Enterprise Reporting Solution

DataBlock Connector Implementation Guide

Product Version 6.3

Last Updated 8/30/2019

Trademark, Publishing Statement, and Copyright Notice

Table of Contents

DataBlock Connector - Introduction

The Argos DataBlock Connector API provides a way to load the results of an Argos report query into third-party applications and business intelligence tools. It uses a script to pass parameters into MAPS and then retrieve the data via a RESTful API. The results of the report query are returned in JSON format to the calling application.

Evisions provides example scripts written using HTML and JavaScript. You are free to write scripts in any language or that work with a variety of third-party applications. We encourage you to upload any custom scripts you create to the CO-OP Share section of our Support site, where users of Evisions software can collaborate and share solutions to common problems. If you are looking for a solution that works with a specific tool or application, you can search on the CO-OP or ask on our forums to see if another institution has already been working on a compatible script.

Installation

Update MAPS License

In the MAPS Config, click the **Check for Updates** button at the bottom of the screen to download and apply your new license.

You can verify that the updated license has been applied by going to the License screen and confirming that the DataBlock Connector is listed as one of the enabled features for Argos.

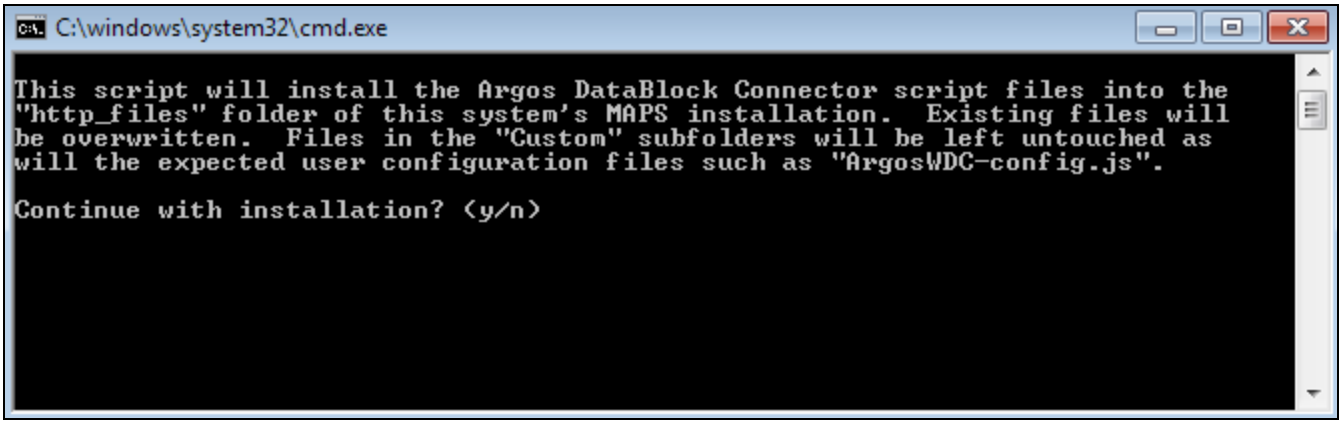
MAPS Security Best Practices

When using the Argos DataBlock Connector, Evisions recommends configuring MAPS to use HTTPS instead of HTTP for its integrated web server. Use of an SSL certificate will ensure that your credentials are encrypted when using the DataBlock Connector. For information on configuring the HTTP/HTTPS file sever, please refer to the MAPS Help.

Download and Install Scripts

Log on or remote in to your MAPS server. Download the Argos DataBlock Connector script files from the Documentation and Software section of the Evisions website.

Extract the .zip file into a local directory on the machine where MAPS is installed. Then, run the install.bat to copy the files into the http_files folder of your MAPS service directory.



You may need to provide administrative permissions in order to perform the installation on your system. If so, enter 'y' at the prompt.

When the installation completes, navigate to the `..\http_files\adbc` folder (typically something like `C:\Program Files (x86)\Evisions\MAPS\Service\http_files\adbc`) and notice that it contains three folders:

- fonts
- Sample
- Tableau

The **Sample** and **Tableau** folders contain working example scripts that can be used either as-is, or as a starting point for your own development. The Sample folder contains a generic example, and the Tableau folder contains a script that is specifically designed to import Argos data into a Tableau® environment. The fonts folder contains font resources (icons) for the sample scripts.

Script Modification

Evisions provides the generic and Tableau sample scripts on an as-is basis. Our support team can help you get the sample scripts up and running on your system, but may not be able to provide assistance with custom modifications or any new scripts that you create for use with the DataBlock Connector. If needed, our Professional Services team can provide a quote for custom development work for additional scripts to meet your requirements.

We welcome you to upload any custom scripts that you develop to our CO-OP share, where you can collaborate with users at other institutions who may be working on similar projects.

Location of Custom Script Files

As a best practice, if you make custom modifications to the scripts provided by Evisions, it is a good idea to store them in `..\http_files\adbc\Sample\Custom` or `..\http_files\adbc\Tableau\Custom`. Creating a Custom folder ensures that your modifications will not be accidentally overwritten if you later reinstall the script files for any reason, such as an update to the baseline script files. If present, files in a Custom folder will automatically be used in place of any files of the same name in the directory immediately above the Custom folder.

User Configuration

Regardless of the script you use, you will need to have at least one user who has been configured with appropriate permissions in MAPS and Argos. This can be a MAPS user or an LDAP user, so long as they are given the necessary permissions either directly or through a group that they are a member of. The user(s) that will be logging in via the script must have the following permissions:

- Authorization to use the Argos application.
- An appropriate user role assigned in MAPS (Report Viewer or higher).
- Appropriate user security permissions within Argos to be able to access the folders and execute the DataBlocks to be run via the DataBlock Connector.
- Authorization in MAPS to use the data connection(s) used by these DataBlocks.
- Appropriate database permissions.

Testing the Script

Depending on which script you are using, refer to the following sections for instructions on configuring and running the scripts:

- Tableau script
- Generic sample script (JavaScript example)

Generic Sample Script

The sample script Evisions provides is intended to be used as a starting point for your own script development. It demonstrates a basic API call to the DataBlock Connector, and shows the JSON data being returned. You will need to modify the script as needed to work with any 3rd-party applications you wish to use with Argos data.

Install Script Files

If not already installed, refer to the previous section for instructions on obtaining the script files and placing them on the MAPS server.

