SAP BI BLOG
all things business intelligence
decisionfirst technologies

Connecting to Salesforce Data Through SAP BusinessObjects

I recently found myself needing to connect my Salesforce® data to SAP® BusinessObjects™. My project required meshing data from multiple sources and BusinessObjects is the best way to do that. Although Salesforce has a native reporting component, it's not as robust as BusinessObjects, which allows me to run powerful analysis, and generate highly formatted, interactive reports and dashboards with extensive visibility, alignment and insight across my entire enterprise.

When I started looking at the best way to accomplish this, I knew I had to do one of two things: extract the data out of Salesforce, or find an ODBC driver to connect to it. Extracting the data required keeping it in a staging area in my database, but I knew that wouldn't produce real-time updates. So I decided an ODBC driver was the path of least resistance.

With several driver options available, I decided to experiment to find the best one. My focus was on the usability and functionality of each without taking cost into consideration.

I searched the internet for Salesforce ODBC drivers and downloaded each from its respective vendor. I installed them on my BusinessObjects server and began my experiment.

Below is a list of the drivers I tested:

- Native SAP Salesforce ODBC installed with BusinessObjects Server and Client tools
- Progress DataDirect Salesforce
- Easysoft Salesforce ODBC Driver
- CDATA ODBC Driver for Salesforce
- Simba Salesforce ODBC Driver

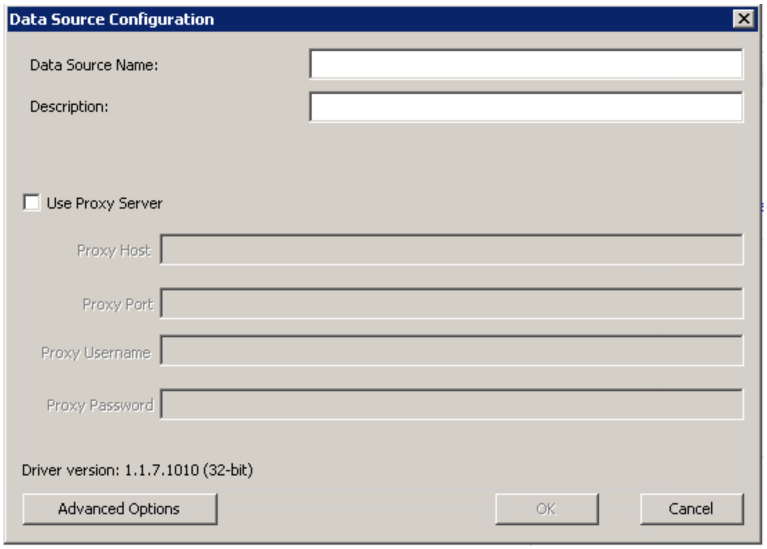
To get started I made sure I had the following:

- Salesforce.com login id, password, and security token*
- Administrator access to my BusinessObjects server
- Server information
  - Operating System - Windows 2008 R2 64 bit
  - Business Objects Version – SAP BusinessObjects BI platform 4.1 SP4 – (14.1.4.1327)

*The security token is needed to access and view certain secured tables (i.e. USER) in the Salesforce schema. Without it you can't read data from secured tables.

Creating an identical universe with each driver was essential. It included Opportunity, Accounts and Users tables, as well as a few columns to extract the Year and Month based on date. Below are my results.

Native SAP Salesforce ODBC



□ Use Proxy Server

Driver version: 1.1.7.1010 (32-bit)

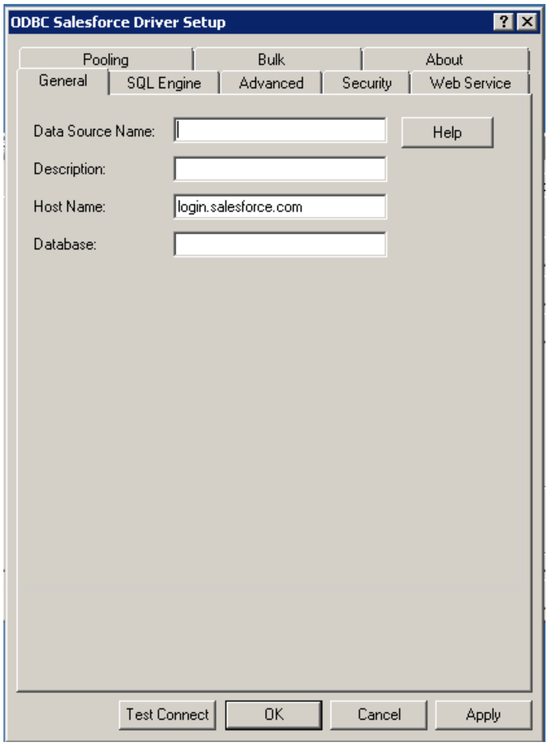
Note: Only the 32bit driver is available through the SAP BusinessObjects install process.

