser.BrowserId AND Session.GlobalSessionId =
GlobalSession.GlobalSessionId`

Alternatively just copy and paste the entire query to create a new *ProfileKeyFilter* report. This unfiltered report can be the starting point for this blog.

```
SELECT IpOwner.IPOwnerId AS IPOwnerIdObject, IpOwner.Name, IpOwner.Country,
Count(DISTINCT Session.GlobalSessionId) AS VisitorCount,
Count(DISTINCT Session.SessionId) AS VisitCount,
Sum(DISTINCT Profile.Total) 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
```

Changes to the Layout of your Report

1. Drag and drop the column *Prof* from the Data Dictionary onto your report and add a text box in the header to display the selected Profile Key name.



2. Double click the text box you added for the Profile Key name.
3. In the Text Editor, **Expression** tab 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.

4. Add a similar expression to the other column headings for *Score* and *Organization*.
5. In your report title, enter the following expression to display the selected profile key name:
Profile Key: {IPOwners.Prof}

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

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

Stimulsoft Reports Net - Demo Version

Jun 01 - okt 15, 2009 Filter Save Filter Design

Print Save Page 1 of 1 100% Whole Report Mail Report

Profile Key: 0-500

Score	Organization	Profile Key
37	PopPhoto, US 2 sessions, 2 visitors	0-500
27	TPG Internet Pty Ltd., AU 1 session, 1 visitor	0-500
24	Nicam Corporation, DK 1 session, 1 visitor	0-500
22	Adorama, US 3 sessions, 2 visitors	0-500
20	Amazon Inc., US 2 sessions, 1 visitor	0-500
12	iiNet Limited, AU 1 session, 1 visitor	0-500
4	Dell UK, GB 1 session, 1 visitor	0-500
2	Nikon AU, AU 1 session, 1 visitor	0-500
43	PopPhoto, US 3 sessions, 2 visitors	2000-5000
43	Vasakronan Service Partner AB, SE 3 sessions, 2 visitors	2000-5000

The unfiltered report displays all the profile keys contained in the Nicam site, listing all organizations by highest score.

Note: When you have made these changes, use the date filter to select a date at least three months before the current date, so that enough data is available from the Nicam site to make your reports meaningful.

Creating a Profile Key Filter

To create a new filter, you need to complete the following steps:

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

Step 1: Create a Filter Definition Item

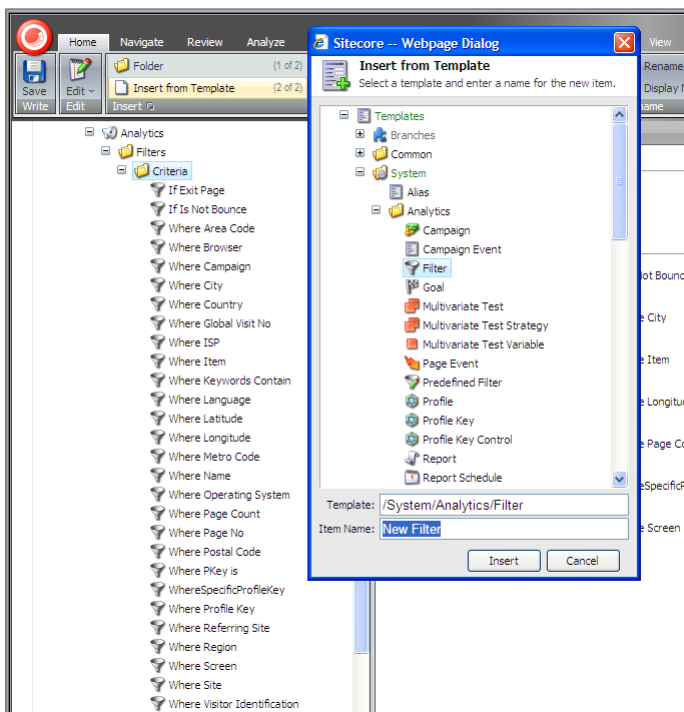
A Sitecore report filter consists of the following components:

- Sitecore filter definition item (stored in the Criteria folder in the content tree)
- C# class .cs file (stored in the 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

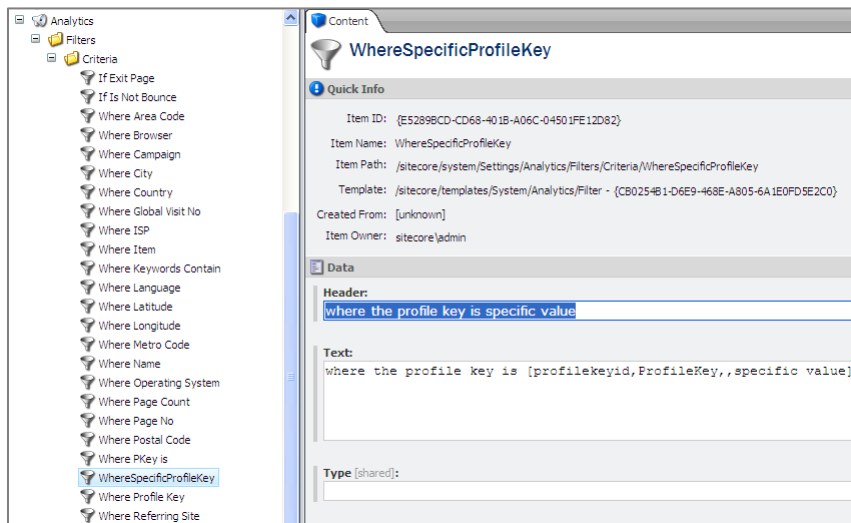
To create a filter definition item:

1. Open the Content Editor and navigate to the following location in the content tree.
/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*.

- In the **Header** field, enter:
where the profile key is specific value
- In the **Text** field, enter:
where the profile key is [profilekeyid,ProfileKey,,specific value]
- The **Type** field contains a reference to your Visual Studio filter implementation. Leave this field blank for the moment.



- Save your changes.

Step 2: Create your Filter Implementation in Visual Studio

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

Create a new C# Class file

- Create a new C# project.
- Create a new C# Class .cs file to add to your project.
- Make sure that you add the following references to your class:
 - Sitecore.Kernel
 - Sitecore.Analytics
- Name your class *WhereSpecificProfileKey*.
- Ensure that your new class inherits from the *Operator Filter Base* class.

```
public class WhereSpecificProfileKey : OperatorFilterBase
```

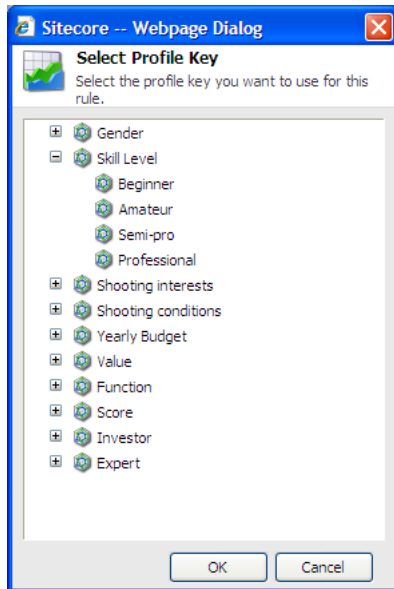
The Operator filter base contains the functionality you need to create different filter conditions. Filter conditions are like rules that make it easy to create SQL statements to select data from the Analytics database. For example, Sitecore Analytics has a pre-configured filter called *Where Profile 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 one condition; the option to choose a profile key from those available in the Nicam site.

The following pop-up window appears when you select this condition in your filter.



