



# AppMeasurement

Windows Phone, Silverlight, and .NET

ADOBE® DIGITAL MARKETING SUITE

IMPLEMENTATION GUIDE

**Copyright 1996-2011.** Adobe Systems Incorporated. All rights reserved. Omniture® is a registered trademark of Adobe Systems Incorporated in the United States, Japan, and the European Community.

[Terms of Use](#) | [Privacy Center](#)

Omniture products and services are licensed under the following Netratings patents: 5,675,510, 5,796,952, 6,115,680, 6,108,637, 6,138,155, 6,643,696, and 6,763,386.

A trademark symbol (®, ™, etc.) denotes an Adobe trademark. An asterisk (\*) denotes a third-party trademark. All third-party trademarks are the property of their respective owners.

# Table of Contents

<b>PREFACE</b> .....	<b>4</b>
<b>CHAPTER 2 - APPMEASUREMENT WITH WINDOWS PHONE</b> .....	<b>6</b>
REQUIREMENTS .....	6
INSTALL WINDOWS PHONE LIBRARIES .....	7
OFFLINE MEASUREMENT .....	7
<b>CHAPTER 3 - APPMEASUREMENT FOR SILVERLIGHT</b> .....	<b>9</b>
REQUIREMENTS .....	9
INSTALL APPMEASUREMENT FOR SILVERLIGHT LIBRARIES .....	10
SILVERLIGHT-SPECIFIC IMPLEMENTATION INFORMATION .....	11
<b>CHAPTER 4 - APPMEASUREMENT WITH MICROSOFT SILVERLIGHT ANALYTICS FRAMEWORK</b> ..	<b>12</b>
PREREQUISITES .....	12
DOWNLOADING AND INSTALLING APPMEASUREMENT FOR MSAF .....	13
USING APPMEASUREMENT WITH MSAF .....	14
<b>CHAPTER 5 - APPMEASUREMENT WITH ASP.NET</b> .....	<b>16</b>
REQUIREMENTS .....	16
INSTALL APPMEASUREMENT FOR ASP.NET LIBRARIES .....	16
ASP.NET-SPECIFIC IMPLEMENTATION INFORMATION .....	17
IMPLEMENTATION EXAMPLES .....	19
<b>CHAPTER 6 - APPMEASUREMENT WITH WPF AND .NET APPLICATIONS</b> .....	<b>21</b>
REQUIREMENTS .....	21
INSTALL APPMEASUREMENT FOR WPF OR .NET LIBRARIES .....	22
<b>CHAPTER 7 - APPMEASUREMENT LIBRARY REFERENCE</b> .....	<b>23</b>
APPLICATION CONFIG VARIABLES .....	23
TRACK CONFIG VARIABLES .....	26
METHODS .....	27
IMPLEMENTATION EXAMPLES .....	28
<b>CHAPTER 8 - VIDEO MEASUREMENT IN SILVERLIGHT APPS</b> .....	<b>30</b>

# Preface

The Adobe® *App Measurement for Windows Phone, Silverlight, and .NET* implementation guide describes using the AppMeasurement interface to measure the usage of your Microsoft\* Windows Phone, Silverlight\*, and .NET applications and websites. It lets you capture user activity and forward that data to Adobe data collection servers where it is available for use by Digital Marketing Suite services, including SiteCatalyst® reports.

While AppMeasurement offers common measurement variables and methods for Windows Phone, Silverlight, and .NET applications, there are some platform-specific differences in the AppMeasurement installation and implementation. For details on these differences, see the platform-specific sections of this document.

This guide is intended for application developers, and assumes that you are familiar with both implementing Digital Marketing Suite data collection code, Silverlight development, and/or .NET development.

## Terms and Conditions of Use

This document and the related software described in this document is proprietary to Adobe Systems, Inc. and is supplied only under license or nondisclosure agreement. The document and software can be used or copied only in accordance with the terms of the agreement ([Enterprise Terms of Use](https://sc.omniture.com/p/l10n/1.0/terms.html) - <https://sc.omniture.com/p/l10n/1.0/terms.html>).

The information in this document is subject to change without notice and does not represent a commitment on the part of Adobe.

## Account Support

ClientCare is available to:

- Answer specific product questions.
- Help you utilize SiteCatalyst reports to the greatest effect.
- Resolve any technical difficulties you might have.
- Help you configure SiteCatalyst variables.

## Service and Billing Information

Depending on the service level you have purchased, some of the options described in this guide might not be available to you. Additionally, each account has unique billing needs. Please refer to your contract for pricing, due dates, terms and conditions. If you would like to add to or otherwise change your service level, or if you have questions regarding your current service, please contact your Account Manager.

Adobe welcomes any suggestions or feedback you might have regarding AppMeasurement for Silverlight or the contents of this guide. Send comments to your Account Manager.

# Contact Information

**[CORPORATE ADDRESS]** Adobe Systems, Inc.  
550 East Timpanogos Circle  
Orem, UT 84097

**[PHONE]** 1.801.722.7000

**[FAX]** 1.801.722.7001

**[TOLL FREE]** 1.877.722.7088 (support, billing and sales)

**[SUPPORT E-MAIL]** [clientcare@omniture.com](mailto:clientcare@omniture.com)

**[SALES E-MAIL]** [sales@omniture.com](mailto:sales@omniture.com)

**[INFORMATION E-MAIL]** [info@omniture.com](mailto:info@omniture.com)

**[CORPORATE URL]** <http://www.omniture.com>

**[LOG-IN URL]** <http://my.omniture.com>

# AppMeasurement with Windows Phone

Adobe® AppMeasurement for Windows Phone, Silverlight, and .NET provides a mechanism to capture usage of native Windows Phone applications in the Adobe Digital Marketing Suite. AppMeasurement lets you capture application events so you can monitor and evaluate the use of your Windows Phone applications.

AppMeasurement leverages the standard Windows Phone development tools to let you integrate Windows Phone application tracking into both new and existing Windows Phone applications.

This section includes the following topics:

- [Requirements](#)
- [Install Windows Phone LibrariesOffline Measurement](#)
- [Offline Measurement](#)

For information about the variables and methods available for use by Windows Phone applications, see ["AppMeasurement Library Reference" on page 23](#).