The setup of this driver was fairly easy since there are no fields to supply a username, password, or security token. This must be done within Information Design Tool (IDT) when you create a secure connection. There is no place to provide the security token, however. This meant I couldn’t query the USER table in my universe. I was able to build a report and successfully query the data from Salesforce. The report retrieved approximately 2,000 records in three seconds, but I needed to find a way to use the USER table. This led me down the path of trying other third-party vendor ODBCs.

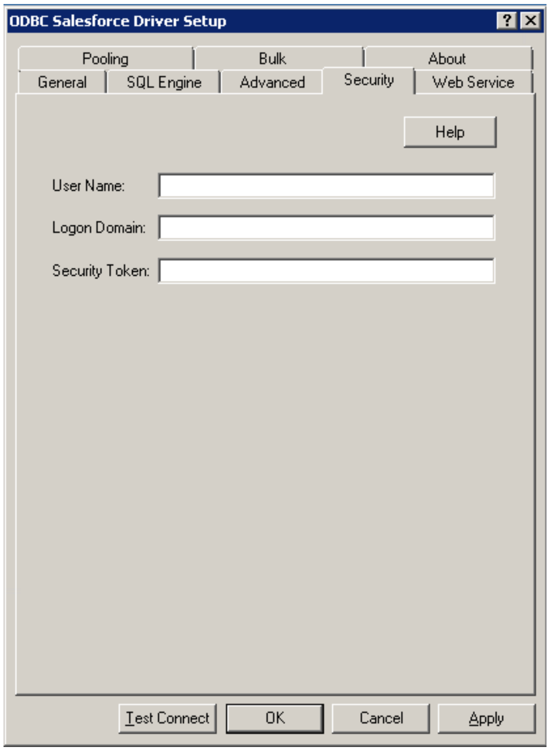
Progress DataDirect Salesforce Driver

https://www.progress.com/data-sources/salesforce

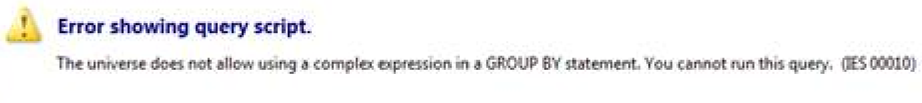
I downloaded all the Progress DataDirect drivers. They bundle them together resulting in 19 different drivers being installed! I only needed the DataDirect7.1 Salesforce driver to create my DSN entry. I installed the 32bit and 64 bit drivers.



Host Name: login.salesforce.com



The setup for Progress did have a field for Username and Security token, but no place for a password. This password needed to be saved in the universe connection within IDT. Their setup was a bit more complicated; luckily Progress provides full support on driver evaluations. Once I configured the driver based on their specifications, I built a simple universe and was able to view data from the USER table. Web Intelligence did return an error when I used the Year and Month dimensions derived from a date. The message explained that Salesforce did not allow complex expressions in group by statements. This is related to using the generic ODBC connection when creating the connection in IDT. But the speed of the query was comparable to the native SAP Salesforce driver.



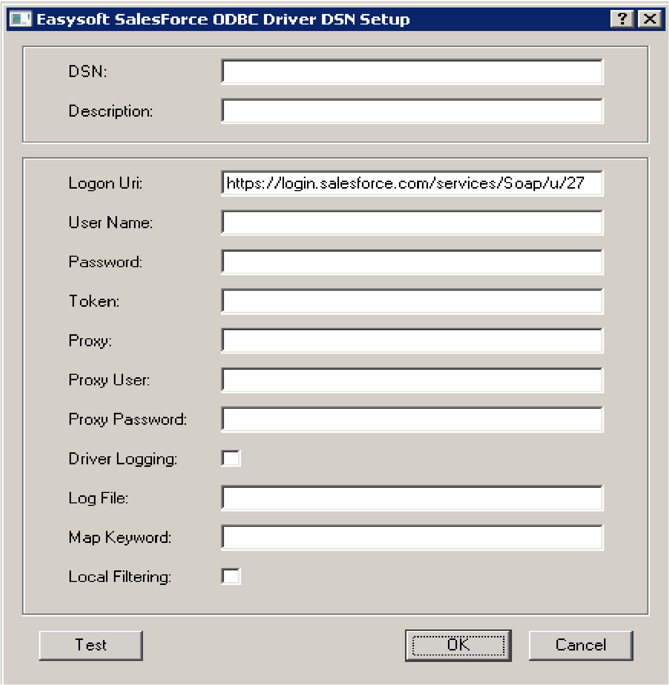
Error showing query script.

