Hands-On Lab

Reporting on Silverlight usage data

Lab version: 1.0.0

Last updated: 3/28/2011



Contents

[Overview 3](#_Toc289092219)

[Exercise 1: Create a Silverlight application to display usage-tracking reports 4](#_Toc289092220)

[Task 1 – Retrieving the Banner Ad Usage-Tracking Data 5](#_Toc289092221)

[Task 2 – Deploying the Silverlight Application and Test the Usage-Reporting Functionality 7](#_Toc289092222)

[Exercise 1 Verification 9](#_Toc289092223)

[Summary 10](#_Toc289092224)

Overview

* 1. Understanding how end users interact with Silverlight applications is just as important as understanding metrics about web pages on web sites. The SharePoint Client Object Model and the Silverlight Toolkit makes it easy to generate visual reports for Silverlight usage-tracking data stored in SharePoint lists.

# Objectives

* 1. This lab will demonstrate how you can generate visual reports for Silverlight usage-tracking data stored in SharePoint lists. To demonstrate these concepts the reader will
  + Create a Silverlight application to display usage-tracking reports.

# System Requirements

* 1. You must have the following items to complete this lab:
  + 2010 Information Worker Demonstration and Evaluation Virtual Machine
  + Visual Studio 2010

# Setup

* 1. You must perform the following steps to prepare your computer for this lab...
  2. Download the [2010 Information Worker Demonstration and Evaluation Virtual Machine](http://www.microsoft.com/downloads/en/details.aspx?FamilyID=751fa0d1-356c-4002-9c60-d539896c66ce&displaylang=en) and create the Hyper-V image.
  3. Perform the steps in ”Creating self services Silverlight applications”, Exercises 1 and 3.
  4. Perform the steps in ”Capturing Silverlight application usage data”.
  5. Download and install the [Silverlight 4 Toolkit](http://silverlight.codeplex.com/releases/view/43528).

# Exercises

* 1. This Hands-On Lab comprises the following exercises:
  2. Create a Silverlight application to display usage-tracking reports
  3. Estimated time to complete this lab: **15 minutes**.

# Starting Materials

* 1. This Hands-On Lab includes the following starting materials.
  + **Visual Studio solutions.** The lab provides the following Visual Studio solutions that you can use as starting point for the exercises.
    - **<INSTALL>\** **Labs\ReportingOnSilverlightUsageData \Source\Begin\SL.Banner.Stats.sln**: A visual Studio 2010 Solution containing the starter code for the Silverlight application that displays usage-tracking reports, a web project to test the Silverlight application, and a SharePoint project to deploy the Silverlight application to a test page in SharePoint.
      1. **Note:** This lab uses the SharePoint Client Object Model and the Silverlight Toolkit (http://silverlight.codeplex.com) to generate visual reports for Silverlight usage-tracking data stored in SharePoint lists. There are also other ways to track, store, and report on usage data for Silverlight applications.
      2. The Microsoft Silverlight Analytics Framework (http://msaf.codeplex.com) allows you to use external web site analytics services to store usage data for Silverlight applications. The Microsoft Silverlight Analyitcs Framework also works with the Microsoft Silverlight Media Framework (http://smf.codeplex.com).
      3. The Microsoft Silverlight Media Framework enables developers to quickly deploy a robust, scalable, customizable media player for IIS Smooth Streaming delivery. The SMF builds on the core functionality of the Smooth Streaming Client (formerly known as the "Smooth Streaming Player Development Kit") and adds a large number of additional features, including an extensibility API that allows developers to create plugins for the framework.

Exercise 1: Create a Silverlight application to display usage-tracking reports

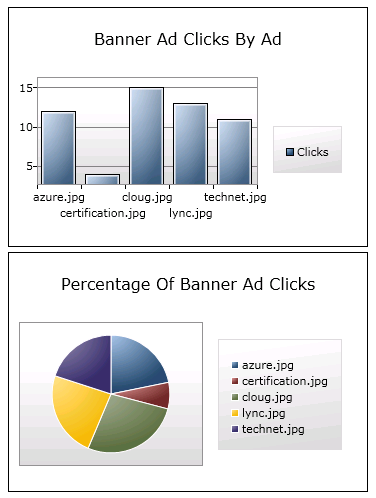
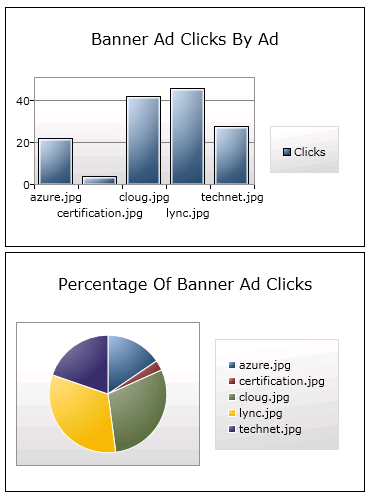
1. In this exercise you will create a Silverlight to display usage-tracking reports. The Silverlight application uses the SharePoint Client Object model to retrieve usage-tracking data from the Ad Tracking list. The charting controls that come with the Silverlight Toolkit are used to generate visual reports based on the usage-tracking data.