## 2.1 Requirements

- AppMeasurement for Windows Phone consists of the following software components. You can download the AppMeasurement software from SiteCatalyst's Code Manager. For information about using Code Manager, see "Code Manager" in the *Admin Console User Guide*, which is available in the Digital Marketing Suite help system.
  - AppMeasurement\_Silverlight\_WP.dll:** A Windows library for use in your shipping Windows Phone application.
  - AppMeasurement\_Silverlight\_WP\_Debug.dll:** A testing version of the Windows Phone library that provides debug information to the Visual Studio output console. Use this library while developing your Windows Phone application.
- After downloading the Windows Phone libraries, copy the reference implementation (in the `Windows Phone Example` tab) onto your system clipboard, or into a text file where you can access it later.
- The current Windows Phone development tools. Currently, you can view the Windows Phone tool documentation, and link to the tool download site, at [http://msdn.microsoft.com/en-us/library/ff402535\(VS.92\).aspx](http://msdn.microsoft.com/en-us/library/ff402535(VS.92).aspx).

## 2.2 Install Windows Phone Libraries

Before using AppMeasurement, you must install the AppMeasurement libraries into your Windows Phone project and configure it as needed to support your specific data collection needs.

**NOTE:** The following task uses the Microsoft Web Developer 2008 Express Edition IDE. If you are using a different IDE, the specifics might vary from what is described here.

### To install and configure the Windows Phone component

1. Launch the Visual Web Developer IDE, then open the Windows Phone project.  
To create a new Windows Phone project, do the following:
  - a. Click **File > New Project**.
  - b. Select **Windows Phone Application**, then click **OK**.
2. Add a reference to AppMeasurement in the Windows Phone project.
  - a. In the Solution Explorer, right-click References, then select **Add Reference**.
  - b. Select the **Browse** tab, browse to and select the appropriate AppMeasurement for Windows Phone library, then click **OK** (for more information, see ["Requirements" on page 6](#)).
3. Add the following line to the Windows Phone project to enable AppMeasurement for the project.

```
using com.omniture;
```
4. Create a variable for the AppMeasurement instance in your project. For example:

```
private AppMeasurement s;
```
5. Create an instance of the AppMeasurement object and add the necessary Digital Marketing Suite variables to the Windows Phone application. For example:

```
public MainPage()  
{  
    InitializeComponent();  
    s = new AppMeasurement();  
    s.account = "myCorp";  
    s.dc = "d1";  
    s.debugTracking = true;  
    s.pageName = "myPage";  
    s.track();  
}
```
6. Build and save the Windows Phone project.
7. When you execute the project in Debugging mode, you can use a Web browser debugger such as Firebug to confirm that the application sends data to Adobe data collection servers as expected.

## 2.3 Offline Measurement

AppMeasurement lets you measure application usage even when the Windows Phone is offline. When enabled, offline measurement does the following:

- The application generates a "hit" (a `track()` or `tracklink()` method call), but the data transmission fails.
- AppMeasurement generates a timestamp for the current hit.
- AppMeasurement buffers the hit data, and backs up buffered hit data to persistent storage to prevent data loss.

- At each subsequent hit, or at 5-second intervals (whichever is shorter), AppMeasurement attempts to send the buffered hit data; maintaining the original hit order. If the data transmission fails, it continues to buffer the hit data.

Offline measurement uses the following methods and variables:

**Table 2.1: Offline Measurement Variables**

NAME	TYPE / DEFAULT VALUE	DESCRIPTION
<code>trackOffline</code>	Boolean / <code>false</code>	Enables/Disables offline AppMeasurement.
<code>offlineLimit</code>	Integer / 10	<p>Sets the maximum number of hits to buffer.</p> <p>When the number of buffered hits reaches <code>offlineLimit</code>, AppMeasurement drops the oldest buffered hit, then buffers the newest hit.</p> <p>Tune this value based on the number of hits, and the amount of data (the number of variables) per hit, sent by your app. For example, if your app generates a lot of offline server calls (lots of link tracking in addition to standard page/screen views), try increasing <code>offlineLimit</code>.</p>
<code>offlineThrottleDelay</code>	Integer / 0	<p>Specifies a cadence (or delay), in milliseconds, for sending buffered hit data when AppMeasurement detects an active network connection. Doing so mitigates the performance impact of sending multiple hits on the application.</p> <p>For example, if <code>offlineThrottleDelay=1000</code>, and it takes 300ms to send the hit data, AppMeasurement waits 700ms before sending the next buffered hit.</p>
<code>timestamp</code>	Integer / 0	<p>Overrides the default timestamp mechanism in AppMeasurement so you can integrate the application's offline measurement with an existing queue system.</p> <p>Adobe recommends letting AppMeasurement manage timestamps, if possible.</p>

**Table 2.2: Offline Measurement Methods**

NAME	DESCRIPTION
<code>forceOffline()</code>	<p>Forces AppMeasurement to behave as if it is offline.</p> <p>This lets you delay sending hit data during a hardware-intensive operation, such as video playback. Rather, you can collect the data offline, then send it once the operation completes.</p>
<code>forceOnline()</code>	Returns AppMeasurement to normal operating mode after using the <code>forceOffline()</code> method.

# AppMeasurement for Silverlight

Adobe® AppMeasurement for Windows Phone, Silverlight, and .NET provides a mechanism to capture usage of native Microsoft Silverlight applications in the Adobe Digital Marketing Suite. AppMeasurement for Silverlight lets you capture application events so you can monitor and evaluate the use of your Silverlight applications.

AppMeasurement for Silverlight leverages the standard Silverlight development tools to let you integrate Silverlight application tracking into both new and existing Silverlight applications.

This section includes the following topics:

- [Requirements](#)
- [Install AppMeasurement for Silverlight Libraries](#)
- [Silverlight-Specific Implementation Information](#)

For information about the variables and methods available for use by Silverlight applications, see ["AppMeasurement Library Reference" on page 23](#).