The universe does not allow using a complex expression in a GROUP BY statement. You cannot run this query. (IES 00010)

Easysoft Salesforce ODBC Driver

http://www.easysoft.com/products/data_access/odbc-salesforce-driver/index.html#section=tab-1

For Easysoft I downloaded and installed the 32bit and 64 bit drivers. Installation and configuration were fairly easy and straightforward.



DSN:

Description:

Logon Url: https://login.salesforce.com/services/Soap/u/27

User Name:

Password:

Token:

Proxy:

Proxy User:

Proxy Password:

Driver Logging: □

Log File:

Map Keyword:

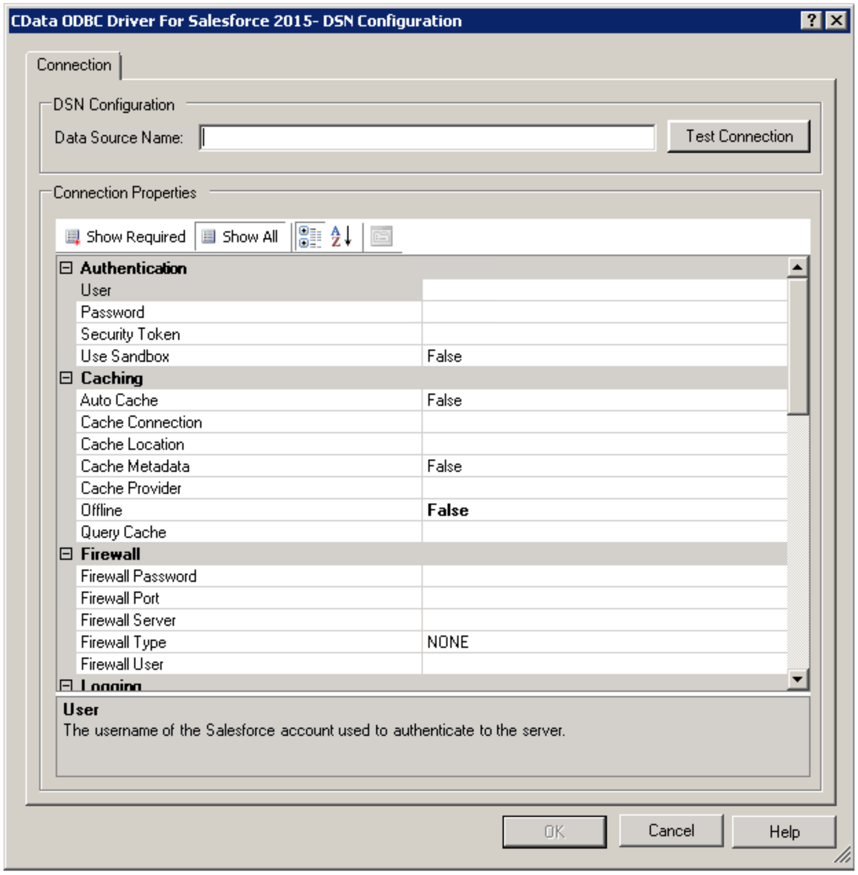
Local Filtering: □

Similar to the Progress driver, I queried and viewed data from the USER table. But the similarities stopped here. As far as execution time for my query, Easysoft performed the worst out of the drivers tested. The simple report I was building took more than 1 minute to return data. I didn't contact their support as I didn't see any tweaking that could be done with the driver.

CDATA ODBC Driver for Salesforce

http://www.cdata.com/drivers/salesforce/download/odbc/

For CDATA, I repeated the download and installation process for their 32bit and 64 bit drivers. Again, both tasks were easy and straightforward. There were many fields, but only three were required - User, Password and Security Token. CDATA support was extremely responsive to my tickets, sending a fix within a couple of hours. But, after four fixes I finally gave up testing their driver. I never got it work.



