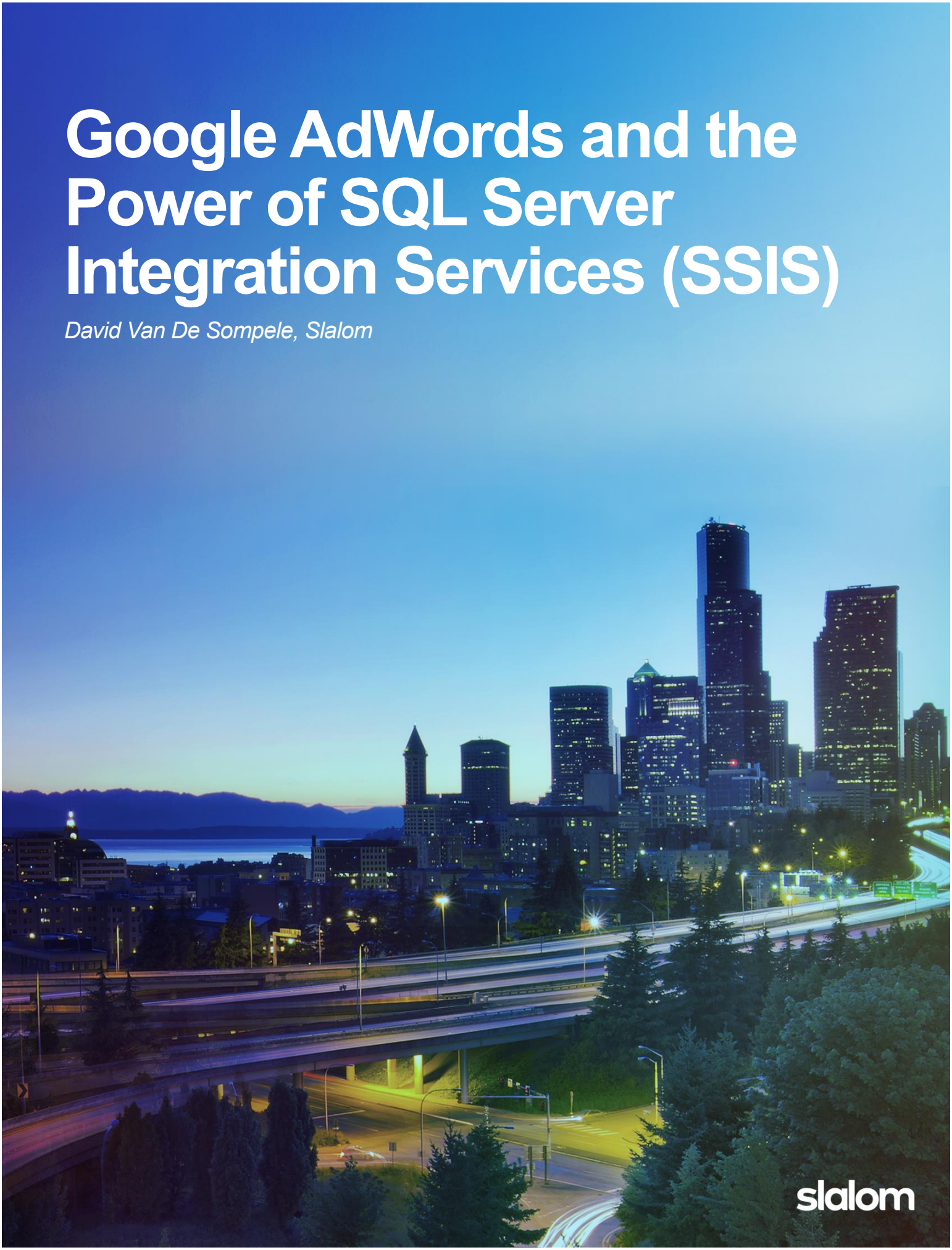
Google AdWords and the Power of SQL Server Integration Services (SSIS)

David Van De Sompele, Slalom



Google AdWords and the Power of SQL Server Integration Services (SSIS)

