

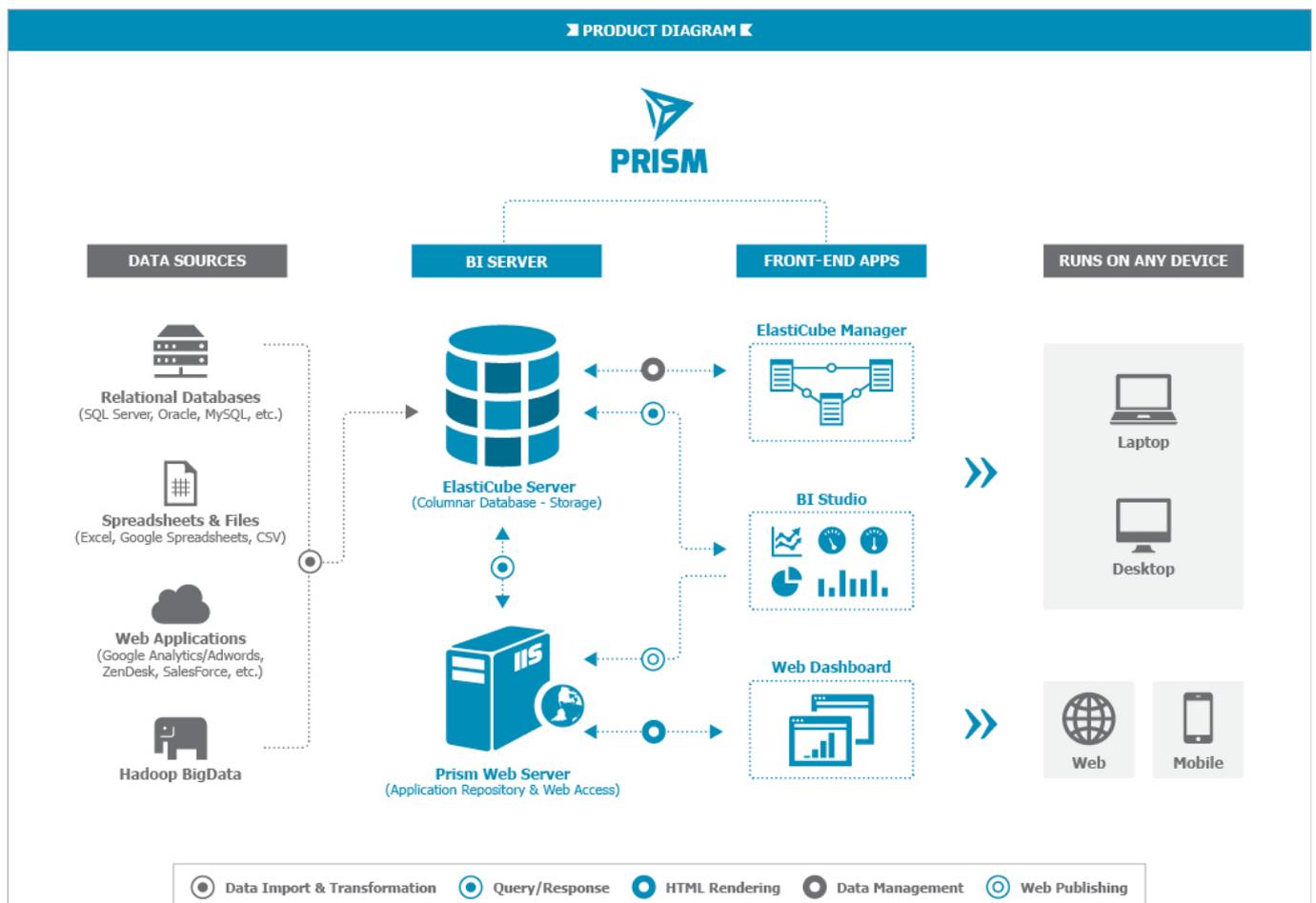


Sisense Prism V4.7 Documentation

Introduction

Sisense Prism is a product suite for building business intelligence solutions. Prism is a single integrated environment providing an extensive toolset to accomplish the following:

- **Analyzing data**
Formulate complex business queries using an intuitive visual interface, add custom calculations.
- **Visualizing data**
Use a wide range of data visualization widgets to visualize data.
- **Preparing the data for analysis**
Merge data from disparate sources, manipulate and cleanse data, full ETL capabilities.
- **Support for large data volumes**
Powered by ElastiCube technology, queries millions of rows of data in seconds.
- **Data Exploration and Ad Hoc Analysis**
Get insight into your data using a wide range of data exploration features.



For first time users, it is recommended to go over this documentation in the order it is presented. Later you can come back to specific sections as you require.

Prism Overview

Prism Documentation

Management and Administration

ElastiCubes are run and maintained in the Prism Server Console. Various administration functions are available to run, edit and manage ElastiCubes in the console. This section also details licensing, user management and security in Prism Server Console and Prism Web.

Prism Installation Guide

Installation Recommendations

Before installation it is recommend to uninstall all previous Prism products before installing the new version.

As best practice for any Platform upgrades, it is recommended first to install the program in a test environment and check all existing ElastiCubes and PSM files. Only once its has been validated that everything is working well, it is suggested to install the new version in a production environment.

Prism can either be deployed automatically using default settings or custom configuration settings can be supplied by clicking 'Change Features' during installation.

Default Installation

This option installs *All* prism products with default configuration, this includes:

- BIStudio
- ElastiCube project manager
- Server Console
- ElastiCube server 64b or 32b according to your machine configuration
- IIS express as the web server hosting PrismWeb platform

In the Installation window, enter your user name and login details to confirm licensing and begin the installation process (note this may take a few minutes to verify details).



SiSense Prism Activation

Sign In [or create a new account](#)

[Forgot Password?](#)

[Behind a Firewall?](#)

[Got a Serial Number?](#)

 Sign In

[Contact Support](#)

Once you have entered login details select *“Get Everything”* for default installation



Custom Installation

You can specify the following settings in the custom installation

i. ElastiCube Server

The 'Long index edition' installation can be selected. This is only recommended in the following instances:

Installing the “Long Index” edition is required only in if one of the following scenarios is true:

- Your data contains tables with over 4 billion rows.
- Your data contains columns with string (text) data with over 4GB of unique values.

It is not recommended to install the “Long Index” edition unless one of these conditions apply, regardless of how much data you have.

ii. Desktop Applications

The following applications can be included or excluded from installation by enabling the

associated checkbox.

- BI Studio
- ElastiCube Manager
- ElastiCube Server Console

iii. Web Server Type

If you do not select a custom Prism Web installation by default IIS Express will be installed.

The Prism Web port and website name will need to be specified. Typically if no other sites are running on the machine port '8081' may be used.

There are three options for the server type.

1. *Trial*

- This automatically deploys the necessary settings, see above.

2. *Microsoft IIS (Recommended for production environments)*

- The Prism Web site name and port may be specified.
- Override existing site name in case one used with the same name.

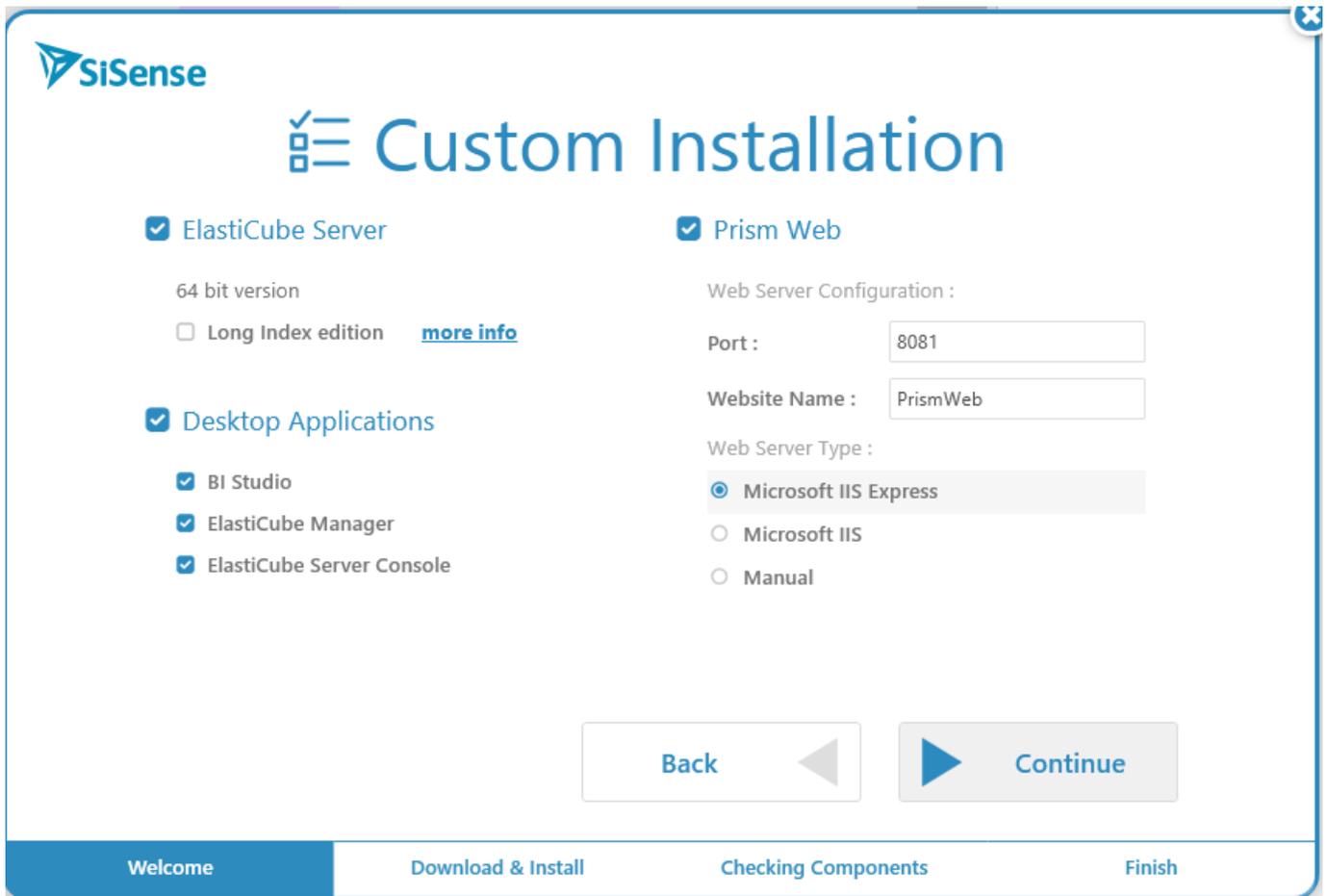
3. *Manual*

- When upgrading the installation will keep the existing site details and configuration in the IIS.
- Need to configure the physical path to the new installation folder, (Example: c:\Program Files\Sisense\PrismWeb)

Please open ports 811,812 in the ElastiCube machine (for remote access to it)

It is recommended to first install in a test environment and check all of your ElastiCubes and PSM files. Only when you are certain that all files are working as expected install the new version in your production environment.

In the Installation window, Select "*Custom Installation*" for default installation



Upgrade from Older Version

Before Installation

It is recommended to backup Prism Web backend data, follow the steps below in order to perform the backup:

1. Copy and backup the entire original prism web installation folder (Program Files \Sisense\PrismWeb). In older versions the folder is 'Program Files(86)\...'
2. Install the new version of Prism Web.
3. You will now need to check your Prism Web data provider

Go to the OLD copy of your prism web folder (App_Data\Configurations) and open the db.config file.

If the DataProvidersName is different from 'PrismEntitiesCompact' () then copy and paste the old db.config file into the new prism web folder into 'Program Files\Sisense\PrismWeb\AppData\Configurations'

If the data provider is compact edition then in the OLD prism web folder copy the Prism.sdf file

(located in PrismWeb\App_Data\MSSQL\Compact).

Go to your new installation,
'Program Files\Sisense\PrismWeb\App_Data\MSSQL\Compact' and rename the existing Prism.sdf file to Prism_Original.sdf file.
Then paste the old Prism.sdf file into App_Data\MSSQL\Compact.

4. Paste the old 'Resources' folder

Go to old installation folder and copy the 'Resources' folder (PrismWeb\Resources).
Go to your new Prism Web folder (C:\Program Files \Sisense\PrismWeb\Resources) and paste the old resources folder.

Installation Behind a Firewall

Bootstrap Version

The standard installation requires internet access in order to download the necessary components. In instances where a machine is behind a firewall, or does not have internet access the following full version can be downloaded and transferred to the machine. The bootstrap version can be downloaded from the link below. It must be replaced with the current version of Prism for example if the latest version number is 4.5.1066 the link .../Sisense%20Prism.4.5.1066.exe

<http://download.sisense.com/PrismInstallations/Full/Sisense%20Prism..exe>

In addition the following supporting application may be needed

- [.Net 4.0](#)
- [vcredist x86](#)
- [vcredist x64](#)
- [IISExpress 7.5 \(only for XP and 2003 server\)](#)
- [IISExpress 8.0 x86](#)
- [IISExpress 8.0 x64](#)

Capacity and System Requirements

Introduction

Sisense Prism easily scales up to billions of records.

When running Prism Server on a PC with the recommended configurations, typical query response times of just a few seconds can be expected. Of course, extreme scenarios may require additional resources.

The system requirements specified below are for the machine functioning as a server.

Prism users access reports and interactive dashboards using a Web browser or Windows client, and do not require any particular PC configuration.

Note: 64-bit Windows can be Windows 7, Windows 8, Windows Server 2003, Windows Server 2008 & Windows Server 2012.



To Analyze a Few Million Records

A few users

- *Minimum:* Standard desktop PC with 64-bit Windows, 8 GB of RAM, 4 logical cores
- *Recommended:* Standard desktop PC with 64-bit Windows, 16 GB of RAM, 4 logical cores

Tens of users

- *Minimum:* Standard desktop PC with 64-bit Windows, multi-core CPU, 16 GB of RAM, 4 logical cores
- *Recommended:* Standard desktop PC with 64-bit Windows, multi-core CPU, 32 GB of RAM, 8 logical cores

To Analyze a Hundreds of Millions of Records

A few users

- *Minimum:* Standard desktop PC with 64-bit Windows, 32 GB of RAM, 8 logical cores
- *Recommended:* Standard desktop PC with 64-bit Windows, 64 GB of RAM, 16 logical cores

cores

Tens of users

- *Minimum:* Standard desktop PC with 64-bit Windows, multi-core CPU, 64 GB of RAM, 8 logical cores
- *Recommended:* Standard desktop PC with 64-bit Windows, multi-core CPU, 128 GB of RAM, 16 logical cores

To Analyze Billions of Records

A few users

- *Minimum:* Standard server PC with 64-bit Windows, 256 GB of RAM, 12 logical cores
- *Recommended:* Standard server PC with 64-bit Windows, at least 512 GB of RAM, 24 logical cores

Tens of users

- *Minimum:* Standard server PC with 64-bit Windows, multi-core CPU, 512 GB of RAM, 24 logical cores
- *Recommended:* Standard server PC with 64-bit Windows, multi-core CPU, at least 512 GB of RAM, 32 logical cores

Getting Started Guide

INTRODUCTION TO THE GETTING STARTED GUIDE

Introduction

This guide provides a high level overview of the steps needed to create your first dashboard and share it with others. It assumes no prior knowledge and points to supporting documentation in the Sisense User Guide.

The aim of this guide is to teach a new user in a practical manner how to connect to their required data, visualize the data in charts and pivots, create interactivity to aid analysis and publish a dashboard to the web in order to share with others.

There are three main steps to create a dashboard. On the left menu panel click each stage and complete the associated steps.

1. **Step 1:** Connect to data in Prism ElastiCube Manager
2. **Step 2:** Create a dashboard in BI Studio
3. **Step 3:** Publish to Prism Web

Objectives

This guide is designed to help new users in the following manner:

- Understand the process of connecting to data as well as designing and sharing a dashboard.
- Show how Prism applications work together.
- Highlight the key components and elements used in design.

Business Story

This data analysis described in this guide investigates a sales process visualizing the number of sales and units sold over a few months in several formats. A dynamic filter is added to segment by client.

Requirements before Starting

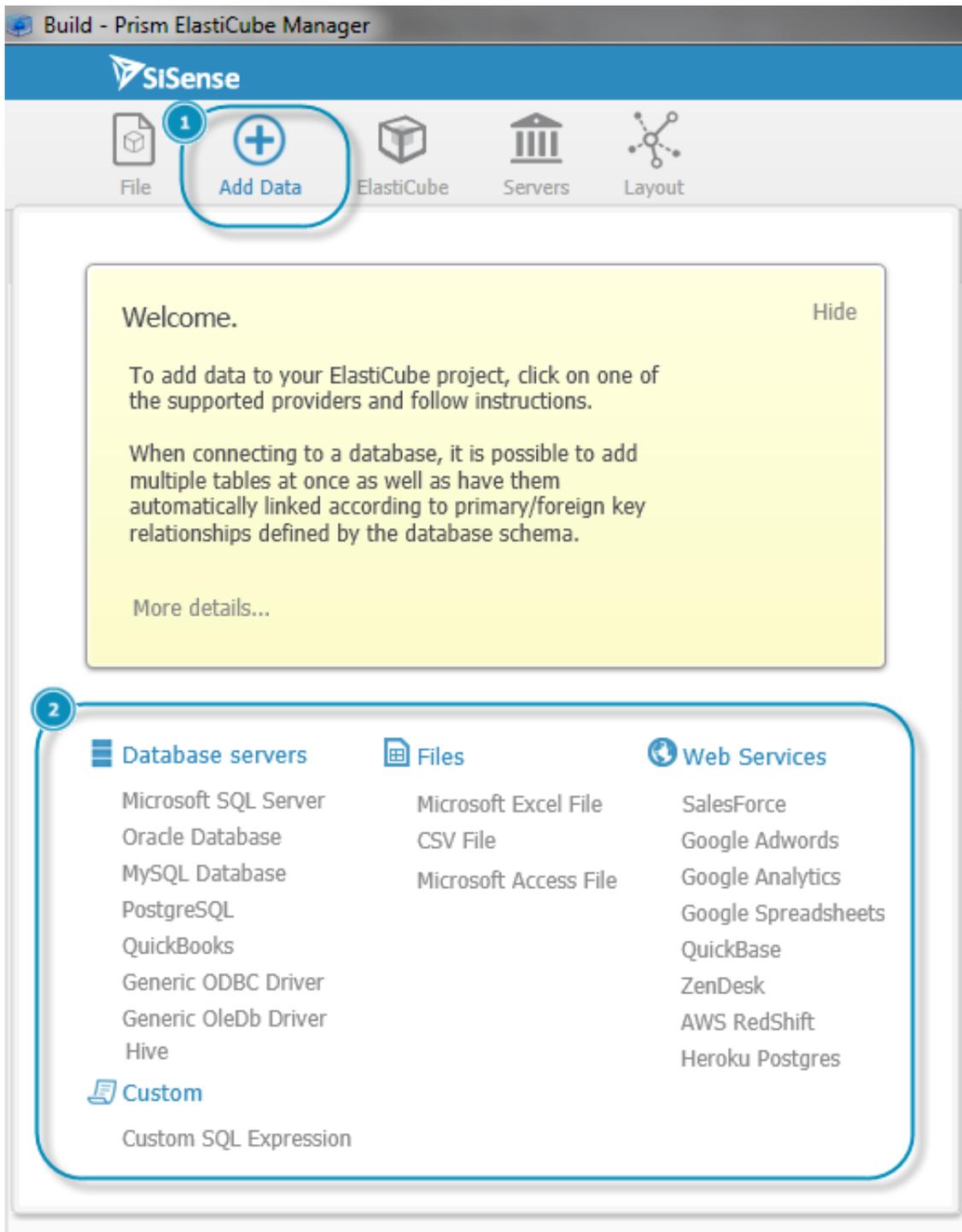
1. Installation of Prism
2. Download this [Excel File](#) which contains all necessary data.

STEP 1: Connect To Data In PrismElasticube Manager

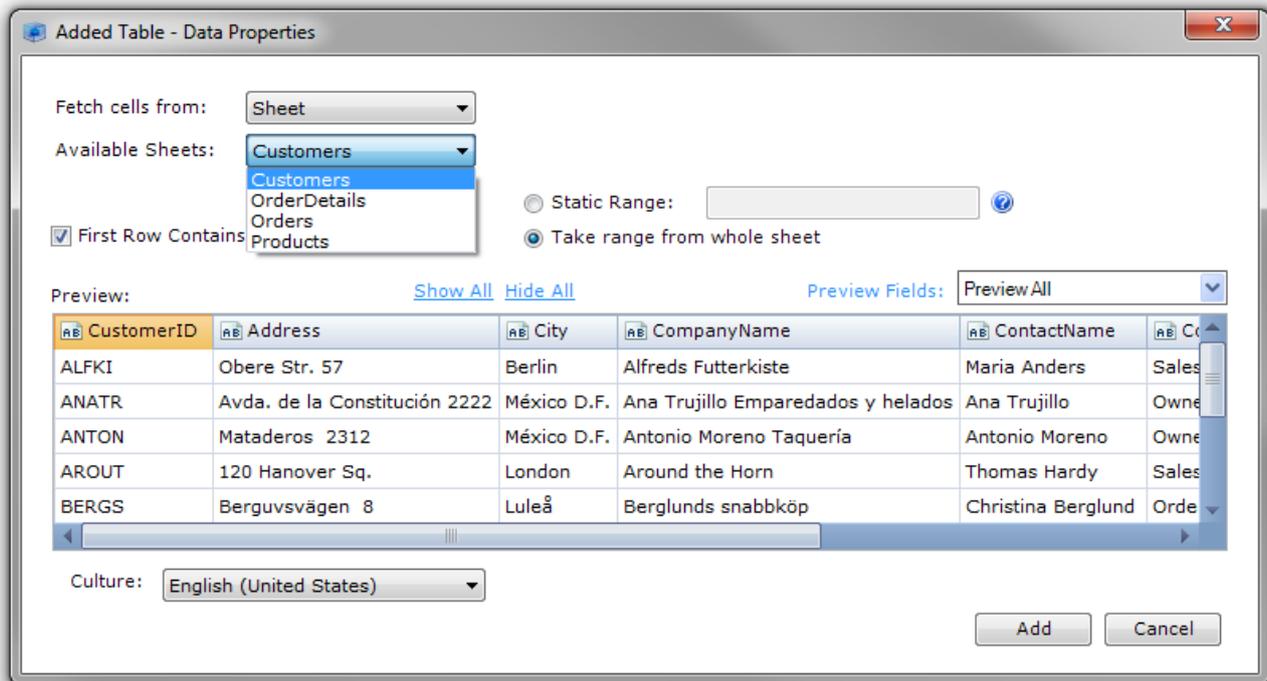
1.1. Connecting to a Data Source

Data forms the basis of dashboards and can be imported from traditional database, data files, web services or custom tables.

1. Download the Getting Started Tables Excel worksheet and save it to the desktop. Download the [Excel File](#)
2. Open Prism Elasticube Manager
3. On the top right click File>>New Elasticube File. Name it "Schema1" and click the tick icon.



1. A new blank schema will open.
2. Click the 'Add Data' icon then under the files category select 'Getting Started Tables.xlsx' file
3. A new window will pop up. Next to the option for 'Available Sheets:' select the Customers sheet, and click Add. A new table with relevant fields will be added to the schema called 'Customers'.

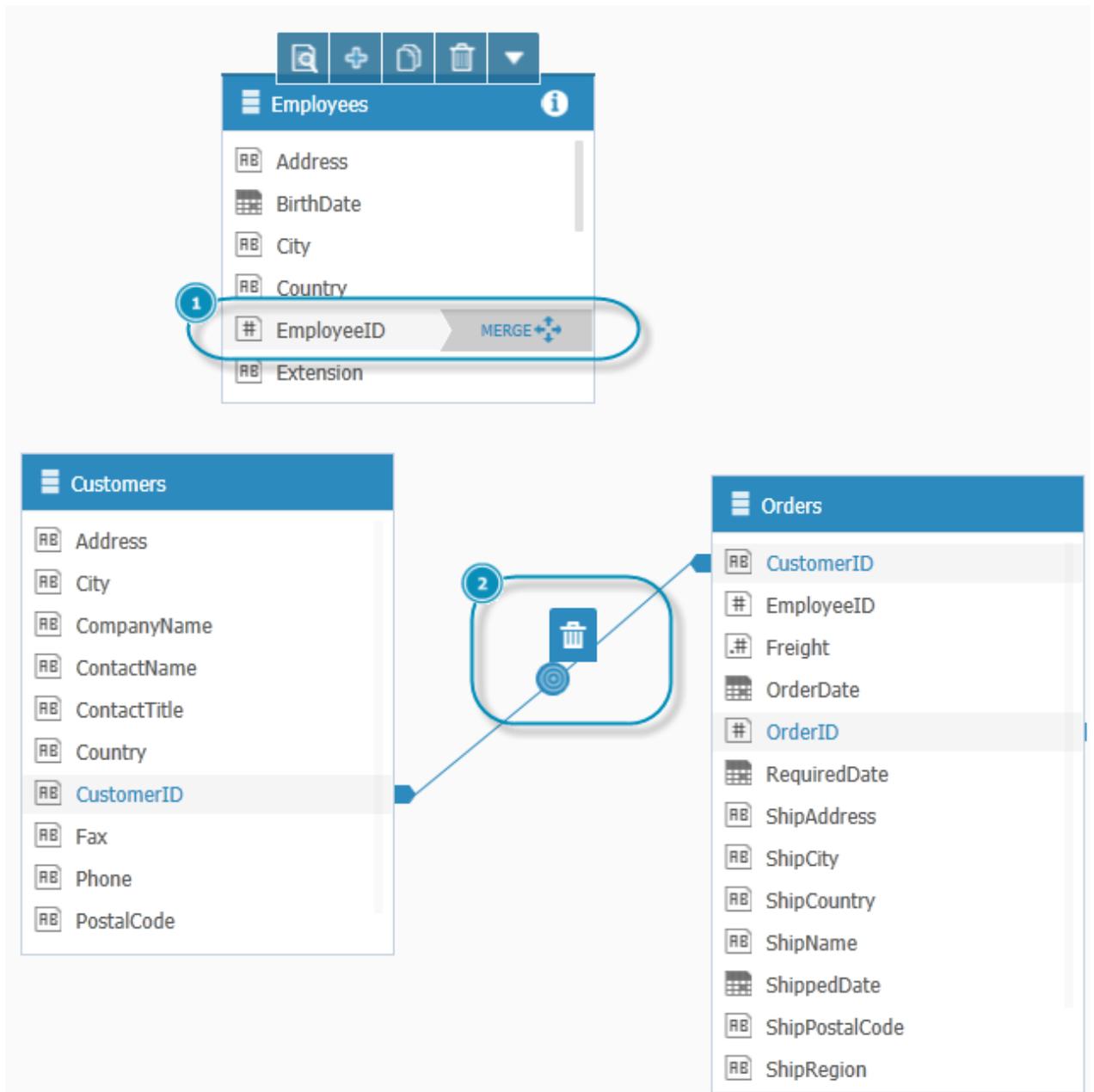


1. Repeat steps 5 and 6 above to import the 'OrderDetails', 'Orders' and 'Products' sheets from the excel file. You can select the appropriate sheet by clicking on the 'Available Sheets' drop down in the top left of the Data Properties window (see image above)

1.2. Creating Relationships between Tables (Data Sources)

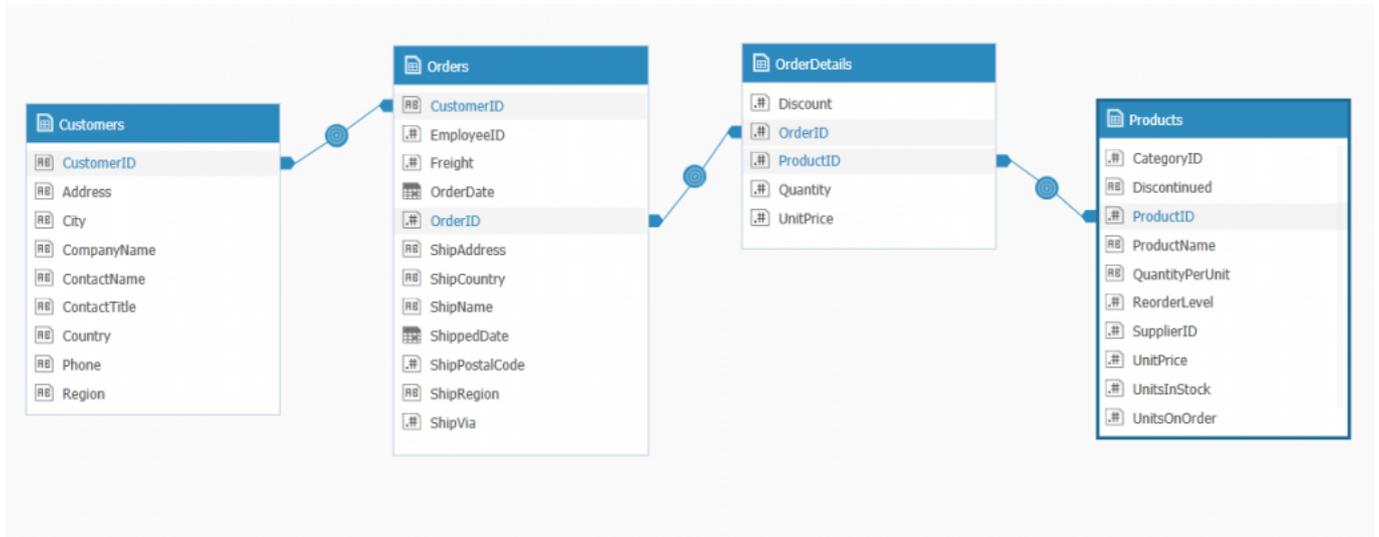
Relationships enable data from two different sources to be linked and used together.