The sample script is comprised of three files:

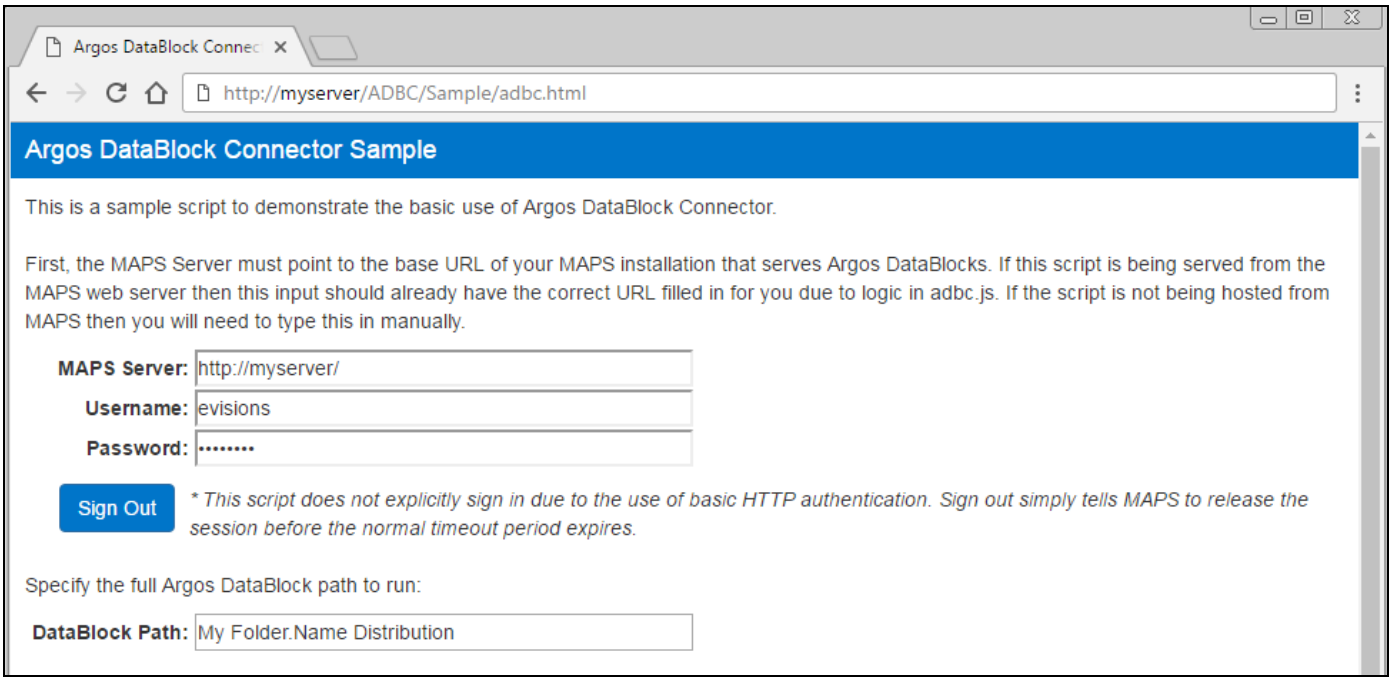
- adbc.html
- adbc.css
- adbc.js

By default, these files are installed in the `..\http_files\adbc\Sample` folder of the MAPS Service directory. If you intend to customize the script or write a new one, we recommend placing your modified files in `..\http_files\adbc\Sample\Custom` to ensure that your changes are not overwritten accidentally in the future. Files in a `Custom` folder will automatically be used in place of files of the same name one directory up.

Connecting to a DataBlock in the Simulator

To load the simulator, open a browser and navigate to the URL of your MAPS server, followed by the path to the adbc.html file from the http_files directory of the MAPS Service folder. For example, `http://myserver/adbc/Sample/adbc.html`

The page provides input fields where you can specify your server locations, a set of credentials, and a path to the DataBlock you want to run.

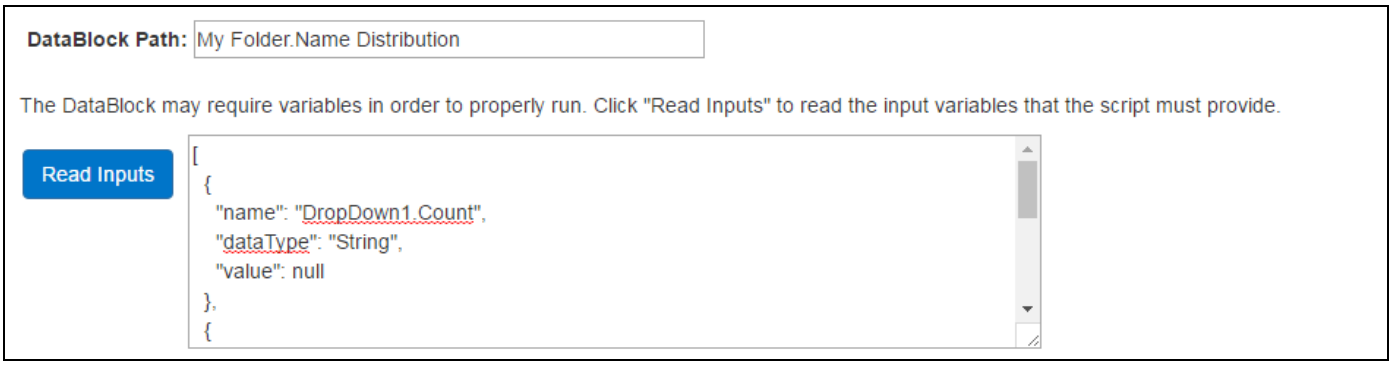


In the **MAPS Server** field, enter the URL or IP address of your MAPS server. Then, enter the username and password for a user who has appropriate permissions to access Argos, the DataBlock you wish to run, and any necessary data connections.

Note that there is no sign in button. The credentials you enter here are passed over in the API call when reading in the input fields or data.

In the **DataBlock Path** field, enter the path to the DataBlock that you want to access. The path uses the Argos path format where folders are separated by a period. In the example above, the DataBlock called "Name Distribution" is located in a folder called "My Folder". In the Argos client, you can find the path of any object by right-clicking on it and selecting **Copy Path to Clipboard**.

The next step is to select **Read Inputs** to retrieve a list of the parameters expected by this DataBlock's report query. The JSON structure for the required inputs will appear in an edit box to the right.



The JSON format includes a name, data type, and value for each parameter. The name and data type are provided for you when reading in the parameters. All values are set to "null" by default.

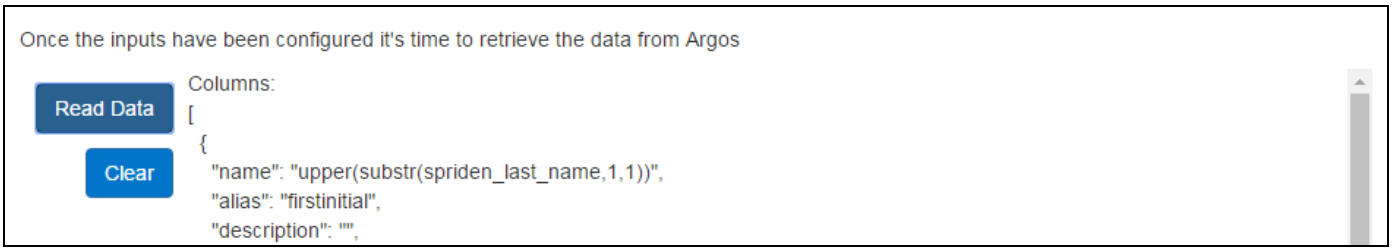
To edit the values or add additional values, you will need to edit the values in the JSON. For example, instead of "null" as the value for Dropdown1.Count, we have replaced this with "20" in the example below. Similarly, for the last name, we have replaced the "null" value with an array containing three names.

```
[
  {
    "name": "DropDown1.Count",
    "dataType": "String",
    "value": "20"
  },
  {
    "name": "Last_Name_Filter",
    "dataType": "String",
    "value": ["Ziegler","Morrison","Reyes"]
  },
  {
    "name": "Start_Date",
    "dataType": "Date",
    "value": "2017-03-16"
  }
]
```

Wildcard Characters

If the SQL in the report query supports it,(i.e., you know that the query is using the parameter in a LIKE statement), you can use a percent sign (%) as a wildcard character. For example, entering "Ab%" for a name field could return names such as "Abby", "Aberforth", etc.