Task 1 – Retrieving the Banner Ad Usage-Tracking Data

* 1. In this task, you will use the SharePoint Client Object Model to query the Ad Tracking list and return the usage-tracking data.
  2. Open the following Visual Studio 2010 Solution by double-clicking on it **<INSTALL>\** **Labs\ReportingOnSilverlightUsageData \Source\Begin\SL.Banner.Stats.sln.**
  3. In the solution Explorer, right-click the **MainPage.xaml** file and select **View Code**.
  4. In the **GetAds** method, add the following code. (Snippet 6.4.1)
     1. C#
     2. //Retrieve the banner ads show in the rotating Silverlight banner control
     3. adsList = clientContext.Web.Lists.GetByTitle("Ads");
     4. CamlQuery adsQuery = new Microsoft.SharePoint.Client.CamlQuery();
     5. adsQuery.ViewXml = @"<View>" +
     6. @"<Query>" +
     7. @"<OrderBy>" +
     8. @"<FieldRef Name='FileLeafRef' Ascending='True' />" +
     9. @"</OrderBy>" +
     10. @"</Query>" +
     11. @"<ViewFields>" +
     12. @"<FieldRef Name='FileLeafRef' />" +
     13. @"</ViewFields>" +
     14. @"</View>";
     15. adsListItemCollection = adsList.GetItems(adsQuery);
     16. clientContext.Load(adsListItemCollection);
     17. **Note:** This code uses the SharePoint Client Object Model to return the list of banner ads from the Ads list.
  5. In the **GetAds** method, under the code you just inserted, add the following code. (Snippet 6.4.2)
     1. C#
     2. //Retrieve the ad tracking data stored in the ad tracking list
     3. adTrackingList = clientContext.Web.Lists.GetByTitle("Ad Tracking");
     4. CamlQuery adDataQuery = new Microsoft.SharePoint.Client.CamlQuery();
     5. adDataQuery.ViewXml = @"<View>" +
     6. @"<Query>" +
     7. @"<OrderBy>" +
     8. @"<FieldRef Name='Title' Ascending='True' />" +
     9. @"</OrderBy>" +
     10. @"</Query>" +
     11. @"<ViewFields>" +
     12. @"<FieldRef Name='Title' />" +
     13. @"</ViewFields>" +
     14. @"</View>";
     15. adTrackingListItemCollection = adTrackingList.GetItems(adDataQuery);
     16. clientContext.Load(adTrackingListItemCollection);
     17. //Execute the query to perform all the batched operations
     18. clientContext.ExecuteQueryAsync(onQuerySucceeded, onQueryFailed);
     19. **Note:** This code uses the SharePoint Client Object Model to return the usage-tracking data from the Ad Tracking list.
  6. In the **MainPage class**, add the following code. (Snippet 6.4.3)
     1. C#
     2. private void onQuerySucceeded(object sender, ClientRequestSucceededEventArgs args)
     3. {
     4. UpdateUIMethod updateUI = DisplayReports;
     5. this.Dispatcher.BeginInvoke(updateUI);
     6. }
     7. private void onQueryFailed(object sender, ClientRequestFailedEventArgs args)
     8. {
     9. errorMessage = args.Message;
     10. UpdateUIMethod displayError = DisplayError;
     11. this.Dispatcher.BeginInvoke(displayError);
     12. }
     13. private delegate void UpdateUIMethod();
     14. private void DisplayError()
     15. {
     16. HtmlPage.Window.Alert(errorMessage);
     17. }
     18. **Note:** This code defines the success and failure event handlers for the SharePoint Client Object Model callback delegate methods that fire when the queries are executed against the Ads and Ad Tracking lists.
  7. In the **MainPage class**, add the following code. (Snippet 6.4.4)
     1. C#
     2. //This collection is bound to the charts
     3. adCollection = LayoutRoot.Resources["AdCollection"] as AdCollection;
     4. //Loop through the collection of ads displayed by the rotator control
     5. foreach (ListItem adItem in adsListItemCollection)
     6. {
     7. int counter = 0;
     8. //Create a new ad
     9. Ad ad = new Ad();
     10. //Set the title
     11. ad.Title = adItem["FileLeafRef"].ToString();
     12. //Loop through the Ad Tracking list data
     13. foreach (ListItem adDataItem in adTrackingListItemCollection)
     14. {
     15. //Parse out the title column
     16. string adTitle = adDataItem["Title"].ToString();
     17. int lastSlashIndex = adTitle.LastIndexOf("/");
     18. adTitle = adTitle.Substring(lastSlashIndex + 1);
     19. //If the title column for the ad tracking data matches the current
     20. //ad being processed from the ads list then iterate the counter
     21. if (ad.Title == adTitle)
     22. {
     23. counter++;
     24. }
     25. }
     26. //Set the count property on the ad to indicate how many times the ad was viewed
     27. ad.Count = counter;
     28. //Ad the ad to the collection bound to the charts
     29. adCollection.Add(ad);
     30. }
     31. **Note:** This code create an instance of the adCollection Observable Collection, creates an instance of the Ad class corresponding to each each in the Ads list, counts the number of times each banner ad was clicked, sets the Count property on each Ad instance, and ads the Ad instances to the adCollection Observable Collection. The chart controls in the Silverlight application are bound to the adCollection Observable Collection.