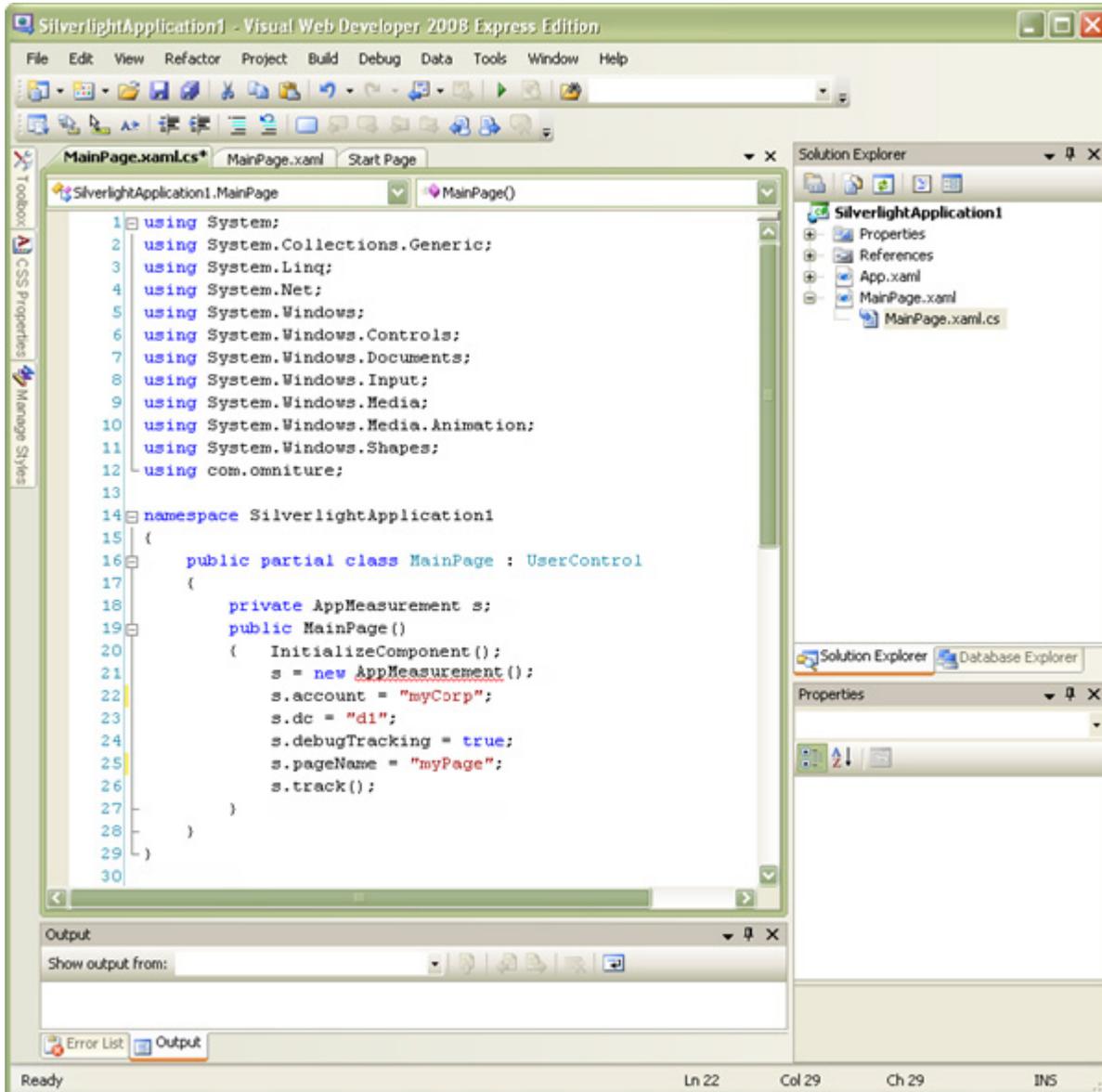
## 3.1 Requirements

- AppMeasurement for Silverlight consists of the following software components. You can download the AppMeasurement software from SiteCatalyst's Code Manager. For information about using Code Manager, see "Code Manager" in the *Admin Console User Guide*, which is available in the Digital Marketing Suite help system.
  - **AppMeasurement\_Silverlight.dll:** A Windows library for use in your shipping Silverlight application.
  - **AppMeasurement\_Silverlight\_Debug.dll:** A testing version of the Windows library that provides debug information to the Visual Studio output console. Use this library while developing your Silverlight application.
- After downloading the Silverlight libraries, copy the reference implementation (in the `Silverlight C# Example` tab) onto your system clipboard, or into a text file where you can access it later.
- Adobe recommends the following for developing Silverlight applications.
  - Microsoft Visual Studio
  - A development system that meets all hardware requirements for Visual Studio

## 3.2 Install AppMeasurement for Silverlight Libraries

Before using AppMeasurement for Silverlight, you must install it into your Silverlight project and configure it as needed to support your specific data collection needs.

**NOTE:** The following task uses the Microsoft Web Developer 2008 Express Edition IDE. If you are using a different IDE, the specifics might vary from what is described here.



### To install and configure the Silverlight component

1. Launch the Visual Web Developer IDE, then open the Silverlight project.  
To create a new Silverlight project, do the following:
  - a. Click **File > New Project**.
  - b. Select **Silverlight Application**, then click **OK**.
  - c. Deselect **Host the Silverlight application in a new Web site**, then click **OK**.

2. Add a reference to AppMeasurement in the Silverlight project.
  - a. In the Solution Explorer, right-click References, then select **Add Reference**.
  - b. Select the **Browse** tab, browse to and select the appropriate AppMeasurement for Silverlight library, then click **OK** (for more information, see ["Requirements" on page 9](#)).
3. Add the following line to the Silverlight project to enable AppMeasurement for the project.
 

```
using com.omniture;
```
4. Create a variable for the AppMeasurement instance in your project. For example:
 

```
private AppMeasurement s;
```
5. Create an instance of the AppMeasurement object and add the necessary Digital Marketing Suite variables to the Silverlight application. For example:
 

```
public MainPage()
{
    InitializeComponent();
    s = new AppMeasurement();
    s.account = "myCorp";
    s.dc = "d1";
    s.debugTracking = true;
    s.pageName = "myPage";
    s.track();
}
```
6. Build and save the Silverlight project.
7. When you execute the project in Debugging mode, you can use a Web browser debugger such as Firebug to confirm that the application sends data to Adobe data collection servers as expected.

### 3.3 Silverlight-Specific Implementation Information

AppMeasurement provides the following Silverlight-specific resources for use with Silverlight applications. For information about other AppMeasurement resources available for use with Silverlight applications, see ["AppMeasurement Library Reference" on page 23](#).

#### setInterface method

Specifies a "root layout" that you can use to find the root of the Silverlight application.