1. To create a relationship between tables, click on the 'merge' button located next to the relevant field and while holding down the mouse button drag to the corresponding field. A line will appear and will turn to green when a relationship can be created between the tables.



2. Create the following 3 relationships, the link below refer to a table and field.

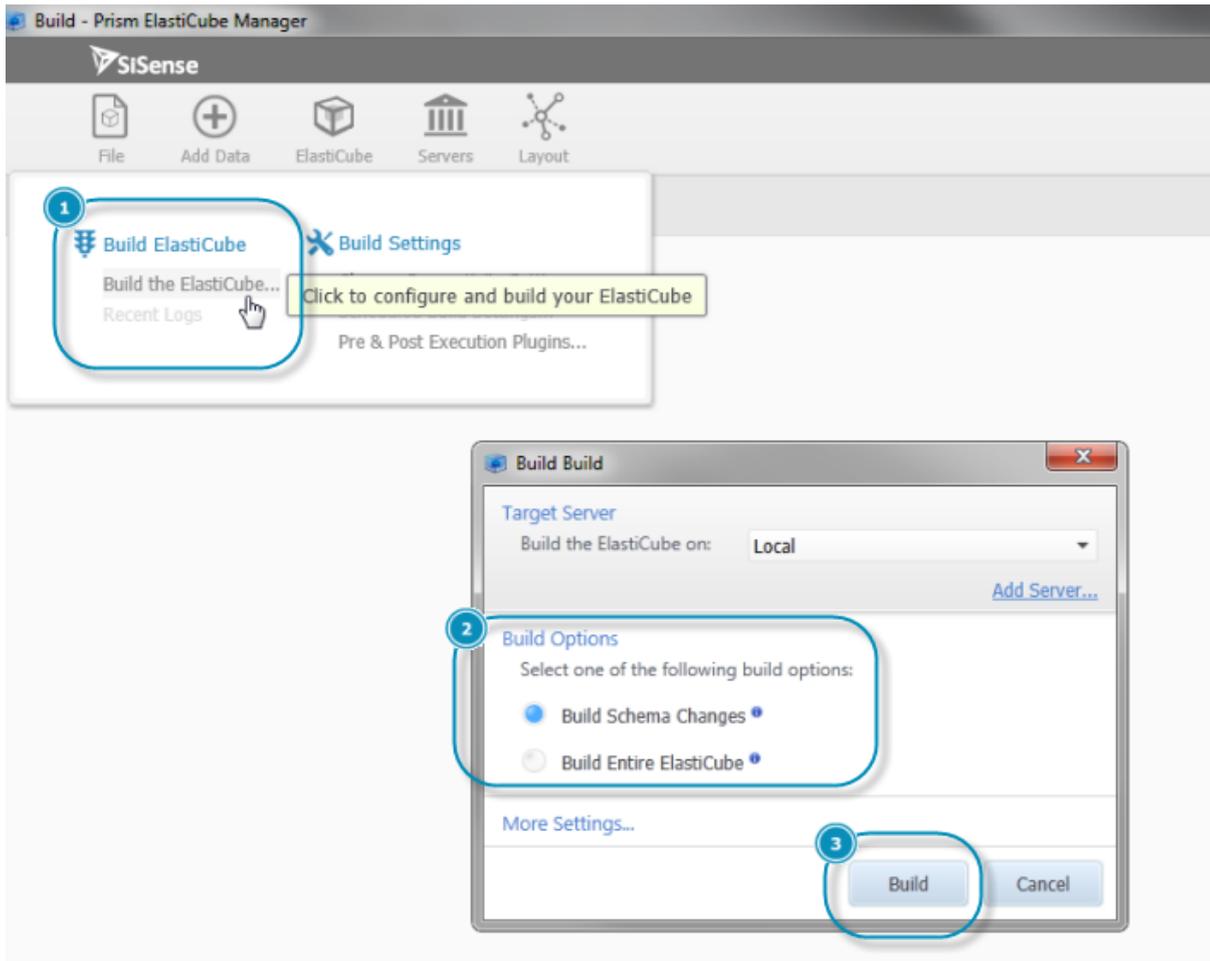
- Orders.OrderID to OrderDetails.OrderID
- OrderDetails.ProductID to Product. ProductID
- Orders.CustomerID to Customers.CustomerID



1.3. Importing Data for the Dashboard

In order to query the data quickly a build needs to be done, which structures and index's the data for optimal performance.

1. On the top right click the ElastiCube icon and select 'Build the ElastiCube'
2. A window will appear will appear with configuration options. Ensure the check-box next to 'Build the Entire ElastiCube' is enabled and click the 'Build' button.



1. In the bottom of the screen a panel will open which describes the steps taken to structure the data. Enable the check box next to the 'Build Entire ElastiCube' option. Once completed a message will appear 'Build successfully ended'
2. The data is now ready to be used in a dashboard.

Step 2: Create a dashboard in BI Studio

Introduction

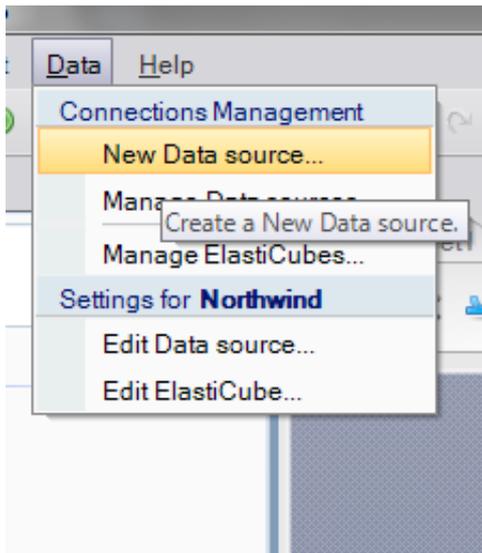
Now you can leave the ElastiCube Management environment and start building your dashboard!

Click on Start>All Programs>Sisense>Prism BI Studio

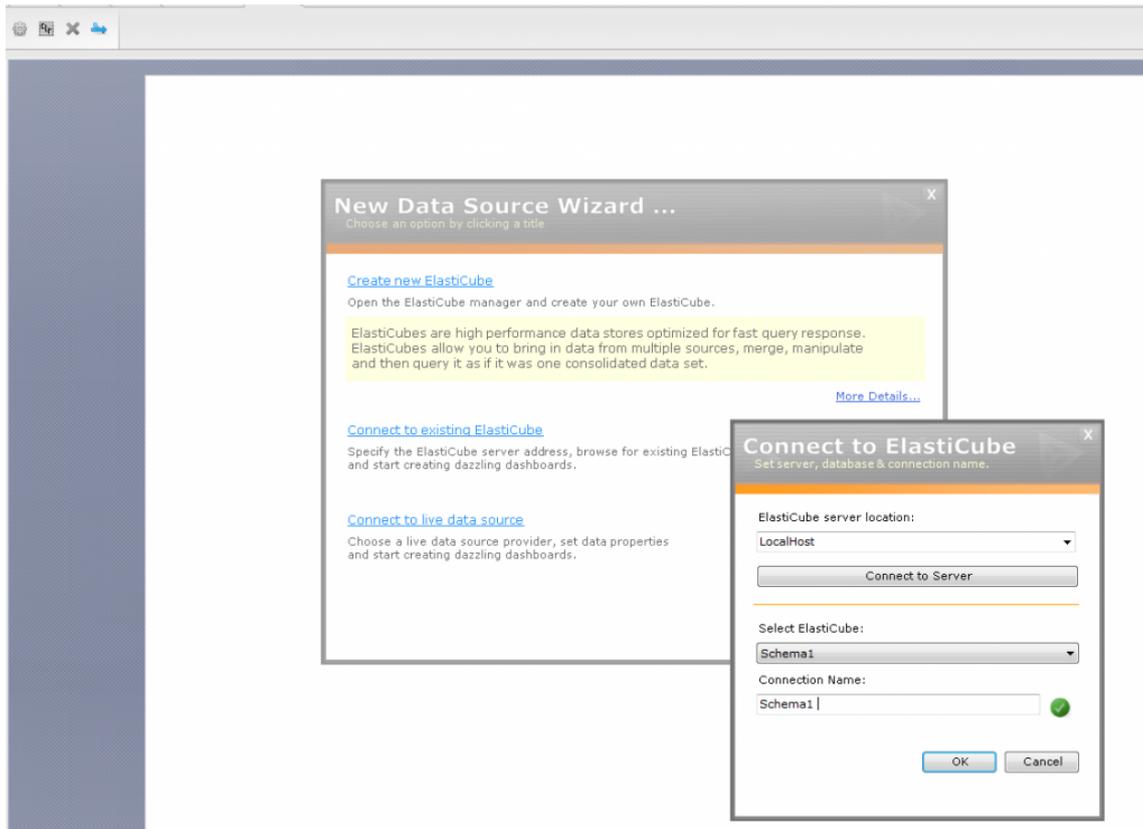
This should load our BI Studio

2.1. Bi Studio Connects to Data Source

1. Open BI Studio
2. In the pop-up window click 'New Document' in the bottom right of the window (alternatively click File>>New). A blank canvas will open on the screen.
3. On the top menu on the left click 'New Data Source'. In the pop-up window select 'Connect to Existing ElastiCube' a new window will open.



4. Click the 'Connect to server' button and next to 'Select ElastiCube' choose the ElastiCube just created called Schema1.



5. The data within the ElastiCube will open on the left in the Data Browser panel. This data can be bonded to widgets such as graphs and pivots to create visualizations and data analytics.

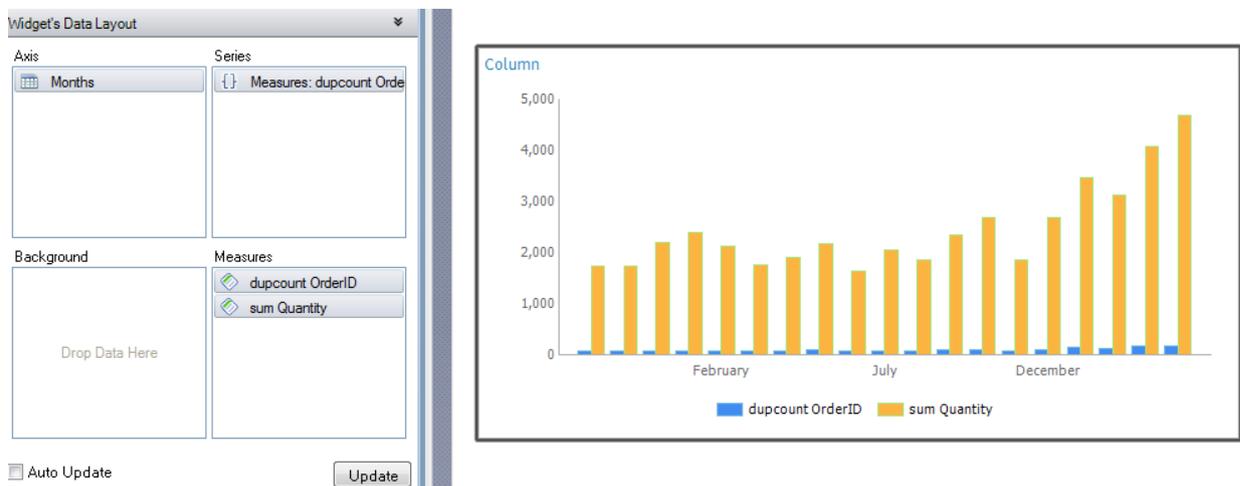
2.2. Create a Pivot Table

1. Open the widgets menu on the bottom left.
2. Under the heading 'Grids' drag and drop the 'Pivot' icon onto the sheet.
3. Click the new pivot table then open the 'Data Browser' on the left to display the 'Widgets Data' Layout panels. The data layout determines how data is displayed in the associated widget.
4. In the data browser scroll to the 'Orders' table and expand the table to see all dimensions. Expand the 'OrderDate' dimension then expand the 'Calendar category'.
5. Drag and drop the Months component into the 'Rows' panel of the data layout.
6. Go back to the data browser and expand the 'OrderDetails' table, drag and drop the 'OrderID' dimension into the measures panel of the data layout and select 'Count Duplicates'. Then drag and drop the 'Quantity' dimension into the measures panel of the data layout and select 'Sum'
7. Click the 'Update' button in the data layout. The pivot table will refresh with data.

More details on settings can be found here:

2.3. Create a Chart

1. Open the widgets menu on the bottom left.
2. Under the heading 'Common Charts' drag and drop the 'Column' icon onto the sheet, next to the pivot table.
3. Click the new chart then open the 'Data Browser' on the bottom left to display the Widgets Data Layout panels.
4. In the data browser scroll to the 'Orders' table and expand the table to see all dimensions. Expand the 'OrderDate' dimension then expand the Calendar category. Drag and drop the Months component into the 'Axis' panel of the data layout.



5. Go back to the data browser and expand the 'OrderDetails' table, drag and drop the 'OrderID' dimension into the 'Measures' panel of the data layout and select 'Count Duplicates'. Then drag and drop the 'Quantity' dimension into the measures panel of the data layout and select 'Sum'
6. Click the 'Update' button in the data layout. The chart will refresh with data.

Details of chart settings can be found here:

2.4. Create a Member Picker

1. Open the widgets menu on the bottom left.
2. Under the heading 'Filtering' drag and drop the 'Members Picker' icon onto the sheet above the pivot and chart.
3. Right click on the new member picker then open the 'Data Browser' on the left to display the Widgets Data Layout panels.
4. In the 'Data Browser' scroll to the 'Customers' table and expand the table to see all

dimensions. Drag and drop the 'ContactName' dimension into the items panel of the picker's Data Layout.

The screenshot shows a window titled "Widget's Data Layout" with a dropdown arrow on the right. It is divided into two main sections: "Items" and "Background".

- Items:** A list box containing one item, "ContactName", with a small icon to its left.
- Background:** A large empty rectangular area with the text "Drop Data Here" centered inside.

At the bottom of the window, there is a checkbox labeled "Auto Update" which is currently unchecked, and a button labeled "Update".

A horizontal dropdown menu with a black border. The text "ContactName" is displayed in the center, and a blue downward-pointing arrow is visible on the right side of the menu.

- 5.
6. Making a selection from the member picker will now filter the data in the pivot and chart.

Details of the member pickers setting can be found here:

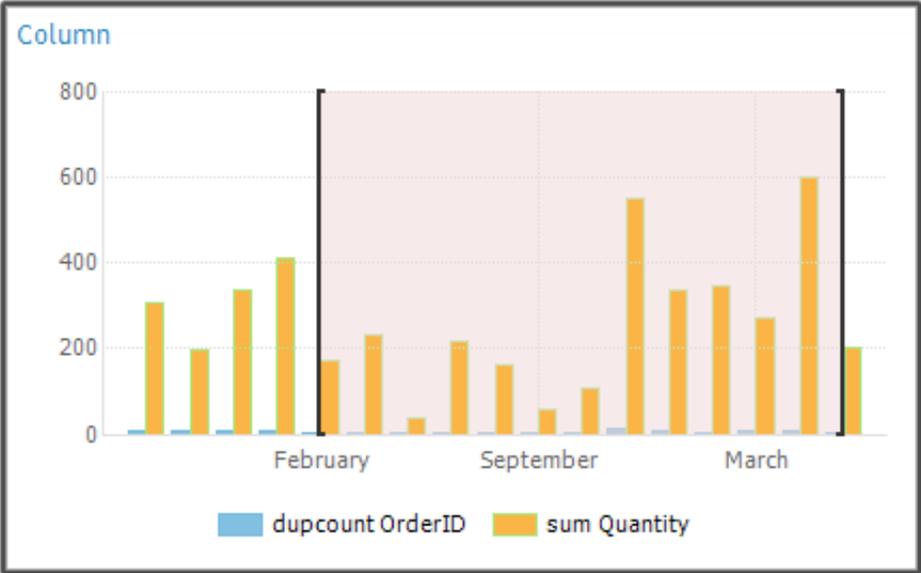
2.5. Create a Connection between the Pivot and Graph

1. Click the column chart and move the mouse cursor to the top of the widget till the menu panel appears, click the icon for 'Widgets Selection Mode'.
2. Right click the pivot table then go to Data>Widget Interconnectivity> Available Widgets - highlight and click the option referencing the chart. Click 'Apply'.

The screenshot shows a software interface with a pivot table on the left and a column chart on the right. The pivot table has columns for 'Months', 'dupcount OrderID', and 'sum Quantity'. The column chart displays data for 'February', 'July', and 'December'. A context menu is open over the pivot table, with 'Widget Interconnectivity' selected. A sub-menu is open over the chart, with 'Available Widgets' selected. The 'Available Widgets' sub-menu shows 'Customers.ContactName' and 'ContactName' selected. The 'Apply' button is visible at the bottom of the sub-menu.

Months	dupcount OrderID	sum Quantity
October	73	1,738
November	66	1,735
December	81	2,200
January	85	2,401
February	79	2,132
March	77	1,770
April	81	1,912
May	96	2,164
June	76	1,635
July	77	2,054
August	84	1,861
September	95	2,343
October	106	2,679
November	89	1,856
December	114	2,682
January	152	3,466
February	122	3,115
March	178	4,065
April	180	4,680

-
- 3. Hover the mouse cursor over the chart, click on the 'Widgets Selection Mode', then left click and highlight a range on the chart. The selected period will filter the values displayed in the pivot table.

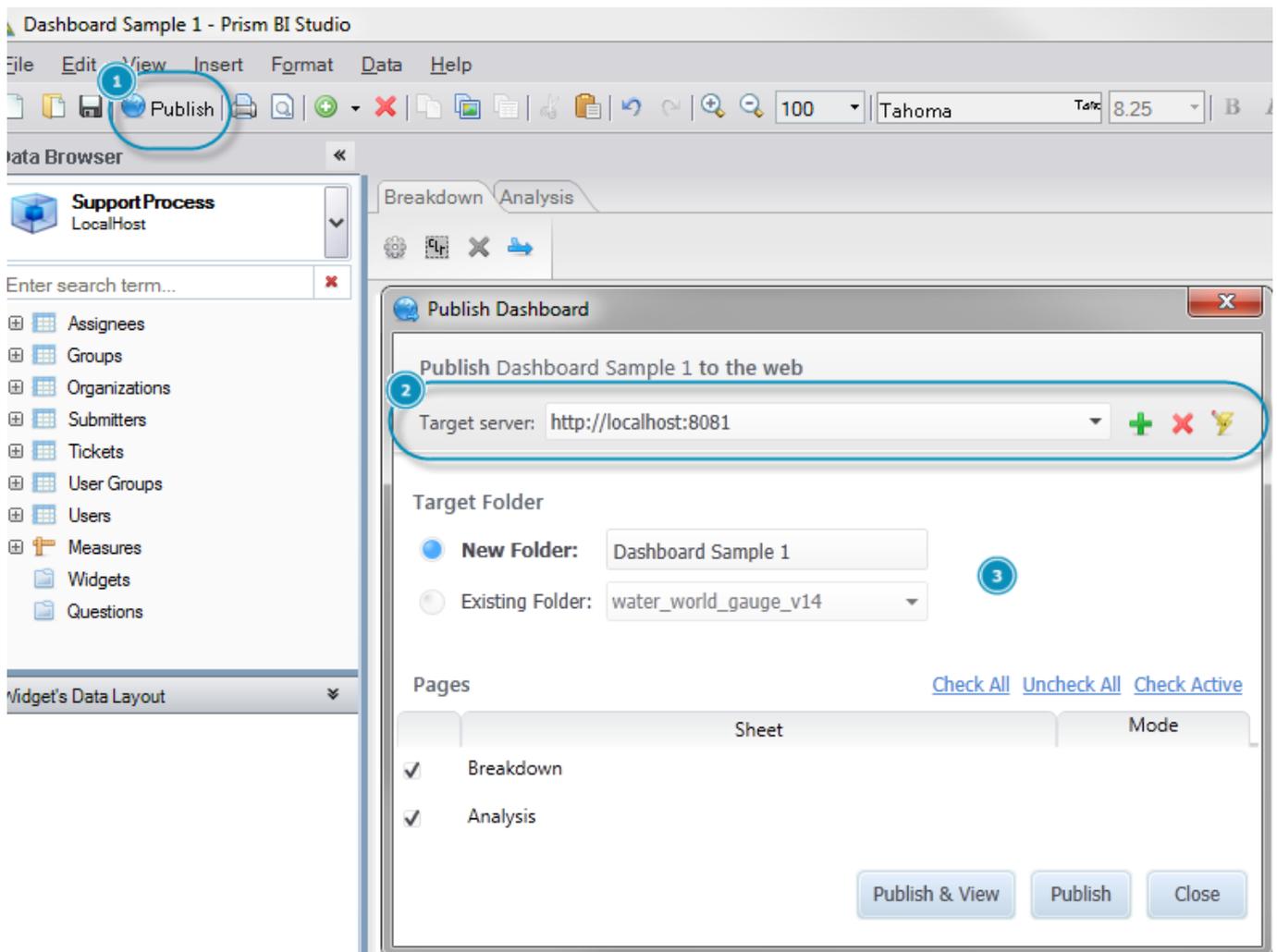


Step 3: Publish to Prism Web

3.1. Publish to the Web

1. On the top main menu panel click File -> Publish
2. A popup will appear with details of the location of the web environment where you can publish to an existing web location or add additional location.
3. Select the default location from the Target Server drop down.
4. Under the Target Folder heading in the New Folder input box type 'MyFirstDashboard'
5. Select all the sheets to publish
6. Click the 'Publish and View' button

Note



For more details on publishing to the web from BI Studio see:

3.2. Share the Dashboard

A dashboard can be shared with multiple users. Shared dashboards can be accessed via a web browser regardless of a user's location.

1. In Prism Web click the dashboards icon on the far right and select the relevant dashboard from the drop down menu.
2. Once the dashboard has loaded click the share icon on the top right.
3. In the input box enter the email address of all recipients with whom you wish to share the dashboard. Press enter after typing each email address. To delete an entered email address click the cross next to the recipients email.
4. Once you have entered all recipients click 'Save Changes'.
5. An email will be sent to the recipient with a link to access the dashboard.
6. Clicking on the link will open a web browser and direct the user to your dashboard. First time users will be prompted to choose a password before logging into the dashboard.

The screenshot shows a web interface with a top navigation bar containing 'admin', 'Dashboards', 'Share', and a settings icon. A modal dialog titled 'Share Dashboard' is open. At the top of the dialog is a yellow warning box: 'You are currently sharing dashboards using your IP Address. Your IP Address might change in the future, resulting in broken links.' Below this, the title 'Share Dashboard' is followed by a lock icon and the text 'Shared only with users listed below:'. To the right of this text is a link 'Make Public'. Below the text is a list of recipients, with one entry showing 'name' and 'name@email.com'. At the bottom of the dialog are two buttons: 'Cancel' and 'Save Changes'.

ElastiCube Manager

The V4 documentation of ElastiCube Manager has been merged with the V5 documentation of ElastiCube Manager.

Please follow [this link to the V5 ElastiCube Manager documentation](#), which is accurate for V4 as well.

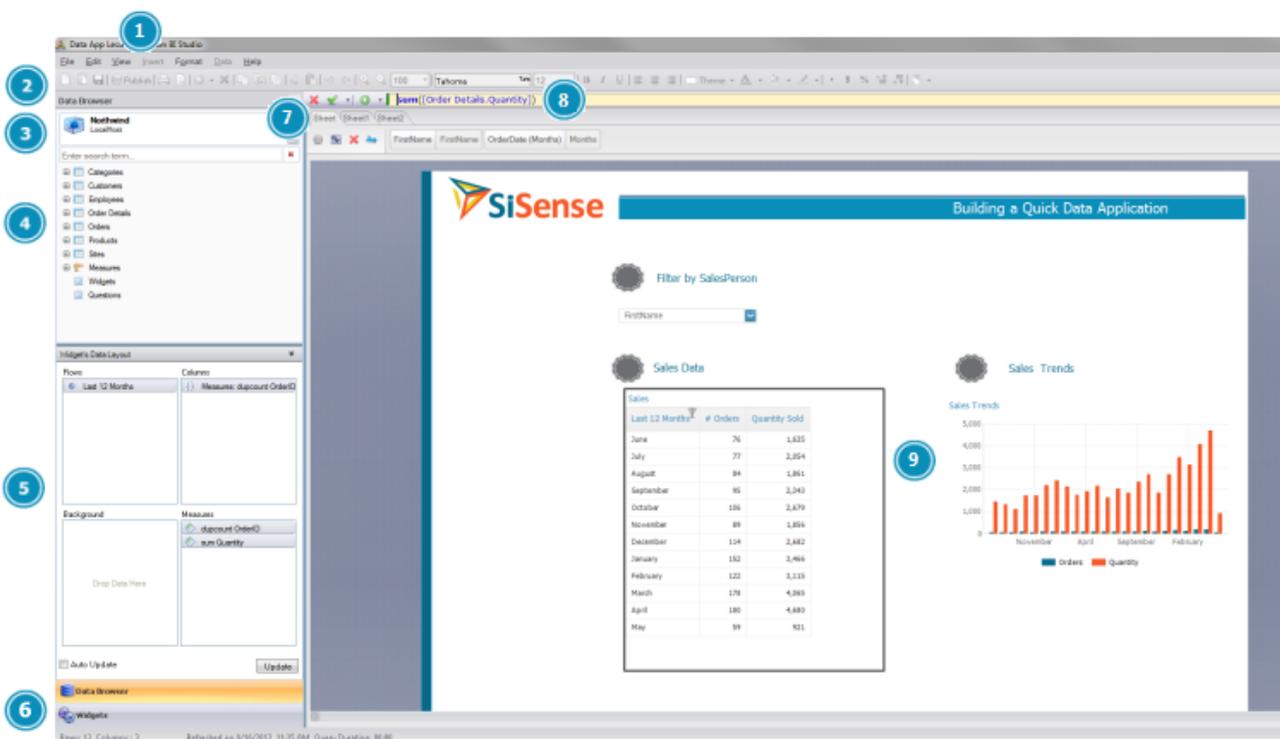
Prism BI Studio

INTRODUCTION TO BI STUDIO

Introduction

BI Studio is the main tool to create visual analysis and reporting solutions using ElastiCubes and live data. BI Studio provides a user interface to easily create complex and beautiful dashboards with no knowledge of databases and share them with others.

BI Studio Environment



1. Main Menu
2. Main Toolbar
3. Data Source Selection List
4. Data Browser (view associated tables and fields)
5. Widget Data Editor (add data to widget)
6. Show Widgets Panel
7. Sheets Tab
8. Measure and Formula Editor
9. Widgets

Video: Creating Your First Data Application

Defining the Dashboard's Data Source

CONNECTING TO EXISTING ELASTICUBES

Before getting started open Prism Server and ensure the relevant ElastiCube is running

1. Open BI Studio.
2. There are several options to connecting to an existing ElastiCube.

Data Browser Connection List

1. In the Data Browser panel on the left click the Data Sources Selection List.
2. All available ElastiCubes will appear in the list.
3. Click the ElastiCube you would like to use in the list then click on the associated ElastiCube.
4. All tables and dimensions of the selected ElastiCube will load in the Data Browser.

Recently Used Connection List

1. In the Data Browser panel on the left all used ElastiCubes are listed under 'Recently used data sources'
2. Click the ElastiCube you would like to use in the list.
3. All tables and dimensions of the selected ElastiCube will load in the Data Browser.