When you are finished entering the parameters, select **Read Data** to get the results of the report query. Again, the results are returned in JSON format.



The columns (fields) being returned are listed first, followed by the rows:

```
Columns:
[
  {
    "name": "upper(substr(spriden_last_name,1,1))",
    "alias": "firstinitial",
    "description": "",
    "dataType": "String"
  },
  {
    "name": "count(spriden_pidm)",
    "alias": "count",
    "description": "",
    "dataType": "Float"
  }
]

Rows:
[
  ["A",12],
  ["G",6],
  ["J",11]
]
```

Tableau Configuration

Tableau® uses the term "Web Data Connector" for any data source retrieved via HTTP or HTTPS. The Argos Web Data Connector, or ArgosWDC, is the Argos DataBlock Connector for Tableau environments.

The ArgosWDC script provided by Evisions can be tested in several ways:

- Tableau
- Tableau Public (free edition)
- Tableau Web Data Connector Simulator

The WDC Simulator is an application provided by Tableau that emulates a Tableau environment specifically for purposes of testing Web Data Connectors. It can be used to test if access to a full Tableau environment is not available. You can also download and install the free edition of Tableau, called Tableau Public. This document explains how to set up the simulator and test the Evisions ArgosWDC script, followed by how to run the script in Tableau Public. The configuration should be similar when using the licensed version of Tableau.

Install Script Files

If not already installed, refer to the previous section for instructions on obtaining the script files and placing them on the MAPS server.

The Tableau script files are installed in the `..\http_files\adbc\Tableau` folder of the MAPS Service directory. If you intend to to customize the script or write a new one, we recommend placing your modified files in `..\http_files\adbc\Tableau\Custom` to ensure that your changes are not overwritten accidentally in the future. Files in a `Custom` folder will automatically be used in place of files of the same name one directory up.

Create Optional Config File

The Tableau script includes a template for a global config file, called ArgosWDC-config.template.js. If desired, you can name a copy of this file ArgosWDC-config.js, and use it to specify certain optional global configuration settings. Available settings include the base URL to MAPS, the name of the session cookie used by the DataBlock Connector, a maximum number of records to be returned, and a default DataBlock and inputs. You can find detailed configuration instructions in the comments section of the ArgosWDC-config.template.js file.

Note: In testing the script file, we have found that the Tableau simulator may perform poorly or crash when selecting extremely large amounts of data. You can use the optional maxRecordCount setting to limit the number of results returned if this becomes an issue for you.

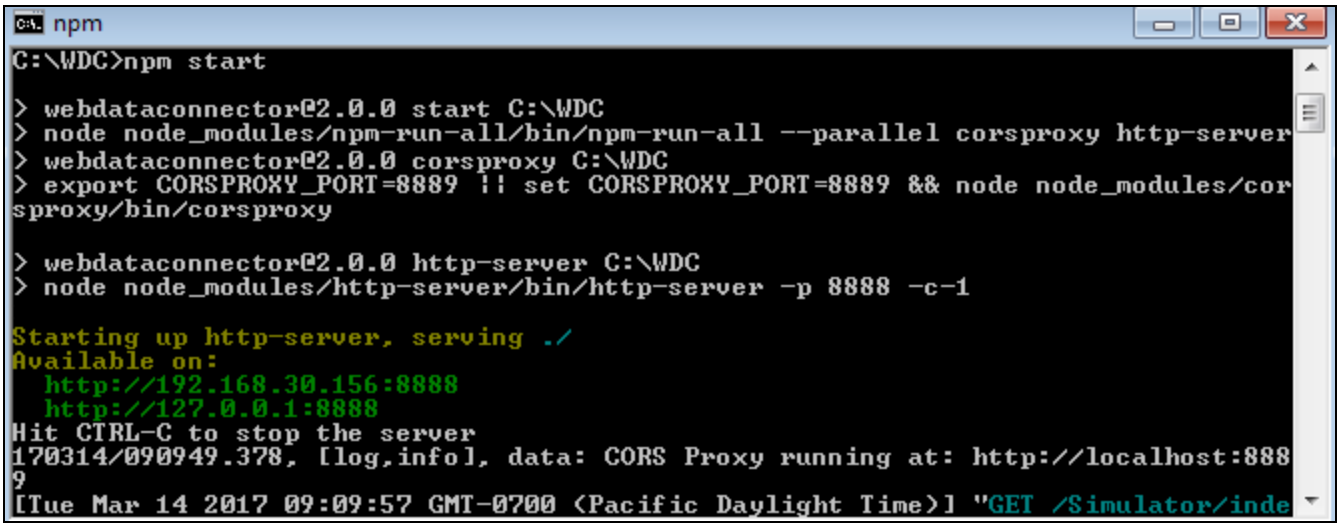
WDC Simulator Configuration

Before beginning, you should review the Tableau Web Data Connector page for an overview of the tool.

Follow the instructions in the Get Started section of their documentation to:

- Install Git
- Install Node.js
- Get the WDC SDK by cloning the repository
- Install npm

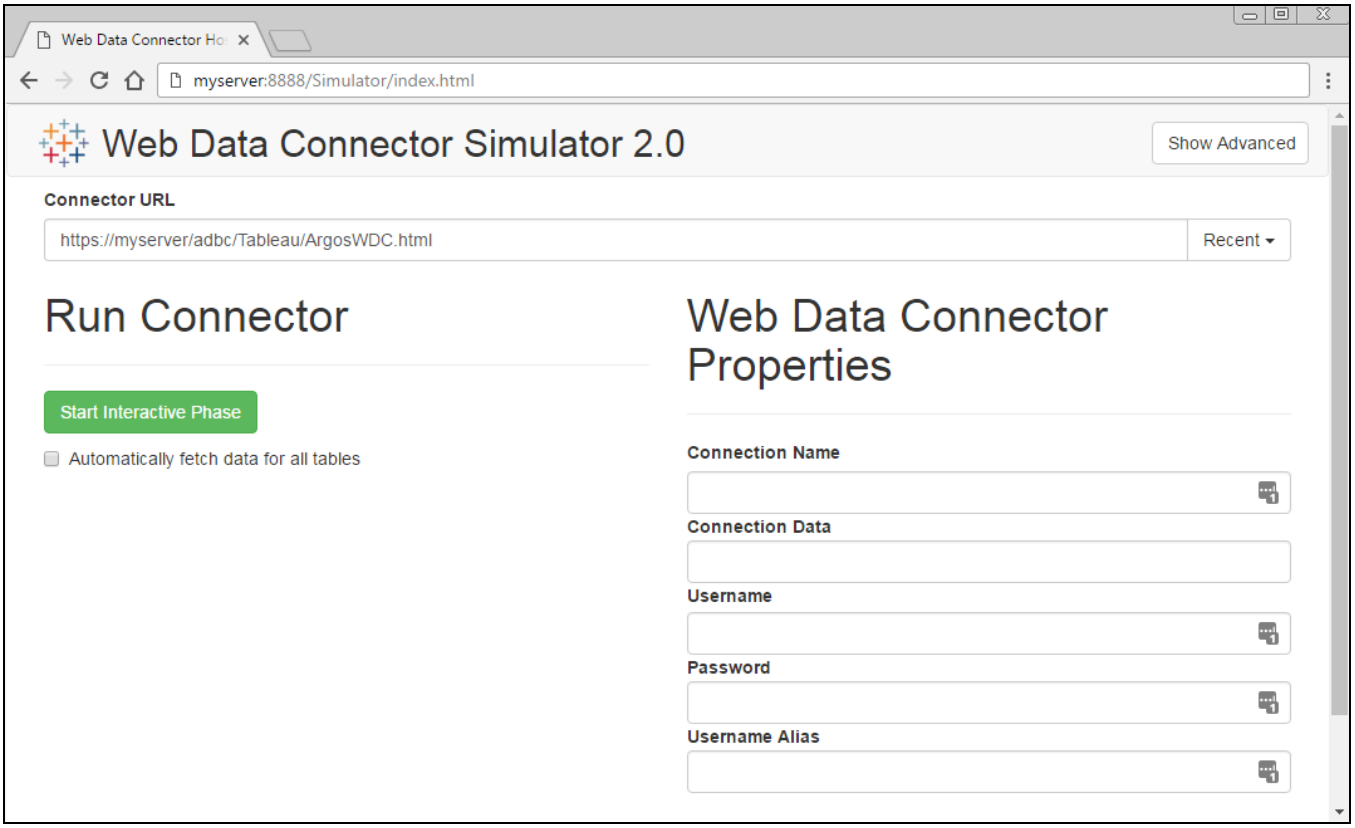
Once this is configured, to start the simulator, first open a command prompt and navigate to the directory where you downloaded the repository. Then, type `npm start` to start the test web server.