6. In your new class, use the following two methods to create your filter:

- *ApplyFilter*
- *IsApplicable*

C# code used in this filter

```
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;

namespace ChrisTest.sitecore_modules.Cbtest
{
    public class WhereSpecificProfileKey : OperatorFilterBase
    {
        // Fields
        private string profileKeyId;

        // Methods
        public override void ApplyFilter(SqlCommand sqlCommand)
        {
            Assert.ArgumentNotNull(sqlCommand, "sqlCommand");
        }
    }
}
```

```

        sqlCommand.AddWhereClause("MyProfileKey",
"ProfileKeyDefinition.ProfileKeyDefinitionId = '" + this.ProfileKeyId + "' ", base.Except);
    }

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

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

```

You can either copy and paste this code into your C# class file or just use these methods to construct your own logic.

Explaining the code:

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

```

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:

```

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.

```
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.

```
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;
    }
}
```

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

C:\inetpub\wwwroot\<site name>\WebSite\bin

Step 3: Add a Placeholder to your 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 the filter variables below the WHERE clause.

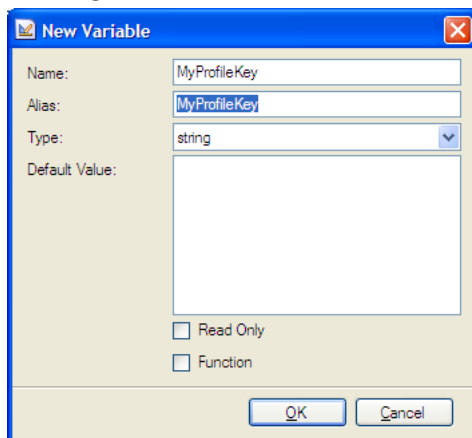
```
IpOwner,  
Ip  
where  
Session.Timestamp >= @St  
Session.Timestamp <= @En  
Session.GlobalSessionId  
Session.IpId = Ip.IpId a  
Ip.IpOwnerId = IpOwner.I  
{Session}  
{GlobalSession}  
{IpOwner}  
{Ip}
```

Tip!

To add an existing Sitecore filter to a report, simply copy the appropriate filter placeholder into the WHERE clause of the report you want to add it to. For example, if you add the {IPOwner} placeholder variable to your report you get all the filters related to visitor country as shown in the example in the introduction.

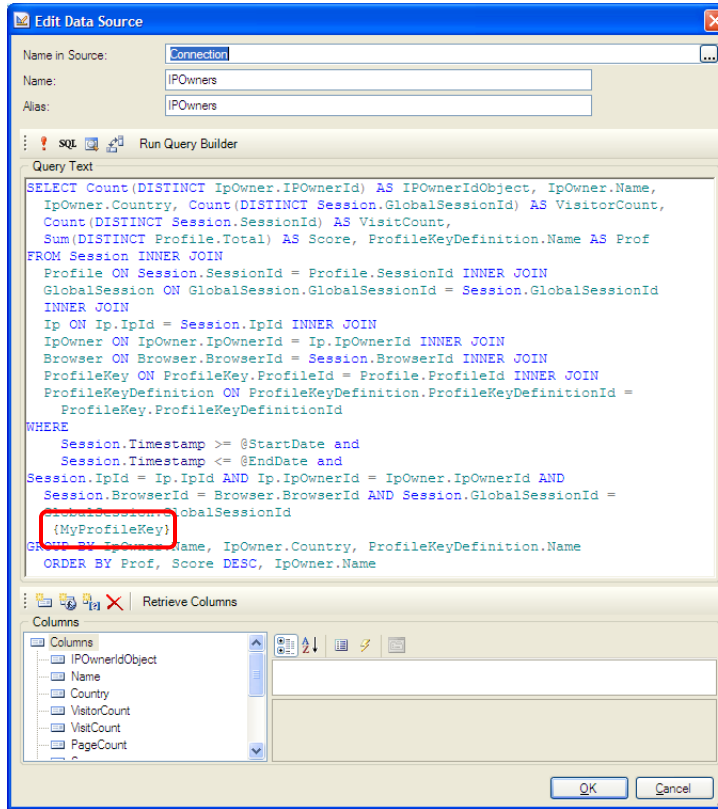
To add your new filter to the *Profile Key Filter* report:

1. In the Report Designer, click *Open Report* and locate the file *ProfileKeyFilter.mrt*.
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*.
3. In the **Name** field enter *MyProfileKey* and enter the same value in the **Alias** field. Keep the Type as string and click OK.



4. Next, add your variable to the SQL query. Click *IPOwners* and then click *Edit* to edit the Data Source. This enables you to make changes directly to your SQL query.

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



6. Click OK and save your changes in the report designer.

Step 4: Configure your Sitecore Filter Definition Item

When you have created your filter class and added a placeholder variable to your SQL query you then must 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`

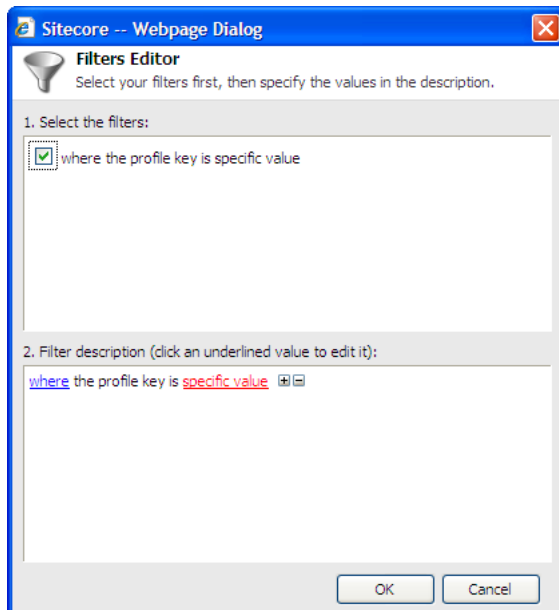
namespace class name assembly

3. Save your changes.

Step 5: Test your Filter

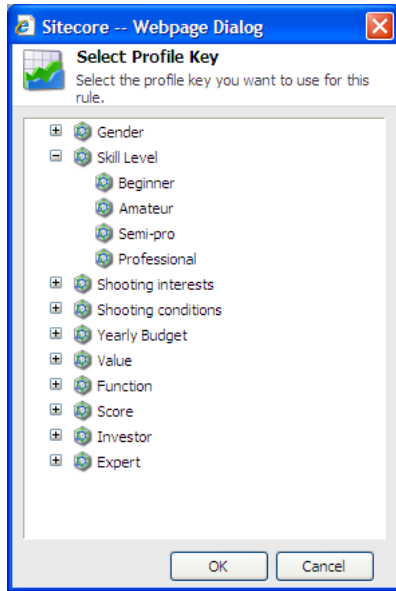
View your new report and test your filter in Sitecore Analytics.

1. In Sitecore Analytics, to refresh your reports, close and then expand 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 profiles and profile keys. Select a profile key, such as *Semi-pro*.



7. Click OK in the **Select Profile Key** dialog box and the **Filter Editor** window. You should see the following report filtered by the *Semi-pro* profile key:

Stimulsoft Reports.Net - Demo Version

Profile Key: Semi-pro

Score	Organization	Profile Key
483	Vasakronan Service Partner AB, SE 5 sessions, 2 visitors	Semi-pro
453	Adorama, US 5 sessions, 2 visitors	Semi-pro
429	Nicam Corporation, DK 3 sessions, 3 visitors	Semi-pro
388	PopPhoto, US 3 sessions, 2 visitors	Semi-pro
271	Dell UK, GB 4 sessions, 4 visitors	Semi-pro
231	Canon NL, NL 1 session, 1 visitor	Semi-pro
202	Amazon Inc., US 4 sessions, 1 visitor	Semi-pro
168	TPG Internet Pty Ltd., AU 1 session, 1 visitor	Semi-pro
127	iiNet Limited, AU 1 session, 1 visitor	Semi-pro
92	SBC Internet Services, US 1 session, 1 visitor	Semi-pro
72	Google, VN 12 sessions, 12 visitors	Semi-pro
71	Ebay, US 1 session, 1 visitor	Semi-pro

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