Technical Note:

# Using the XMLA connector with proxies and SSL

Revision History:

| | Version | Author | Details |
|---|---|---|---|
| 1 | 1.0 | Laurentiu Iordache | Created this document |
| | | | |
| | | | |

# Table of Contents

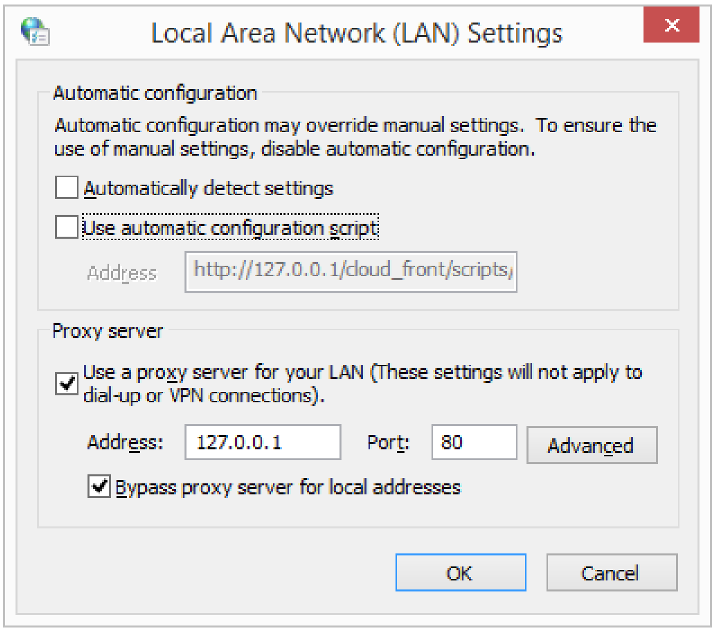
Introduction ........................................ 3
XMLA Connector and proxies ........................................ 4
HTTP vs HTTPS ........................................ 5
Proxy Bypass ........................................ 6
Automatic configuration ........................................ 7
Authoritative Proxy ........................................ 7
SSL Support ........................................ 8

## Introduction

This document details the usage of the XMLA Connector in conjunction with proxies. It also details SSL specific topics.

## XMLA Connector and proxies

The XMLA Connector is designed to automatically detect proxies. From the Windows perspective, the proxy settings are accessed from **Control Panel** by clicking **Network and Internet** and then **Internet Options**. In the dialog box that pops-up go to the **Connections** tab. On the bottom right of the tab click on **LAN settings** and the following dialog appears:



Local Area Network (LAN) Settings

Automatic configuration

Automatic configuration may override manual settings. To ensure the use of manual settings, disable automatic configuration.

□ Automatically detect settings

□ Use automatic configuration script

Address: http://127.0.0.1/cloud_front/scripts/

Proxy server

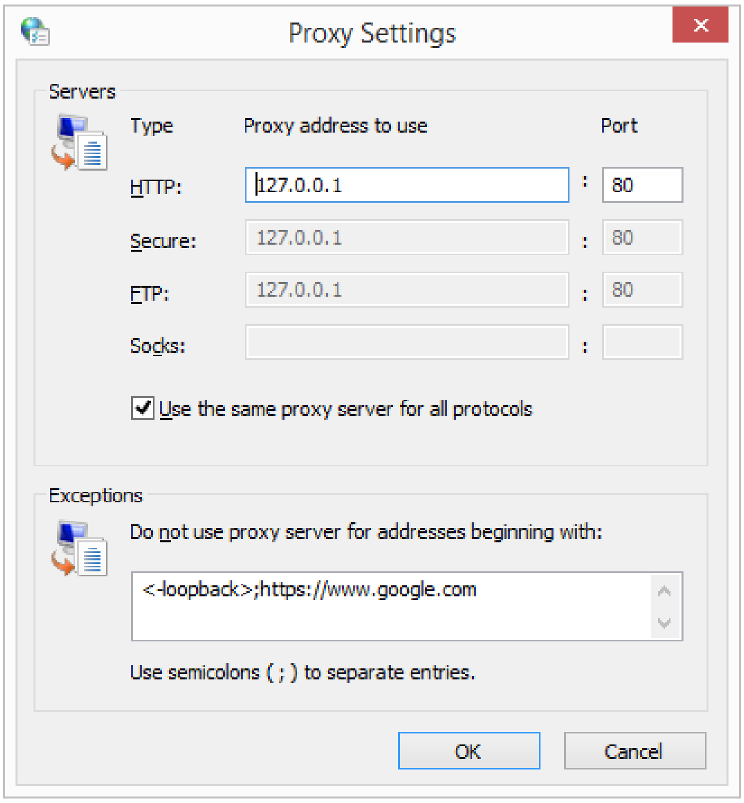
☑ Use a proxy server for your LAN (These settings will not apply to dial-up or VPN connections).