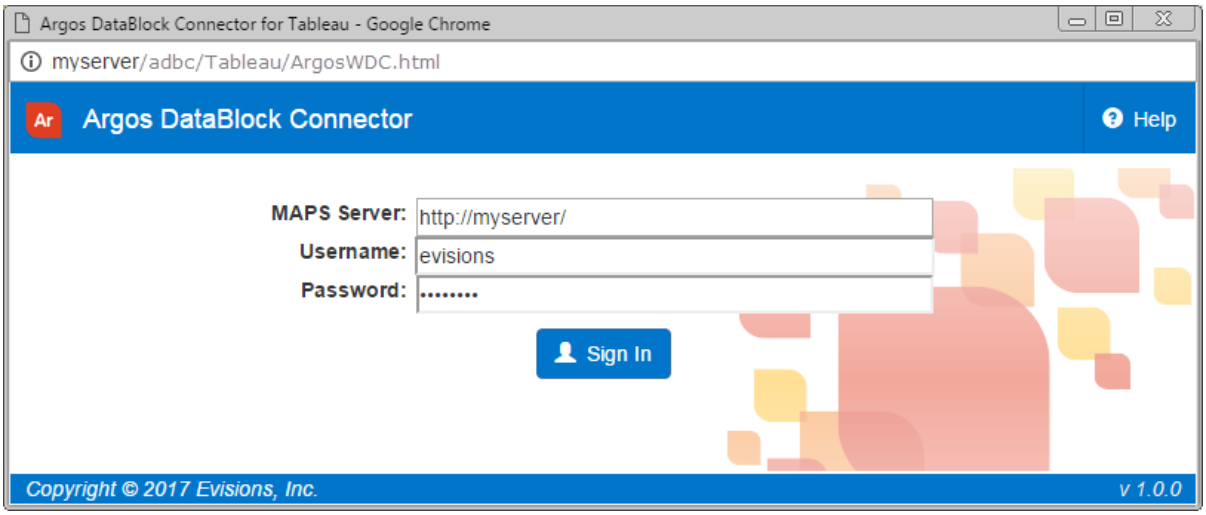
Connecting to a DataBlock in the Simulator

In a browser, navigate to http://localhost:8888/Simulator/index.html to load the simulator page.

In the **Connector URL** field, enter the address of your MAPS server followed by the path to the `ArgosWDC.html` file within the `http_files` folder in your MAPS Service directory. The path should look like `/adbc/Tableau/ArgosWDC.html`.

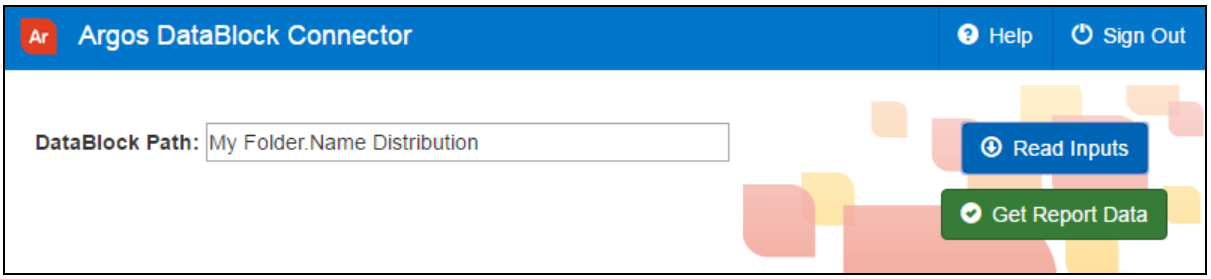


Click the **Start Interactive Phase** button to launch the Argos DataBlock Connector simulator:

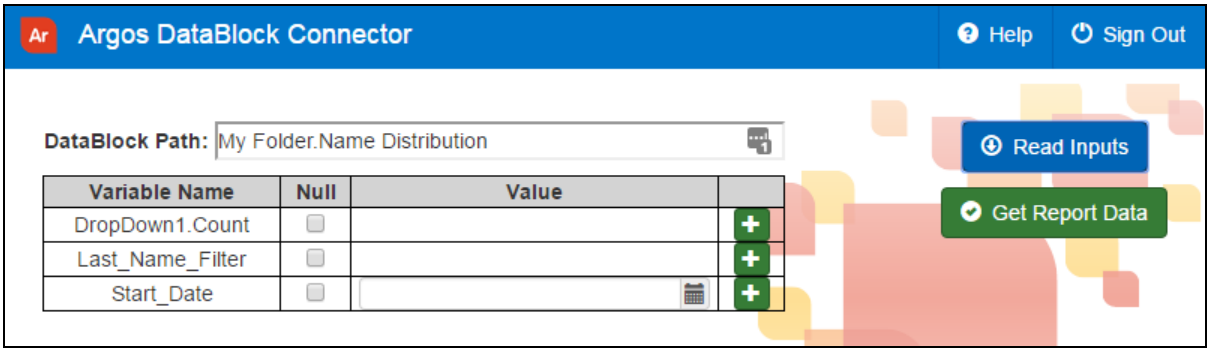


In the **MAPS Server** field, enter the URL or IP address of your MAPS server. Then, sign in as a user who has appropriate permissions to access Argos, the DataBlock you wish to run, and any necessary data connections.

On the next screen, enter the path to the DataBlock that you want to access. The path uses the Argos path format where folders are separated by a period. In the example below, the DataBlock called "Name Distribution" is located in a folder called "My Folder". In the Argos client, you can find the path of any object by right-clicking on it and selecting **Copy Path to Clipboard**.



Select **Read Inputs** to load a list of the parameters expected by this DataBlock's report query.