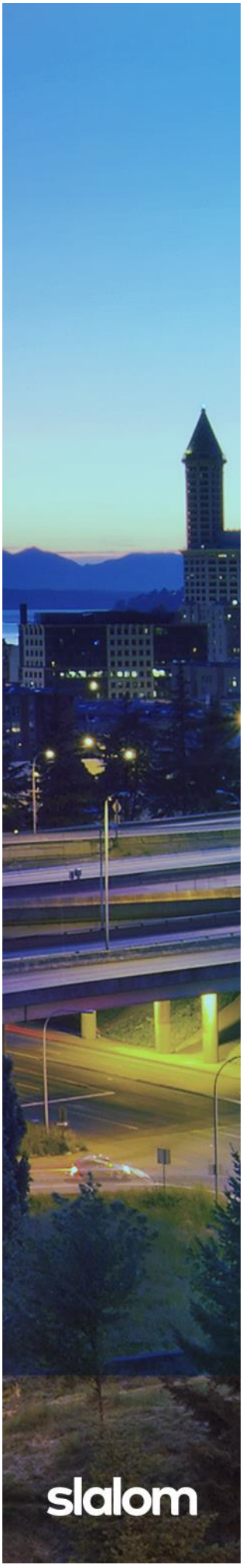
By David Van De Sompele, Slalom

Do you advertise on Google using the AdWords platform? Do you have a need to download metrics data generated by AdWords campaigns into your enterprise data environment? Importing reporting data from Google AdWords into a database involves some manual steps, and if you do this frequently, the cost in the form of people's time and effort may be significant. I recently automated this procedure for a client; leveraging the flexibility of SQL Server Integration Services and earned gratitude and smiles from some financial business analysts in the process.

I have been working directly with SQL Server, through much iteration, for almost twenty years and I am truly amazed at the product's evolution during that time. For example, the introduction in SQL Server 2000 of Data Transformation Services-an ETL development platform–and its later refinement and transformation into SQL Server Integration Services (SSIS)–the incredibly powerful and flexible .NET enabled data manipulation (extract, transform and load) tool that ships with SQL Server. It has connection components for just about every data source you will encounter in the business world, and it has some very robust data transformation components that make advanced data manipulation a relatively easy task. For me, a great day is one on which I discover a new capability or a new, possibly better, way of doing something within Integration Services. Little did I know that one great day was about to become several, and lead to this article, when my current client asked if I could use Integration Services to help them solve a problem.

My client is a mobile advertising company. The Google AdWords platform is one of several tools they use to meet their customers' needs. The platform is an online advertising service that places advertising copy at the top, bottom, or beside the list of search results Google displays for a particular search query. Business analysts in my client's finance group have been downloading various statistical and performance reports from AdWords, manipulating the data in a spreadsheet, and then importing it to a SQL Server database– a manual and time-consuming process. They asked if I could use Integration Services to automate the entire process, saving them time and positioning them to provide timelier reporting back to their customer base. This is the kind of challenge I thoroughly enjoy and it meant I would have to quickly become familiar with the Google AdWords platform.

Google has published an API for AdWords and included a developer portal from which you can download everything needed to build custom applications for accessing an AdWords account. According to the developer's guide, "The AdWords API is a collection of web services that you can use to build applications that manage AdWords accounts and their associated campaign data. While the AdWords API is based on SOAP 1.1, high-level client libraries are provided to help you develop applications more quickly." There are client libraries and code examples available for a number of languages and platforms including Java, .NET (4.0 and higher), PHP, PERL, and RUBY. The API uses the OAuth2 authentication protocol and since SSIS does not currently have a connection component that supports OAuth2, custom coding at least the authentication piece is necessary.



After downloading and unzipping the .NET libraries you will find an entire Visual Studio solution complete with code examples for interacting with the API written for both VB.NET and C#. After reviewing the code examples, it was apparent that these were intended to work within the context of a full web application. The libraries are intended to leverage either a web.config or app.config file for a lot of settings related to the OAuth2 authentication credential and the SOAP Listener Extension-there are stubbed out examples in both files. Integration Services, while being a .NET-enabled platform, is structured differently than a standard web application and that means using the AdWords .NET assemblies within the context of Integration Services requires custom configuration and additional code beyond that required for the OAuth2 authentication.