---

**NOTE:** Using `setInterface` is required for certain advanced features such as video auto-tracking.

---

SYNTAX	PARAMETERS	RESPONSE
<code>s.setInterface(interface);</code>	<b>interface:</b> (Required) The "root layout" object from the Silverlight application.	None

# AppMeasurement with Microsoft Silverlight Analytics Framework

The Microsoft Silverlight Analytics Framework provides an Extensible Web Analytics framework for Microsoft Silverlight and WPF Applications. For detailed information about the Microsoft Silverlight Analytics Framework, visit <http://msaf.codeplex.com/>.

The Adobe Digital Marketing Suite provides code that lets you use AppMeasurement with the Microsoft Silverlight Analytics Framework. This section contains the following topics:

- [Prerequisites](#)
- [Downloading and Installing AppMeasurement for MSAF](#)
- [Using AppMeasurement with MSAF](#)

## 4.1 Prerequisites

Using AppMeasurement for Microsoft Silverlight Analytics Framework (MSAF) requires the following:

- Microsoft Visual Studio 2010 or Microsoft Visual Web Developer 2010 Express (visit [www.microsoft.com/express/Web](http://www.microsoft.com/express/Web))
- Microsoft Expression Blend 4 (visit [www.microsoft.com/expression/products/Blend\\_OverView.aspx](http://www.microsoft.com/expression/products/Blend_OverView.aspx))
- Microsoft Silverlight Analytics Framework for Blend 4 (visit [msaf.codeplex.com](http://msaf.codeplex.com))
- AppMeasurement for MSAF (see ["Downloading and Installing AppMeasurement for MSAF" on page 13](#))

## 4.2 Downloading and Installing AppMeasurement for MSAF

You can download AppMeasurement for Microsoft Silverlight Application Framework from Code Manager in the Digital Marketing Suite Administration Console.

### To download and install AppMeasurement for MSAF

1. Log in to the Digital Marketing Suite.
2. In the Suite, click **Admin > Admin Console**.
3. In the left-side navigation, select **Admin Console > Code Manager**.
4. On the Code Manager page, select **Windows Phone, Silverlight, and .NET** in the **Select the type of code to generate** dropdown list.
5. In the Options pane, provide the requested information to generate the Silverlight code, then click **Generate Code**.

**Report Suite** Select your report suite.

**Character Encoding** Select the desired character-encoding format for the generated code. This should match the character encoding on your Web site.

Selecting the wrong character encoding might cause permanent data loss because text is improperly converted upon receipt.

**Currency** Specify your Web site's base currency for tracking conversion statistics.

The Base Currency is the currency used to store data in SiteCatalyst. All currency amounts are converted the base currency, if necessary, based on the current conversion rate when SiteCatalyst receives the data.

**Number of periods in domain name** Specify the number of periods in your domain name.

**Use https:// instead of http://** Select to generate code using secure HTTP (SSL) instead of standard HTTP.

The Code Manager generates code based on your selections, and allows you to download it for use on your Web site or in your applications.

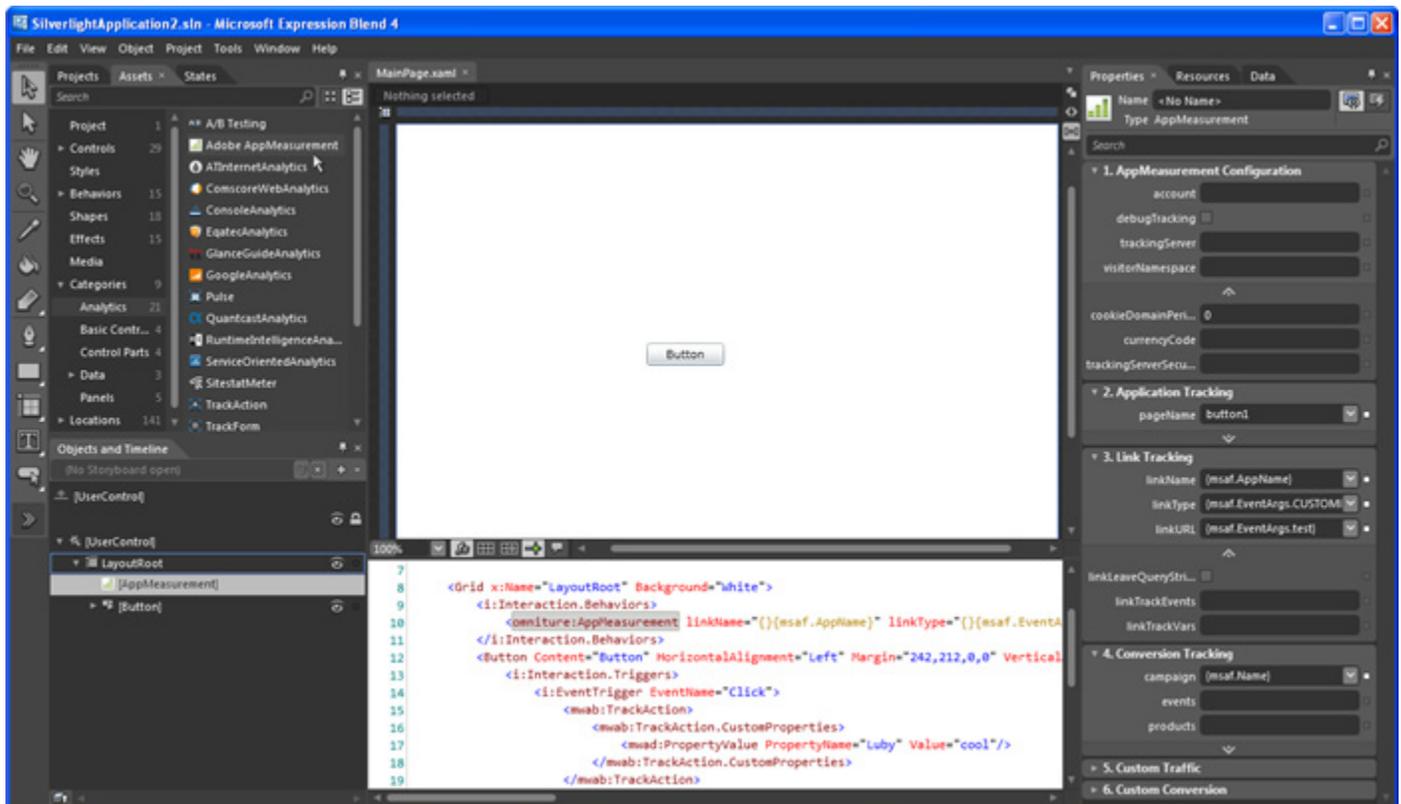
6. On the Generated Code page, select the **Component Files** tab, then select **AppMeasurement\_MSAF.msi**. This installer includes the code necessary to integrate AppMeasurement with the Microsoft Silverlight Analytics Framework.
7. Launch `AppMeasurement_MSAF.msi` to install AppMeasurement for MSAF into your Silverlight development environment.