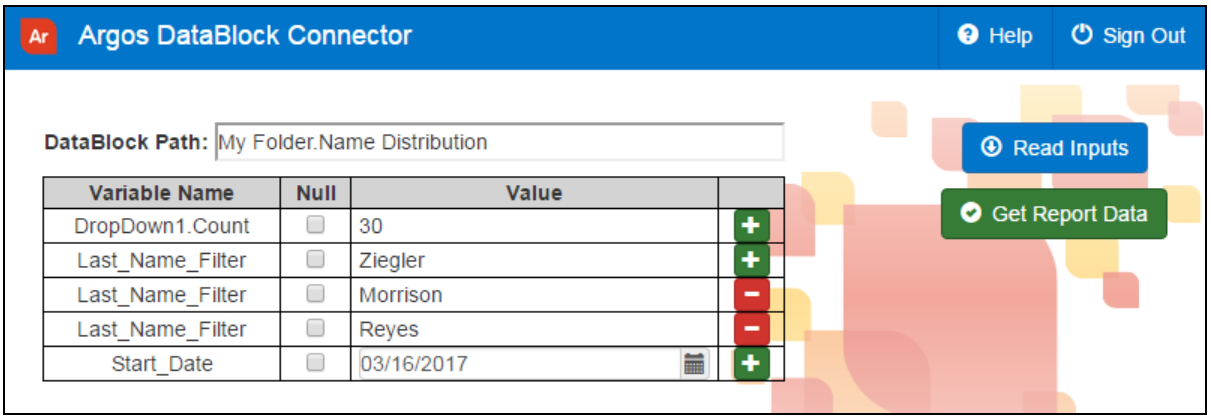


Enter the value for each variable in the **Value** field next to its name. If you want to specify more than one value for a variable, click the green + button next to the variable name to add an additional entry field. Use the red − button to remove a value that is no longer needed.

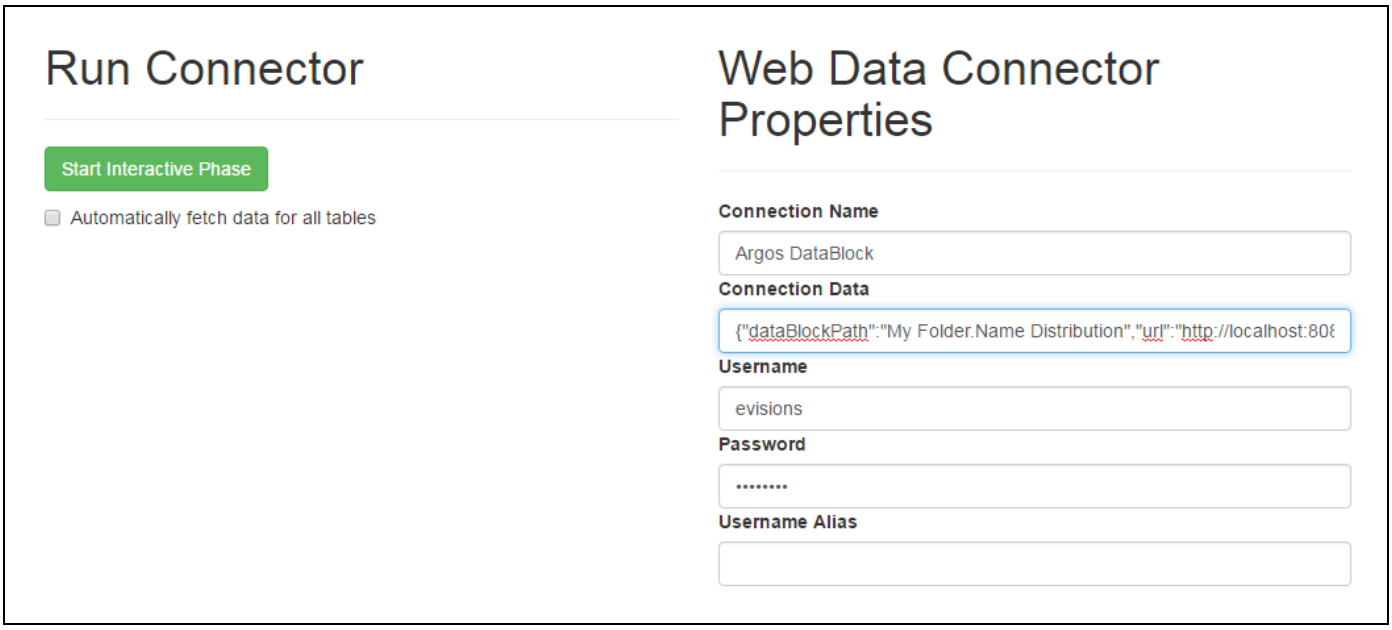
Wildcard Characters

If the SQL in the report query supports it,(i.e., you know that the query is using the parameter in a LIKE statement), you can use a percent sign (%) as a wildcard character. For example, entering "Ab%" for a name field could return names such as "Abby", "Aberforth", etc.

To enter a `null` value, check the box in the **Null** column instead of entering a value.



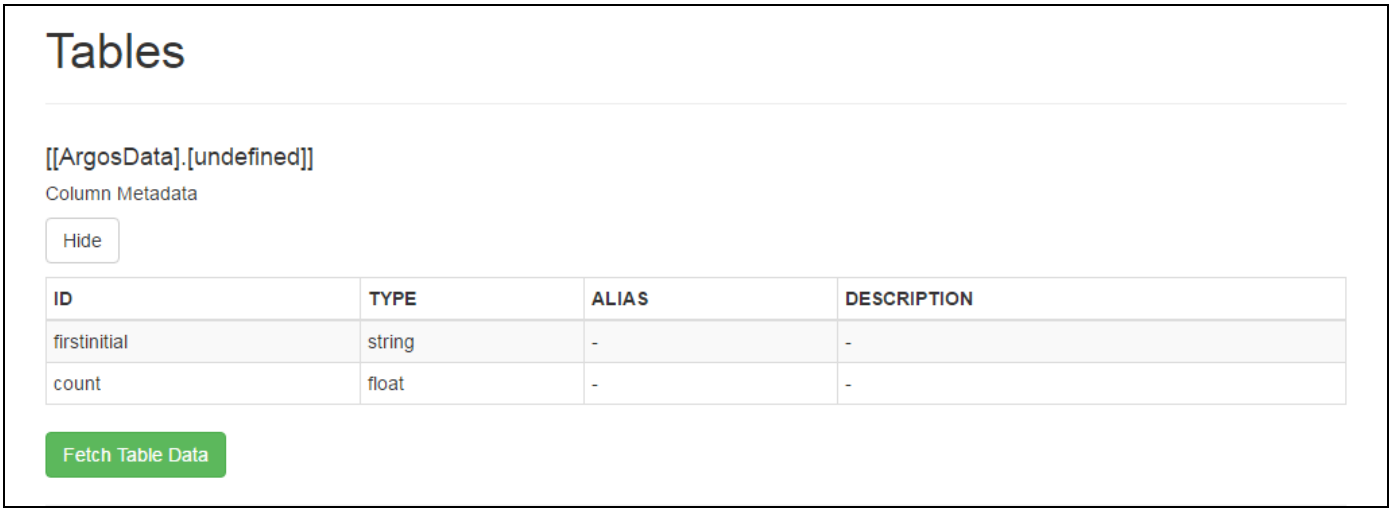
When finished, select **Get Report Data** to generate the JSON and copy the Web Data Connector properties into the Tableau simulator. Notice that the **Connection Name**, **Connection Data**, **Username**, and **Password** fields are now filled out.