The focus of this paper is the specific code and customizations required to make SSIS authenticate correctly with the AdWords API and call an instance of the ManagedCustomer component which will position you to automate calling and downloading AdWords reports. Specifically, I will demonstrate/discuss the following:

- How to make a custom .NET assembly available to SSIS Script Component task.
- How to programmatically pass OAuth2 authentication credentials to an API.
- How to locate and modify Integration Services execution host configuration files for necessary runtime settings.

A takeaway from this is that some of the configuration principals are transferrable/reusable with other custom .NET assemblies that you may want to use within the realm of SSIS. As mentioned, Integration Services is a highly flexible tool. The Script Task component available at the Control Flow level is one of the most flexible components within Integration Services.

We will use a Script Component task to write the custom code needed to properly interact with the AdWords API, but before coding can start you must complete a couple of critical configuration steps to make the AdWords assemblies available to the Script component. It is worth noting that the first two steps are required for any custom .NET assembly you want to use within the Script Task component.

Add the .NET assemblies to the Global Assembly Cache

This is necessary on both the machine you will use to develop the SSIS package and the SQL Server on which the package will reside and execute. You will need the GACUtil.exe utility to add the assemblies to the global assembly cache on your development machine. If you have the full version of Visual Studio 2010 with default install the GACUtil.exe can be found in this location:

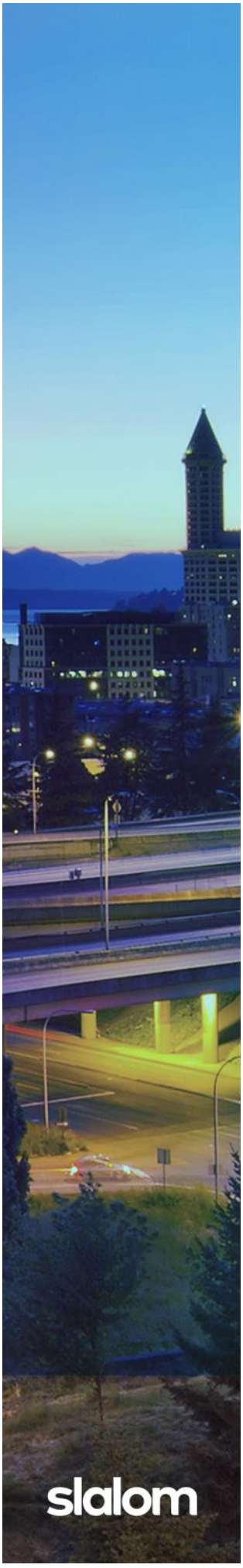
C:|Program Files (x86)|Microsoft SDKs|Windows|v7.0A|Bin|NETFX 4.0 Tools

If you only have the VS2010 shell that ships with SQL Server 2012 you will not have the GACUtil utility. You can get it by installing the Windows 7.1 SDK–a free download from Microsoft. You will need the 7.1 version as this supports the .NET 4.0 Framework. For specifics on using the GACUtil.exe please see this Microsoft Article. Note that Microsoft does not recommend using GACUtil.exe for adding assemblies to the GAC on production servers. Rather, they suggest using the Windows Installer.

To use the Google AdWords API there are two specific assemblies that must be added to the Global Assembly Cache:

Google.Ads.Common.dll

Google.Adwords.dll



Add a reference to the .NET assemblies in your Script Task

Add a new Script Task to the Control Flow tab in your SSIS project. Double-click on it to open the Script Task Editor. Select the Script Language you will use (I'm using VB.NET) and then click on the Edit Script button to open the code editor. Click on the Project Menu and select Add Reference from the drop down. Click the Browse tab from the Add Reference pop-up and navigate to the location at which you unzipped the Google API package. You will want to add the following two files as references:

Google.Ads.Common.dll

Google.Adwords.dll

Now you are ready to custom code the script task. Start by adding the following three lines to the Imports section of the script:

```
Imports Google.Api.Ads.AdWords.Lib

Imports Google.Api.Ads.Common.Lib

Imports Google.Api.Ads.AdWords.v201406 – The version referenced in this
line, v201406, may vary depending upon which version of the AdWords
assemblies you are using.
```