For information about using AppMeasurement for MSAF for data collection in your Silverlight application, see ["Using AppMeasurement with MSAF" on page 14](#).

## 4.3 Using AppMeasurement with MSAF

Once you have installed AppMeasurement for MSAF, you can configure data collection your Silverlight applications. AppMeasurement for MSAF can capture data about page views and link clicks in your Silverlight application.

**NOTE:** The following task assumes you are familiar with Microsoft Expression Blend 4 and Microsoft Silverlight Analytics Framework (see ["Prerequisites" on page 12](#)).



### To use AppMeasurement for MSAF

1. Launch Microsoft Expression Blend 4.
2. Open a Silverlight 4 project where you want to add data collection.  
AppMeasurement for MSAF supports only Silverlight 4 applications.
3. In the Assets tab, select **Categories > Analytics**, then drag **Adobe AppMeasurement** onto your project's LayoutRoot (in the Objects and Timeline tab).

This makes AppMeasurement data collection available to all components of your Silverlight project. For data collection to work, you must add the Adobe AppMeasurement asset to LayoutRoot.

4. Drag the appropriate Tracker object onto each component in your application where you want to collect data.  
AppMeasurement for MSAF supports the following Tracker objects: Pulse, TrackAction, and TrackTouch. For more information about each of these objects, see the Microsoft Silverlight Analytics Framework documentation.
5. Configure each Tracker object as needed.  
For example, you might need to create custom property key/value pairs to use in the data collection process. For more information about the Tracker objects, see the Microsoft Silverlight Analytics Framework documentation.
6. Assign values to the AppMeasurement properties that you want to collect.

This determines what data AppMeasurement collects through the Microsoft Silverlight Analytics Framework. Each AppMeasurement for MSAF property corresponds to one of the following:

- An Application Config variable (see ["Application Config Variables" on page 23](#))
- A Track Config variable (see ["setInterface method" on page 11](#))
- Parameters needed for a `trackLink` method call (see ["trackLink" on page 27](#)).

AppMeasurement provides the following UI controls and options for assigning property values:

<b>Expand or Collapse Categories</b>	Click the Arrow icon  next to a category name to collapse or expand its property list.
<b>Display Advanced Properties</b>	By default, AppMeasurement for MSAF displays only Basic (most commonly used) properties. Click the Expand icon  to display all properties in that category.
<b>Static Value</b>	Select a property field to type in a static value for that property. Each time AppMeasurement collects data, it includes this constant property value.
<b>Dynamic Value</b>	Select the dropdown button in a property field to display a list of MSAF properties to assign to the AppMeasurement property. Each time AppMeasurement collects data, it includes the current value of the AppMeasurement property.
<b>Custom Property</b>	Select the dropdown button in a property field, then select <code>{msaf.EventArgs.CUSTOMPROPERTY}</code> to assign a custom property, defined in the Tracker object, to the AppMeasurement property. For more information about custom properties, see the Microsoft Silverlight Analytics Framework documentation.  Once selected, replace "CUSTOMPROPERTY" with the custom property name, as defined in your Tracker object. Each time AppMeasurement collects data, it includes the value of the Tracker object's custom property as the value of the AppMeasurement property.

7. Save and rebuild the project to incorporate the AppMeasurement for MSAF data collection into the application.

When you load the application in a browser, you can use an HTML debugger such as Firebug\* to view AppMeasurement data collection as it occurs.

# AppMeasurement with ASP.NET

Adobe® AppMeasurement for .NET and Silverlight lets you capture user activity on ASP.NET websites and pages in the Adobe Digital Marketing Suite. AppMeasurement leverages the standard ASP.NET development tools to let you integrate ASP.NET measurement into both new and existing ASP.NET websites and pages.

This section includes the following topics:

- [Requirements](#)
- [Install AppMeasurement for ASP.NET Libraries](#)
- [Implementation Examples](#)

## 5.1 Requirements

- AppMeasurement for ASP.NET consists of the following software components. You can download the AppMeasurement software from SiteCatalyst's Code Manager. For information about using Code Manager, see "Code Manager" in the *Admin Console User Guide*, which is available in the Digital Marketing Suite help system.

**AppMeasurement\_DotNET.dll:** A Windows library for use in your production ASP.NET site.

**AppMeasurement\_DotNET\_Debug.dll:** A testing version of the Windows library that provides debug information to the Visual Studio output console. Use this library while developing your ASP.NET site.

- Adobe recommends the following for developing ASP.NET applications.
  - Microsoft Visual Studio
  - A development system that meets all hardware requirements for Visual Studio

## 5.2 Install AppMeasurement for ASP.NET Libraries

Once you have downloaded the AppMeasurement for ASP.NET library, save it into your IIS `/bin` directory so it can be referenced in your ASP.NET website or page.

## 5.3 ASP.NET-Specific Implementation Information

AppMeasurement provides the following ASP.NET-specific resources for use with ASP.NET websites and pages. For information about other AppMeasurement resources available for use with ASP.NET websites and pages, see [“AppMeasurement Library Reference” on page 23](#).

**Table 5.1: AppMeasurement Configuration Variables**

RESOURCE	DESCRIPTION
<pre>public AppMeasurement (HttpRequest request, HttpResponse response)</pre>	<p><b>Syntax:</b> <code>s = new AppMeasurement (Request, Response);</code></p> <p>The AppMeasurement for ASP.NET constructor. Use the constructor to pass in the request and response objects for your ASP.NET page. Typically these are <code>Request</code> and <code>Response</code>.</p>
<pre>sendFromServer</pre>	<p><b>Syntax:</b> <code>s.sendFromServer = false;</code>  <code>Response.Write (s.track ());</code></p> <p>When <code>false</code>, the visitor’s Web browser handles the data collection (with <code>track()</code> or <code>trackLink()</code>) by making an image request to Digital Marketing Suite servers. Your application must then write out the returned <code>&lt;img&gt;</code> tag.</p> <p>When <code>true</code>, AppMeasurement sends the data collection request itself.</p> <p><b>Note:</b> The <code>sendFromServer</code> flag defaults to <code>false</code> when using the ASP.NET constructor.</p>
<pre>addVisitorIDToLinksAndForms ()</pre>	<p><b>Syntax:</b> <code>s.addVisitorIDToLinksAndForms (myHTMLContent);</code>  <code>Response.Write (s.addVisitorIDToLinksAndForms (myHTMLContent));</code></p> <p>Adds <code>s_visitorID=[s.visitorID]</code> to all links and forms in a block of HTML that you pass in, then returns the updated HTML to you. The response takes the form:</p> <p>This method lets you use internal links and forms on the ASP.NET website to pass a visitor ID from page to page instead of using a cookie.</p> <p><b>Note:</b> After using this method, you can call <code>manageVisitorID</code> to identify the visitor based on the <code>s_visitorID</code> value added to the HTML links and forms.</p>