The Connection Data field contains the JSON information required by the Argos DataBlock Connector API, which includes the MAPS server information, the location of the DataBlock, and all of the specified parameters and their values. In the example above, this takes the form:

```
{"dataBlockPath":"My Folder.Name Distribution","url":"http://localhost:8080/","inputs":
[{"name":"DropDown1.Count","dataType":"String","value":"30"},{"name":"Last_Name_
Filter","dataType":"String","value":["Ziegler","Morrison","Reyes"]},{"name":"Start_
Date","dataType":"Date","value":"2017-03-16"}]}
```

Below the Web Data Connector Properties, the **Tables** section lists the table fields that will be returned by the query.



Select **Fetch Table Data** to pull in the query results and verify that you are seeing the expected data.

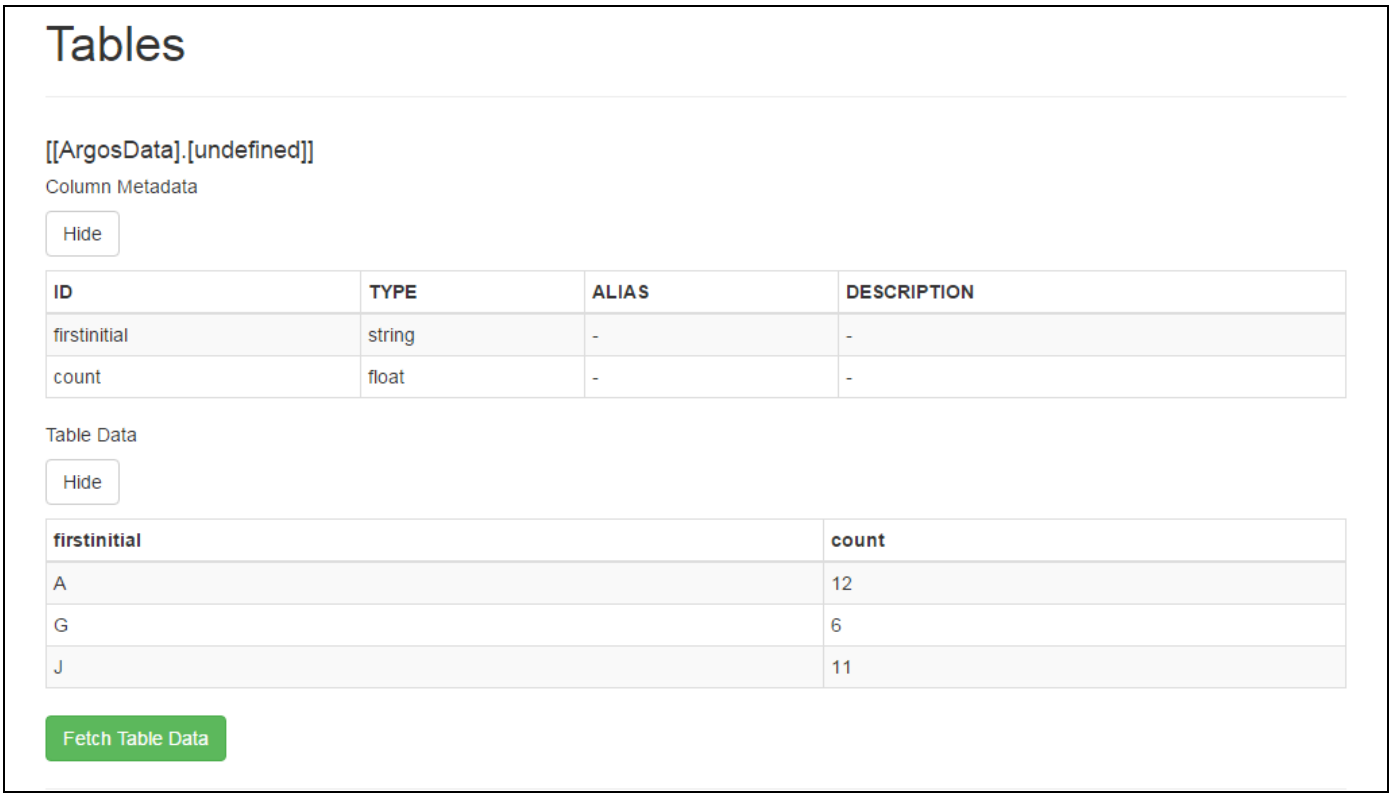
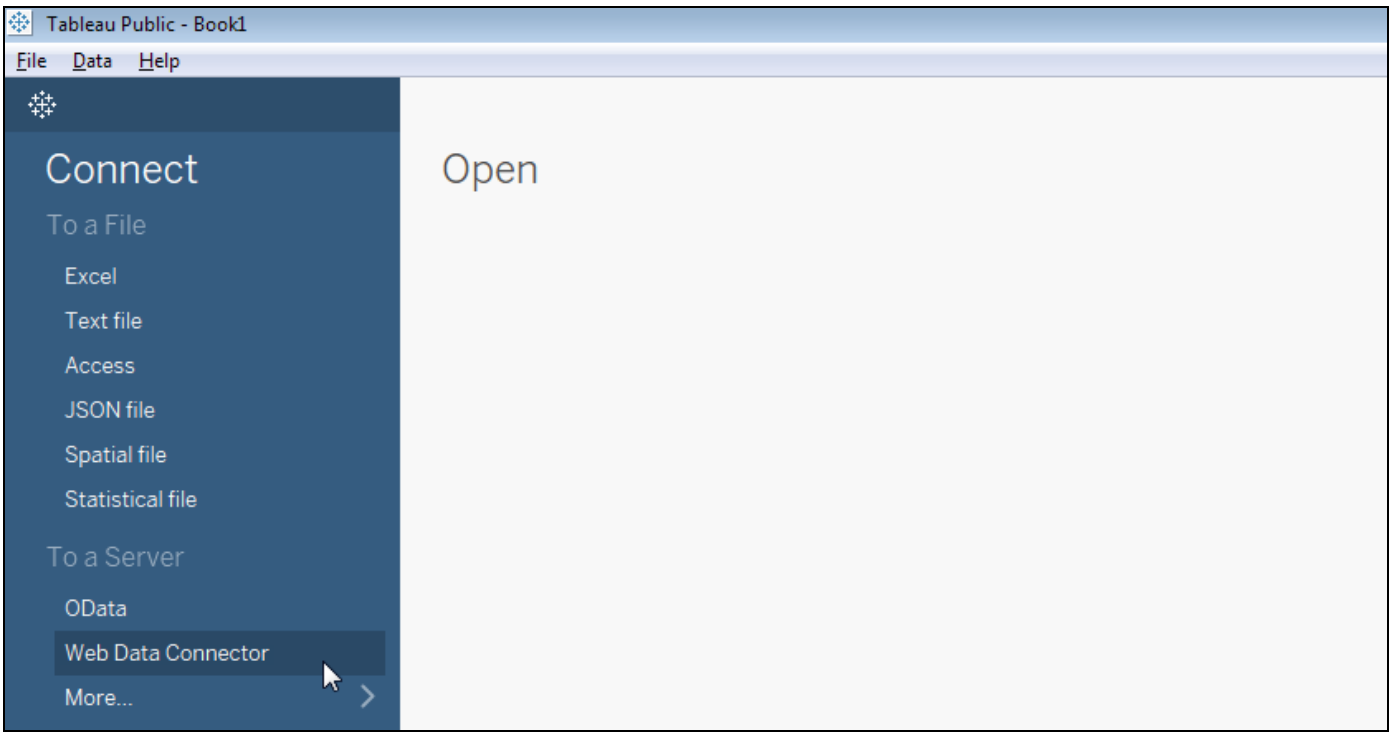
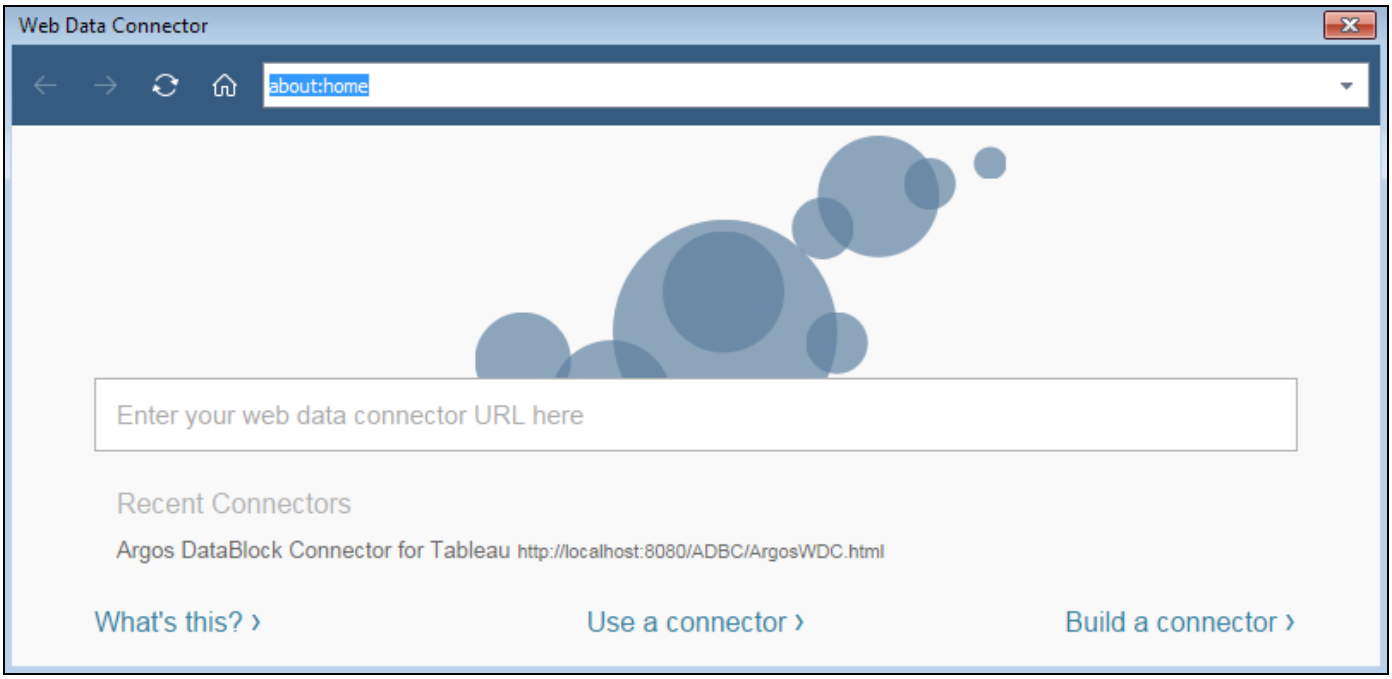