Data Connection Menu

1. In the Data Browser panel on the top left under 'Data Management' click on 'New Data Source'. Alternatively you can click on the top menu panel 'Data'>'New Data Source' A popup will appear.
2. Click 'Connect to existing ElastiCube'
3. The following details will need to be entered.
4. ElastiCube server location: This is the machine ip or address on which the ElastiCube is hosted. To connect to ElastiCube on your own machine enter 'localhost'
5. Click 'Connect to Server'
6. A list of all available ElastiCubes at that location will be populated. Select the ElastiCube you wish to use and click
7. All tables and dimensions of the selected ElastiCube will load in the Data Browser on the left.

Changing an ElastiCube IP in BI Studio

There may be instance when a ElastiCube has been moved to a different location. The steps below describe how to reconnect to the ElastiCube from the dashboard.

1. In the Data Browser panel on the left click the Data Sources Selection List.
2. All available ElastiCubes will appear in the list.
3. Click the ElastiCube you would like to use in the list then click on the associated ElastiCube.
4. In the Data Browser right click on the ElastiCube name and select 'Edit Data Source'.
5. A pop-up window will appear. In the 'ElastiCube server location' enter the ip of the machine hosting the ElastiCube and click 'Click to test Connectivity' to ensure a connection can be made.
6. Click 'Ok' to save changes.

Browsing and Managing Metadata

Metadata refers to the data contained within an ElastiCube or live data source. Metadata is displayed on the Data Browser panel to the left as dimensions and measures.

Browsing and Viewing Metadata

The data set associated to a dimension can be browsed by following the steps below:

1. Click on the relevant table node to explore its associated dimensions.
2. All existing dimensions contained in the node will appear in a list.
3. Clicking on a relevant dimension to browse associated data.
4. Click on the dimension name to list all data contained within the dimension.



Tip:

Browsing and Viewing Filters

A filter associated with a dimension can be browsed by following the steps below:

1. Click on the relevant table node to explore its associated dimensions.
2. All existing dimensions contained in the node will appear in a list.
3. Click on a relevant dimension to browse associated filters.
4. Click on the Filter node to view filters associated with the dimension.
5. Click on the relevant filter to return the data set determined by the filter.
6. Right click on the filter to access the following menu items:
 - **Edit:** View and update the filter setting.
 - **Rename:** Edit or rename the filter.
 - **Duplicate:** Duplicate the filter.
 - **Delete:** Delete the filter.
 - **Refresh:** Refresh underlying result set generated by the filter.
 - **Select Members:** Select all results generated by the filter.
 - **Aggregate:** Aggregate the results set generated by the filter.

Browsing and Viewing Measures

Aggregation associated with a dimension can be browsed by following the steps below:

1. Click on the relevant table node to explore its associated dimension measures.
2. All existing dimensions contained in the node will appear in a list.
3. Click on a relevant dimension to browse associated measures.
4. Click on aggregations to list all standard measures associated with the dimension.
5. The list below explains the relevant aggregations.

Aggregation	Explanation	Applicable Fields
Sum	Calculates the total of values	Fields containing numeric values
Average	Calculates the average value	Fields containing numeric values
Minimum	Calculates the minimum value	Fields containing numeric values
Maximum	Calculates the maximum value	Fields containing numeric values
Count	Counts the number of unique values	All Fields
Count Duplicates	Counts the number of rows (all values)	All Fields

Let's take a look at simple example. Consider the following tabular data:

First Name	Last Name	Month	Work Hours
John	Smith	January	120

John	Williams	January	130
John	Smith	February	110
John	Williams	February	135
Mary	Jones	February	140

Possible Dimensions for this data are:

- **First Name:** containing 2 Members, John and Mary
- **Last Name:** containing 3 Members, Smith, Williams and Jones
- **Month:** containing 2 Members, January and February

Possible Measures for this data are:

- **Total Work Hours:** Calculates the total work hours by applying the Sum aggregation on the values in the Work Hours field, resulting in $120+130+110+135+140=635$
- **Average Work Hours:** Calculates the average amount of work hours by applying the Average aggregation on the values in the Work Hours field, resulting in $(120+130+110+135+140)/5=127$
- **Number of Work Months:** calculates the number of different month values in the Month field, resulting in 2 (January and February).

The most powerful thing about Dimensions and Measures is that they can be easily combined to perform calculations over isolated data at any granularity. Combining the 'Month' Dimension with the 'Average Work Hours' Measure would calculate the average work hours per month.

Similarly, combining the 'Last Name' Dimension with the 'Number of Work Months' Measure would calculate the number of different months each 'Last Name' worked.

Binding Metadata to Widgets

Metadata such as dimensions and measures from the ElastiCube can be bound to widgets to create visualizations within a dashboard.

Each widget has its own data layout which determines how data is displayed in the associated widget. When a widget is selected, the widget Data Layout with the data configuration appropriate to that widget appears in the data browser panel on the left.

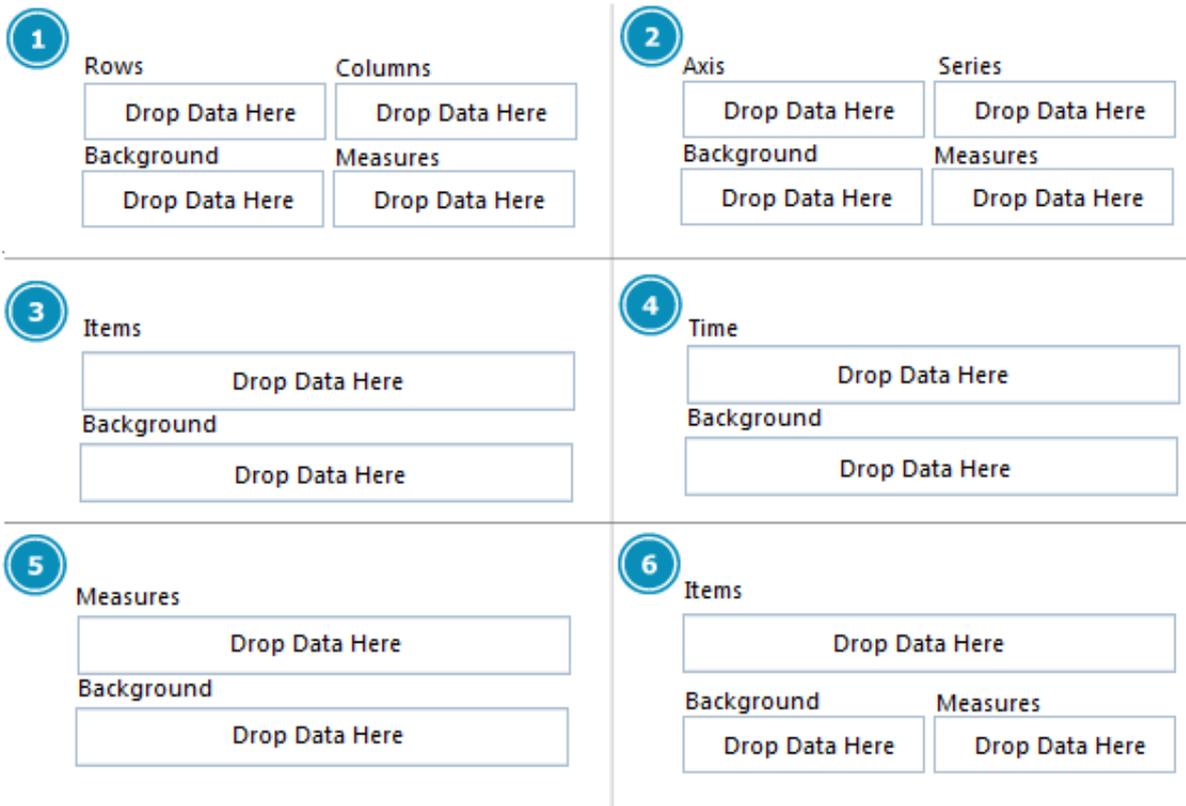


Illustration: Data layout for different widgets.

1. Pivot Table
2. Chart
3. Member Picker
4. Date-Range Picker
5. Indicator
6. List

Below is a description of each Data Layout components.

Data Layout	Description
<i>Row</i>	Members placed on this panel will make up the rows of the pivot table.
<i>Column</i>	Members placed on this panel will make up the fields (columns) of the pivot table.
<i>Items</i>	Members placed on this panel will make up the list items in lists or member picker.
<i>Axis</i>	Defines members that are displayed as labels in the chart.
<i>Series</i>	Defines members that are displayed as series in the chart.
<i>Time</i>	Members placed on this panel will make up the time range of the date picker or indicator.
<i>Measures</i>	Members placed on this panel will make up the fields (columns) of the pivot table.
<i>Background</i>	The members that slice the data, but will not show as rows or fields.

Data Layout	Widget				
	Pivots	Chart	Indicators	Member	Date Range

				Picker	Pickers
Row	Y				
Column	Y				
Items				Y	
Axis		Y			
Series		Y			
Time					Y
Measures	Y	Y	Y		
Background	Y	Y	Y	Y	Y

*'Y'- Is associated with widget

Binding Data to a Widget

When a Widget is selected, the Widget Data Editor will load up with the data configuration appropriate to that Widget. If there is already data bound to the Widget, it will show in the editor

Add a new Dimension or Measure

To **add a Dimension or Measure** to a panel, drag it from the Data Browser and drop it onto the appropriate panel. Once placed in the data layout the following options are available:

- To **reposition** a Dimension within a panel, click on the Dimension/Measure and reposition it while the left mouse button is clicked.

- To **move** a Dimension or Measure from one panel to another, drag it from the original panel and drop it onto the new panel.
- To **remove** a Dimension from a panel, right-click on the dimension and click 'Remove'.
- To **clear** a panel, right click and select 'Clear'.
- To **filter** a Dimension, right-click the Dimension and click 'Filter'.

Modify existing Dimension or Measure

Right clicking on the member brings up one or more of the menu items below.

- **Rename:** Rename the member.
- **Remove:** Remove the member.
- **Add to repository:** Add the dimension, measure or filter to the measures or filter repository.
- **Filter:** Create a filter (See Filter)
- **Clear:** Clear all members in the panel.
- **Edit:** Edit the existing measure.
- **New:** Create a new measure.

Updating the Widget

Click the **Update** button or check 'Auto Update' to apply the Data Editor's current data layout to the widget and query the Data Source.

The widget will be refreshed and appear with data based on the data layout.

Replace, Union, Intersect and Exclude

Dropping a Dimension/Filter onto a panel that already contains data from that Dimension will bring up several options:

- **Replace** the existing data with the new data.
- **Union** the new data with the existing data.
- **Intersect** new data with the existing data.
- **Exclude** the new data from the existing data.

Creating Measures from a Dimension: Dropping a Dimension onto the Measures panel, will bring up the quick Measure creation menu. Selecting one of the aggregation types will automatically add a Measure with the specified aggregation on the dropped dimension.

Applying Scope on a Measure:

When a Measure exists in the Measures panel you can restrict the results to a given scope. For example restricting total sales to a particular region.

1. Right-clicking on the member and click on 'Apply Scope'.
2. A list of all dimensions will open.
3. Clicking on the drop down next to a dimension will display filters and values associated with the dimension.
4. Enable the check box next to the value or filter to restrict measures results based on the selected value or filter. Click 'OK'
5. Click 'OK' to apply the scope.
6. Click the 'Update' button to save scope changes to the widget.

Notes for Placing Members in the Data Layout:

- Any Dimension, Group or Filter can be placed on all panels except for the Measures panel that accepts Measures only.
- Placing data on the Background panel while data from the same Dimension is placed on either Rows or Columns panels will result in intersection operation between the two panels. Only Members appearing in both will be returned.
- The order in which fields appear is determined by the order of their appearance in the panel. You can reorder the fields by clicking on the appropriate Dimension and dragging it to its new location.

Calculations and Formulas

QUICK FORMULAS

Quick Formulas enable you to instantly perform additional calculations on a measure.

1. Place the mouse cursor over the header of a measure column until the menu icon appears.
2. Click the 'Function' icon.
3. A drop down will appear with a list of quick formulas.
4. Placing the mouse cursor over a formula will open an additional menu .
5. Clicking on the relevant formula will perform the associated calculation on the measure.

Below is a list describing Quick Formulas.

Quick Formula	Description
Growth	Growth from previous period, or same period previous year/quarter/month
Difference	Difference from previous period, or same period previous year/quarter/month
Contribution	Contribution of current period to the total duration.
Past Values	Change in value, year/quarter/month/week/day to current period.
Change over Time	Change in value, growth/period/year/quarter/month/week to current period.
Change over Time (%)	Change in percent, growth/period/year/quarter/month/week to current period.
Running Sum	Running total, year/quarter/month to current period
Running Average	Running average, year/quarter/month to current period
Rolling Period Sum	Rolling total, 'n' periods back until the current period
Rolling Period Average	Rolling average, 'n' periods back until the current period

Custom Measures

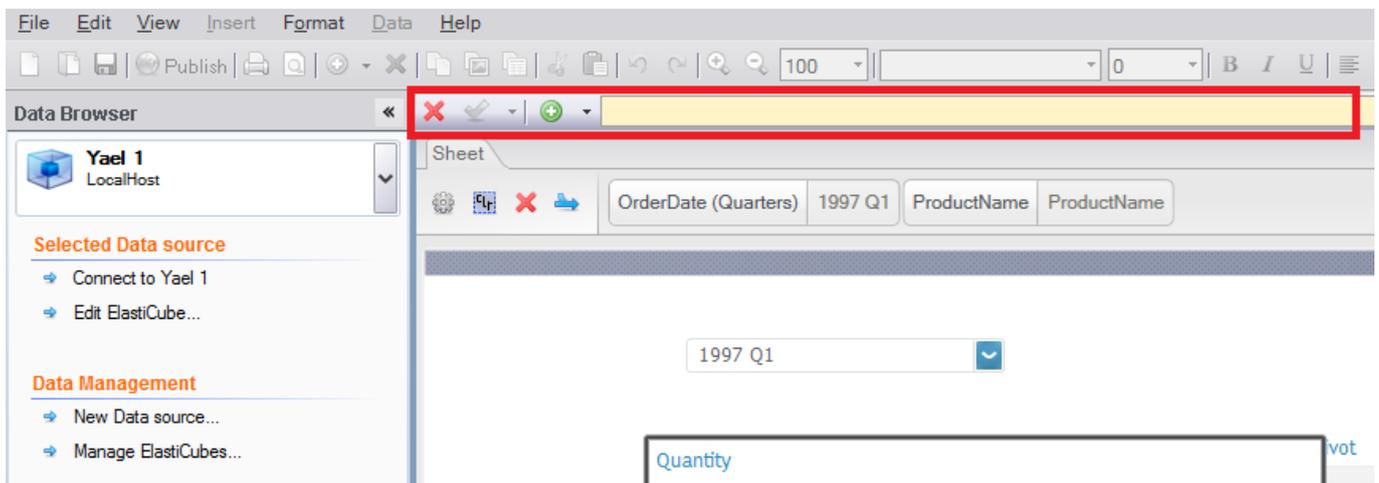
Introduction to Custom Measures

A custom measure is a mathematical or analytic calculation combining one or more dimensions, aggregations and mathematical operators.

Custom measures can be used anywhere and in the exact same way as regular measures.

Formula Editor

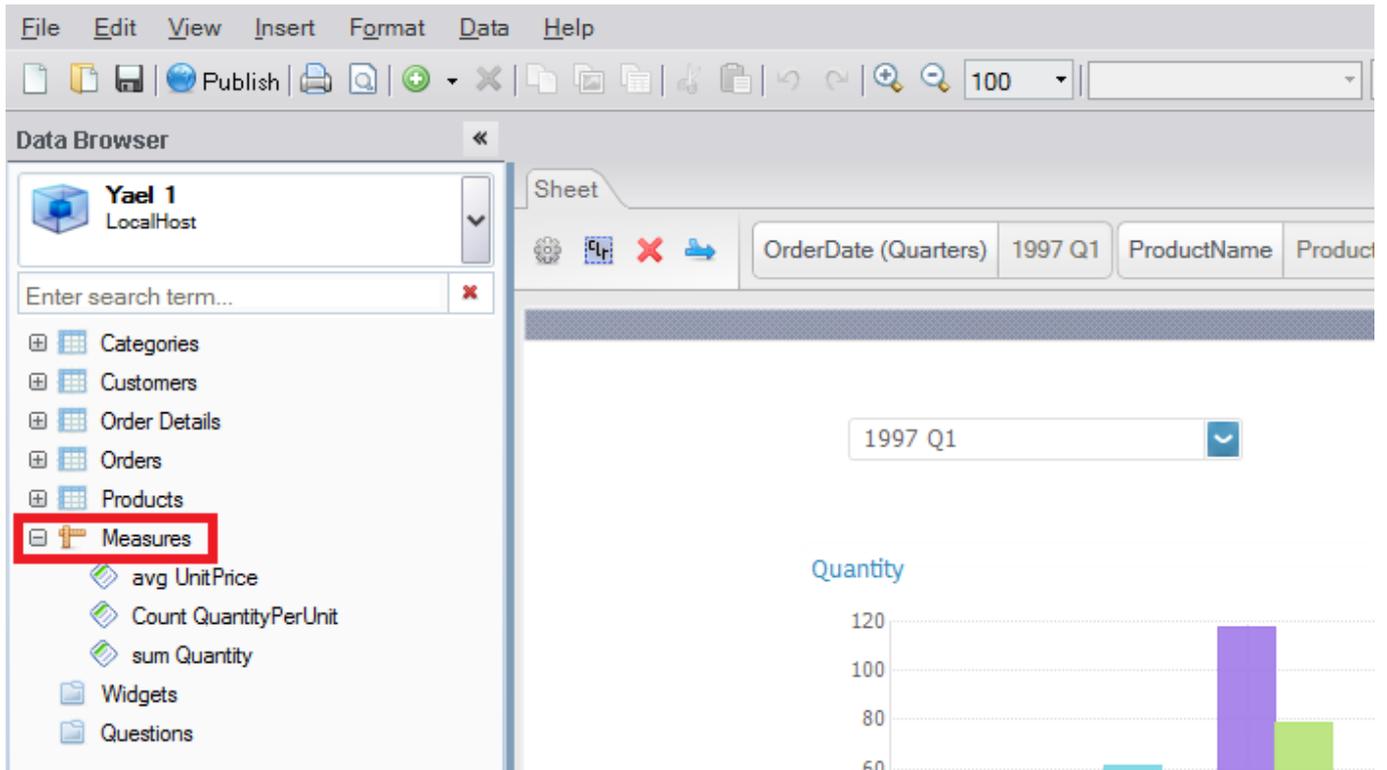
Custom measures are created in the Formula Editor which is a panel for defining, modifying a measure's formula. It is located immediately under the main toolbar. The editor is only visible when working on a custom measure or when a measure is selected on a widget. See the steps below on how to create a measure.



Measures Repository

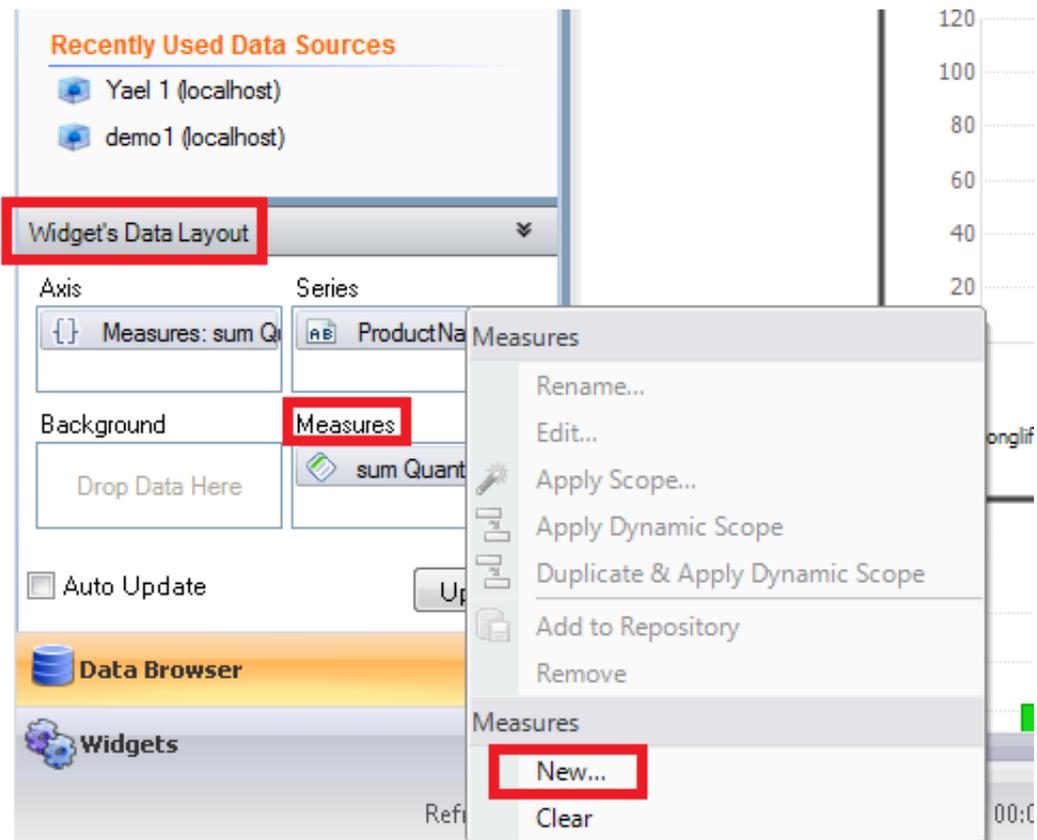
The measures repository enables any measure to be saved and used within other widgets and dashboards.

The measures repository is located in the data browser under the 'Measures' node. Clicking on the node displays a list of all pre-aggregated and custom measures.

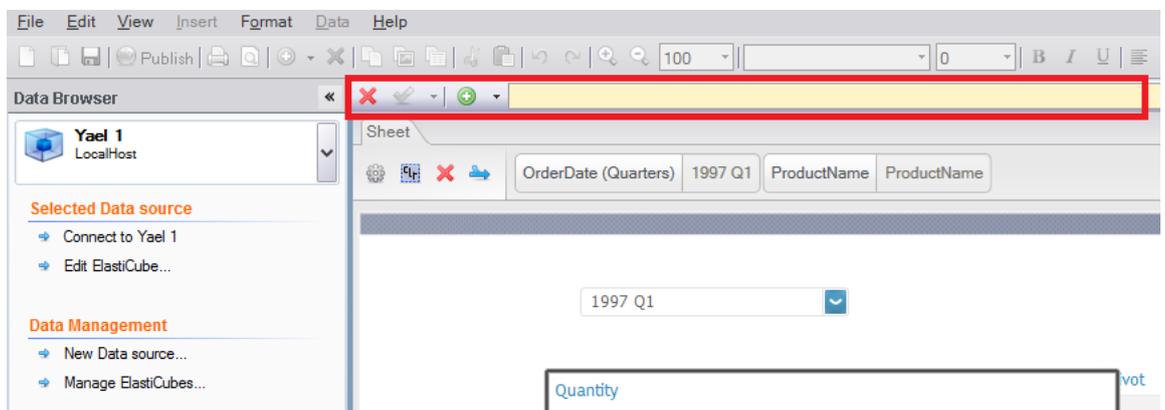


Create a Custom Measure

1. Right click on the Measures node in the data browser and select 'new'.
Alternatively click a widget to add a custom measure, in the widgets data panel right click the measures component and select 'new'.

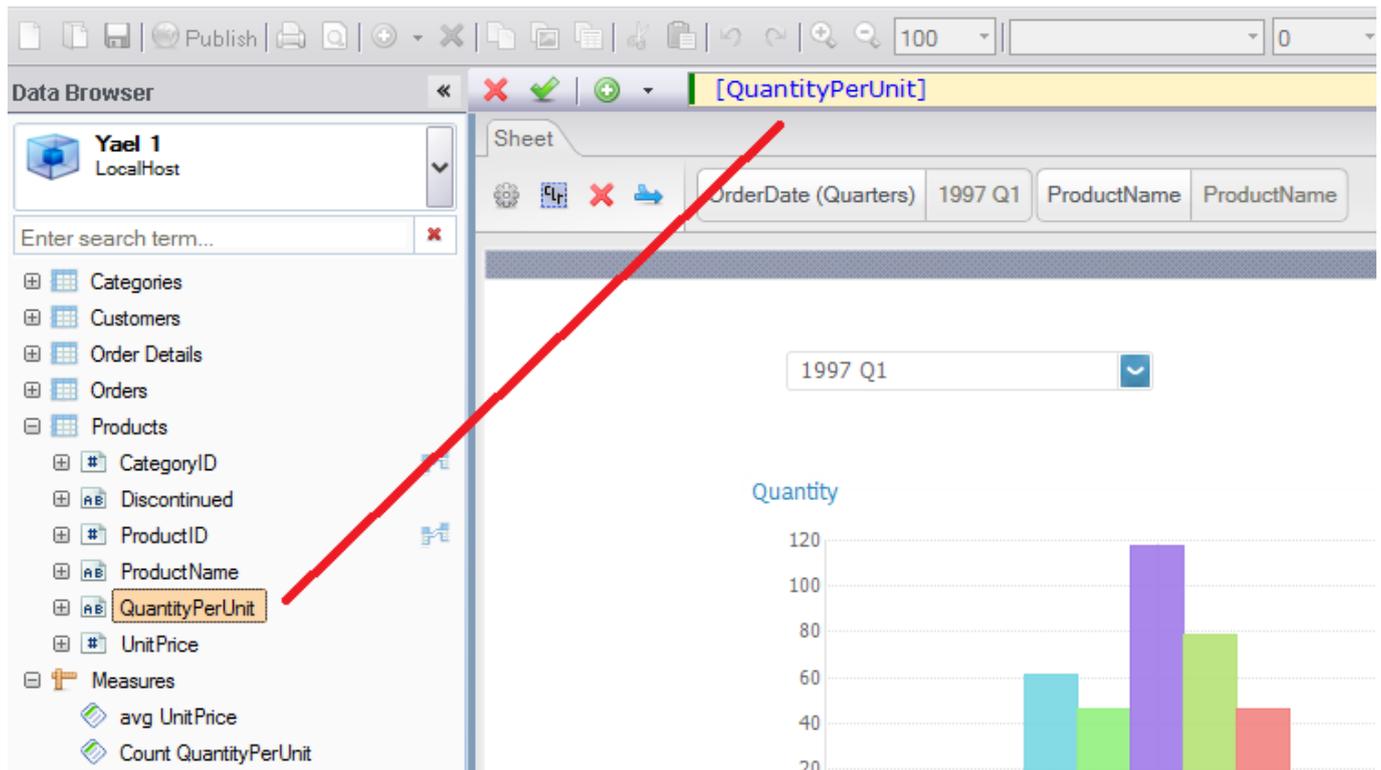


2. The formula editor will open in a panel above the sheet.



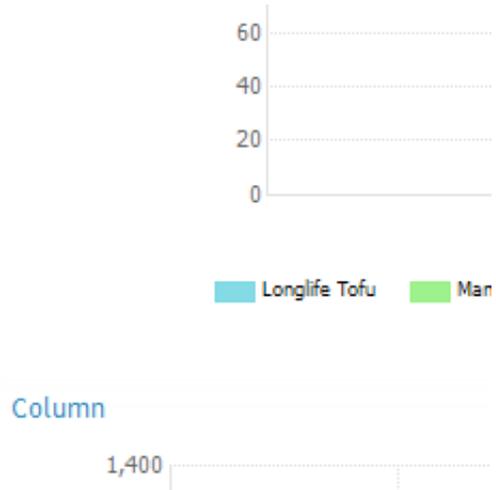
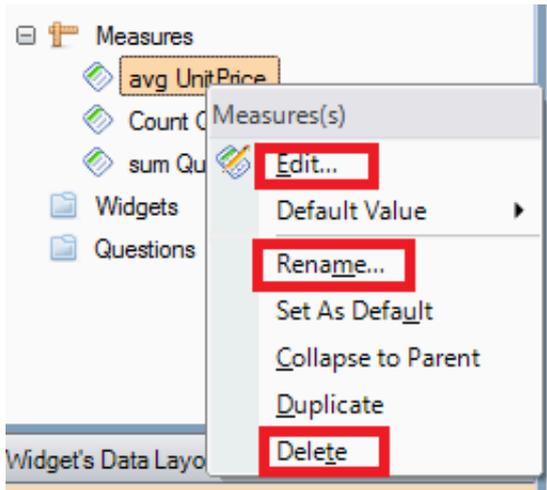
Note

When working with existing measures and nodes drag and drop the relevant dimensions from the data browser into the formula editor.



3. Click the 'tick' icon or press enter to give the custom measure a descriptive name and click OK to save it to the measures repository.

- **Rename:** Right click the measure in the measures repository and select 'Rename'.
- **Edit:** Right click the measure in the measures repository and select 'Edit'. The measure will be displayed in the formula editor panel. Modify the formula and press enter to save the changes.
- **Delete:** Right click the measure in the measures repository and select 'Delete'.