RESOURCE	DESCRIPTION
manageVisitorID	<p><b>Syntax:</b> <code>s.manageVisitorID();</code>  or  <code>s.manageVisitorID(false, true);</code>  <code>Response.Write(s.addVisitorIDToLinksAndForms(myPageContent));</code></p> <p>Attempts to identify the website visitor without using a 3rd party cookie, using the following techniques:</p> <ol style="list-style-type: none"> <li>Using a first party <code>s_visitorID</code> cookie containing the value of <code>s.visitorID</code> or a generated unique ID. When the cookie is set this method causes the page to redirect back to itself to detect if the cookie could be set. If the cookie cannot be set, <code>manageVisitorID</code> continues with the next technique.</li> <li>Using a mobile device unique identifier from several known HTTP headers.</li> <li>Pulling <code>s_visitorID</code> from the page query-string or POSTed variables. This option requires you to use the <code>addVisitorIDToLinksAndForms</code> method.</li> </ol> <p>If all of these methods fail no visitor ID is generated and visitor identification is left to the server-side data collection system (Mod-Stats).</p> <p>This method supports the following optional arguments:</p> <p><code>useFirstPartyCookie</code>: Default = true. Enables setting first-party cookies. If you set this argument to <code>false</code>, technique (a) above is not attempted.</p> <p><code>forceGeneration</code>: Default = false. Used in conjunction with <code>addVisitorIDToLinksAndForms()</code> to force the page to generate a unique ID if <code>s.visitorID</code> is not already set and cookies are not being used.</p>

## 5.4 Implementation Examples

The following examples demonstrate some common use cases for AppMeasurement for ASP.NET:

- [\(Instance instantiation\) Declare instance](#)
- [\(Instance instantiation\) Instantiate instance and setup account configuration variables](#)
- [\(track\) Send a Page View Only](#)
- [\(track\) Send pageView data using variable overrides](#)
- [\(trackLink\) Send a Custom Link](#)

### (Instance instantiation) Declare instance

Declare instance as either a member of your page class.

```
AppMeasurement s;
```

### (Instance instantiation) Instantiate instance and setup account configuration variables

The most likely place to do this is in the constructor of your page class, but can be done elsewhere.

```
<%@ Page Language="C#" Debug="true" %>
<%@ Import Namespace="com.omniture" %>
<%
//Instantiate instance
s = new AppMeasurement();
//Setup application config variables
s.account = "myrsid";
s.visitorNamespace = "myco";
s.trackingServer = "myco.dl.sc.omtrdc.net";
```

### (track) Send a Page View Only

The most likely place to do this is in the constructor of your page class, but can be done anywhere.

```
<%@ Page Language="C#" Debug="true" %>
<%@ Import Namespace="com.omniture" %>
<%
//Instantiate instance
s = new AppMeasurement();
//Setup application config variables
s.account = "myrsid";
s.visitorNamespace = "myco";
s.trackingServer = "myco.dl.sc.omtrdc.net";
// Set variables and track Page View
s.pageName = "Some Page Name";
s.track();
```

### (track) Send pageView data using variable overrides

Specifies a temporary variable value used only for track call.

```
<%@ Page Language="C#" Debug="true" %>
<%@ Import Namespace="com.omniture" %>
<%
//Instantiate instance
s = new AppMeasurement();
```

```
//Setup application config variables
s.account = "myrsid";
s.visitorNamespace = "myco";
s.trackingServer = "myco.dl.sc.omtrdc.net";
// Set variables and track Page View
Dictionary<string,object> trackData;
trackData["pageName"] = "Some page name";
trackData["prop1"] = "Some Prop Value";
s.track(v);
```

### **(trackLink) Send a Custom Link**

```
s.trackLink("http://www.myco.com", "o", "Home Link");
```

# AppMeasurement with WPF and .NET Applications

AppMeasurement provides a mechanism to capture usage of native Microsoft WPF or .NET applications in the Adobe Digital Marketing Suite. AppMeasurement for WPF or .NET lets you capture application events so you can monitor and evaluate the use of your WPF or .NET applications.

AppMeasurement for WPF or .NET leverages the standard WPF or .NET development tools to let you integrate WPF or .NET application tracking into both new and existing WPF or .NET applications.

This section includes the following topics:

- [Requirements](#)
- [Install AppMeasurement for WPF or .NET Libraries](#)

For information about the variables and methods available for use by WPF and .NET applications, see ["AppMeasurement Library Reference" on page 23](#).

## 6.1 Requirements

- AppMeasurement for WPF or .NET consists of the following software components. You can download the AppMeasurement software from SiteCatalyst's Code Manager. For information about using Code Manager, see "Code Manager" in the *Admin Console User Guide*, which is available in the Digital Marketing Suite help system.

**AppMeasurement\_DotNET.dll:** A Windows library for use in your shipping WPF or .NET application.

**AppMeasurement\_DotNET\_Debug.dll:** A testing version of the Windows library that provides debug information to the Visual Studio output console. Use this library while developing your WPF or .NET application.

- Adobe recommends the following for developing WPF or .NET applications.
  - Microsoft Visual Studio
  - A development system that meets all hardware requirements for Visual Studio

## 6.2 Install AppMeasurement for WPF or .NET Libraries

Before using AppMeasurement for WPF or .NET, you must install it into your project and configure it as needed to support your specific data collection needs.

**NOTE:** The following task uses the Microsoft Visual Studio 2010 IDE. If you are using a different IDE, the specifics might vary from what is described here.