Tableau Public Configuration

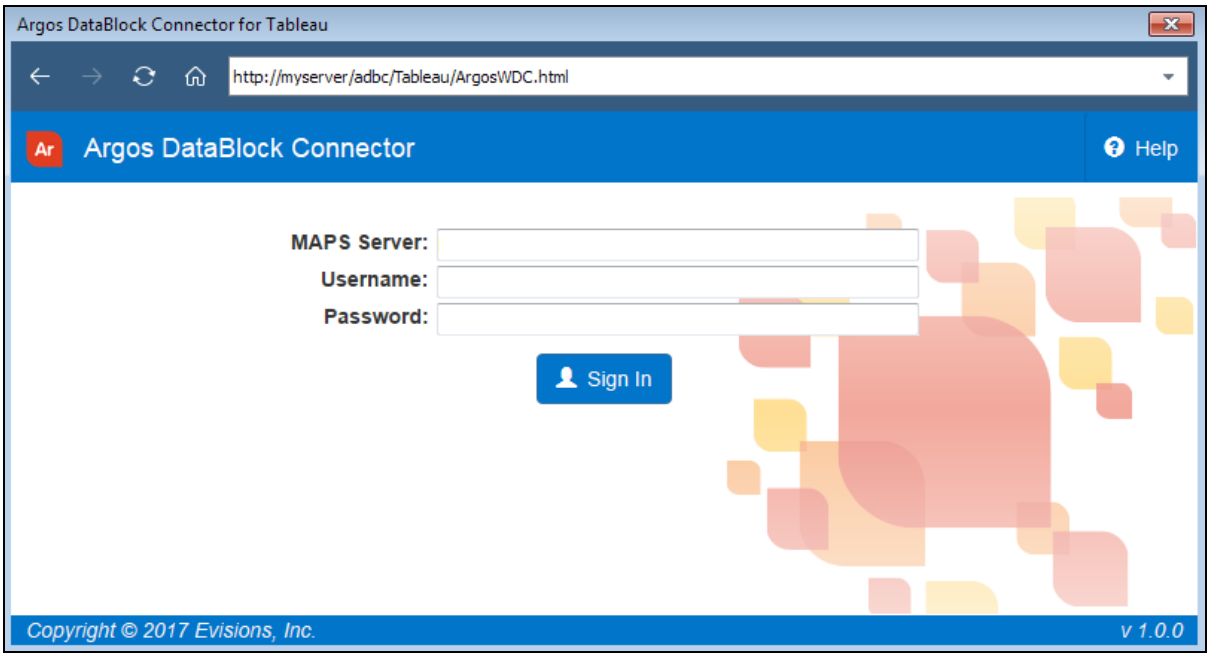
After installing Tableau Public, launch the application and select **Connect** -> **To a Server** -> **Web Data Connector** in the menu on the left.



Enter the URL to your MAPS server, followed by the path to the ArgosWDC.html file from the http_files directory of the MAPS Service folder. For example, `http://myserver/adbc/Tableau/ArgosWDC.html`.