Measured Values

Introduction

A measured value is a way to segment the output of a measure.

A measured value combines a measure with one or more dimensions, members or filters in order to segment the measure. For example sales for a specific region or sales restricted to a specific year.

Measured values can be used anywhere and in the exact same way as regular measures.

Create a Measured Value via Apply Scope

A measured value can be generated by applying a scope on any measure contained within a widget.

1. Select a widget, in the data layout right click the appropriate measure and select 'Apply Scope'.
2. A list of all dimensions will appear.
3. Select the relevant dimension to segment the measure. Dimensions nodes can be expanded to view all associated filters and members. Enabling the check box next to a member or filter will segment the measure based on the selection.
4. Click 'Ok' to save the measured value and if necessary click update to refresh the widget data with the measured value.

Create a Measured Value via Formula Editor

1. Right click on the measures node in the data browser and select 'new'. Alternatively click a widget to add a custom measure, in the widgets data panel right click the measures component and select 'new'.
2. The formula editor will open in a panel above the sheet.
3. Click on the measure editor panel and type '()' then drag the appropriate dimension aggregation from the data browser into the **round brackets** in the panel.
4. The following logic will need to be applied to create the measured value:

Syntax

(, [, ...])

Parameters

: A measure or a custom measure.

: A member, group of members or a filter. Each must belong to a different dimension.

Example

([Sum Sales], [Region])

5. Hit enter to save the measured value.

Example: Measured Value

Product	Store	Sales
Shoes	Store A	10
Shoes	Store B	20
Shoes	Store C	30
Shirt	Store A	40
Shirt	Store B	50
Pants	Store A	60
Pants	Store C	70

$([\text{Sales}], [\text{Store A}]) = 110 \{10+40+60\}$

$([\text{Sales}], [\text{Store A and Store B}]) = 180 \{10+20+40+50+60\}$
[Store A and Store B] is a Group containing Store A and Store B.

$([\text{Sales}], [\text{Store A and Store B}], [\text{Shoes}]) = 30 \{10+20\}$
[Store A and Store B] is a Group containing Store A and Store B.

$([\text{Sales}], [\text{Products with Sales} > 60]) = 220 \{90+130\}$
[Products with Sales > 60] is a Filter by Criteria returning only products with totals sales greater than 60 (Shirt=90 and Pants=130)

Simple Arithmetic

ARITHMETIC FUNCTIONS

The following mathematical operators can be used with custom measures and measured values.

- **Addition:** $[\text{Fixed Cost}] + [\text{Variable Cost}]$
- **Subtraction:** $[\text{Sales}] - [\text{Cost}]$
- **Division:** $[\text{Unit Sold}] / [\text{Inventory}]$
- **Multiplication:** $[\text{Unit Sold}] * [\text{Price}]$
- **Parentheses:** $([\text{Unit Sold}] * [\text{Price}]) - [\text{Cost}]$

Aggregations

AGGREGATIONS

Value Aggregation functions are used to perform calculations on a list of values.

Syntax

AggregationType()

Parameters

: A Dimension or an arithmetic expression made up of Numeric Dimensions and/or numbers.

The following aggregations can be used with custom measures and measured values.

Aggregation	Syntax	Explanation	Applicable Fields
Sum	Sum()	Calculates the total of values	Fields containing numeric values
Average	Avg()	Calculates the average value	Fields containing numeric values
Minimum	Min()	Calculates the minimum value	Fields containing numeric values
Maximum	Max()	Calculates the maximum value	Fields containing numeric values
Count	Count()	Counts the number of unique values	All Fields
Count Duplicates	DupCount()	Counts the number of rows (all values)	All Fields

Example: Simple Aggregations

Consider the following raw data:

Product	Store	Sales
Ice Cream	Store A	100
Ice Cream	Store B	200
Bread	Store A	150
Milk	Store B	400

The following table describes how each function works. The functions are marked in bold and dimensions are marked in brackets [].

[Product]	Sum([Sales])	Avg([Sales])	Min([Sales])	Max([Sales])	Count([Sales])	DupCount([Sales])
Ice Cream	300	{(100+200)/2}	100	200	1	2
Bread	150	150	150	150	1	1
Milk	400	300	400	400	1	1

Example: Complex Aggregations: One

Consider the following transactional sales data:

Transaction Time	Product	Price Per Unit	Quantity Sold
12:00	Shoes	10	2
13:00	Shoes	10	4

Transaction Time	Product	Price Per Unit	Quantity Sold
14:00	Shirt	5	3
15:00	Shirt	5	6

Sales is calculated by multiplying Price Per Unit with Quantity Sold. However, if you multiplied the 'Sum Per Per Unit' **Measure** with the 'Sum Quantity Sold' **Measure**, you would get incorrect results.

[Product]	Sum([Price Per Unit])	Sum([Quantity Sold])	Sum([Price Per Unit]) * Sum([Quantity Sold])
Shoes	20 {10+10}	6 {2+4}	120 {20*6}
Shirt	10 {5+5}	9 {3+6}	90 {10*9}

According to the calculation above, shoes were sold for 120 dollars. This is incorrect because the price of a pair of shoes is 10 dollars, not 20. Total sales for shoes should have been 6 * 10 dollars = 60 dollars. The wrong value is because the price is aggregated before multiplication with the quantity is done.

To generate the correct results, multiply the 'Price Per Unit' Dimension and 'Quantity Sold' Dimension values first and only then perform the aggregation:

Sum([Price Per Unit] * [Quantity Sold])

Example: Complex Aggregations: Two

Consider the following transactional sales data:

Products	Years	Sum([Quantity])
Beverages	2010	1842
	2011	3996
	2012	3694
	Average	3177
Condiments	2010	962
	2011	2895

		2012	1441
	Average		
			1766
Confections			
		2010	1357
		2011	4137
	Average	2012	2412
			2635
Dairy Products			
		2010	2086
		2011	4374
	Average	2012	2689
			3050

The above data shows the total unit sold per year and average over a three year period between 2010 to 2012. The original transactional data for unit sales is daily and the above calculation uses the **sum()** to combine the data into an annual total.

However, in order to create a table which just shows the product and average units sold over the same three year period the standard *average* or *avg*([Units Sold]) syntax would return the daily average. This is because the transactional data for unit sold is recorded on a daily basis and therefore the average would be the daily average over the three years.

To calculate the annual average the following syntax will be needed:

avg([OrderDate (Years)], [sum Quantity])

Using this syntax will return the following result:

Products	sum([Quantity])	Avg([OrderDate (Years)] , sum([Quantity]))
Beverages	9532	3177
Condiments	5298	1766
Confections	7906	2635
Dairy Products	9149	3050

The ALL Function

This function represents all Members in a Dimension.

Description

Returns a single Member representing the aggregation of all Members in .

Syntax

All()

Parameters

: A Dimension.

Example

Consider the following raw data:

Product	Store	Sales
Shoes	Store A	10
Shoes	Store B	20
Shoes	Store C	30
Shirt	Store A	40
Shirt	Store B	50
Pants	Store A	60
Pants	Store C	70

([Sales], ALL([Product])) = 280 {10+20+30+40+50+60+70}

$$([\text{Sales}], [\text{Shoes}]) / ([\text{Sales}], \text{ALL}([\text{Product}])) = 0.214 \{60/280\}$$

The PREV, NEXT and PARALLEL Functions

These functions return a time period Member that relate to the current Time period Member.

Current

Description

Returns the Member in which is currently in context in the query.

Syntax

Current()

Parameters

: A Dimension

Prev

Description

Returns the Time period Member in which is N periods back from the current Member.

Syntax

Prev([,])

Parameters

: A Calendar Time Dimension

: Number of periods.

Next

Description

Returns the Time period Member in which is N periods after the current Member.

Syntax

Next([,])

Parameters

: A Calendar Time Dimension

: Number of periods.

Parallel

Description

Returns the Time period Member in that represents the same time period as the current Member, in the previous year, quarter or month.

Syntax

ParallelYear([,])

ParallelQuarter([,])

ParallelMonth([,])

Parameters

: A Calendar Time Dimension

: Number of periods.

Examples

Consider the following raw data:

Time	Sales
January 1, 2008	10
January 2, 2008	20
February 1, 2008	30
February 2, 2008	40
March 1, 2008	50

Results:

[Time Month]	[Sales]	([Sales], Prev([Time Month]))	([Sales], Next([Time Month]))
January	30		70
February	70	30	50

March	50	70
-------	----	----

Growth Rate = ([Sales], PREV([Time Month])) / ([Sales], CURRENT([Time Month]))

[Time Days]	[Sales]	([Sales], ParallelMonth([Time Days]))
January 1, 2008	10	
January 2, 2008	20	
February 1, 2008	30	10
February 2, 2008	40	20
March 1, 2008	50	30

Difference from same day previous month =
 ([Sales], CURRENT([Time Days])) - ([Sales], PARALLELMONTH([Time Days]))

Running To-Date Totals and Averages

These functions calculate running total/average, up till the current time period member.

Year to Date Sum

Description

Returns the running total starting from the beginning of the year, quarter or month up to the current time period member. The time dimension to be used is automatically determined by the context of the query.

Parameters

: A Measure or Custom Measure.

Syntax

Year to Date Sum: **YTDSum**()

Quarter to Date Sum: **QTDSum**()

Month to Date Sum: **MTDSum**()

Year to Date Average

Description

Returns the running average starting from the beginning of the year, quarter or month up to the current time period member. The time dimension to be used is automatically determined by the context of the query.

Parameters

: A Measure or Custom Measure.

Syntax

Year to Date Average: **YTDAvg**()

Quarter to Date Average: **QTDAvg**()

Month to Date Average: **MTDAvg**()

Syntax

(, [, ...])

Parameters

: A Measure or a Custom Measure.

: A Member, Group of Members or a Filter. Each must belong to a different Dimension.

Examples

Consider the following raw data:

Time	Sales
February 1, 2008	10
February 2, 2008	20
March 1, 2008	30
March 2, 2008	40
April 1, 2008	50

Results

[Time Month]	[Sales]	YTDSum([Sales])
February	30	30 {30}
March	70	100 {30+70}
April	50	150 {30+70+50}

[Time Month]	[Sales]	QTDSum([Sales])
February	30	30 {30}

March	70	100 {30+70}
April	50	50 {50}

[Time Month]	[Sales]	YTD Avg([Sales])
February	30	30 {30/1}
March	70	50 {(30+70)/2}
April	50	50 {(30+70+50)/3}

[Time Month]	[Sales]	QTDAvg([Sales])
February	30	30 {30/1}
March	70	50 {(30+70)/2}
April	50	50 {50/1}

Rolling Totals and Averages

These functions calculate running total/average starting N periods back up until the current time period Member.

Rolling Period Sum

Description

Returns the running total of starting periods back up until the current time period Member.

Syntax

RPSum(,)

Parameters

: A Measure or Custom Measure.

: Number of periods back (Optional. The default value is 1).

Rolling Period Average

Description

Returns the running average of starting periods back up until the current time period Member.

Syntax

RPAvg(,)

Parameters

: A Measure or Custom Measure.

: Number of periods back (Optional. The default value is 1).

Examples

Consider the following raw data:

Time	Sales
February 1, 2008	10
February 2, 2008	20
March 1, 2008	30

March 2, 2008	40
---------------	----

April 1, 2008	50
---------------	----

Results:

[Time Month]	[Sales]	RPSum([Sales], 1)
February	30	30 {30}
March	70	100 {30+70}
April	50	120 {70+50}

[Time Month]	[Sales]	RPAvg([Sales], 1)
February	30	30 {30/1}
March	70	50 {(30+70)/2}
April	50	60 {(70+50)/2}

Growth from Previous/Parallel Periods

These functions calculate the growth/growth rate of a measure from the previous period

Growth

Description

Calculates growth over time. Assuming the query is at a monthly level, it is the same as the following formula: $([Measure] - \text{PastMonth}([Measure])) / \text{PastMonth}([Measure])$.

Syntax

Growth()

Parameters

: A Measure or Custom Measure.

Growth Rate

Description

Calculates growth rate over time. Assuming the query is at a monthly level, it is the same as the following formula: $[Measure] / \text{PastMonth}([Measure])$.

Syntax

GrowthRate()

Parameters

: A Measure or Custom Measure.

Examples

Consider the following raw data:

Date	Store	Sales
November 2007	Store A	10
November 2007	Store B	20
December 2007	Store A	30

December 2007	Store B	40
January 2008	Store A	50
January 2008	Store B	60
February 2008	Store A	70
February 2008	Store B	80

Growth Over Previous Month (or Year|Quarter|Day Let's calculate growth and growth rate from the previous month.

[Date Month]	[Sum Sales]	PastMonth([Sum Sales])
December 2007	70 {30+40}	30 {previous month is November 2007}
January 2008	110 {50+60}	70 {previous month is December 2007}
February 2008	150 {70+80}	110 {previous month is January 2008}

To calculate growth rate using the GrowthRate() function:

[Month]	GrowthRate([Sum Sales])
November 2007	

December 2007	2.33 {70/30}
January 2008	1.57 {110/70}
February 2008	1.36 {150/110}

To calculate growth using the Growth() function:

[Month]	Growth([Sum Sales])
November 2007	
December 2007	1.33 {(70-30)/30}
January 2008	0.57 {(110-70)/110}
February 2008	0.36 {(150-110)/110}

Growth From Parallel Quarter (or Year|Month) Let's calculate the growth and growth rate from the same month, previous Quarter.

[Date Month]	[Sum Sales]	PastQuarter([Sum Sales])
November 2007	30 {10+20}	{August 2007 does not exist}
December 2007	70 {30+40}	{September 2007 does not exist}
January 2008	110 {50+60}	{October 2007 does not exist}

February 2008	150 {70+80}	30 {same month previous quarter is November 2008}
---------------	-------------	---

To calculate growth rate using the PastQuarter() (or similarly, the PastYear()) function:

[Month]	[Sum Sales] / PastQuarter([Sum Sales])
November 2007	
December 2007	
January 2008	
February 2008	5 {150/30}

To calculate growth using the PastQuarter (or similarly, the PastYear()) function:

[Month]	(([Sum Sales] - PastQuarter([Sum Sales])) / PastQuarter([Sum Sales]))
November 2007	
December 2007	
January 2008	
February 2008	4 {(150-30)/30}

The PastYear, PastQuarter and PastMonth Functions

These functions calculate the value for the same period, during the past year, quarter or month.

Description

Calculates the value for the same period, during the past year, quarter or month.

Syntax

PastYear()

PastQuarter()

PastMonth()

Parameters

: A Measure or Custom Measure.

Examples

Consider the following raw data:

Date	Sales
February 2007	10
March 2007	20
April 2007	30
May 2007	40
June 2007	50
July 2007	60
August 2007	70

September 2007	80
October 2007	90
November 2007	100
December 2007	110
January 2008	120
February 2008	130
February 2008	140

The values for PastYear(), PastQuarter() and PastMonth() would be:

Date	Sales	PastYear([Sales])	PastQuarter([Sales])	PastMonth([Sales])
February 2007	10			
March 2007	20			10
April 2007	30			20
May 2007	40	10		30
June 2007	50	20		40
July 2007	60	30		50

August 2007	70		40	60
September 2007	80		50	70
October 2007	90		60	80
November 2007	100		70	90
December 2007	110		80	100
January 2008	120		90	110
February 2008	130	10	100	120
March 2008	140	20	110	130

Date Difference Functions

These functions calculate the difference between two dates.

Description

Returns the difference between and in years, quarters, months, days, hours, minutes and seconds respectively. YDiff, QDiff and MDiff always return whole numbers, while the rest may return fractions.

Syntax

YDiff(,)

QDiff(,)

MDiff(,)

DDiff(,)

HDiff(,)

MnDiff(,)

SDiff(,)

Parameters

: A Time Dimension containing start time values.

: A Time Dimension containing end time values.

Examples

Consider the following raw data:

Store	Start Time	End Time	Amount
Store A	February 1, 2007 10:00am	February 1, 2007 07:00pm	100
Store B	February 1, 2007 09:00am	February 1, 2007 08:35pm	200
Store C	February 1, 2007 10:30am	February 1, 2007 05:00pm	300

Results:

Store	YDiff([Start Time],[End Time])	QDiff (...)	MDiff(...)	DDiff(,,)	HDiff (...)	MnDiff(...)	SDiff(...)
Store A	0	0	0	0.38	9	540	32400
Store B	0	0	0	0.48	11.58	695	41700
Store C	0	0	0	0.27	6.50	390	23400

Contribution Calculations

Contribution calculations are typically done using a combination of Measured Values and arithmetic operations.

Consider the following raw data:

Product	Store	Month	Sales
Shoes	Store A	January	10
Shoes	Store A	February	20
Shirt	Store A	January	30
Shirt	Store B	February	40
Pants	Store A	January	50

Contribution of Product Sales to Total Sales We use the ALL function to indicate that [Sum Sales] needs to be calculated in the scope of all Members of the Products Dimension.

[Product]	[Sum Sales]	([Sum Sales], ALL([Product]))
Shoes	30 {10+20}	150 {30+70+50}
Shirt	70 {30+40}	150
Pants	50 {50}	150

Notice that the value of this formula is identical for all Products. To calculate the contribution of sales to total sales:

[Product]	[Sum Sales] / ([Sum Sales], ALL([Product]))
Shoes	0.20 (30/150)
Shirt	0.47 (70/150)
Pants	0.33 (50/150)

Contribution of Product Sales to Total Sales of Filtered Products Similarly, we can use a Filter inside a Measured Value to calculate a Measure in the scope of the Members returned by the Filter. Assume the Filter [Top 2 Products] returns the following results:

[Top 2 Products]	[Sum Sales]
Shirt	70
Pants	50

If we place this Filter inside a Measured Value, we can calculate total sales for the top 2 products:

[Top 2 Products]	[Sum Sales]	([Sum Sales], [Top 2 Products])
Shirt	70	120 (70+50)

Pants	50	120
-------	----	-----

Notice that the value of this formula is identical for all Products. To calculate the contribution of sales to total sales of the top 2 products:

[Top 2 Products]	[Sum Sales] / ([Sum Sales], [Top 2 Products])
Shirt	0.58 {70/120}
Pants	0.42 {50/120}

Contribution of Product Sales Per Store to Total Product Sales Assuming the raw data above, Product sales per store is:

[Store]	[Product]	[Sum Sales]
Store A	Shoes	30
Store A	Shirt	30
Store A	Pants	50
Store B	Shirt	40

We use the ALL function to indicate that [Sum Sales] needs to be calculated in the scope of all Members of the Product and Store Dimensions.

--

[Store]	[Product]	[Sum Sales]	([Sum Sales], ALL([Product]), ALL([Store]))
Store A	Shoes	30	150 {30+30+50+40}
Store A	Shirt	30	150
Store A	Pants	50	150
Store B	Shirt	40	150

Notice that the value of this formula is identical for all rows. To calculate the contribution of sales to total sales:

[Store]	[Product]	[Sum Sales] / ([Sum Sales], ALL([Product]), ALL([Store]))
Store A	Shoes	0.20 {30/150}
Store A	Shirt	0.20 {30/150}
Store A	Pants	0.34 {50/150}
Store B	Shirt	0.26 {40/150}

Dynamic Input and Placeholders

Introduction

Placeholders can be used to manually input a numeric or text value. The inputted values can then be incorporated into measures and filters. Some examples of placeholders include:

- Conduct what-if scenarios based on an input value. For example explore changes in sales based on manually entered growth percentage.
- Dynamically restrict values to an inputted range.
- Filter data in a pivot by means of a value in an input box.
- Use placeholder value within a measure.

There are two main types of input widgets for data – input numeric box and input text box. These are located in the ‘Filtering ’ category of the widgets panel. See [Input widgets](#) for more details.



Note

Using Placeholders

All filters and placeholders can be referenced within a measure or filter by using the following syntax

Reference a **single widget**

@widget constant

Syntax: (Measure, [**@WidgetName**])

Reference the **global scope**

@global constant, entire global scope

Syntax: (Measure, [**@global**])



Note

Placeholder Examples

Dynamically Filter Top Ranking Items

Example: Use a placeholder in a Top Ranking Filter

The following example enables a user to dynamically choose the number of top ranking items.

1. Add a new input numeric box placeholder.
2. Right click the numeric box select rename and enter a new name, for example 'top'.
3. Select the pivot widget required.
4. Right click a relevant row dimension in the widgets data panel and click 'Filter'.
5. In the filter window select 'Top Ranking' from the 'Select filter type' list.
6. In the count section enter the placeholder widget name in the 'count' field for example [`@top`] and select a measure to filter by for example [Sum Quantity].Note: Save your widgets measure to the repository to make it easier to work with.
7. Click 'OK'
8. If necessary click 'update' in the data layout panel.
9. Entering a value in the numeric box will now return the associated top ranking values in the pivot.

Dynamically Restrict the Range of a Measure

Example: Use a placeholder to filter the range of values a measure displays. For example filter sales between 1 to 2 million dollars.

The following example enables a user to dynamically choose the range of a measure.

1. Add two new input numeric box placeholders.
2. Right click each numeric box select rename and enter a new name, for example 'from' and 'to'.
3. Select the pivot widget required.
4. Add the relevant measure to the measures repository (in the widgets data layout right click the measure and select 'Add to Repository')
5. Right click a relevant row dimension in the widgets data layout and click 'Filter'.
6. In the filter window select 'Filter by Measure' from the 'Select filter type' list.
7. In the 'By' section select the saved measure from the drop down, an appropriate logical operator, for example '>' and enter the placeholder widget name for example [`@from`].
8. Click the 'And' link and select measure criteria.
9. Repeat the process from step 7 except use another appropriate logical operator, for example '<'

Quick Formulas

Quick Formulas enable you to instantly perform additional calculations on a measure.

1. Place the mouse cursor over the header of a measure column until the menu icon appears.
 2. Click the 'Function' icon.
 3. A drop down will appear with a list of quick formulas.
 4. Placing the mouse cursor over a formula will open an additional menu .
 5. Clicking on the relevant formula will perform the associated calculation on the measure.
- Below is a list describing Quick Formulas.

Quick Formula	Description
Growth	Growth from previous period, or same period previous year/quarter/month
Difference	Difference from previous period, or same period previous year/quarter/month
Contribution	Contribution of current period to the total duration.
Past Values	Change in value, year/quarter/month/week/day to current period.
Change over Time	Change in value, growth/period/year/quarter/month/week to current period.
Change over Time (%)	Change in percent, growth/period/year/quarter/month/week to current period.
Running Sum	Running total, year/quarter/month to current

	period
Running Average	Running average, year/quarter/month to current period
Rolling Period Sum	Rolling total, 'n' periods back until the current period
Rolling Period Average	Rolling average, 'n' periods back until the current period

Filtering

Create and Apply a Filter

Create a new Filter:

1. In the Data Browser open the relevant Dimension's node in the Data Browser
2. Right-click on the 'Filter's node and navigate to 'New'. (To edit after creating a filter right click and select 'Edit.).
3. To create a new filter one or more of the following details will need to be entered:
 - Select the relevant type of filter in the 'Select filter type' drop down . (See description below of filters).
 - Name the filter for future reference by editing the input box next to 'Caption'.
 - Configure filter-specific properties

For values in a widgets data layout right click the dimension and select 'Filter'

- Right click the relevant dimension and click 'Filter'.
- To edit after creating a filter right click and select 'Edit'.

Apply Filter on an existing Filter:

1. Open the relevant Dimensions node in the Data Browser
2. Find the Filter under the 'Filters' node, right-click on it and navigate to 'New Filter'
3. To edit after creating a filter right click and select 'Edit'.

Filters Types

The table below describes the filter types available.

Measure	Filter Members by comparing the value of a Measure to another value.
Value	Filter Members by comparing their values to another value.
Attribute	Filter Members that intersect with Members from other Dimensions.
Top\Bottom Ranking	Returns the Members whose value for a specified Measure is the highest (Top) or lowest (Bottom).
Top\Bottom Percentile	Return the Members whose cumulative value for a specified Measure is within a given percent of the total value.
Exclude	Exclude explicit Members (or Members

returned by another Filter) from the results of a Dimension or a Filter.

Filter Types

VALUE AND LABEL FILTERS AND CRITERIA

Value filters and criteria let you filter Members according to their value. Value criteria apply to Dimensions with numeric-value Members and Label criteria apply to Dimension with text-value Members.

Value Filter - Required details:

Logical Operator: Logical assessment to perform based on input value (operators: ?,=, >,

Attribute Filters and Criteria

Attribute criteria let you filter Members that intersect with Members from other Dimensions. Intersecting Members means that there is at least one row in the data that contains both the Member from the filtered dimension and the Members specified in the criteria.

Examples:

- To specify the intersection Members or Filters, click on the '**Intersecting With**' box.
- Filter products that were sold to male customers
- Filter products that were sold in January 2008.

Attribute Filter - Required details:

Intersecting member (Optional): Restrict the filter to the scope of specified Members.

Measure Filters and Criteria

Measure filters perform filtering on members by comparing the value of a Measure to another value.

Example

- Filter sales reps where total sales (sum of sales) greater than \$10,000
- Filter products where unit sold (sum of quantity) less than 5000

Measure Filter - Required details:

- **Measure:** Measure to use as basis for logical assessment.
- **Logical Operator:** Logical assessment to perform based on input value (operators: ?,=, >,
<)

Top\ Bottom Ranking Filter

The Top Ranking filter returns the Members whose value for a specified Measure is the highest, while the Bottom Ranking filter returns the Members whose value for a specified Measure is the lowest.

In effect, what this filter does is sort the Members according to the specified Measure (highest-to-lowest in the case of Top Ranking and lowest-to-largest in the case of Bottom Ranking) and returns a number of Members specified.

Examples:

- Products with the most sales
- Customers with the least purchased products
- Products with the most sales in 2008

Top\Bottom Ranking - Required details:

Count: Number of top\ bottom values to return.

(Measure) By – (optional): Measure to use as basis for top\bottom assessment.

Background (Optional): Restrict the calculation of the Measure's value to the scope of specified Members.

The Measure's Background

It is possible to restrict the calculation of the Measure's value to the scope of specified Members. This allows you to filter Members whose Sales are the highest/lowest in May-2008 for example.

To define the background (scope) of the Measure, click on the box titled 'Click to define background', navigate to the relevant Dimensions and select the Members or Filters to use.

Top Percentile and Bottom Percentile Filters

The Top Percentile and Bottom Percentile filters return the Members whose cumulative value for a specified Measure is within a given percent of the total value.

In effect, what these filters do is sort the Members according to the value of the Measure (largest-to-smallest in the case of Top Percentile and smallest-to-largest in the case of Bottom Percentile) and starts adding the values of the Measure for each Member. The process stops as soon as the cumulative value grows larger than the specified percent of total value.

Examples:

Customers that amount to 80% of total sales

Top\Bottom Percentile - Required details:

- **Count:** Number of top\ bottom values to return.
- **Measure By** (optional): Measure to use as basis for top\bottom assessment.
- **Background** (optional): Restrict the calculation of the Measure's value to the scope of specified Members.

The Measure's Background

It is possible to restrict the calculation of the Measure's value to the scope of specified Members. This allows you to filter Members whose Sales in May-2008 amount to 80% of total sales in May-2008.

To define the background (scope) of the Measure, click on the box titled '**Click to define background**', navigate to the relevant Dimensions and select the Members or Filters to use.

Exclude Filters

The Exclude filter lets you exclude explicit Members (or Members returned by another Filter) from the results of a Dimension or a Filter.

Examples:

- All Members from the Customers Dimension, except for Customer A and Customer B
- All Members from the Products Dimension, except for Products sold more than once
- All Members from the 'Products with Sales > 1000' filter, except for Products sold in the USA

Exclude - Required details:

Excluding: Members or filter to exclude in filter operation.

Groups

Groups enable you to group members from a dimension and utilize the group as a filter. For example you may have a dimension containing countries, all countries in Europe can be selected and added to a group called 'Europe'.

Creating a New Group

1. In the Data Browser expand the relevant dimension.
2. Expand the default node to show all associated members.
3. Select the relevant members, holding down control while clicking the left mouse button enables multiple values to be selected.

Tip:

1. Once all required members are selected, right click over one of the highlighted members and select 'Group'.
2. Give the group a name and click OK
3. The group will then be created in the filters node of the associated dimension.

Adding new members to an Existing Group

1. In the Data Browser expand the relevant parent dimension.
2. Expand the default node to show all associated members.
3. Select the relevant member(s).
4. Right click one of the highlighted dimensions and select 'Add to Group' and select the relevant group.

Note

Groups Menu Options

Right clicking the relevant group provides the following menu options

- **Edit** - Groups can be edited by dragging and dropping the relevant group into a question page.
- **Delete** - Deletes the group
- **Duplicate** - Duplicates the group
- **Refresh** - Refreshes members associated with group
- **Select Members** - Selects all members within the group

Question Composition

Introduction

Question Composition is a tool for creating complex filters using a visual workflow. It provides an intuitive interface for creating the filters in a step-by-step process, as well as observing the structure of more complex filters.