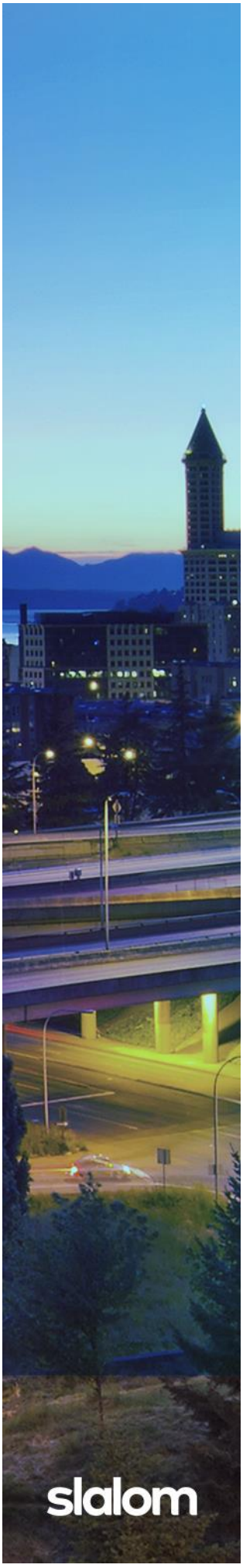
As mentioned earlier, the API uses the OAuth2 authentication protocol. You can read more about this at the OAuth2 website. You will need eight specific pieces of information in order to authenticate with the protocol. The AdWords .NET assemblies expect to find the authentication credentials in either the app.config or web.config files that are supplied with the code examples. Since SSIS does not use these configuration files you must programmatically set the OAuth2 credentials at runtime. Using VB.NET, here is how this is done:

First, you must call a new instance of the AdWordsUser class:

```
Dim bt(0) As Byte

Dim AdWordsMCC As New AdWordsUser
```

Then set the OAuth2 credentials:

```
TryCast(AdWordsMCC.Config, AdWordsAppConfig).UserAgent = "Your Assigned
User Agent"

TryCast(AdWordsMCC.Config, AdWordsAppConfig).OAuth2ClientId ="Your
OAuth2ClientId"

TryCast(AdWordsMCC.Config, AdWordsAppConfig).OAuth2ClientSecret = "Your
Client Secret"

TryCast(AdWordsMCC.Config, AdWordsAppConfig).OAuth2AccessToken ="Your
Access Token"

TryCast(AdWordsMCC.Config, AdWordsAppConfig).DeveloperToken = "Your
Developer Token"

TryCast(AdWordsMCC.Config, AdWordsAppConfig).AdWordsApiServer =
"https://adwords.google.com"

TryCast(AdWordsMCC.Config, AdWordsAppConfig).OAuth2RefreshToken = "Your
refresh Token"
```



```
TryCast(AdWordsMCC.Config, AdWordsAppConfig).ClientCustomerId = "The
client Id you are working with"
```

Now you are ready to call a new instance of the Managed Customer:

```
Dim mcService As New ManagedCustomerService


Try

     mcService =
CType(AdWordsMCC.GetService(AdWordsService.v201406.ManagedCustomerServic
e), _       ManagedCustomerService)


    mcService.RequestHeader.clientCustomerId = Nothing


Catch mc As Exception

     Dts.Log(Err.Description, 0, bt)

     Dts.Log(Err.GetException.Message.ToString(), 4, bt)

End Try
```

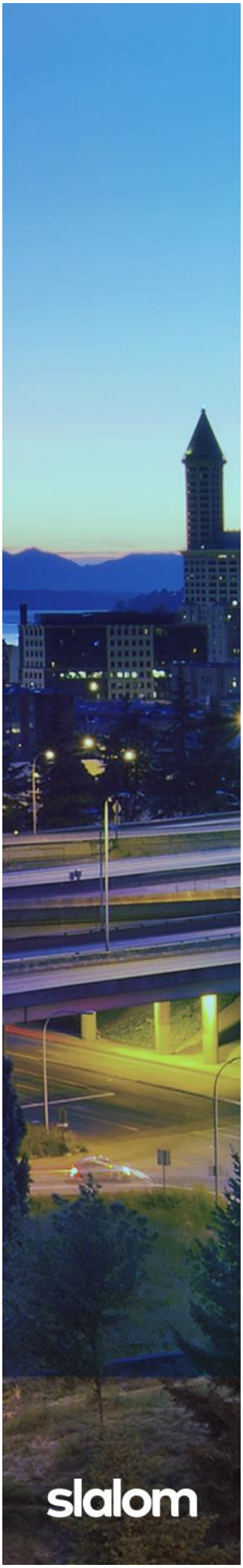
Now you are ready to write the code needed to accomplish your download tasks within the API. Google has supplied plenty of self-explanatory code samples to assist you with navigating the Managed Customer Service component and using it to download reports so I am not going to discuss that here. Instead, I will conclude by discussing the final required configuration change within SSIS.