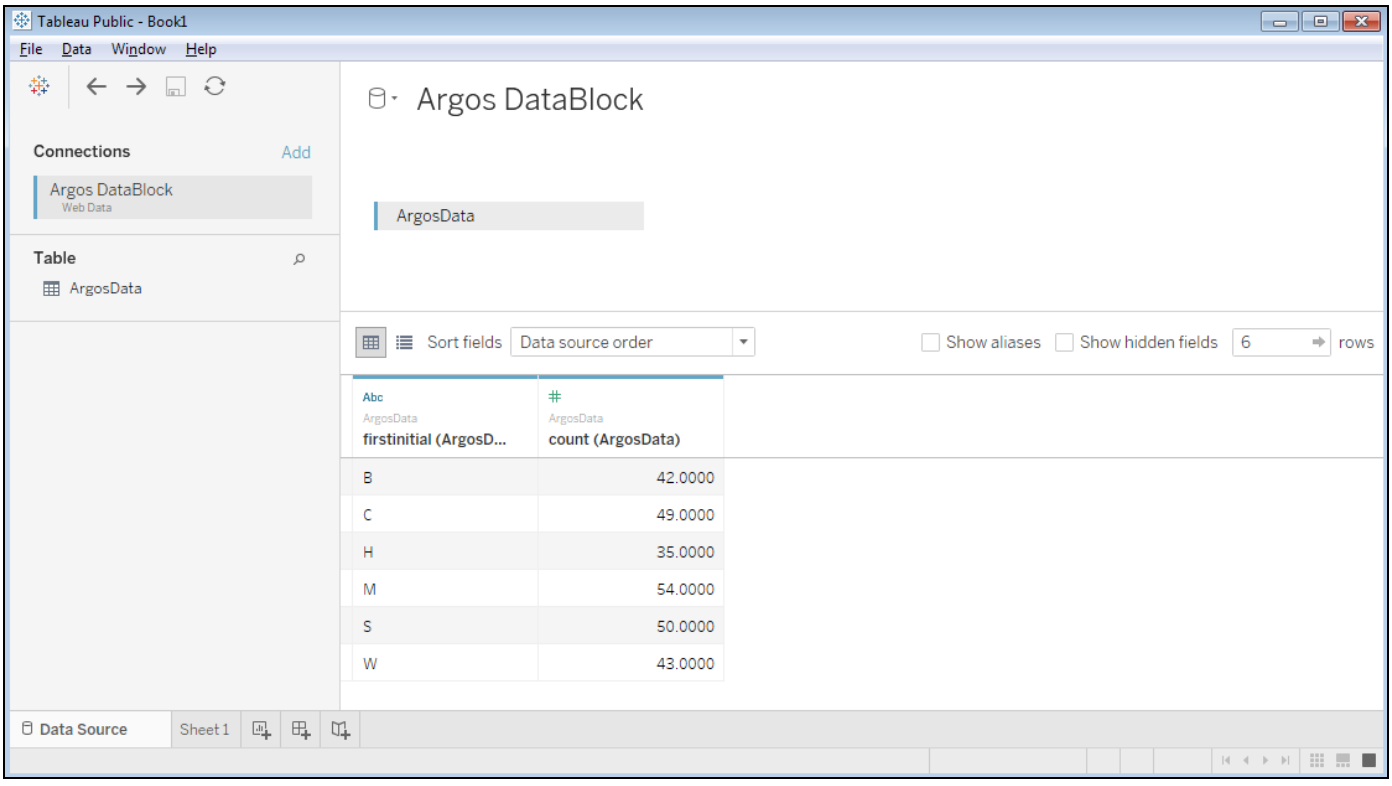


Press Enter after typing in the URL. You will see the login screen for the Argos DataBlock Connector.



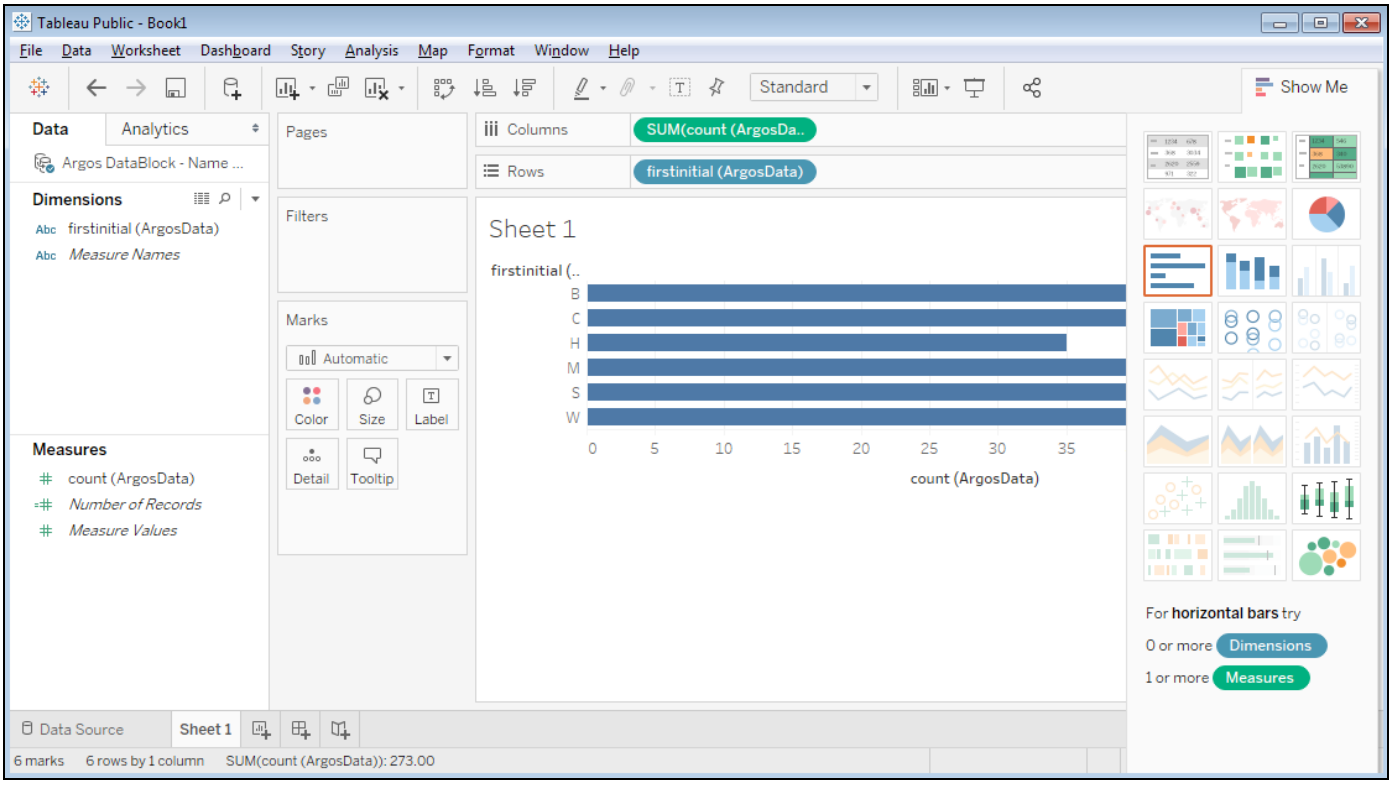
Follow the instructions in the previous section for signing in, retrieving the list of parameters, and entering parameter values.

After clicking Get Report Data, the dialog will close and you will see the new Web Data Connector added in Tableau.



If desired, you can rename the connection from "Argos DataBlock" to something more descriptive, such as the name of the particular DataBlock, by clicking within the name field.

You can now manipulate the data for this connector as you would with any other in Tableau. To get started, click on Sheet1 at the bottom of the screen to open a blank worksheet. You can then add visualizations and manipulate the dimensions and measures available through the connector.



For more information on how to use Tableau Public, please refer to the Tableau Help.

Getting Help

For information on using the software, please refer to the in-product Help, which contains detailed information on all aspects of the product.

If you are having problems with the installation or configuration, you can search our support site, which includes a knowledge base of common issues. If you are unable to find the solution, submit a HelpDesk request with a detailed explanation of the problem you are experiencing.

Please do not hesitate to contact the Evisions HelpDesk if any questions or problems arise. We are here to help you and want to ensure your success.

If you find that areas of this documentation could benefit from additional detail or clarification, please let us know. We are constantly trying to improve the installation process to make it as easy as possible.