Filters are created inside special types of sheets called 'Questions'.

Creating a Question

1. To create a new question, navigate to the Insert menu in the main menu and select 'Question'. Alternatively right click on the sheets panel and select 'Question'.
2. The new question will appear in the same tab-strip as sheets appear, above the main canvas.
3. To remove a question, right-click on it and select 'Remove'.

Defining the Base Dimension or Filter

1. To start filtering a dimension, drag-and-drop the dimension from the data browser onto the question composition workspace.
2. To apply a new filter on the existing element in a question, hover with the mouse over the element, showing the connection boxes
3. Right-click on one of the connection boxes () and select the filter you wish to apply. The arrows represent the direction of the flow of the filters. The flow starts at the initial dimension and moves through levels of filters.
4. Configure the filter's parameters and click 'Ok'

Edit a Filter: To edit or modify an existing filter, drag-and-drop it from the Data Browser onto the question composition screen:

Renaming a Filter: Every element in a question can be given a descriptive name by double-clicking the filter title and entering a relevant name.

Saving a Filter: Every element in a question can be saved as a separate filter by clicking the save button. Upon saving, the filter will be available for use in the data browser, under the relevant Dimension's

Publish Dashboard to the Interactive HTML

Prism Web enables dashboards to be published and shared in an interactive web environment accessible via any browser. The steps below describe how to publish a dashboard.

Note

1. On the top main menu panel click File -> Publish (Alternatively you can click the Publish icon on the top panel).
2. A popup will appear with details of the location of the web environment where you can either publish to an existing location or add additional location.

The following options are available:

Publish to a New web location

- If this is the first time you are publishing a Connect to new server window will open. If you have published previously click the Add New Server icon.
- You will need to enter the following details:

Address: This is the ip or domain name of the server location.

User: This is the user name for the Prism Web environment (Publish rights will need to be enabled).

Password: This is the password for the Prism Web environment associated with the user name.

- Click Connect

Publish to an Existing web location

- Select the location from the Target Server drop down.
- Wait till the connection has been made and the folder and page selectors appear.

Edit or Delete an existing web location

- Click the Edit icon to view and edit server details
- Click the Delete icon to remove the web server connection.

Publish to an existing folder

To publish to an existing folder select a folder from the Existing Folder drop down.

Publish to a new folder

To publish to a new folder enter a name in the New Folder input box.

Select one or more sheets to publish

Select the sheets you wish to publish by enabling the check box next to the relevant sheet name in the Pages panel.

Publish to web

Click Publish to the web environment or Publish and View to open the web environment after publishing is complete.

Dashboard View Mode

Prism Viewer is a presentation mode to present dashboards without editing options available.

1. To enter view mode click the 'View'>'Open in Viewer' (F7) in top main menu panel.
2. The viewer will open with the associated dashboard.

Below is a list of menu option available in Prism Viewer.

- **Help:** Opens the help and features menu.
- **Full Screen:** Opens the dashboard in full screen mode (F11).
- **Open:** Browse the computer to locate and open additional dashboards in view mode.
- **Print:** Print the current dashboard.
- **Zoom:** Zoom in or out of current dashboard.
- **Fit:** Fit the dashboard to the screen.
- **Copy as Image:** Copy the selected widget as an image.
- **Copy as Value:** Copy values from the selected widget.
- **Export to PDF:** Export the dashboard as a PDF.
- **Export to Excel:** Export the dashboard to Excel.
- **Refresh:** Query the data sources to refresh all data in the dashboard.

Embed External Web Content in Dashboard

Prism Web supports embedding external content into your dashboard. This can be done by dropping HTML code into the dashboard in BI Studio. Follow the steps below to embed content.

- Drag and drop a text widget onto the relevant sheet.
- Double click the text box and type or paste the HTML code you would like to use in Prism Web.
- The size of the text box fixes the size of the HTML in Prism Web. Resize the text box to create the appropriate dimensions.
- When the dashboard is published to Prism Web the HTML will be run and embedded in the dashboard.

Conditional Formatting & Exception Highlighting

Usage and Application of Conditional Formatting

Introduction

Conditional formatting helps you in the following ways;

- Visually explore data
- Analyze data
- Detect critical issues
- Identify patterns and trends.

Whenever you analyze data, a visual representation of underlying data makes it easier to identify exceptions and visually compare aggregations values.

Conditional formatting allows you to create one or more rules for when and how pivot cells or axis in charts are formatted. If the rules (conditions) that you specified are met, then the formatting is applied.

The list below details the settings you can control:

Conditional Format	Description
Cell Highlight	Highlight color and settings based on a criteria.
Cell Format	Combination of background, numeric and font settings based on a criteria.
Icon Set	Show different in-cell icons based on a criteria.
Data Bars\Circles\Squares	Show different in-cell bars, circles or squares based on criteria and cell value in comparison to current range of values.

Create a New Conditional Format

1. Hover over the relevant widget until the formatting and function menu appears.
2. Click the 'Manage Data Formatting' icon.
3. Select the measure to apply conditional formatting from the drop down.
4. Select the relevant formatting option or click 'Manage Data Formatting'.
5. Click the 'Add Formatting' link.
6. On the left click the type of formatting to apply.
7. For additional formatting criteria click 'Add'. To remove formatting click 'Remove'. To edit background, numeric and text formatting click 'edit' to access the formatting menu.
8. Set conditional criteria and formatting based on meeting criteria. Below is a list of formatting options based on formatting type.
9. Once complete click 'OK' then 'Apply'

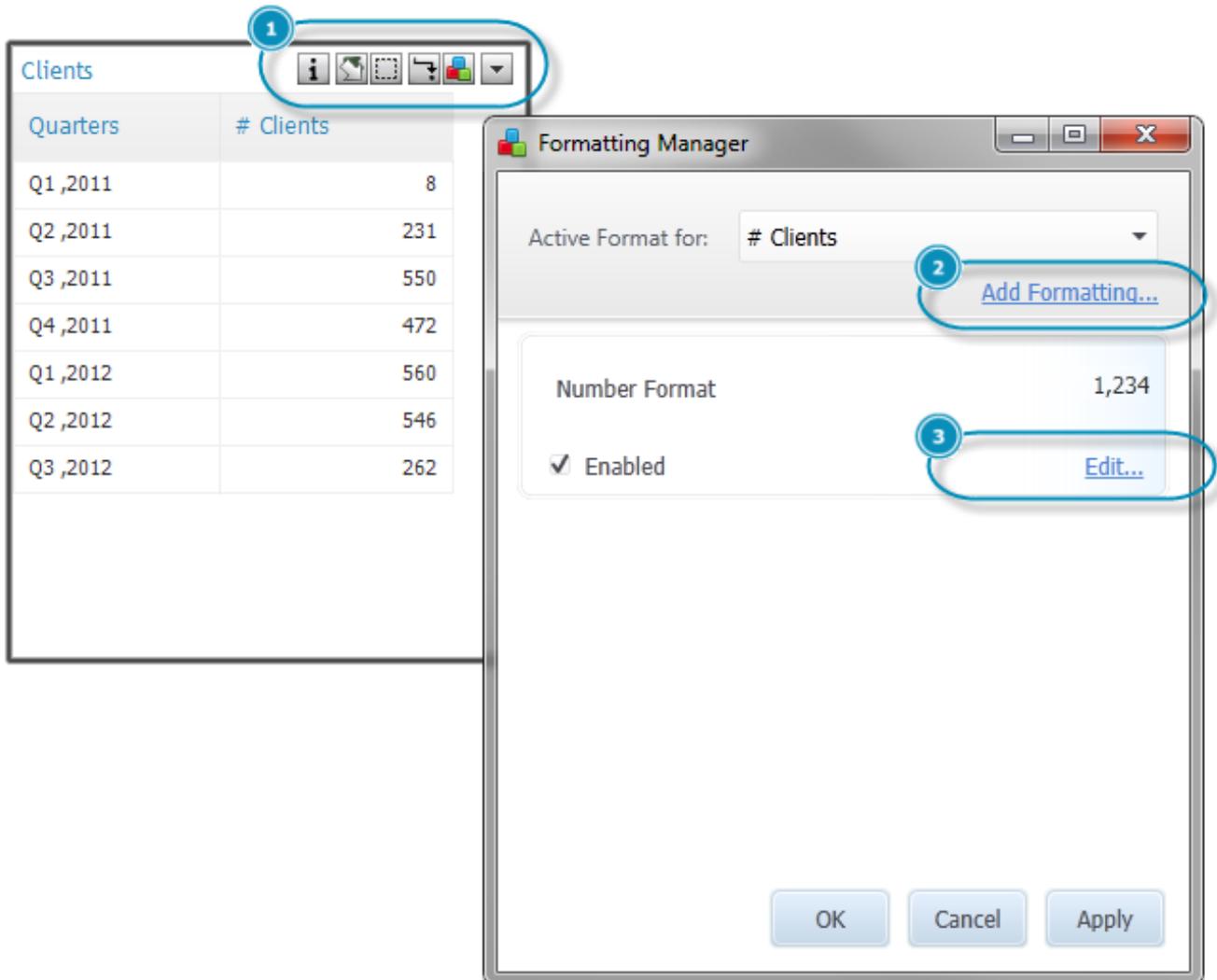


Image: Apply Formatting

1. Widget preference menu
2. Add new conditional formatting
3. Edit current formatting

Modify or Remove Conditional Formatting

It is possible to modify, remove and add multiple formatting and visual styles by bringing up the Formatting Manager via the 'Manage Data Formatting...' menu.

1. Hover over the measure header until the formatting and function menu appears.
2. Click the 'Manage Data Formatting' icon.
3. Click 'Manage Data Formatting'.
4. Click 'Edit' to modify or 'Remove' to delete. To keep formatting settings but not apply format click 'Enabled'

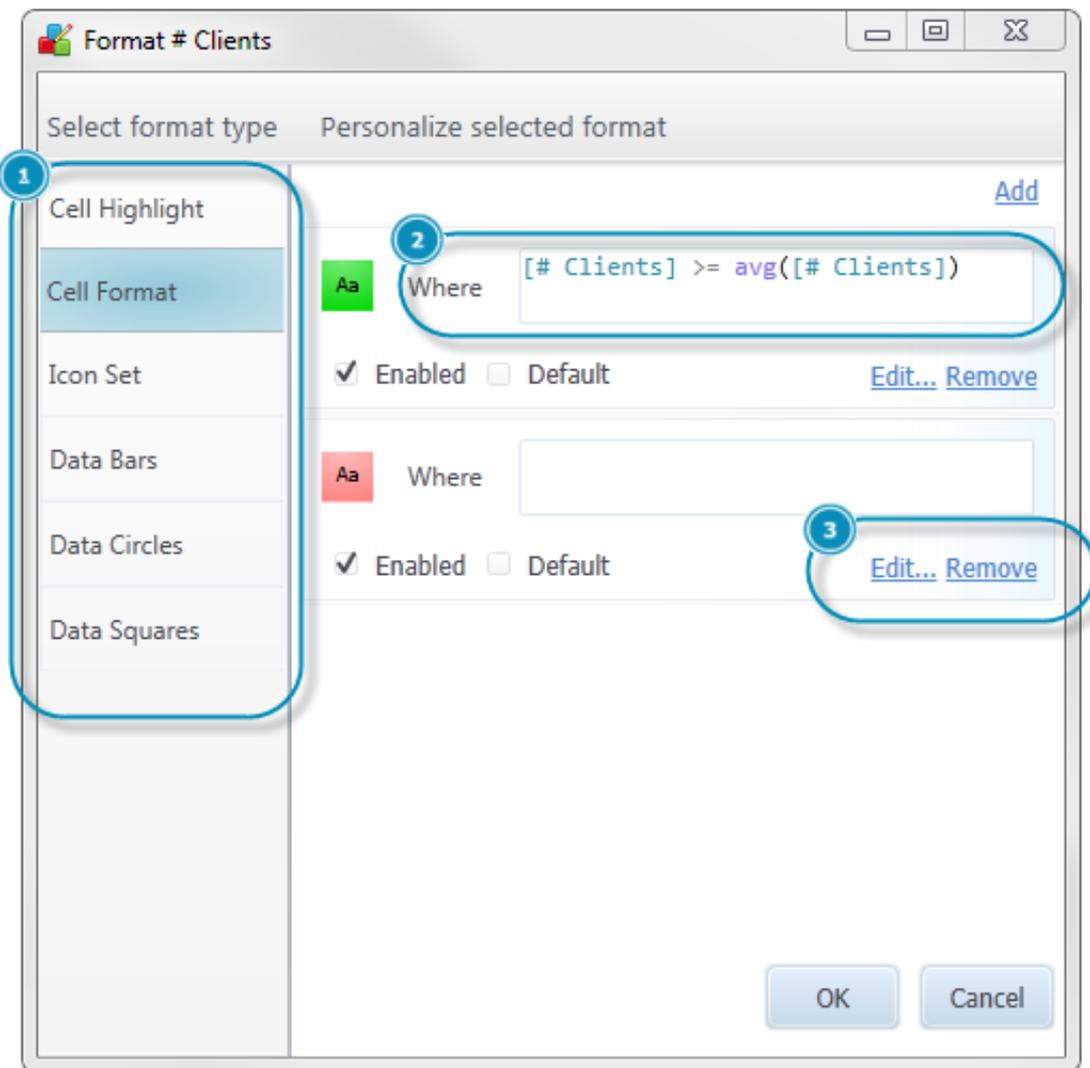


Image: Conditional Formatting Menu

1. Format type
2. Format criteria
3. Edit\Remove formatting criteria

Conditional Formatting Types

CELL FORMAT

Cell Format

To more easily find specific cells within a range of cells, you can format those specific cells based on a comparison operator.

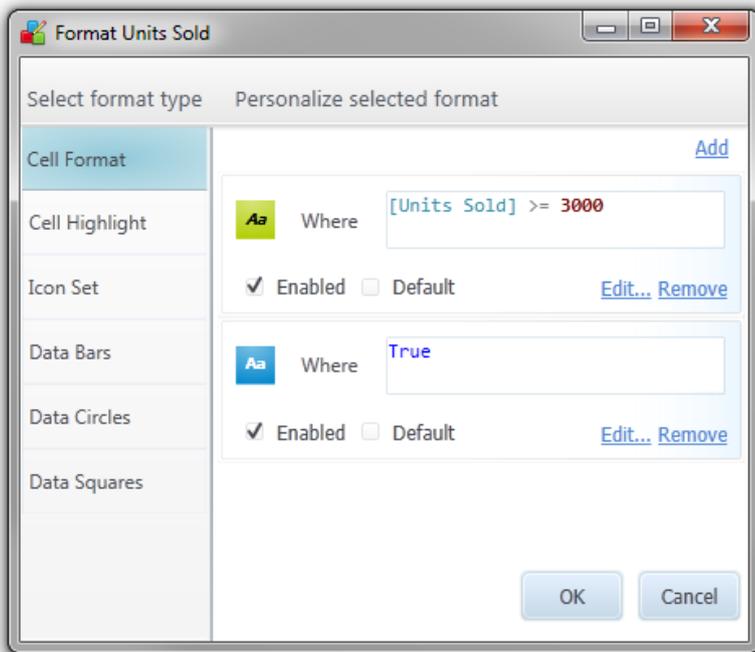
Cell Format enables formatting to be applied to both the cell background and text. For example, a green background color and bold font style can be applied to all cells where sales is smaller than previous month sales.



Formatting Menu Options

- **Background:** Background color to apply based on meeting associated criteria.
- **Where:** Criteria to use in logical assessment. For example `[Sales]>= avg([Sales])`.
- (Click 'Edit' next to the associated criteria to access the additional formatting options below.)
- **Background:** Defines the background color
- **Number:** Format, Type, Precision, Culture and thousand separator.
- **Font:** Type, color, size and style

Example: Cell Format



CategoryName	Units Sold
Beverages	3,996
Condiments	2,895
Confections	4,137
Dairy Products	4,374
Grains/Cereals	2,636
Meat/Poultry	2,189
Produce	1,583
Seafood	3,679

Example Formatting Criteria: Where total units sold are larger than 3000, apply green background color and bold, otherwise - apply blue background color and bold white text.

Cell Highlight

CELL HIGHLIGHT

Cell Highlight

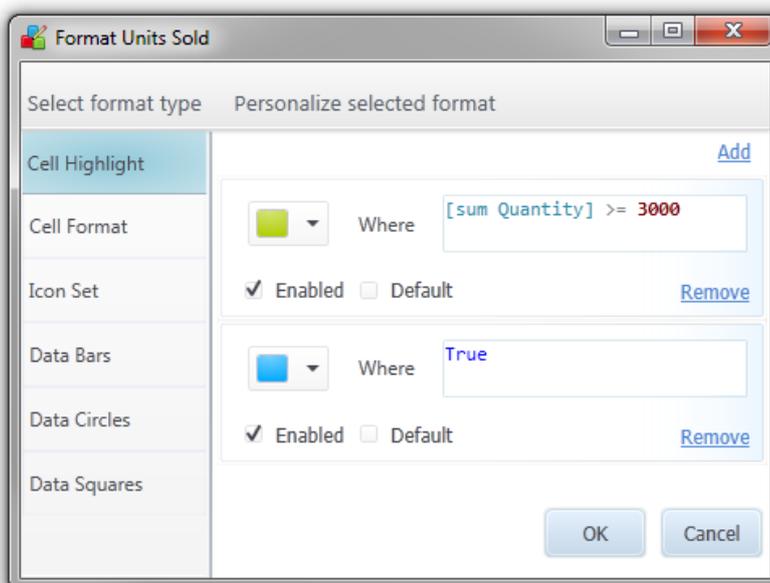
To more easily find specific cells within a range of cells, you can format those specific cells based on a comparison operator. For example, a green background color can be applied to all cells where sales is larger than average sales.



Formatting Menu Options

- **Background:** Background color to apply based on meeting associated criteria.
- **Where (Criteria):** Criteria to use in logical assessment. For example `[Sales]>= avg([Sales])`.

Example: Cell Highlight



CategoryName	Units Sold
Beverages	3,996
Condiments	2,895
Confections	4,137
Dairy Products	4,374
Grains/Cereals	2,636
Meat/Poultry	2,189
Produce	1,583
Seafood	3,679

Formatting Criteria: Where units sold greater than 3000 apply green background color, otherwise - apply blue.

Icon Set

ICON SET

Icon Set

An icon set can be used to annotate and classify data into two to six categories separated by a threshold value.

Each icon represents a range of values. For example, in the 3 Flags icon set, the green up arrow represents higher values, the yellow sideways arrow represents middle values, and the red down arrow represents lower values. Icons can be set to only display for cells that meet a condition.

For example, a warning icon can be displayed in those cells that fall below a critical value and no icons for those that exceed it. To do this, set the desired condition for the warning icon, and set all other expressions to false.



Formatting Menu Options

- **Where (Criteria):** Criteria to use in logical assessment. For example `[Sales]>=avg([Sales])`.

Example: Icon Set

The screenshot shows the 'Where' menu in Excel with three criteria:

- ✓ Where `ApplyScope([sales], [@combo1]) > ApplyScope([sales], [@combo2])`
- ⚠ Where `ApplyScope([sales], [@combo1]) = ApplyScope([sales], [@combo2])`
- ✗ Where `true`

Below the menu is a Pivot+ Selectors table:

	December, 2007	October, 2007
products		
Acer 2400	✗	429
Acer 2500	✓	380
Acer 3600	⚠	180
Acer 3800	✗	172
Dell P400	✗	367
Dell P500	✓	424

Editor Pivot+ Selectors

Data Bars, Squares & Circles

Overview

Below are explanations and examples of data bars, squares and circles.



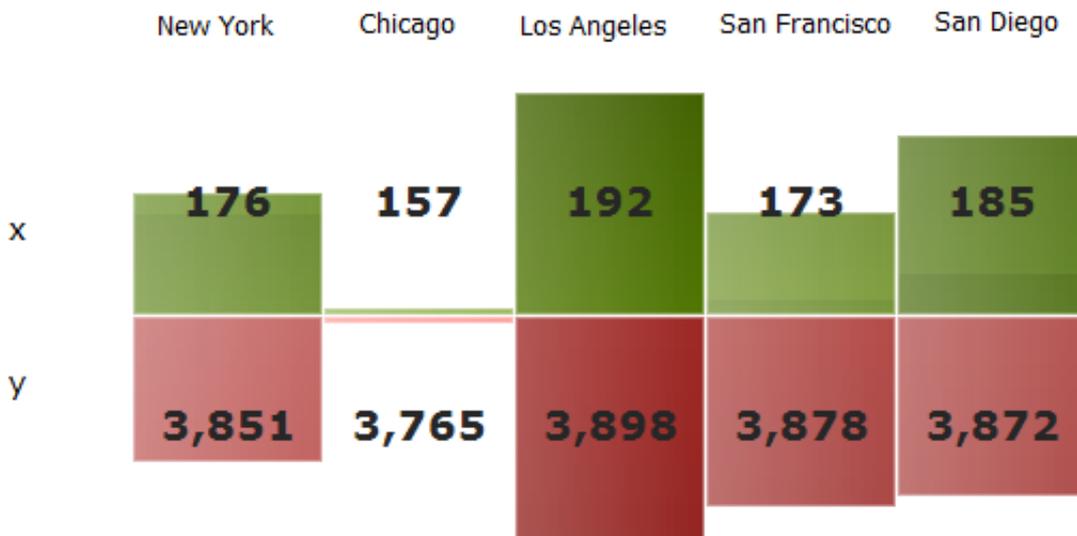
Data Bars

A data bar helps you see the value of a cell relative to other cells. The length of the data bar represents the value in the cell. A longer bar represents a higher value, and a shorter bar represents a lower value. Data bars are useful in spotting higher and lower numbers, especially with large amounts of data, such as top selling and bottom selling toys in a holiday sales report.

Example

Product	Average Revenue		Average Costs
Beverages	\$3,996		\$1,580
Condiments	\$2,895		\$2,618
Confections	\$4,137		\$2,332
Dairy Products	\$4,374		\$1,725
Grains/Cereals	\$2,636		\$2,256
Meat/Poultry	\$2,189		\$1,153
Produce	\$1,583		\$1,332
Seafood	\$3,679		\$2,286

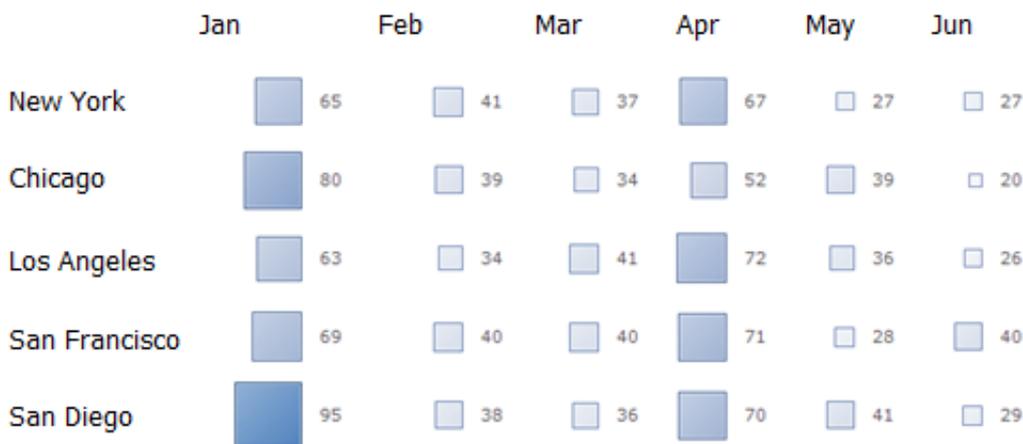
Data Bar Settings: Both are Horizontal bars with vertical alignment = center, Green's horizontal alignment = right, and Red's horizontal alignment = left



Data Bar Settings: These are Vertical bars with horizontal alignment = stretch, Green's vertical alignment = bottom, and Red's horizontal alignment = top

Data Squares

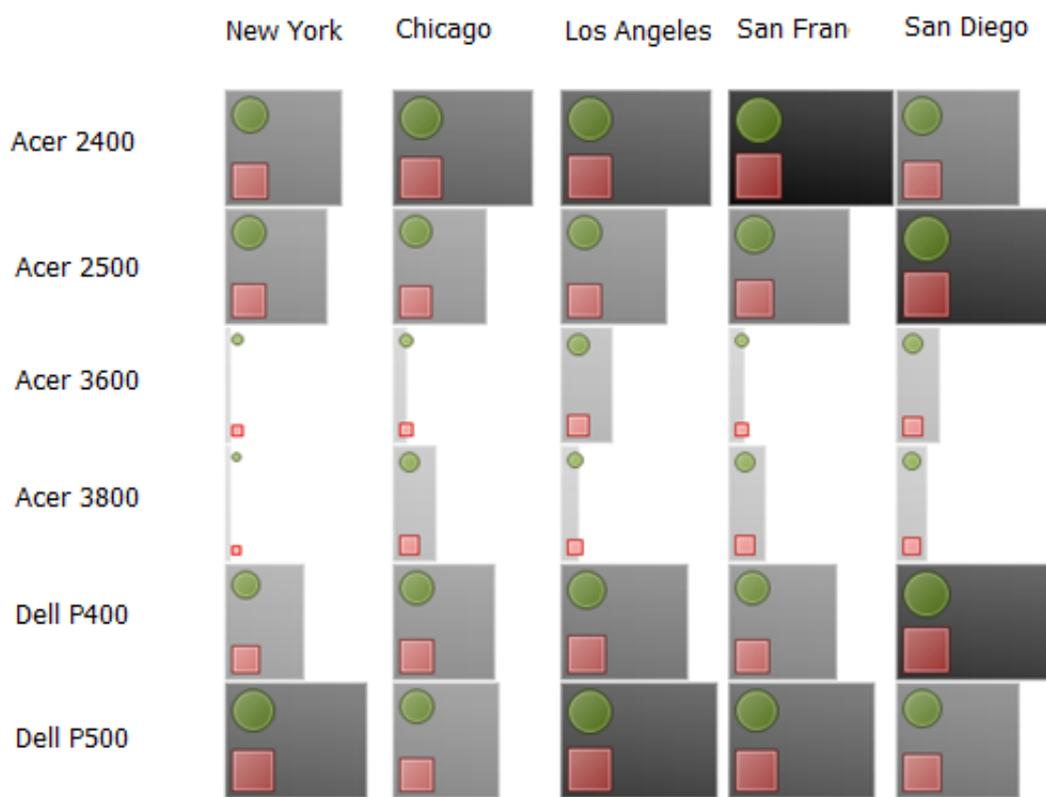
A data square has the same qualities as the data bar. The only difference is that the value is represented via a square which can be applied with a custom layout for visual impact.



Data Square Settings: Squares color is set to from white to blue depending on data value.

Data Circles

A data circles has the same qualities as data bars and squares. The only difference is that the value is represented by a circle which can be applied with a custom layout for visual impact



Data Circle Settings: Bars, squares and circles – combined at once

Formatting Menu Options

- **Where (Criteria):** Criteria to use in logical assessment. For example `[Sales]>= avg([Sales])`.
- **Size:** Set the conditions that affect the size of the data bars, circles or squares. Set values between 0 and 1 that indicate what portion of the entire cell the visualization will take up.
- **Color range or condition:** Range of colors from the minimum to maximum value. Set condition criteria which determine the color.
- **Layout:** Position of the bar, circle or square in the cell according to horizontal/vertical alignment and orientation.

Editing Expressions

Expression Editor

The expression editor is an input text box which enables one to write an arithmetic or logical expression, based on arithmetic and logical operations, data source members, widget placeholder, functions, calculations and fixed values.

It provides a useful environment for recognized text coloring, auto complete mechanism and error notifications.

Members

Input members can be any of the global scope measures (viewed on the data browser), or the edited widget measures (a measure defined in the widget itself).

The members' syntax is the member name.

When the name contains spaces, it must be surrounded by rectangular brackets (bracketing should be always considered as a good practice).

Sample: [myMeasure] > 0

Keywords

Expressions may contain many reserved keywords. Those keywords are familiar from the SQL syntax language keywords, such as TRUE, FALSE, LIKE, BETWEEN and so forth.

Functions

Functions are a quick way to manipulate data for most common usage. A function gets a list of parameters delimited by commas and yields the manipulated value.

The function syntax is the function name, followed by an opening round bracket, followed by function parameters (delimited by commas) and terminated by a closing round bracket.

Sample: Abs([myMeasure])

Calculations

Calculations are a way to manipulate data, when the data is not limited to the current evaluated row, but to a wider scope.

Calculations may be of an aggregation type such as the Min, Max, Avg and Sum aggregations, or measured value via the ApplyScope function.