### To install and configure AppMeasurement component

1. Launch the Visual Studio IDE, then open the WPF or .NET project.  
To create a new project, do the following:
  - a. Click **File > New Project**.
  - b. Select **WPF Application**, or another appropriate .NET application type, then click **OK**.
2. Add a reference to AppMeasurement in the project.
  - a. In the Solution Explorer, right-click References, then select **Add Reference**.
  - b. Select the **Browse** tab, browse to and select the appropriate AppMeasurement library, then click **OK** (for more information, see ["Requirements" on page 21](#)).
3. Add the following line to the project to enable AppMeasurement:

```
using com.omniture;
```
4. Create a variable for the AppMeasurement instance in your project. For example:

```
private AppMeasurement s;
```
5. Create an instance of the AppMeasurement object and add the necessary Digital Marketing Suite variables to the WPF or .NET application. For example:

```
s = new AppMeasurement();  
s.account = "myCorp";  
s.dc = "d1";  
s.debugTracking = true;  
s.pageName = "myPage";  
s.track();
```
6. Build and save the project.
7. When you execute the project in Debugging mode, you can use the Visual Studio Output Console to confirm that the application sends data to Adobe data collection servers as expected.

# AppMeasurement Library Reference

Adobe® AppMeasurement for Windows Phone, Silverlight, and .NET provides a mechanism to capture usage of native Microsoft Windows Phone, Silverlight, and .NET applications in the Adobe® Digital Marketing Suite. AppMeasurement lets you capture events so you can monitor and evaluate the use of your applications.

Regardless of the specific .NET platform you are using, AppMeasurement provides the following common set of variables and methods that you can use to configure application and website measurement:

- [Application Config Variables](#)
- [Track Config Variables](#)
- [Methods](#)
- [Implementation Examples](#)

## 7.1 Application Config Variables

AppMeasurement sets the following variables when the application or website launches, and typically do not need to change. However, you can change them at any time, if necessary.

**Table 7.1: AppMeasurement Configuration Variables**

VARIABLE	REQUIRED	DESCRIPTION
account	Yes	<p><b>Syntax:</b> <code>s.account="mycompanycom";</code></p> <p>The report suite or report suites (multi-suite tagging) that you want to track.</p>
dc	Yes	<p><b>Syntax:</b> <code>s.dc="d1";</code> or <code>"d2";</code></p> <p>Identifies the Adobe data center. Do not change this value unless so instructed by your Adobe Account Manager.</p> <p>d1 = San Jose data center d2 = Dallas data center</p> <p>For reference, the <code>dc</code> variable is also specified in your Website's core JavaScript file (<code>s_code.js</code>).</p>
visitorNamespace	Yes	<p><b>Syntax:</b> <code>s.visitorNamespace="mycompany";</code></p> <p>Must match the <code>visitorNamespace</code> used in your JavaScript tracking file (<code>s_code.js</code>). If your JavaScript file does not have a <code>visitorNamespace</code> variable defined, do not define it in the AppMeasurement DLL. Using the wrong value for <code>visitorNamespace</code> causes incorrect reporting.</p>

VARIABLE	REQUIRED	DESCRIPTION
pageName or pageURL	Yes	<p><b>Syntax:</b> <code>s.pageName="Some Page Name";</code>  <code>s.pageURL="http://www.myurl.com";</code></p> <p>Specify <code>pageName</code> so that ClickMap can properly overlay data.</p> <p>If users do not access the application through a Web browser, specify a <code>pageURL</code> as well. If users access the application through a Web browser, leave <code>pageURL</code> blank and AppMeasurement automatically sets <code>pageURL</code> to the Web browser's URL.</p>
charSet	No	<p><b>Syntax:</b> <code>s.charSet="UTF-8";</code></p> <p>Identifies the character encoding used in the application or website. For more information about character encoding, see the <i>SiteCatalyst Implementation Guide</i>, which is available in the Digital Marketing Suite help system.</p>
cookieDomainPeriods	No	<p><b>Syntax:</b> <code>s.cookieDomainPeriods=2;</code></p> <p>Specifies the number of elements in your domain. SiteCatalyst uses this variable when setting the <code>visitorID</code> cookie. For example, <code>s.trackingServer="www.domain.co.uk"</code> has 3 domain periods, and <code>s.trackingServer="www.mycorp.com"</code> has 2 domain periods.</p> <p>For more information, see the <i>SiteCatalyst Implementation Guide</i>, which is available in the Digital Marketing Suite help system.</p>
cookieLifetime	No	<p><b>Syntax:</b> <code>s.cookieLifetime="600";</code></p> <p>Specifies the amount of time, in seconds, to store the <code>visitorID</code> cookie.</p> <p>Set <code>cookieLifetime="session"</code> to have the cookie expire with the Web browser session.</p>
currencyCode	No	<p><b>Syntax:</b> <code>s.currencyCode="USD";</code></p> <p>Specifies the currency code that Adobe uses to denominate purchases or currency events collected in the application or website.</p>
linkLeaveQueryString	No	<p><b>Syntax:</b> <code>s.linkLeaveQueryString = true;</code></p> <p>Determines whether the query string should be included in the Exit Links and File Download reports. By default, SiteCatalyst excludes query strings from all reports. However, for some Exit Links and File Download reports, the important portion of the URL might be in the query string.</p> <p>For example, the download file name might be defined in the query string, making the query string necessary for an accurate File Downloads report.</p> <p><b>Note:</b> If query strings contain session IDs or other personally identifiable information, contact your Account Manager to have SiteCatalyst strip out those query string elements.</p>