Address: 127.0.0.1 Port: 80

☑ Bypass proxy server for local addresses

Fig. 1

This example shows a system configured to use a proxy on 127.0.0.1:80. It also bypasses server for local addresses. Clicking on advanced will produce a popup like the one in the next figure:



Proxy Settings

Servers

| Type | Proxy address to use | Port |
|---|---|---|
| HTTP: | 127.0.0.1 | 80 |
| Secure: | 127.0.0.1 | 80 |
| FTP: | 127.0.0.1 | 80 |
| Socks: | | |

☑ Use the same proxy server for all protocols

Exceptions

Do not use proxy server for addresses beginning with:

<-loopback>;https://www.google.com

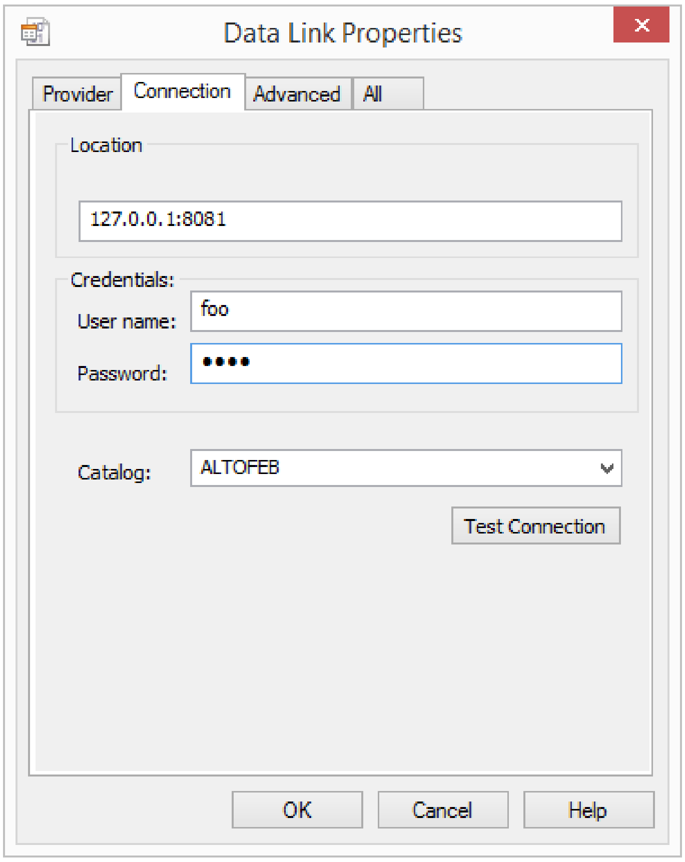
Use semicolons ( ; ) to separate entries.

Fig. 2

The configuration in the previous figure shows the system using the same proxy for all protocols. It also shows an exclusion list for the loopback address and https://www.google.com.