Mostly, Calculations receives a measure to manipulate and a placeholder.

The calculation syntax is the calculation name, followed by an opening round bracket, followed by calculation parameters (delimited by commas) and terminated by a closing round bracket.

Sample: ApplyScope([Sales], [@CountryCitiesPivot])

Expression Resulting Types

The edited expression may be a conditional expression or numeric expression, depending on its resulting value, numeric expressions are mostly used in the Data bars, circles and squares for their Value or threshold .

Sample: Conditional Expression: [Sales] > Avg([Sales])

Sample: Numeric Expression: [Sales] – [Cost]

Auto Complete (Ctrl + Space)

By hitting the Ctrl + Space, the auto complete popup list will be shown. The list provides all available recognized tokens for:

- Reserved Keywords
- Members
- Functions
- Calculations
- Placeholders

Every item shows its name, along with its group icon. Additionally, by moving the mouse over the item, it will show a descriptive tip. This can be very useful in order to get familiar with functions' syntax and resulting type.

Working with the auto complete list is really handy, since it keeps in track with the typed token and filters all irrelevant items.

- In order to accept the member, just press Enter.
- In order to hide the popup list, just hit Ctrl + Space again.
- When the Ctrl + Space is hit and only one match is available for the current token, this match will be accepted.

Errors

The syntax editor provides error indications for invalid expressions. There are two kinds of indications:

1. Red error waves, beneath the error text range.
2. Yellow exclamation mark on the upper-left hand side.

The syntax editor validates the typed expression when:

1. A new expression is loaded.

2. Text changes were made.
3. The editor loses focus.

Viewing the current errors description can be made by moving the mouse over the yellow exclamation mark.

Widgets

Widgets are the visual building blocks of a dashboard. They are visual and interactive components that are used to visualize data, perform analysis, accept input from users and add business logic to your documents.

The Widgets are categorized in the following groups:

Group	Description
Grids	Pivots and tables
Common Charts	Standard charts
Indicators	Display a Single value in the Widget
Selection	Widgets used to change selection
Special Charts	Special types of charts
Actions	Pre-defined actions

Data Analysis and Visualization Widgets

PIVOT

Pivot tables are one of the most useful widgets for visualizing data. They allow you to quickly summarize and analyze large amounts of data, independent from the original data layout.

1. To access the pivots settings highlight the widget till the settings menu appears and click the 'Show Widgets Context Menu' icon.
2. Below is a detailed description of the pivots setting options.

Shortcuts

- **Copy:** Click to copy the widget.
- **Cut:** Click to cut the widget.
- **Remove:** Delete the widget from the sheet.
- **Ordering:** Order the widgets position in relation to other widgets. See widget ordering.

Data

- **Edit Data Hierarchies:** Create, edit or delete data hierarchies. See data hierarchies.
- **Switch Axes:** Switch the row dimension based on dimensions in the data hierarchy
- **Widget Interconnectivity:** Sets the connectivity of the widget to other widgets. See widget inter-connectivity settings.

Actions

- **Show the Data Browser:** Opens the data browser panel.
- **Create Linked Chart:** Creates a linked chart based on the data in the pivots data layout which can be pasted into other applications.
- **Copy Data Behind:** Copies the underlying data in the pivot which can be pasted into other applications.
- **Copy as Image:** Copies the widget and data as an image which can be pasted into other applications.
- **Save to Repository:** Saves the widget to the widget repository in the data browser. Widgets in the repository can be used in other sheets or dashboards.

Widget Settings

- **Show Title:** Enable or disable to show or hide the widgets title.
- **Widget Size:** Control the size of the widget based on the widget contents or page size
- **Auto Fit** content based on width, height or both. This can also be disabled to manually adjust the widget.
- **Fit to page** scales the widget to fit the whole page or the visible page.
- **Manage Data Formatting:** Access the data formatting menu: See *Conditional*

Formatting & Exception Highlighting.

- **Fixed Column Width:** Manually fix the width of the columns.
- **Left Column Width:** Enter a number in () to adjust the left columns. Enter '-1' to disable.
- **Right Column Width:** Enter a number in () to adjust the right columns. Enter '-1' to disable.

Note: Click 'Apply' to save column width settings.

Pivot

Pivot tables are one of the most useful widgets for visualizing data. They allow you to quickly summarize and analyze large amounts of data, independent from the original data layout.

1. To access the pivots settings highlight the widget till the settings menu appears and click the 'Show Widgets Context Menu' icon.
2. Below is a detailed description of the pivots setting options.

Shortcuts

- **Copy:** Click to copy the widget.
- **Cut:** Click to cut the widget.
- **Remove:** Delete the widget from the sheet.
- **Ordering:** Order the widgets position in relation to other widgets. See widget ordering.

Data

- **Edit Data Hierarchies:** Create, edit or delete data hierarchies. See data hierarchies.
- **Switch Axes:** Switch the row dimension based on dimensions in the data hierarchy
- **Widget Interconnectivity:** Sets the connectivity of the widget to other widgets. See widget inter-connectivity settings.

Actions

- **Show the Data Browser:** Opens the data browser panel.
- **Create Linked Chart:** Creates a linked chart based on the data in the pivots data layout which can be pasted into other applications.
- **Copy Data Behind:** Copies the underlying data in the pivot which can be pasted into other applications.
- **Copy as Image:** Copies the widget and data as an image which can be pasted into other applications.
- **Save to Repository:** Saves the widget to the widget repository in the data browser. Widgets in the repository can be used in other sheets or dashboards.

Widget Settings

- **Show Title:** Enable or disable to show or hide the widgets title.
- **Widget Size:** Control the size of the widget based on the widget contents or page size
- **Auto Fit** content based on width, height or both. This can also be disabled to manually adjust the widget.
- **Fit to page** scales the widget to fit the whole page or the visible page.
- **Manage Data Formatting:** Access the data formatting menu: See *Conditional Formatting & Exception Highlighting*.
- **Fixed Column Width:** Manually fix the width of the columns.

- **Left Column Width:** Enter a number in () to adjust the left columns. Enter '-1' to disable.
- **Right Column Width:** Enter a number in () to adjust the right columns. Enter '-1' to disable.

Note: Click 'Apply' to save column width settings.

Charts

CHARTS

Charts are useful when it is more important to identify trends and patterns than observe the actual values.

Chart Widget Types

- **Column Charts:** Each column in the chart represents the value of one item of data. Column charts are good ways to show comparisons between items of data.
- **Bar Charts:** Similar to column charts, except they run horizontally on the page instead of vertically like column charts.
- **Area Charts:** Reflect changes in values by filling in the portion of the graph beneath the line connecting various data points.
- **Line Charts:** Each line in the graph shows the changes in the value of one item of data. Line charts are a good way to show trends over time.
- **Pie Charts:** Show the size of a dimension compared to other dimensions and used to show percentages.

Special Chart Widget Types

- **Bubble Chart:** A Bubble chart is similar to scatter charts except the data points are replaced by bubbles. The bubbles offer a way to displaying a third variable in the two dimensional chart by color and size.
- **Doughnut:** Doughnut charts are similar to pie charts in that they show the proportions of parts to a whole and can show more than one series of data.
- **Point:** Point charts reflect data as points along the x and y axis.
- **Spline:** Spline charts are similar to line charts except the data has been smoothed out.
- **Spline Area:** Reflect area data with the smoothing effect of spline charts.
- **Stacked Area\Bar\Column:** Stacked area, bar and column charts are identical to area, bar and column charts, except the areas are stacked on top of each other, rather than overlapping. This can make the chart easier to read.
- **Stacked Area 100\Bar 100\Column 100:** Stacked 100 charts are the same as stacked charts except data dimension are stacked and represented by the portion they represent out of 100%

Chart Settings

To access the charts settings highlight the widget till the settings menu appears and click the 'Show Widgets Context Menu' icon.

Below is a detailed description of the chart setting options.

Shortcuts

- **Copy:** Click to copy the widget.
- **Cut:** Click to cut the widget.
- **Remove:** Click to delete the widget from the sheet.
- **Ordering:** Order the widgets position in relation to other widgets. See widget ordering.

Data

- **Edit Data Hierarchies:** Create, edit or delete data hierarchies. See data hierarchies.
- **Switch Axes:** Switches the row dimensions based on dimensions in the data hierarchy
- **Widget Interconnectivity:** Sets the connectivity of the widget to other widgets. See widget inter-connectivity settings.

Actions

- **Show the Data Browser:** Opens the data browser panel.
- **Copy Data Behind:** Copies the underlying data in the pivot which can be pasted into other applications.
- **Copy as Image:** Copies the widget and data as an image which can be pasted into other applications.
- **Save to Repository:** Saves the widget to the widget repository in the data browser. Widgets in the repository can be used in other sheets or dashboards.

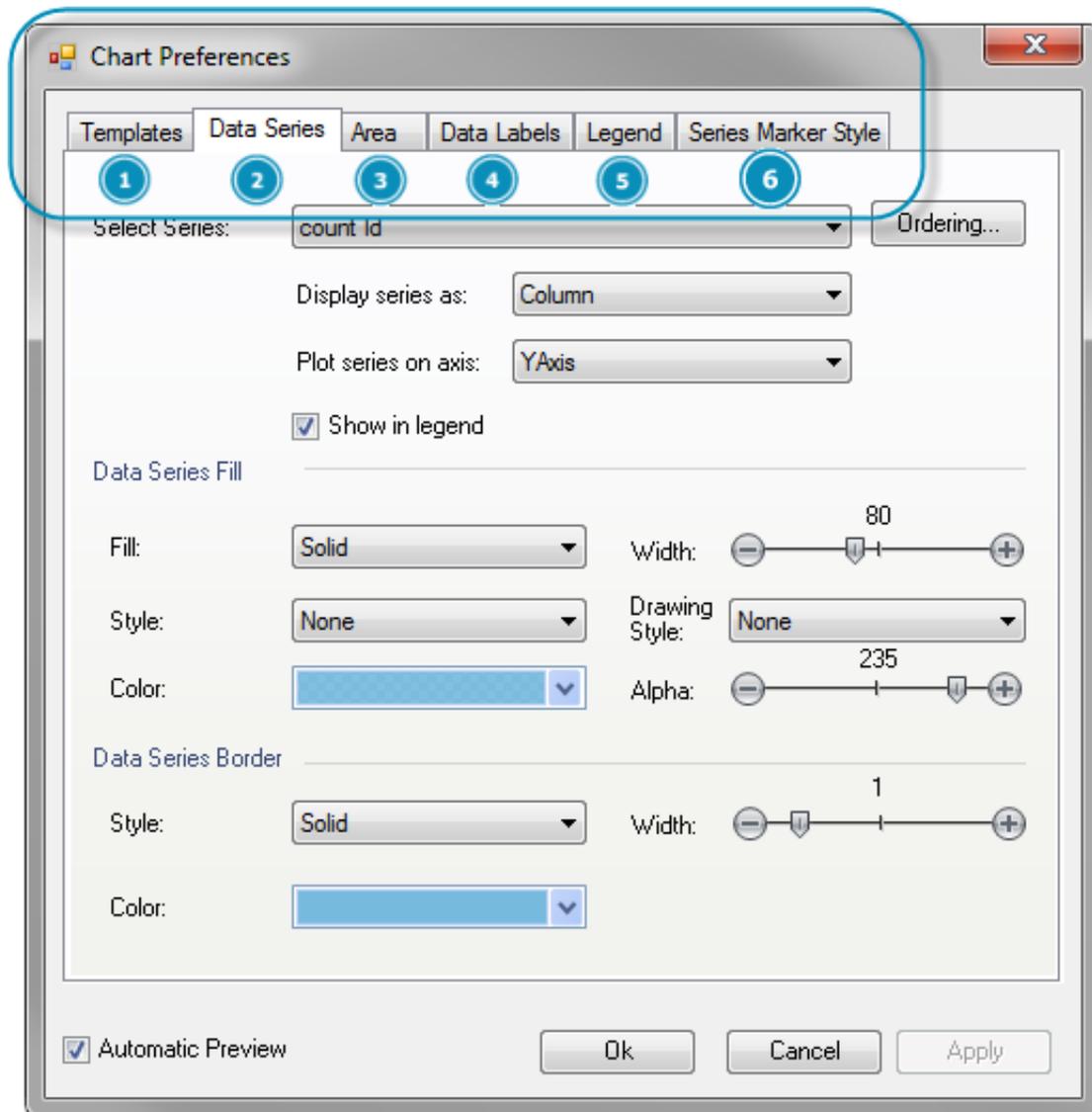
Widget Settings

- **Show Title:** Enable or disable to show or hide the widgets title.
- **Show Tooltips:** Enable or disable to show or hide the tooltips.
- **Change Series Ordering:** Arrange the order in which measures are presented in the chart.
- **Date-time Formatting:** Set the format of date-time members in the chart. See date-time formatting.
- **Format Chart:** Format chart elements: See Format Chart Menu
- **Format Chart Axis:** Format chart axis: See Format Axis Menu
- **Manage Data Formatting:** Access the data formatting menu: See Conditional Formatting & Exception Highlighting.
- **Set Page Size:** Limits the number of concurrent results per page. There are two options available.
- **Disable Paging:** Disables paging.
- **Enable Paging by Defining Page Size Value:** Determines max page size.
- **Change Type:** Change the type of chart to another chart type.

Formatting Charts

FORMATTING CHARTS

The chart menu enables formatting of the chart type, data series, chart area, labels, legend and series markers. The image below shows the main areas and formatting options in the chart menu.



1. **Templates:** Toggle chart type and color scheme.
2. **Data Series:** Format series type, axis and format.
3. **Area:** Toggle area layout and fill.
4. **Data Labels:** Show/hide series label and format.
5. **Legend:** Edit chart legend format and position.
6. **Series Marker Style:** Apply and format series markers.

Placing Dimension on Y2 Axis

Dimensions can be placed on the Y or Y2 axis. To select which axis a dimensions is displayed on follow the steps below:

1. Right click the chart and select 'Format Chart'.
2. Click the 'Data Series' tab.
3. Select the series dimension (to select all dimensions click 'All Series').
4. Click the 'Plot series on axis' drop-down.
5. Select how the dimension should be plotted in the graph.
6. Click 'Apply' then 'Ok' to save changes.

Configure Series Display

Each dimension can be displayed in a different format. For example a graph with a sales dimension and a cost dimension can display sales as a line and costs as a column in the same chart. To toggle how series are displayed follow the steps below.

1. Right click the chart and select 'Format Chart'.
2. Click the 'Data Series' tab.
3. Select the series dimension (to select all dimensions click 'All Series').
4. Click the 'Display series as' drop-down.
5. Select how the dimension should be displayed in the graph.
6. Click 'Apply' then 'Ok' to save changes.

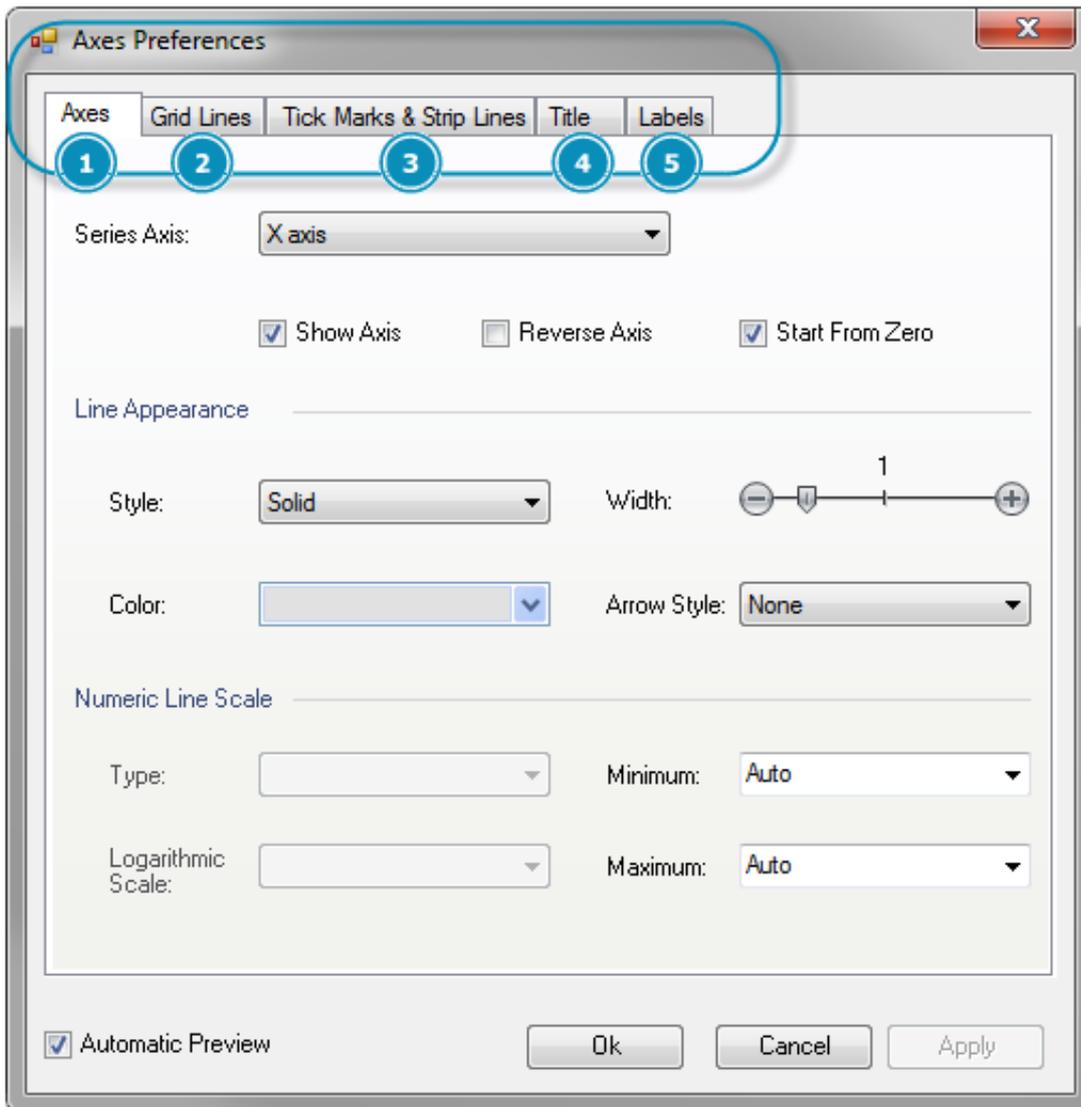
Add Data Labels

Values, percent's, numeric formats or axis label can be displayed next to the chart series. To toggle how series labels are displayed follow the steps below.

1. Right click the chart and select 'Format Chart'.
2. Click the 'Data labels' tab.
3. Next to the 'Select series' label choose the appropriate series from the drop down (to select all dimensions click 'All Series').
4. Enable the 'Show label' checkbox.
5. Under the 'Label text' select the appropriate 'Format' from the drop down.
6. Toggle with settings in the 'Label Text', 'Label Fill' and 'Label Alignment' options to format the way the labels are displayed.
7. Click 'Apply' then 'Ok' to save changes.

Formatting Chart Axes

The Format Axis menu enables formatting of the chart axis, including scale, color, formatting, axis titles and item labels.



1. Axes: Show\ hide series axis and configure appearance and scale.
2. Grid Lines: Show\ hide grid lines and configure appearance and scale.
3. Tick Marks & Strip Lines: Show\ hide tick markets and configure appearance.
4. Title: Configure axis title.
5. Labels: Configure axis labels.

Gauges & Indicators

Indicators are useful when you need to show a single value, and possibly whether it meets a pre-defined threshold. There are several Indicators that are available.

Indicator Widget Types

- **Calendar:** Calendar indicators visualize data on a calendar highlighting data based on time and dates.
- **Round Scale:** Round Scale Indicators are used to visualize values in the form of a speedometer, where the needle indicates the current value.
- **Round Numeric:** Round Numeric indicators are a formatted circle with the current value showing in the middle.
- **Rectangular Scale:** Round Scale Indicators are used to visualize values in the form of a speedometer, where the needle indicates the current value.
- **Rectangular Numeric:** Rectangular Numeric indicators are a formatted rectangle with the current value in the middle.
- **Light:** Light indicators light up in different colors according to specified thresholds.
- **Traffic Light:** Traffic lights Indicators turn on one of three lights when the value is within the particular light's configured threshold.

Indicator Settings

1. To access the charts settings highlight the widget till the settings menu appears and click the 'Show Widgets Context Menu' icon.
2. Below is a detailed description of the chart setting options.

Shortcuts

- **Copy:** Click to copy the widget.
- **Cut:** Click to cut the widget.
- **Remove:** Click to delete the widget from the sheet.
- **Ordering:** Order the widgets position in relation to other widgets. See widget ordering.

Data

- **Widget Interconnectivity:** Sets the connectivity of the widget to other widgets. See widget inter-connectivity settings.

Actions

- **Show the Data Browser:** Opens the data browser panel.
- **Copy Data Behind:** Copies the underlying data in the pivot which can be pasted into other applications.

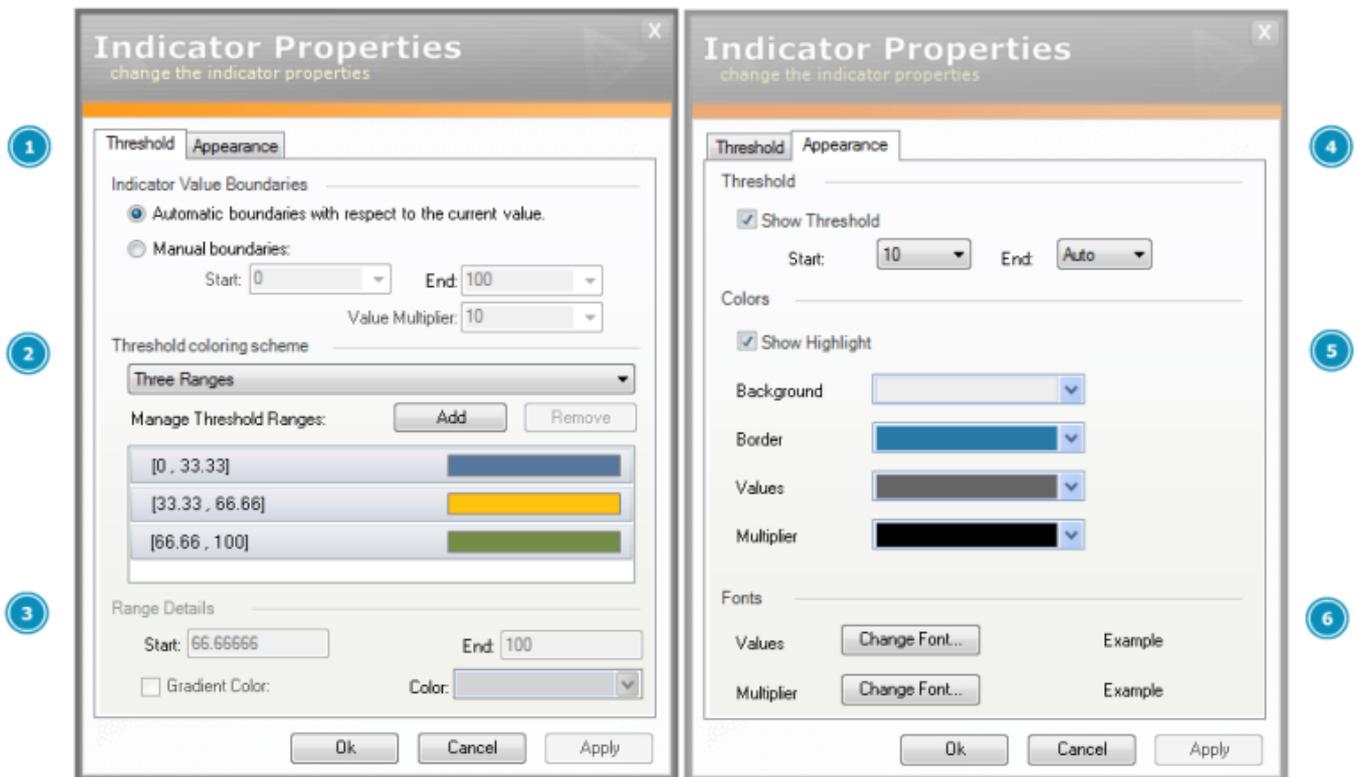
- Copy as Image: Copies the widget and data as an image which can be pasted into other applications.
- Save to Repository: Saves the widget to the widget repository in the data browser. Widgets in the repository can be used in other sheets or dashboards.

Widget Settings

- Lock for Layout Changes: Fixes the position of the widget.
- Show Shortcuts Header: Enable or disable to show or hide the shortcuts header.
- Show Title: Enable or disable to show or hide the widgets title.
- Preferences: Set color thresholds and conditional settings.

Indicator Preference Menu

The preference menu enables configuration of threshold ranges and appearance as well as format of the indicators appearance.



1. Value Boundaries: Specify the value boundaries for the indicator. (If the measure is a percentage enter 0 to 1 from 0 to 100%.)
2. Threshold Range: Add or remove thresholds. Format Threshold range and color setting.
3. Range Details: Specify start and end value of the threshold range.
4. Threshold display: Specify start and end value of range.
5. Colors: Set background, border, value and multiplier colors.
6. Fonts: Set value and multiplier colors.

Creating a Threshold

1. Right click on the indicator and select preferences.
2. Click the 'Thresholds' tab and select the number of thresholds from the 'Thresholds coloring scheme' drop down. Select custom to manually specify the number of thresholds.
3. Click 'Add' to create a threshold.
4. Click the relevant threshold and in the 'Range details' specify the start value, end value of the threshold and the threshold color.
5. Click 'Apply' then 'Ok' to save the threshold.

Selection & Input Widgets

SELECTION & INPUT WIDGETS

Input Selection widgets are widgets that are used for allowing users to change the data they are looking out by picking values in a list. In effect, what these Widgets do is dynamically apply Background on other widgets. There are several of Widgets like these, which you can create from the Widget Pane.

Filtering Widget Types

- Input Text Box: Enable manual input of text.
- Input Numeric Box: Enable manual input of numeric values.
- Input Member Search Box: Search Selectors are used to select Members based on text search patterns.
- Date Picker: Date Pickers are used to select one or more dates within a single month.
- Date Range Picker: Date Range Pickers are used to select a range of dates, starting and ending at specified periods.
- Members Picker: A Members Picker is used to select one or more Members from a drop down list.
- List: A List widget allows you to select one or more values out of a list, and optionally see the value of a measure for each particular item.

Filtering Widget Settings

To access the charts settings highlight the widget till the settings menu appears and click the 'Show Widgets Context Menu' icon.

Below is a detailed description of the selection & input setting options.

Shortcuts

- Copy: Click to copy the widget.
- Cut: Click to cut the widget.
- Remove: Click to delete the widget from the sheet.
- Ordering: Order the widgets position in relation to other widgets. See widget ordering.

Data

Widget Interconnectivity: Sets the connectivity of the widget to other widgets. See widget interconnectivity settings.

Actions

- Show the Data Browser: Opens the data browser panel.

- Copy Selected Text: Copies the widget text which can be pasted into other applications.
- Clear Selection: Clear selection in member picker
- Invert Selection: Select inverse items from current selection.

Widget Settings (One or more of the settings are available depending on the type of widget)

- Lock for Layout Changes: Fixes the position of the widget.
- Show Shortcuts Header: Enable or disable to show or hide the shortcuts header.
- Show Title: Enable or disable to show or hide the widgets title.
- Default Value: Default value to display in input box.
- Water Mark: Default text to display in input box.
- Minimum Value: Minimum value that can be entered in numeric input box.
- Maximum Value: Maximum value that can be entered in numeric input box.
- Rename: Rename widget
- Edit Tooltip Text: Edit text to display as tool tip.
- Show Find Button: Show/Hide find button in search box.
- Search Type: Search type to perform based on input.
- Set Auto Date Range: Set dynamic dates for example last 24 months.

More Settings: Selection type: Enable the selection type, includes:

- Single or multiple selection
- First or last member selection
- Show/ hide total and drop down area

Formatting

Text boxes, images and panels are used to give extra graphics to your dashboard such as labels, colored frames, company logos, etc. These widgets can be created from the widget pane, under the Formatting group.

Formatting Widget Types

Panel: Panels are used to add additional formatting and layout to a dashboard, such as frames and colored areas.

Text Box: Text Boxes are used to add formatted text to a dashboard.

Image: Image Widgets are used to add images to a dashboard.

Formatting Widget Settings

To access the charts settings highlight the widget till the settings menu appears and click the Show Widgets Context Menu icon.

Below is a detailed description of the chart setting options.

Shortcuts

- Copy: Click to copy the widget.
- Cut: Click to cut the widget.
- Remove: Click to delete the widget from the sheet.
- Ordering: Order the widgets position in relation to other widgets. See widget ordering.