VARIABLE	REQUIRED	DESCRIPTION
linkTrackEvents	No	<p><b>Syntax:</b> <code>s.linkTrackEvents="purchase,event1";</code></p> <p>A comma-separated list of events to send with a custom, exit, or download links.</p> <p>Populate <code>linkTrackEvents</code> in the <code>onClick</code> event. If an event is not specified in <code>linkTrackEvents</code>, <code>AppMeasurement</code> does not send it to SiteCatalyst, even if it is populated in the link's <code>onClick</code> event.</p>
linkTrackVars	No	<p><b>Syntax:</b> <code>s.linkTrackVars= "events,prop1,eVar49";</code></p> <p>A comma-separated list of variables to send with custom, exit, and download links.</p> <p>Set <code>linkTrackVars=""</code>; to send all variables that have values.</p> <p>To avoid inflation of instances or page views associated with other variables, populate <code>linkTrackVars</code> in the link's <code>onClick</code> event that is used for link tracking.</p>
trackDownloadLinks	No	<p><b>Syntax:</b> <code>s.trackDownloadLinks = true;</code></p> <p>Lets you track links to downloadable files in your application or website. When enabled, <code>AppMeasurement</code> uses <code>linkDownloadFileTypes</code> to determine which links are downloadable files.</p> <p>Set <code>trackDownloadLinks</code> to <code>false</code> only if there are no downloadable file links your Web site, or if you don't want to track file downloads.</p>
trackExternalLinks	No	<p><b>Syntax:</b> <code>s.trackExternalLinks = true;</code></p> <p>Lets you track clicks on links that lead to Web pages external to your Web site. When enabled, <code>AppMeasurement</code> uses <code>linkInternalFilters</code> and <code>linkExternalFilters</code> to determine if a clicked link is an exit link.</p> <p>Set <code>trackExternalLinks=false</code> only if there are no exit links on your Web site, or if you don't want to track clicks on exit links.</p>
trackingServer trackingServerSecure	No	<p><b>Syntax:</b> <code>s.trackingServer = "metrics.mycompany.com";</code> <code>s.trackingServerSecure = "smetrics.mycompany.com";</code></p> <p>Identifies the collection domain when you are using a first-party cookie implementation. These values should match the collection domain in your <code>s_code.js</code> file, if one exists.</p>
visitorMigrationKey	No	<p><b>Syntax:</b> <code>s.visitorMigrationKey=[omniture_provided_value];</code></p> <p>Specifies the migration key used to migrate visitor cookies from a third-party to a first-party cookie implementation.</p> <p>To get this value, contact your Account Manager.</p>

VARIABLE	REQUIRED	DESCRIPTION
debugTracking	No	<b>Syntax:</b> <code>s.debugTracking=true;</code> Lets you view debug information in your platform's debugger. For example, the Silverlight editor trace window or a Firebug console (Firefox).

## 7.2 Track Config Variables

The following variables are typically set before calling one of the track methods. For more information about these tracking variables, see the *SiteCatalyst Implementation Guide*.

pageName	server	zip
pageURL	pageType	events
referrer	dynamicVariablePrefix	products
purchaseID	variableProvider	prop (1-75)
transactionID	campaign	eVar (1-75)
channel	state	hier (1-5)

---

**WARNING:** When capturing variables such as `campaign` that should be set only one time per visit, make sure you reset the variable to NULL (or empty string "") immediately after calling `track` or `trackLink`. Otherwise, these variables continue to increment with each subsequent transaction, which corrupts your tracking data.

---

## 7.3 Methods

The AppMeasurement DLL supports the following methods for tracking Windows Phone, Silverlight, and .NET applications and websites:

- [track](#)
- [trackLink](#)

### track

Sends a standard page view to Adobe data collection servers, including any tracking variables that have values. The track method takes no parameters and returns no response.

SYNTAX	PARAMETERS	RESPONSE
<code>s.track();</code>	None	None

### trackLink

Sends custom, download, or exit link data to Adobe data collection servers. Track all micro-level activity that should not capture a page view with `trackLink`.

**NOTE:** When using `trackLink`, AppMeasurement sends the `pageName` variable, but SiteCatalyst does not record it as a page view.

SYNTAX	PARAMETERS	RESPONSE
<code>s.trackLink(url, type, name);</code>	<p><b>url:</b> (Required) The URL that identifies the clicked link. If you do not want to specify a URL, use an empty string ("" ) for the <code>url</code> parameter. If no URL is specified, <code>trackLink</code> uses the <code>name</code> parameter to identify the clicked link.</p> <p><b>type:</b> (Required) A letter code that specifies the link report that should display the URL or name. Supported values include:</p> <ul style="list-style-type: none"><li>o: Custom Links report</li><li>d: File Downloads report</li><li>e: Exit Links report</li></ul> <p><b>name:</b> (Required) The name that appears in the link report. If you do not want to specify a name, use an empty string ("" ) for the <code>name</code> parameter. If no name is specified, <code>trackLink</code> uses the <code>url</code> parameter to identify the clicked link.</p> <p>You must provide a value for either <code>name</code> or <code>url</code>.</p>	None

## 7.4 Implementation Examples

The following examples demonstrate some common use cases for AppMeasurement for WPF or .NET:

- [\(Instance instantiation\) Declare instance](#)
- [\(Instance instantiation\) Instantiate instance and setup account configuration variables](#)
- [\(track\) Send a Page View Only](#)
- [\(track\) Send pageView data using variable overrides](#)
- [\(trackLink\) Send a Custom Link](#)

### (Instance instantiation) Declare instance

Declare instance as either a member of your page class.

```
AppMeasurement s;
```

### (Instance instantiation) Instantiate instance and setup account configuration variables

The most likely place to do this is in the constructor of your page class, but can be done elsewhere.

```
//Instantiate instance
s = new AppMeasurement();
//Setup application config variables
s.account = "myrsid";
s.visitorNamespace = "myco";
s.trackingServer = "myco.dl.sc.omtrdc.net";
```

### (track) Send a Page View Only

The most likely place to do this is in the constructor of your page class, but can be done anywhere.

```
//Instantiate instance
s = new AppMeasurement();
//Setup application config variables
s.account = "myrsid";
s.visitorNamespace = "myco";
s.trackingServer = "myco.dl.sc.omtrdc.net";
// Set variables and track Page View
s.pageName = "Some Page Name";
s.track();
```

### (track) Send pageView data using variable overrides

Specifies a temporary variable value used only for track call.

```
//Instantiate instance
s = new AppMeasurement();
//Setup application config variables
s.account = "myrsid";
s.visitorNamespace = "myco";
s.trackingServer = "myco.dl.sc.omtrdc.net";
// Set variables and track Page View
Dictionary<string,object> trackData;
trackData["pageName"] = "Some page name";
trackData["prop1"] = "Some Prop Value";
s.track(v);
```

### **(trackLink) Send a Custom Link**

```
private void someEventHandler(object sender, RoutedEventArgs e) {  
    s.trackLink("http://www.myco.com", "o", "Home Link");  
}
```

# Video Measurement in Silverlight Apps

AppMeasurement for Silverlight lets you capture data about video usage in your Silverlight application. To do this, AppMeasurement for Silverlight gathers basic information from the media player, then builds a session of events and sends it to Adobe data collection servers for processing. After the collection servers process the video data, it is accessible through a set of SiteCatalyst video reports.

Complete information to implement video measurement using Silverlight is available in the [Measuring Video in SiteCatalyst](#) guide.