Data Source Name:

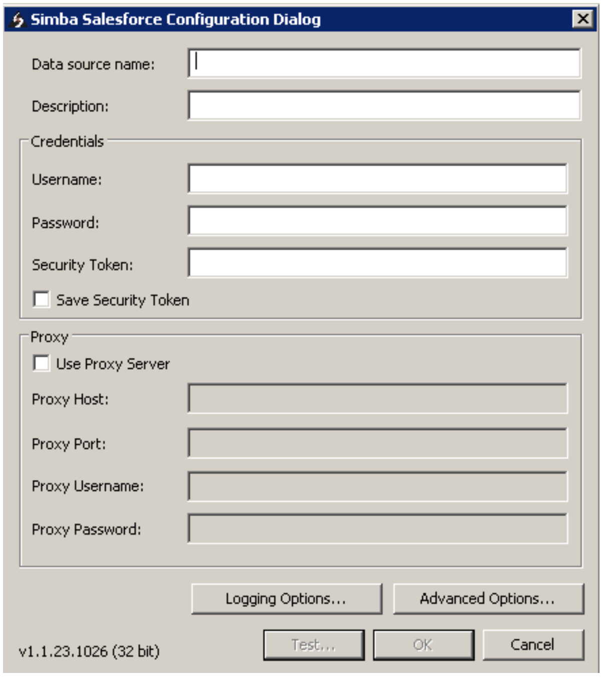
Authentication	
User	
Password	
Security Token	
Use Sandbox	False
Caching	
Auto Cache	False
Cache Connection	
Cache Location	
Cache Metadata	False
Cache Provider	
Offline	**False**
Query Cache	
Firewall	
Firewall Password	
Firewall Port	
Firewall Server	
Firewall Type	NONE
Firewall User	
Logging	

User
The username of the Salesforce account used to authenticate to the server.

Simba Salesforce ODBC Driver

http://www.simba.com/connectors/salesforce-driver

I downloaded and installed the 32 bit and 64 bit drivers for Simba version 1.1.23. This driver was fairly easy to configure. It had fields for Username, Password, and Security token. It even allowed you to save the Security Token!



☐ Save Security Token

☐ Use Proxy Server

v1.1.23.1026 (32 bit)

I built my simple universe and report with great success, except I still received an error message when trying to use the Year and Month dimensions. I was frustrated and began looking for a work around. I found that within Salesforce tables you can create custom fields that are dynamically populated based on a formula. (I'm not going to go into that process, but if I get enough comments on this post maybe the DFT Marketing team will bug me to write on that too!)

So I created two dynamic fields in Salesforce based on a date column. When I refreshed my structure in my universe, the new columns showed up! Sweet success! Now I can use those fields in my report.

But I still wanted more info on this issue. I contacted Simba and they told me, "Our ODBC and JDBC connections to the BOBJ Universes are supported directly through them."

This made me think… when I installed my server I didn't specify the Simba drivers. After further research I realized that when I added the native SAP Salesforce ODBC Driver, it is actually a Simba driver! Who knew?

But, as is the case in IT, there was still the issue of not being able to see the USER table with the native driver because there was no place for the security token.

That's when I found out that each version of BusinessObjects comes with a different version of the Simba driver. The version of BusinessObjects on my server came with Simba Driver version 1.1.7, also a 32 bit driver. The version I installed from Simba's website was 1.1.23, 16 versions ahead of the native driver!

Luckily I had another server with a BusinessObjects SP5 Patch6 installed on it. That version was 1.1.23. Huzzah! From this updated server I created a 32bit connection (only the 32 bit connection comes installed with BusinessObjects) and created my universe. I was able to see the USER table and query from it. The query response time was fast with no degradation.

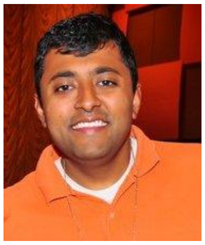
It turns out the best choice was actually the native driver that installs with BusinessObjects server and client tool installations. I just took the long road to get there. Winner Winner Tofu Dinner!

Additional Info on the Complex Expression error

- When creating the secured connection in IDT, you need to use the Salesforce connection for OpenAccess to go along with Simba driver 1.1.23 or greater otherwise you will not be able to use complex group-by's in your reports.
- If you use the generic option, you will get this error on any dimension that is derived on another field.

Additional notes about connecting your Salesforce environment to BusinessObjects

- API calls are made to your Salesforce environment to retrieve data.
- Each Salesforce license has a limited amount of calls that can be made in a rolling 24 hour period.
- The number of calls made depends on number of user and type of license you have.
- More information on number of API calls can be found here: https://help.salesforce.com/apex/HTViewHelpDoc?id=integrate_api_rate_limiting.htm



Mitesh Shah
Sr. Business Intelligence Consultant
Decision First Technologies
Mitesh.Shah@decisionfirst.com

Mitesh Shah is a senior Business Intelligence consultant for Decision First Technologies specializing in data services, web services, front-end reporting and analysis, dashboards and Information Governance for numerous enterprise customers across the country. Specifically he works with SAP® Data Services, Information Steward, BusinessObjects™ Tool Suite and HANA.