Reference the SOAP Listener Extension in the SSIS configuration files

The AdWords assemblies require a reference to a SOAP listener service and you can find examples of this in the supplied app.config and web.config files. The entry looks like this:

```
<system.web>
              <webServices>
                <soapExtensionTypes>
                  <add
type="Google.Api.Ads.Common.Lib.SoapListenerExtension,
Google.Ads.Common, Version=3.2.0.0, Culture=neutral,
PublicKeyToken=52807268f2b614dc" priority="1" group="0"/>
                </soapExtensionTypes>
              </webServices>
</system.web>
```

Note that if you are using a different version (I am using v201406) of the AdWords Assemblies; the Version tag will be a different value that you will need to obtain from Google. Integration Services has a set of execution hosts that run Integration Services



packages in various settings (command line, SQL Agent, debug mode from Business Intelligence Development Studio, etc.). Each of these execution hosts has its own configuration file and it is to each of those configuration files that you must add the Soap Listener Extension entry I listed above.

Listed below are the four most common SSIS execution hosts and their corresponding configuration files and locations.

DTExec.exe.config: DTExec.exe is the standalone command tool used for executing SSIS packages; it is an execution host so it has a configuration file. This file can be found in both the 32 and 64 bit folders. The default install locations for SQL Server 2012 are:

C:|Program Files|Microsoft SQL Server|110|DTS|Binn|DTExec.exe.config

C:|Program Files (x86)|Microsoft SQL Server|110|DTS|Binn|DTExec.exe.config

DtsDebugHost.exe.config: DtsDebugHost.exe is the execution host used by Business Intelligence Design Studio (BIDS)/The Visual Studio Shell when executing a package from the Design Studio in Debug Mode. This file can be found in both the 32 and 64 bit folders. The default install locations for SQL Server 2012 are:

C:|Program Files|Microsoft SQL Server|110|DTS|Binn|DtsDebugHost.exe.config

C:|Program Files (x86)|Microsoft SQL Server|110|DTS|Binn|DtsDebugHost.exe.config

DtsHost.exe.config: DtsHost.exe is the execution host for many SQL Server internal processes like SQL Agent. This file can be found in both the 32 and 64 bit folders. The default install locations for SQL Server 2012 are:

C:|Program Files|Microsoft SQL Server|110|DTS|Binn|DtsHost.exe.config

C:|Program Files (x86)|Microsoft SQL Server|110|DTS|Binn|DtsHost.exe.config

ISServerExec.exe.config: ISServerExec.exe is the execution host for the Integration Services server. This is new to SQL 2012 as part of the redesigned Integration Services platform enhancements. This file is used when you execute packages from the SSIS Catalog as a SQL Agent job step. This file can be found in both the 32 and 64 bit folders. The default install locations for SQL Server 2012 are:

C:|Program Files|Microsoft SQL Server|110|DTS|Binn|ISServerExec.exe.config

C:|Program Files (x86)|Microsoft SQL Server|110|DTS|Binn|ISServerExec.exe.config

If you do not correctly reference the SOAP Listener Extension or if it is missing from one of the configuration files you will receive the following error at execution time: **The SoapListenerExtension class is not loaded. The most possible cause for this error is that you haven't registered Google.Api.Ads.Common.Lib.SoapListenerExtension as a soap extension under configuration/system.web/webServices/soapExtension Types in your App.config or Web.config.**

If you advertise on the Google AdWords platform and find you are expending a lot of time and resources to manually import the metrics data into your enterprise data environment, consider using SQL Server Integration Services to automate the process. Integration Services is a powerful and flexible tool that offers a wealth of standard and customizable components for extract transform and load processes. The Script Task component is especially useful for customizations that go beyond the already-robust capabilities found in the arsenal of standard components. Finding new ways to leverage these will add value to the work you do for your clients as you help them become more efficient in their business operations.