Widget Settings (One or more of the settings are available depending on the type of widget)

- Lock for Layout Changes: Fixes the position of the widget.
- Show Shortcuts Header: Enable or disable to show or hide the shortcuts header.
- Show Title: Enable or disable to show or hide the widgets title.
- Edit Tooltip Text: Edit text to display as tool tip.
- Load Image: Select location of image to load into image widget.
- Reload Image: Select location of image to replace current image in image widget.

Formatting (Look & Feel) Widgets

Text boxes, images and panels are used to give extra graphics to your dashboard such as labels, colored frames, company logos, etc. These widgets can be created from the widget pane, under the Formatting group.

Formatting Widget Types

Panel: Panels are used to add additional formatting and layout to a dashboard, such as frames and colored areas.

Text Box: Text Boxes are used to add formatted text to a dashboard.

Image: Image Widgets are used to add images to a dashboard.

Formatting Widget Settings

To access the charts settings highlight the widget till the settings menu appears and click the 'Show Widgets Context Menu' icon.

Below is a detailed description of the chart setting options.

Shortcuts

- **Copy:** Click to copy the widget.
- **Cut:** Click to cut the widget.
- **Remove:** Click to delete the widget from the sheet.
- **Ordering:** Order the widgets position in relation to other widgets. See widget ordering.

Widget Settings (One or more of the settings are available depending on the type of widget)

- **Lock for Layout Changes:** Fixes the position of the widget.
- **Show Shortcuts Header:** Enable or disable to show or hide the shortcuts header.
- **Show Title:** Enable or disable to show or hide the widgets title.
- **Edit Tooltip Text:** Edit text to display as tool tip.
- **Load Image:** Select location of image to load into image widget.
- **Reload Image:** Select location of image to replace current image in image widget.

Widget Ordering

When two (or more) Widgets overlap, the order at which they appear is determined by the Widgets' relative order on the Z axis.

Generally, the Widget created last will appear on top of a Widget created earlier.

Apply Widget Ordering

From the widget settings menu

1. Click on the relevant widget and hover the mouse till the settings menu appears and click the '**Show Widgets Context Menu**' icon.
2. Under the '**Quick Shortcuts**' category click the 'Ordering' icon until the menu appears.
3. Select the appropriate ordering option. See the table below for ordering settings.

From the main menu

1. Click on the relevant widget to select.
2. On the top main menu click '**Format**'.
3. Hover over the '**Ordering**' option until the menu appears.
4. Select the appropriate ordering option. See the table below for ordering settings.

Option	Description
Bring To Front	Brings the Widget to the front
One Step Up	Brings the Widget one step towards the front
Send To Back	Sends the Widget to the back
One Step Down	Sends the Widget one step towards the back

Widget Alignment and Spacing

The alignment of a group of widgets can be formatted automatically.

Click on the relevant widget to select. For multiple widgets left click the mouse and draw a border around all relevant widgets. Alternatively

On the top main menu click 'Format'.

Hoover over the 'Alignment' option for aligning or 'Spacing' option for spacing until the menu appears.

Select the appropriate ordering option. See the table below for alignment and spacing settings.

Alignment		
Direction	Option	Description
Horizontal	Align Lefts	Aligns the left edges of the selected Widgets
Horizontal	Align Centers	Aligns the centers of the selected Widgets horizontally
Horizontal	Align Rights	Aligns the right edges of the selected Widgets
Vertical	Align Tops	Aligns the top edges of the selected Widgets
Vertical	Align Centers	Aligns the centers of the selected Widgets vertically
Vertical	Align Bottoms	Aligns the bottom edges of the selected Widgets

Spacing

Direction	Option	Description
Horizontal	Minimal Spacing	Sets minimal horizontal spacing between selected Widgets
Horizontal	Same Spacing	Sets the same horizontal spacing between selected Widgets
Vertical	Minimal Spacing	Sets minimal vertical spacing between selected Widgets
Vertical	Same Spacing	Sets the same vertical spacing between selected Widgets

Exporting Data and Dashboards

EXPORT WIDGET DATA TO EXCEL

To export a Pivot Table to Excel, follow the steps below:

Right click the relevant pivot table.

Highlight 'Export To Excel' until the menu appears.

Click the relevant export option. Available options are listed below.

Option	Description
All Cells	Exports all rows and values
Grouped	Uses one value to reference a Member when it appears multiple times consecutively
Merged	Merges adjacent cells that reference the same Member
Export	Begins the export process

Export Dashboard to PDF

EXPORT DASHBOARD TO PDF

On the main menu click 'File'> Click the 'Export to Pdf'
Click 'Browse' to specify the location to export the pdf.
Select the pages to export by enabling the associated checkbox.
Click 'Ok' to export to pdf.

Actions

Actions are widget which enable greater interactivity between sheets within a dashboard making it possible to apply settings from one sheet to another, navigate between sheets and work with data in multiple filters.

Refresh Document, Sheet & Widget

Refreshing queries the ElastiCube (data source) and returns the most recent results to widgets in the dashboard. Below is a list of refresh types.

Option	Description
Selected Widgets	Refreshes the Widgets currently selected
Active Sheet	Refreshes the entire active sheet
All Document	Refreshes the entire document
Specific Widgets	Refreshes a specific Widget

Refreshing can be done on three levels or automated.

Refresh Widgets

Refresh actions enable the user to click a button in order to refresh a widget, document or sheet.

1. In the widgets panel drag and drop either a refresh sheet, widget or document action.
2. For refresh widgets actions, right click the refresh button and on the 'Action parameter' click the widgets to refresh.

Refresh Menu

1. On the top main menu select the 'refresh' icon on the far right.
2. Click the appropriate widget, sheet or document to refresh.

Automatic Refresh

1. On the top main menu select 'File'>' Schedule Automatic Refresh'
2. A popup menu will appear.
3. Enter the day, time month or interval for the refresh.
4. Click 'OK' to save changes.

Note: Automatic refresh only works when the application is running and in View mode (F7 or through the View->Open In Viewer).

Clear Selection

Clear actions enable the user to click a button in order to clear the selection in a widget(s).

1. In the widgets panel drag and drop the 'Clear selection' widget onto the sheet.
2. Right click the clear selection button and in the 'Action parameter' click the widgets to clear.

Jump To Page

Action - Jump to Page

The 'Jump to Page' actions enable the user to click a button in order to move to another sheet.

1. In the widgets panel drag and drop the 'Jump to Page' widget onto the sheet.
2. Right click the jump to page button and in the 'Sheet-to-jump-to' click the sheet to jump to.

Apply Scope

ACTION - APPLY SCOPE

The 'Apply Scope' actions enable the user to click a button in order to apply a scope from one widget to another widget.

1. In the widgets panel drag and drop the 'Apply Scope' widget onto the sheet.
2. Right click the apply scope button and in the 'Input widgets' click the widget to act as the scope and in the 'Affected widget' click the widget to apply to scope on.

Drill To Sheet

ACTION - DRILL TO SHEET

The 'Drill to Sheet' actions enable the user to click a button in order to apply the selection from one or more widgets to another sheet.

1. In the widgets panel drag and drop the 'Drill to Sheet' widget onto the sheet.
2. Right click the drill to sheet button and in the 'Destination sheet' select the sheet to apply the selection and in the 'Input selection widgets' click the widget selections to apply to the sheet.

Drill to New Sheet

ACTION - DRILL TO NEW SHEET

The 'Drill to New Sheet' actions enable the user to click a button in order to apply the selection from one or more widgets to a duplication of another sheet.

1. In the widgets panel drag and drop the 'Drill to New Sheet' widget onto the sheet.
2. Right click the drill to new sheet button and in the 'Destination sheet' select the sheet to duplicate and apply the selection and in the 'Input selection widgets' click the widget selections to apply to the sheet.

Browse To

ACTION - BROWSE TO

The 'Browse To' actions enable the user to click a button in order to take selections from widgets and browse to a website or perform a search.

1. In the widgets panel drag and drop the 'Browse To' widget onto the sheet.
2. Right click the browse to button and in the 'Input parameter widgets' click the widget to act as the input for browsing in the 'URL Structure' enter the URL structure to browse to.
3. To perform a Google search based on the input of widgets user the following syntax:
`http://www.google.com/search?q={0} + {1} + {2}`
For each additional widget to include in the search string us '+ {n}' where n is the next incremental widget being used.

Export to Excel

ACTION - EXPORT TO EXCEL

The 'Export to Excel' actions enable the user to click a button in order to export widget data to Excel.

1. In the widgets panel drag and drop the 'Export to Excel' widget onto the sheet.
2. Right click the export to excel button and in the 'Widget to export' click the widget to export data.

Creating and Editing Data Hierarchies

Data hierarchies enable you to create dynamic views to a broader or narrower data segment by drilling into the data set.

1. To create data hierarchies in a pivot right click on the Pivot or chart and select '**Edit Data Hierarchies**'
2. The Data Hierarchies editor will open allow you to add dimensions under the selected dimension.
3. Click '**Add Hierarchy**', a new hierarchy will appear on the left panel.
4. Next to the new hierarchy click '**Rename**' to name or '**Delete**' to remove.
5. To create the levels of a new hierarchy, click the relevant hierarchy then drag and drop dimensions from the data browser into the right panel.
 - Dimension placed at the top of the hierarchy constitute the first level of the hierarchy with each proceeding dimension creating a lower level.
6. To delete or edit a level click the hierarchy and in the right panel select the relevant dimension and select '**Remove**'
7. Once completed click '**OK**' to save the hierarchy.

Accessing Existing Data Hierarchies

- Select the relevant pivot or chart.
- Right click the relevant column or axis and select '**Drill Down**' or '**Drill Up**' to segment the data and explore the hierarchy. Selecting '**Drill Down To**' or '**Drill Up To**' enables you to specify the level of the hierarchy to apply.
- The '**Expand**' or '**Expand To**' options can be clicked to add a hierarchy level into the pivot or chart.
- The '**Switch**' or '**Switch To**' option can be clicked to change the column with a dimension from the data hierarchy.

Widget Interconnectivity

A direct link can be created between widgets whereby a selection in one widget can affect the data presented in another.

For example a selection from a member picker may be set to affect the results displayed in a pivot table but not influence data in any other widget. The steps below describe the process to enable

1. Right click on the widget and select 'Widget Interconnectivity'>'Available Widgets'
2. Select the relevant widget and click to enable or if checked disable..
3. Click 'Apply' to save changes.

For pivots and charts ensure the 'Widget Selection Mode' is enabled if you are using the pivot or chart as the selector. Selections in the inter-connected widget will now affect the data represented in the widget.

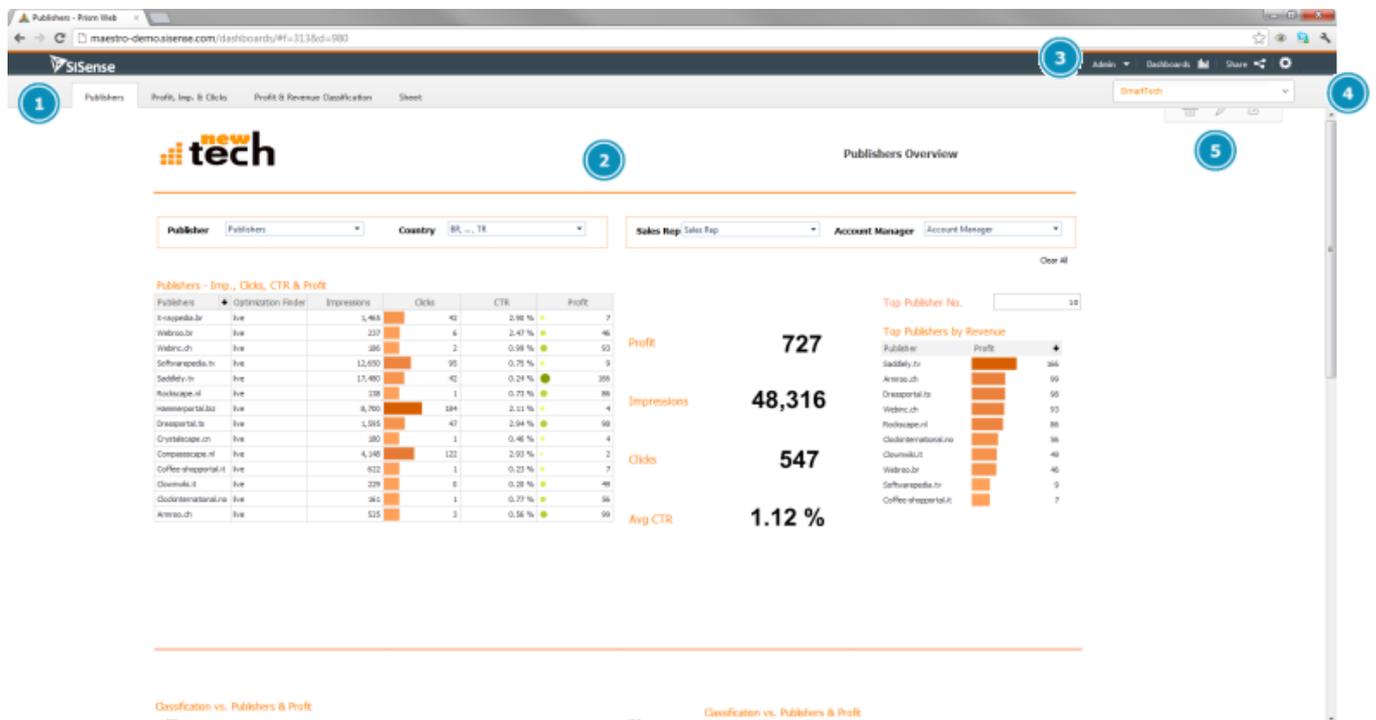
Prism Web

INTRODUCTION TO PRISM WEB

Introduction

Prism Web is an interactive web environment to share and interact with dashboards. The web environment is accessible using any major web browser. It provides control to manage users, dashboards and data sources.

Prism Web Environment



1. Sheets tab
2. Interactive web dashboard
3. Configuration and sharing menu
4. Dashboard selector
5. Edit and print controls

Re-arrange Widgets

Widgets can be moved and resized dynamically in the web environment.

Edit Widgets

1. On the top right click the 'Edit' icon.
2. All editable widgets will be highlighted and a border will appear around each widget indicating its current width and height settings.

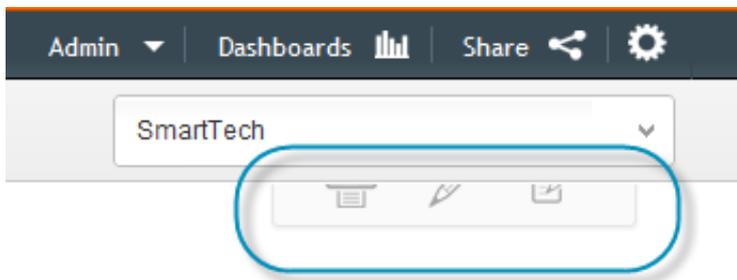


Image: Click 'Edit' to re-arrange dashboard widgets

Adjusting a Widgets Position:

Place the mouse cursor over the widget then left click and hold the button while moving the widget to a new position.

Resizing a Widget:

- Re-size Height: Place the mouse cursor at the bottom border of the widget until the cursor changes to a dual arrow cursor. Left click and hold the mouse button while simultaneously moving the mouse up or down to adjust the height.
- Re-size Width: Place the mouse cursor at the left or right border of the widget until the cursor changes to a dual arrow cursor. Left click and hold the mouse button while simultaneously moving the mouse left or right to adjust the width.
- Re-size Height and Width Simultaneously: Place the mouse cursor at any corner of the widgets border until the cursor changes to a dual arrow cursor. Left click and hold the mouse button while simultaneously moving the mouse in a diagonal direction to adjust both the height and weight.

3. To save any changes made to the widgets position or size click 'Exit Edit Mode' in the top right of the screen.

Share Dashboards with Others

A dashboard can be shared with multiple users. Shared dashboards can be accessed via a web browser regardless of a user's location.

1. In Prism Web click the dashboards icon on the far right and select the relevant dashboard from the drop down menu.
2. Once the dashboard has loaded click the share icon on the top right.
3. In the input box enter the email address of all recipients with whom you wish to share the dashboard. Press enter after typing each email address. To delete an entered email address click the cross next to the recipient's email.
4. Once you have entered all recipients click 'Save Changes'.
5. An email will be sent to the recipient with a link to access the dashboard.
6. Clicking on the link will open a web browser and direct the user to your dashboard. First time users will be prompted to choose a password before logging into the dashboard.

SEE ALSO

[Private and Public Dashboards](#)

[User Management](#)

Download Widget Data as Excel

In certain cases you may wish to export data contained within a dashboard to perform further analysis or use in presentations. Widget data from pivots and charts can be exported to Excel.

1. To export data contained within a pivot or chart place the mouse cursor over the widget until a set of icons appears on the top right of the pivot.
2. Click the icon to the right 'Export to Excel'.
3. In your browser either an Excel file will be downloaded or you will be prompted to save or open the file. In certain cases you may have to allow the download.
4. In most cases files are automatically saved in the Downloads folder. This folder is usually located on the drive where Windows is installed (for example, C:\users\your name\downloads).
5. Clicking on the downloaded file will open the data in Excel.

The screenshot shows the SiSense dashboard interface. At the top, there is a navigation bar with the SiSense logo and several menu items: 'Publishers', 'Profit, Imp. & Clicks', 'Profit & Revenue Classification', and 'Sheet'. Below the navigation bar is the 'new tech' logo. The main content area features a filter bar with 'Publisher' set to 'Publishers' and 'Country' set to 'BR, ..., UK'. Below the filter bar is a table titled 'Publishers - Imp., Clicks, CTR & Profit'. The table has columns for Publishers, Optimization Finder, Impressions, Clicks, CTR, and Profit. A blue circle highlights the 'Export to Excel' icon (represented by a document with a download arrow) in the top right corner of the table.

Publishers	Optimization Finder	Impressions	Clicks	CTR	Profit
Webroo.br	live	237	6	2.47 %	46
Webinc.ch	live	186	2	0.99 %	93
Shippedia.com	live	22,536	110	0.49 %	75
Rockscape.nl	live	138	1	0.73 %	86
Ringrati.pl	live	584	13	2.31 %	52
Mistscape.es	live	2,634	49	1.85 %	31

Image: Export widget data to Excel

Manage your Profile

Users can view and edit their profile and security settings by following the steps below:

1. Click the 'Admin' icon on the top right then select 'Edit Profile'.
2. A popup will appear with profile and security information.
3. The following settings can be edited.

- **Update Email and Personal Details:**

In the tab labeled 'The Basics' you can edit details on your name and email address. Additional profile information on department, job and gender details can be added or edited in the 'Additional Info' tab. Click 'Save Changes' once you have complete updating your profile.

- **Change Password:**

Click the tab labeled 'Security' and select 'Change Password' enter your current password followed by your new password and confirmation of the new password. Click 'Save Changes' to update your password or 'Cancel' to revert to existing settings.

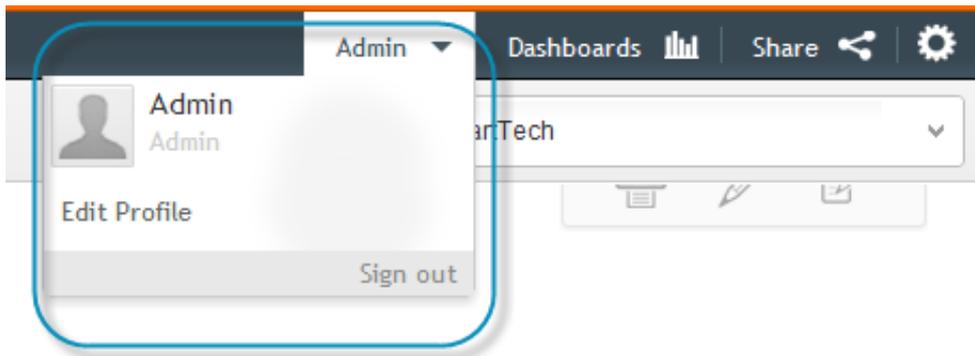


Image: Manage profile

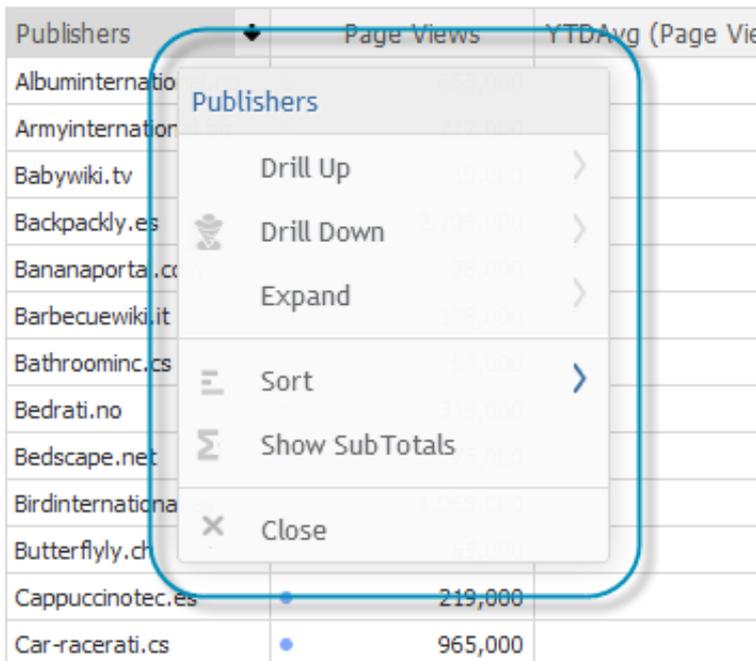
Adding and Removing Fields

Data hierarchies and drill downs enable users to dynamically add or remove fields as well as drill up or down to explore data sets. These features can be accessed in pivot tables and charts in Prism Web.

Right click on the column header in a pivot table or measure within a graph you wish to explore.

1. A menu will appear with options to view additional fields or more segmented data. One or more of the options below can be selected. Outcomes from the selection below are determined by existing data hierarchies.
 - Drill Down: View a narrower segment of the data. For example drilling from years to months.
 - Drill Up: View a broader segment of the data. For example drilling from days to months.
 - Expand: Inserts an additional column alongside existing columns. For example viewing a quarterly breakdown alongside annual data.
 - Switch To: Replaces the current column with another dimension. For example replace a view of region and revenue data with a view of sales representative and revenue data.
 - Hide: Hide the selected column.
 - Sort: Sort column data in ascending or descending order.
 - Show Totals: Show column totals.
 - Hide Totals: Hide column totals.
 - Close: Close the menu.

Profit vs Page Views & Visitors



Publishers	Page Views	YTD Avg (Page Vie
Albuminternatio		
Armyinternation		
Babywiki.tv		
Backpackdy.es		
Bananaportal.co		
Barbecuewiki.it		
Bathroominc.cs		
Bedrati.no		
Bedscape.net		
Birdinternational		
Butterflyfly.ch		
Cappuccinotec.es	219,000	
Car-racerati.cs	965,000	

Image: Adding and removing fields

Manage Users Accounts

Administrators can view and manage all users with access to Prism Web Dashboards.

1. Login to Prism Web as an administrator.
2. Click the top right configuration icon (cog) and select 'Manage users'.
3. One or more of the following actions can be performed.
 - **Add** a New User: click the 'Add New User' button, enter the user email address and press the plus icon.
 - **Edit** an Existing User: Locate the relevant user account, highlight the user and click the 'Edit' icon.
 - **Delete** a User: Locate the relevant user account, highlight the user and click the 'Delete' icon.
 - **Search**: Enter the user details or email on the top right search box and press enter.

4. When editing a user the following settings are available:

Update Email and Personal Details: In the tab labeled 'The Basics' you can edit details on your name and email address. Additional profile information on department, job and gender details can be added or edited in the 'Additional Info' tab.

Permissions : The following permissions can be enabled by ticking the associated check box.

- *Edit* - Edit dashboards
- *Delete* - Delete dashboards
- *Share* - Share dashboards
- *Publish* - Publish dashboards to web environment
- *Admin* - Give administrator rights.

Security: Change Password : Click the tab labeled 'Security' and select 'Change Password' enter a new password and confirmation of the new password. Click 'Save Changes' to update the password or 'Cancel' to revert to existing settings.

5. Click 'Save Changes' once the profile has been updated.

Prism Web Folder Management

Prism Web catalogues all published dashboards into folders.

Access a Folder

1. On the top right click the folder drop down.
2. Select and click the relevant dashboard to open it.
3. The selected dashboard will open in the browser.

Delete a Folder or Sheet

1. On the top right click the folder drop down.
2. Highlight the relevant dashboard and click the cross icon to the right.
3. Click the delete button that appears to remove the dashboard folder.



Manual Configuration for Prism Web



Instruction below outline deployment in the following environments:

- Windows 2003 R2 with IIS6
- Windows Server 2008 with IIS 7, 7.5
- Windows Vista\7

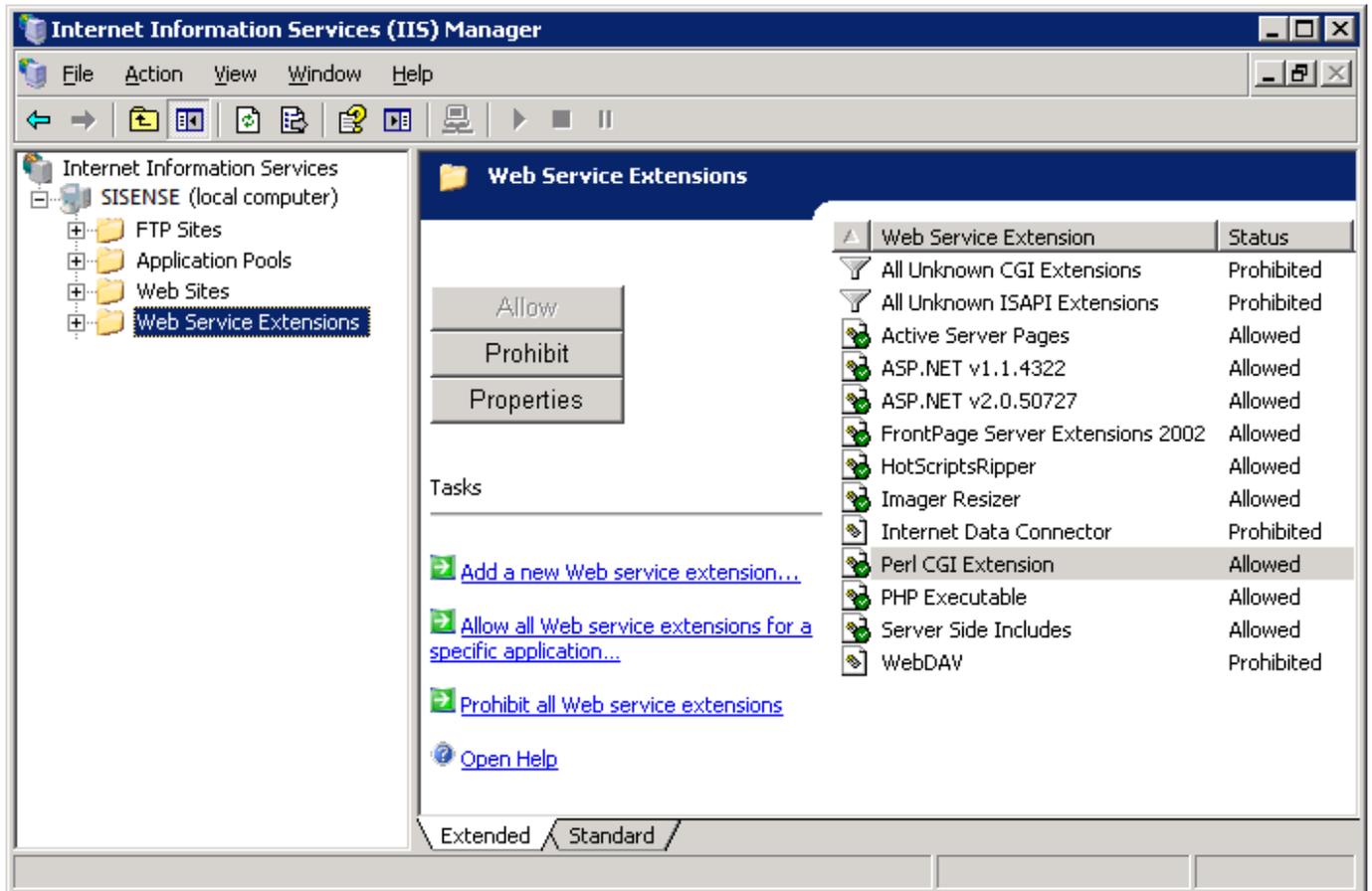
Windows 2003 R2 with IIS6

The following IIS configurations are based on the server operating system:

Windows Server 2003 R2 with IIS 6 Installing IIS in Windows Server 2003 requires your installation CD-ROM or original installation files. Please have the installation files ready before trying to setup IIS. Sample Screen shots are taken from Windows Server 2003 Standard Edition. Please follow the steps shown below to setup

To open Internet Information Services Manager in your start menu click Run and type inetmgr. Expand your server node and click Web Service Extensions. In the right pane, make sure ASP.NET v4.0 extension is in status Allowed.