From the perspective of the XMLA Connector only HTTP and Secure ports are used. The FTP and Secure proxies are ignored. Apart from this all the remaining settings have an effect to the functionality of the provider.

The usage of a proxy server is a combination between the parameters aforementioned and the location, as it is typed in the **Location** text input when creating a new Connection. An example is given below:



Location: 127.0.0.1:8081

Credentials:

User name: foo

Password: ••••

Catalog: ALTOFEB

Fig. 3

## HTTP vs HTTPS

By default the provider tries to connect to the server using http. Using an http prefix is optional. With respect to the figure above, the address

```
127.0.0.1:8081
```

is equivalent to:

```
http://127.0.0.1:8081
```

In order to use SSL the address must be prefixed with `https://`. For example, in the previous configuration, in order to connect using SSL he address should be:

```
https://127.0.0.1:8081
```

## Proxy Bypass

There are a number of options to bypass the proxy as follows:

1. With the checkbox **Bypass proxy server for local addresses** active any address starting with an IP in the local network will not use the proxy.

For Example:

If the computer running Excel has the IP `192.168.0.123` and the computer running the XMLA server has the IP `192.168.0.124`, the address:

`http://192.168.0.124/saiku/xmla`

will bypass the proxy. Similarly, the address

`https://192.168.0.124/saiku/xmla`

will also bypass the proxy.

However, if the XMLA server would have the IP 192.168.1.124 (notice the different networks: 192.168.1.0/24 for the server and 192.168.0.0/24 for the client), then the address:

`http://192.168.0.124/saiku/xmla`

will use the proxy as it is configured.

2. When the address is listed in the **Exceptions** (Fig. 2, in the bottom) list and the begging of the address matches the exception.

For Example:

If the Exceptions list contains the address `https://contoso.com` then the address:

`https://contoso.com :8080/mondrian/xmla`

will bypass the proxy. However the address

`http://contoso.com :8080/mondrian/xmla`

will use the proxy as it starts in `http://` and not `https://.`

3. For loopback addresses when <-loopback> is present in the Exceptions.

For example:

With the configuration as shown in Fig. 2 when the server runs on the same machine as the client. The address:

`http://127.0.0.1:8081`

will bypass the proxy. Similarly, an address as follows:

`https://localhost:8081`

will also bypass the proxy.

## Automatic configuration

With respect to Fig. 1, if **Using automatic configuration script** is checked then the provider will download and use the proxy auto-config file specified in Address. Using this option overrides all the other settings enumerated before. For details about PAC files read the following http://en.wikipedia.org/wiki/Proxy_auto-config (for example).

The current version of provider has support for interpreting the following specific functions

| Pos | Function |
|---|---|
| 1 | isInNet |
| 2 | isPlainHostName |
| 3 | dnsDomainIs |
| 4 | shExpMatch |
| 5 | dnsResolve |
| 6 | myIpAddress |
| 7 | isResolvable |

## Authoritative Proxy

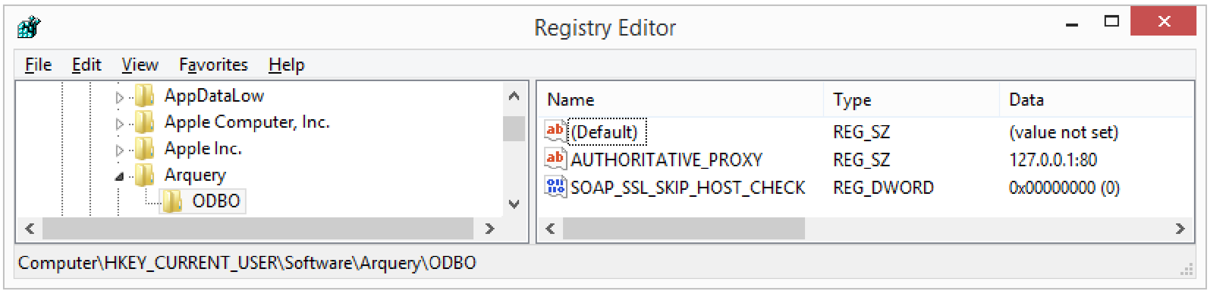
This feature allows bypassing the entire Windows configuration with respect to proxy. In order to enable this the following key should be present in the registry:

```
[HKEY_CURRENT_USER\Software\Arquery\ODBO]
```