1. Task 2 – Deploying the Silverlight Application and Test the Usage-Reporting Functionality

In this task, you will deploy the Silverlight application to the http://intranet.contoso.com site collection and test the usage-reporting functionality.

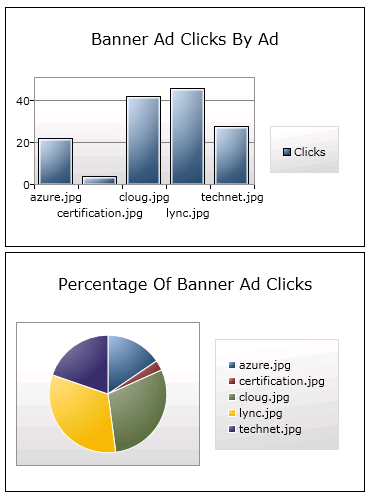
* 1. In the **Visual Studio 2010 Solution Explorer**, right-click the **SL.Banner.Stats.WP** **project** and select **Deploy**.
  2. When Visual Studio 2010 indicates the deployment has succeeded, open **Internet Explorer** and navigate to **http://intranet.contoso.com/SitePages/SL.Banner.StatsWebPartPage.aspx.**
  3. Notice the charts display the usage-tracking data from the Ad Tracking list.
     1. 
     2. Figure 1
     3. Usage-Tracking Charts
     4. **Note:** Your charts may have different data in them, depending how many times you clicked each banner ad.
  4. In the top portion of the page, click the **banner ad** several times.
  5. Wait until a different banner ad appears and click the **banner ad** several times.
  6. **Repeat** this process for each banner ad.
  7. **Refresh** the web page.
  8. Notice the charts display the updated usage-tracking data from the Ad Tracking list.
     1. 
     2. Figure 2
     3. Usage-Tracking Charts
     4. **Note:** Your charts may have different data in them, depending how many times you clicked each banner ad.

Exercise 1 Verification

* 1. In order to verify that you have correctly performed all steps of exercise 1, proceed as follows:

#### Verification 2

In this verification, you will browse to the web page in the SharePoint site where you deployed the Silverlight application and verify it works.

* 1. Open **Internet Explorer** and navigate to **http://intranet.contoso.com/SitePages/SL.Banner.StatsWebPartPage.aspx.**
  2. Notice the charts display the usage-tracking data from the Ad Tracking list.
     1. 
     2. Figure 3
     3. Usage-Tracking Charts
     4. **Note:** Your charts may have different data in them, depending how many times you clicked each banner ad.

Summary

* 1. In this lab, you have seen how to use the SharePoint Client Object model to retrieve usage-tracking data from the Ad Tracking list. You have also learned how to use the charting controls that come with the Silverlight Toolkit to generate visual reports based on the usage-tracking data.