Under this key a string value as follows control the proxy override:

| Name | Value |
|---|---|
| AUTHORITATIVE_PROXY | address:port |

The figure below shows the system configured for a proxy on 127.0.0.1 listening on port 80:



| Name | Type | Data |
|---|---|---|
| (Default) | REG_SZ | (value not set) |
| AUTHORITATIVE_PROXY | REG_SZ | 127.0.0.1:80 |
| SOAP_SSL_SKIP_HOST_CHECK | REG_DWORD | 0x00000000 (0) |

Computer\HKEY_CURRENT_USER\Software\Arquery\ODBO

Fig. 4

## SSL Support

The XMLA Connector works seamless with SSL servers. Prefixing the server address with `https://` will switch to SSL communication.

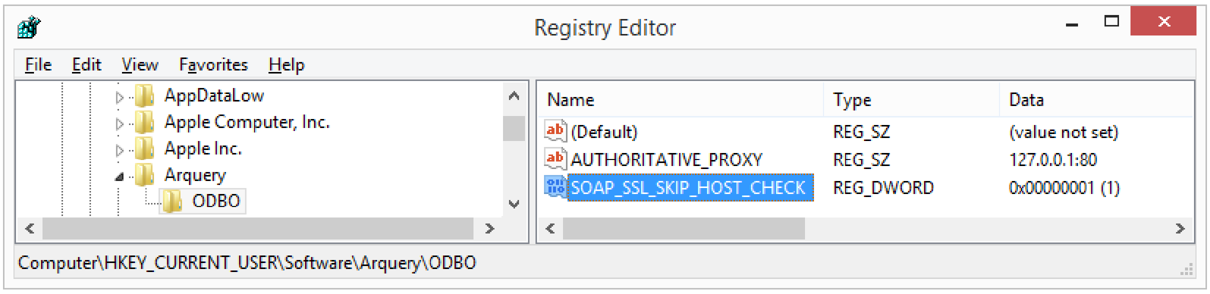
By default, the connector **does validate** the server certificate. This means that self-signed certificates will be rejected. In order to bypass the validation and allow working with self-signed server certificate the following key should be present in the registry:

```
 [HKEY_CURRENT_USER\Software\Arquery\ODBO]
```

Under this key a DWORD value as follows controls the proxy override:

| Name | Value |
|---|---|
| SOAP_SSL_SKIP_HOST_CHECK | 1 |

Write here any value different of 0 to bypass the validation. Check the next figure for an example:



| Name | Type | Data |
|---|---|---|
| (Default) | REG_SZ | (value not set) |
| AUTHORITATIVE_PROXY | REG_SZ | 127.0.0.1:80 |
| SOAP_SSL_SKIP_HOST_CHECK | REG_DWORD | 0x00000001 (1) |

Computer\HKEY_CURRENT_USER\Software\Arquery\ODBO

Fig. 5

**Please note:** Disabling server validation exposes the client to man in the middle attacks. This feature should be used with care. The recommended way of working with a given self-signed certificate is the following option:

An alternate way of using a self-signed certificate is by importing it in the Windows certificate store. For details about importing a certificate read here: https://technet.microsoft.com/en-us/library/cc754489.aspx?f=255&MSPPError=-2147217396 . The provider honors the trust exactly as it is honored by Windows.

With respect to trusted Root CA the provider implements a mixed model: The most common CA certificated are preregistered with the provider. In addition to this the provider registers all the certificates on the Windows Trusted Root Certificate Authorities list.