Creating the Website

1. After installing PrismWeb, create a new website named "PrismWeb" using the Web Site Creation Wizard. - IP: All Unassigned

- port: available TCP port (default :80)

- Host header: leave blank or host name

- Path:

64-bit version : "C:\Program Files \(\x86\)Sisense\PrismWeb"

32-bit version : "C:\Program Files \Sisense\PrismWeb"

- Access Permissions: Read, Execute

Click Application Pools and create a new one called "PrismWeb".

Web Site Creation Wizard [X]

Web Site Description
Describe the Web site to help administrators identify it.

Type a description of the Web site.

Description:

< Back **Next >** Cancel

Web Site Creation Wizard [X]

IP Address and Port Settings
Specify an IP address, port setting, and host header for the new Web site.

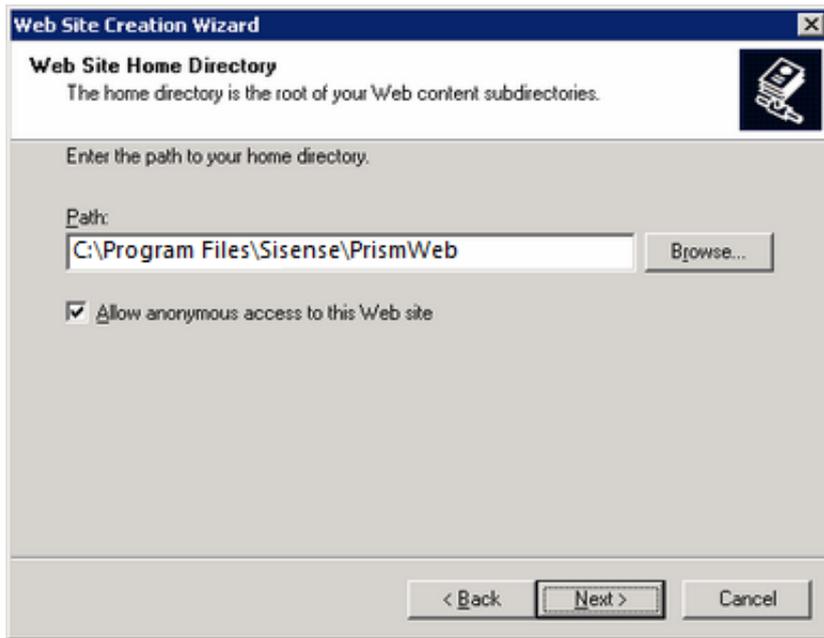
Enter the IP address to use for this Web site:

TCP port this Web site should use (Default: 80):

Host header for this Web site (Default: None):

For more information, read the IIS product documentation.

< Back **Next >** Cancel



Creating the Website

2. Right-click on the PrismWeb Website and Properties:

Tab: ASP.NET

ASP.NET version: 4.0.30319

Tab: Home Directory

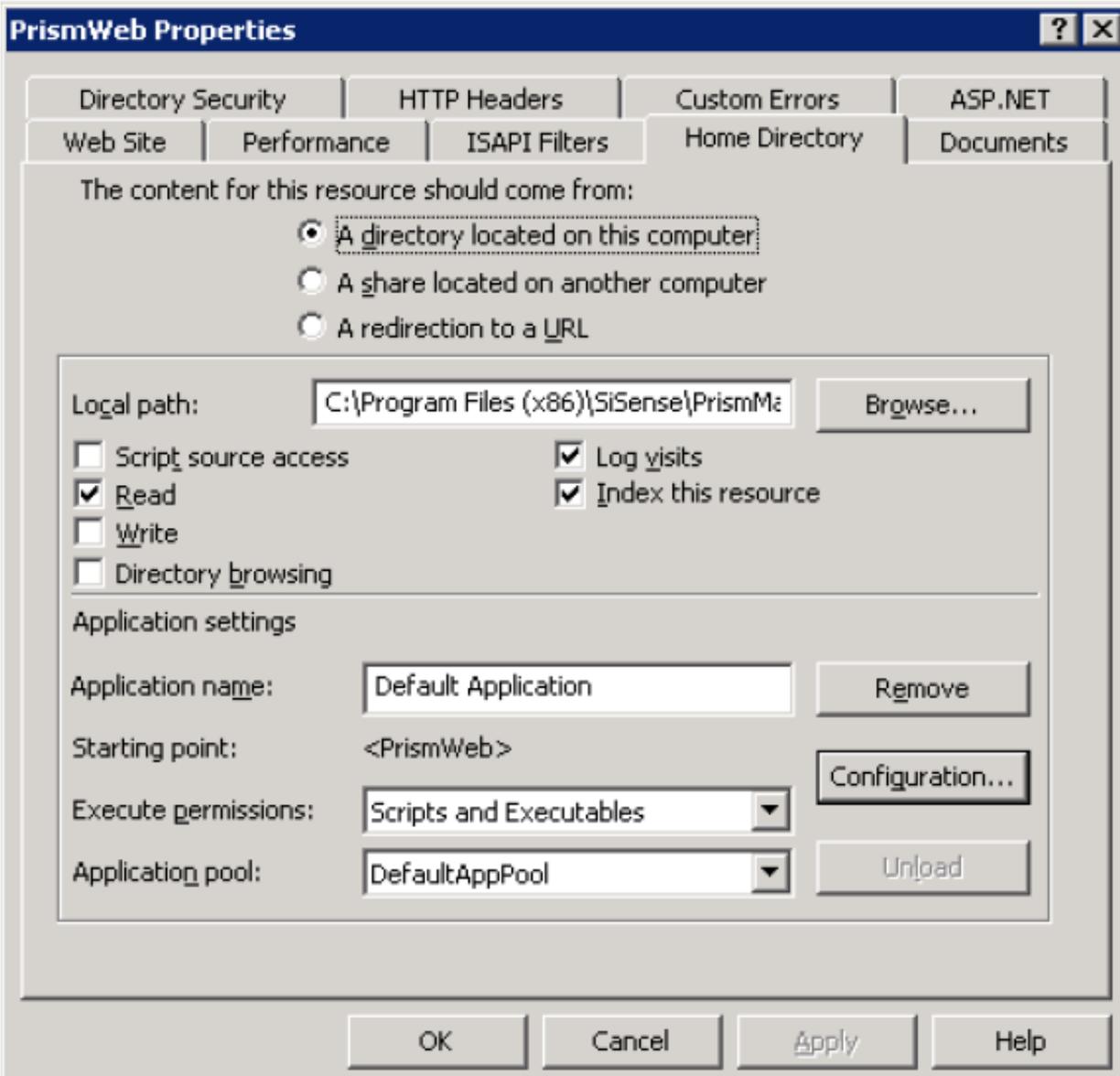
1. Click Configuration... button.

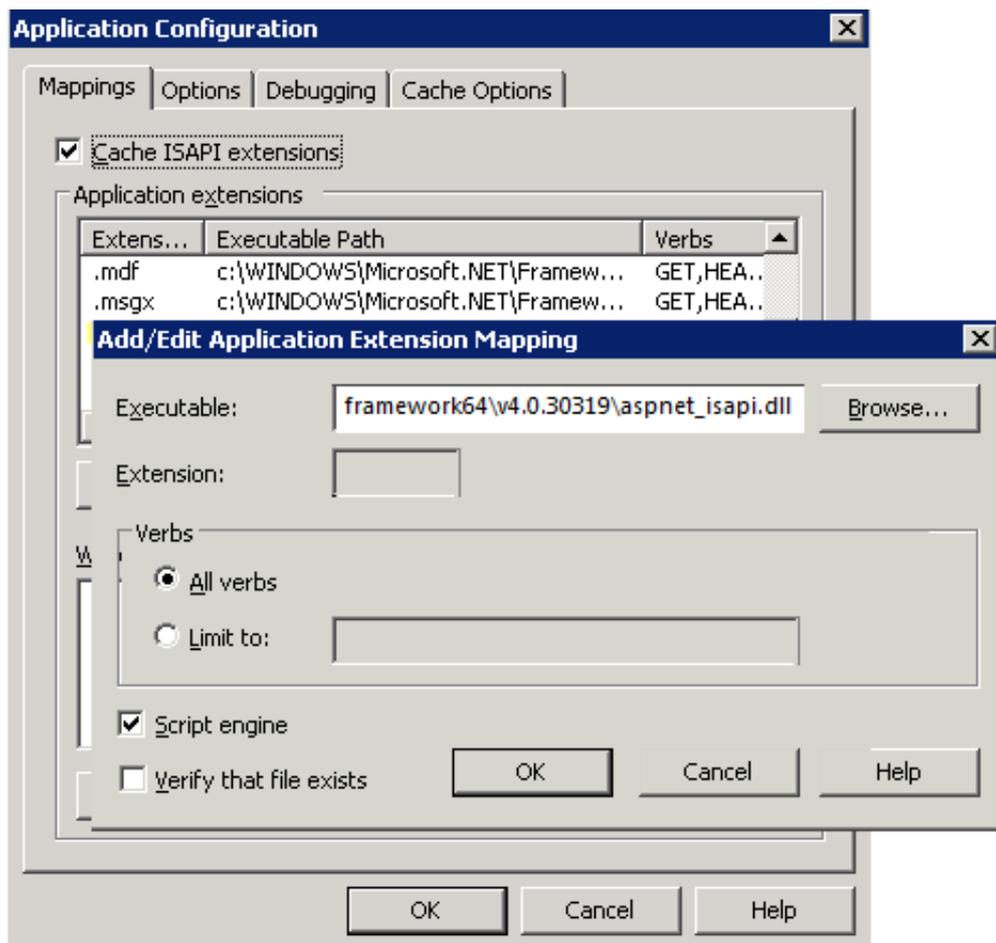
2. On Mappings tab, make sure you have mappings to the .svc extension and make sure their Executable Path is set to ~\aspne_isapi.dll. Click Edit... and copy the path.

3. Click Insert on the buttons below.

4. Paste the executable path and uncheck the "Verify if file exists".

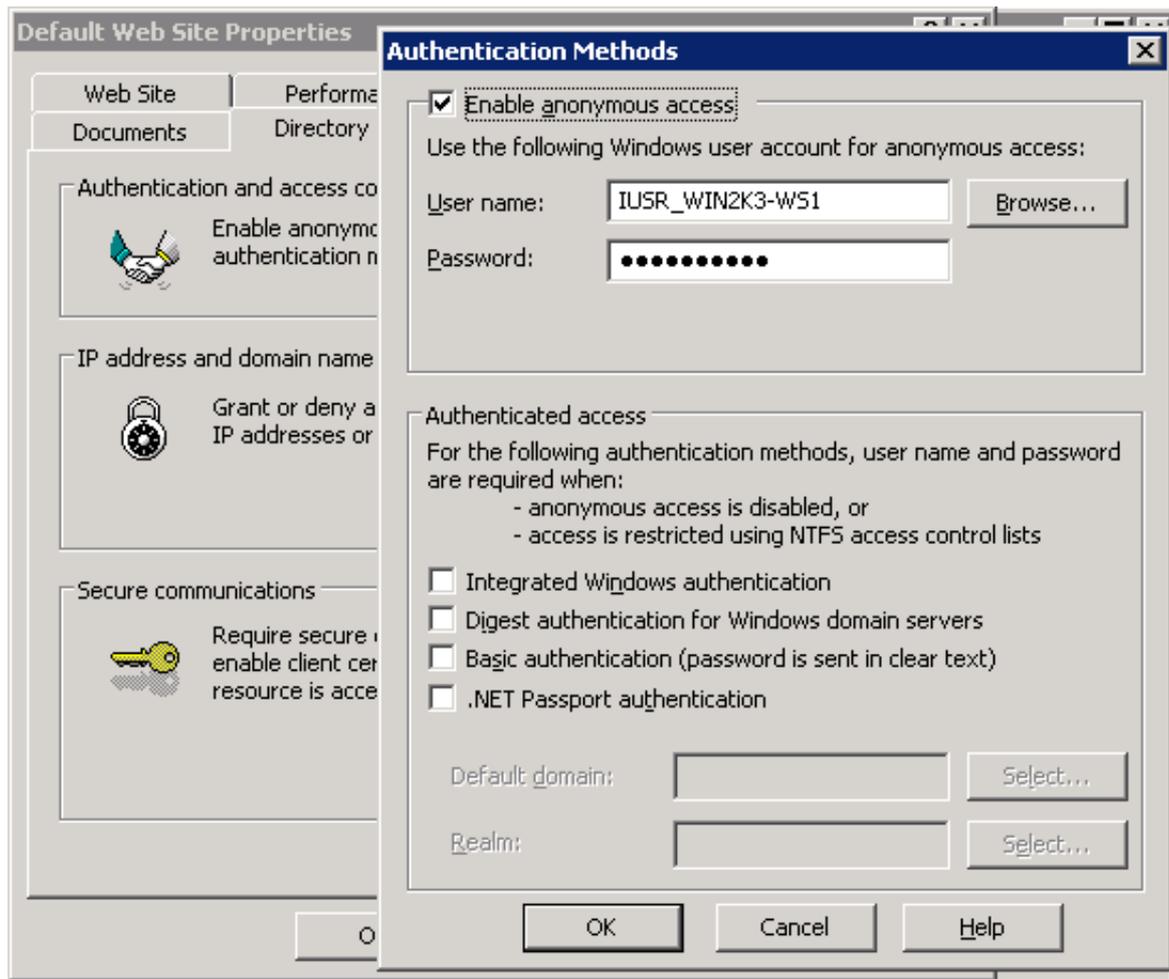
5. On Application pool, select PrismWeb





Tab: Directory Security

1. In Authentication and access control, click Edit. The Authentication Methods window appears.
2. Make sure that Enable anonymous access checkbox is selected and that none of the checkboxes in Authenticated access is selected.



Windows Server 2008 with IIS 7, 7.5

IIS comes preloaded in Windows 2008 Server. Please follow the steps shown below to setup. Sample Screenshots are taken from Windows 2008 Server R2 Enterprise Edition.

1. Go to Start -> Control Panel -> Programs -> Turn Windows Features on or off

2. Select and expand Roles tab.

3. If you do not see Web Server (IIS) role, you must add it using the Add Roles button:

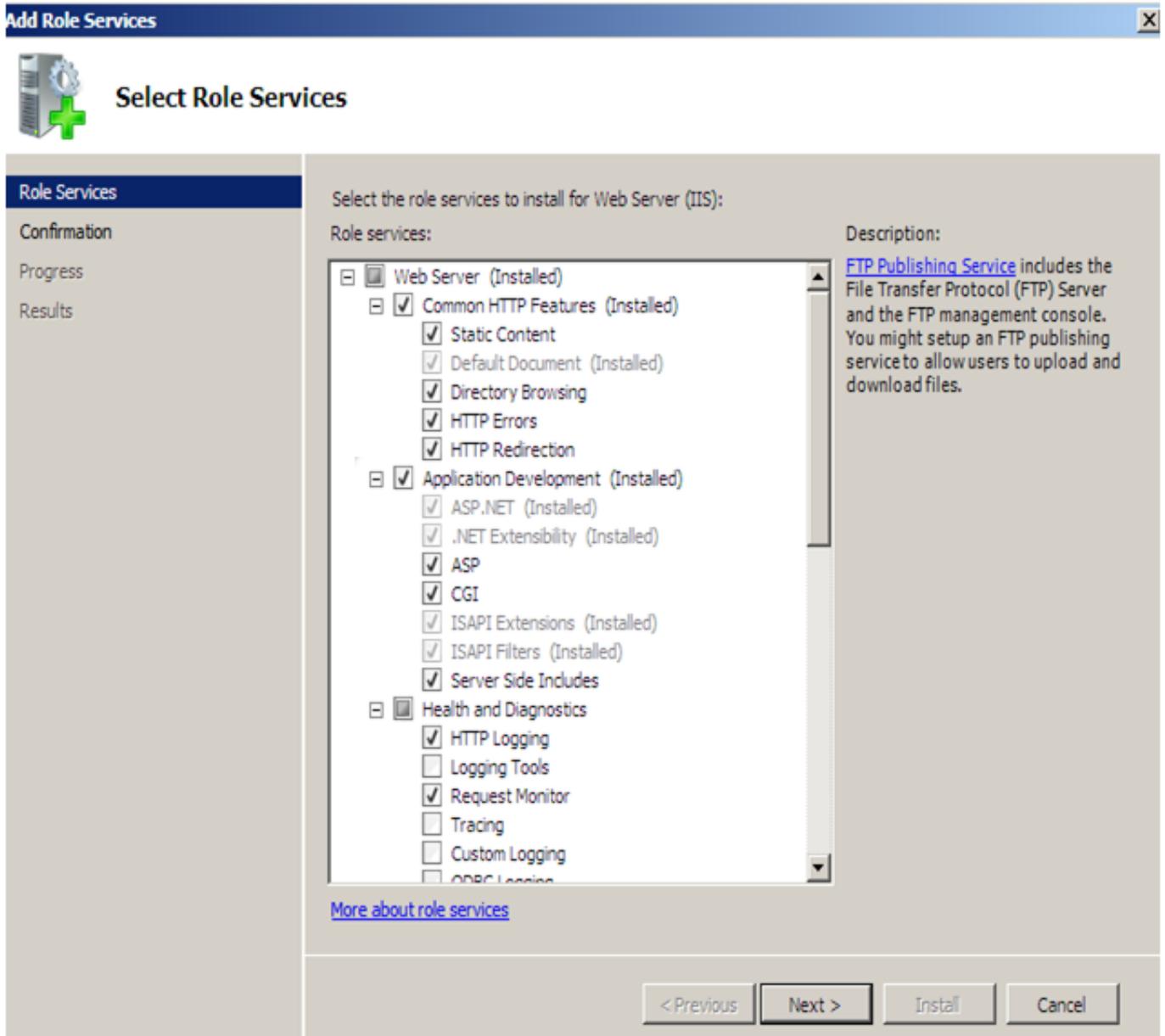
- Click Web Server (IIS) and in the right pane, expand Role Services.
- Make sure the following services are installed:
 - Common HTTP Features? Static Content.
 - Application Development? ASP.NET, .NET Extensibility, ISAPI Extensions and ISAPI Filters.

- Management Tools ?IIS Management Console, IIS Management Scripts and Tools, and Management Service.

4. In the navigation tree on the left, click Features.

5. Make sure the following feature added:

.NET Framework 3.5.1 Features ?WCF Activation? HTTP Activation.



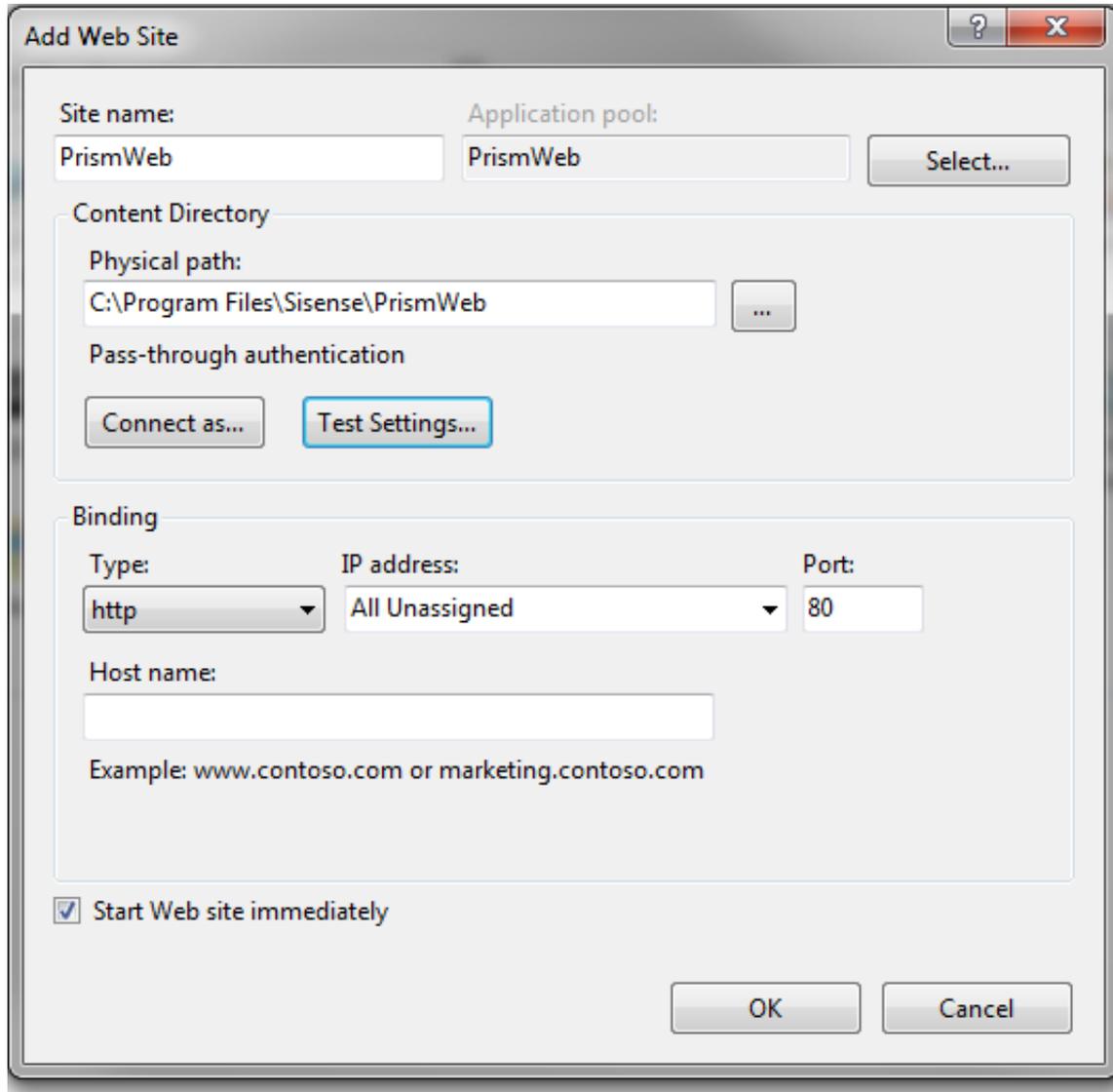
Creating the Website

1. Expand your server node. Right click "Sites"?Add Web Site. - Site name: PrismWeb - Port: available TCP port (default :80)

- Physical Path:

64-bit version: "C:\Program Files (x86)\Sisense\PrismWeb"

32-bit version: "C:\Program Files\Sisense\PrismWeb"



2. On the left pane select "Application pools"

3. Click "PrismWeb".

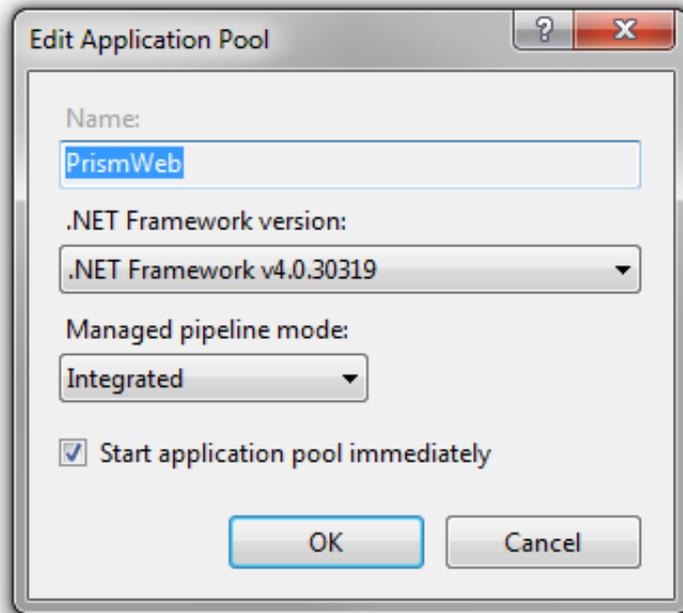
4. Select:

“.Net Framework version” : Net Framework v4.0.0319

“Managed pipeline mode” : Integrated

5. Select on the right pane “Advanced Settings”.

Note : when using 64 bit installation make sure that under General ? Enable 32-Bit Application is False



6. Expand Sites node and click PrismWeb.

7. In section IIS on the right, double-click Authentication.

8. Make sure Anonymous Authentication and Forms Authentication are enabled and all other authentications are disabled.

Authentication

Name	Status	Response Type
Anonymous Authentication	Enabled	
ASP.NET Impersonation	Disabled	
Basic Authentication	Disabled	HTTP 401 Challenge
Digest Authentication	Disabled	HTTP 401 Challenge
Forms Authentication	Enabled	HTTP 302 Login/Redirect
Windows Authentication	Disabled	HTTP 401 Challenge

Windows 7

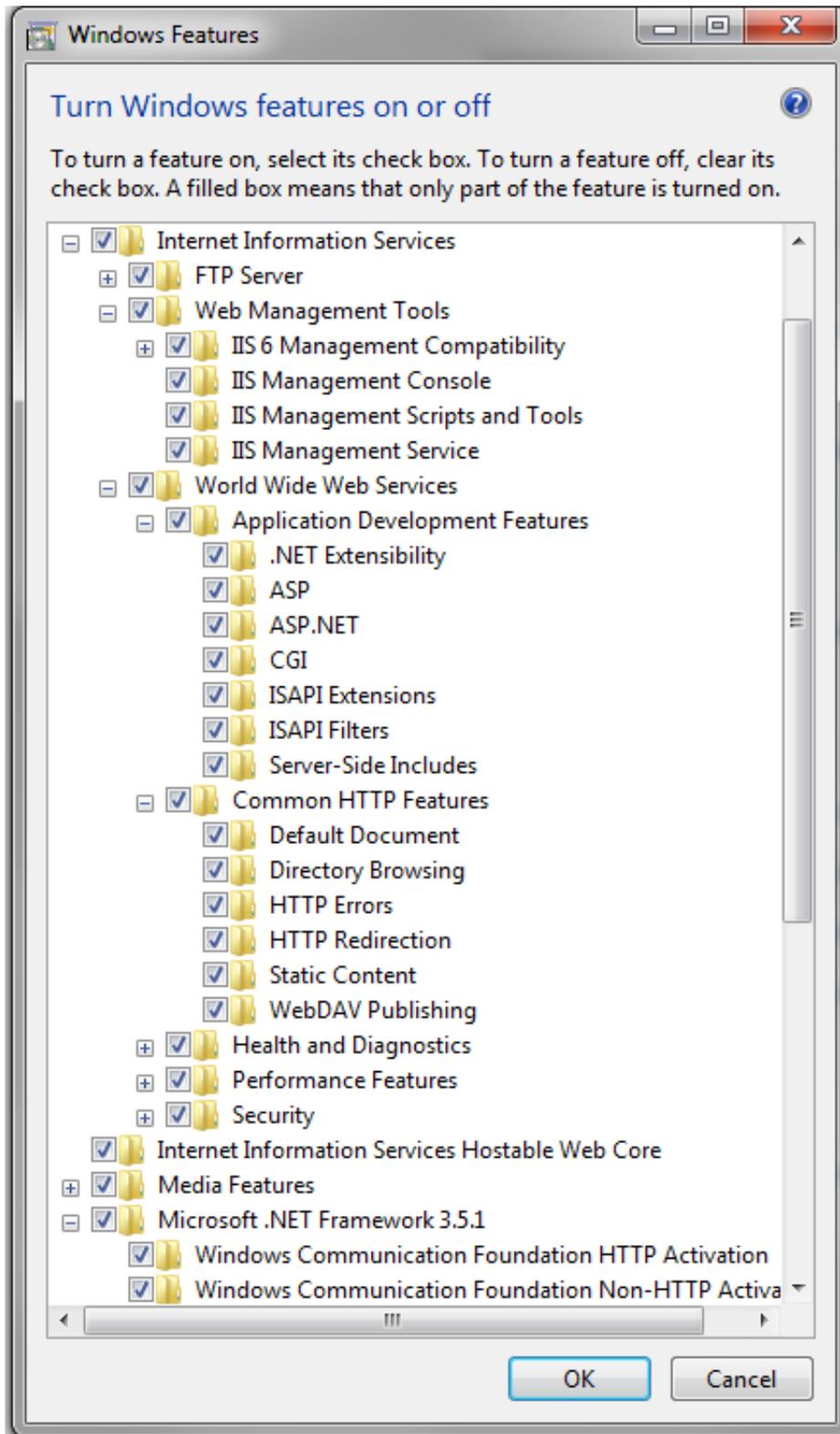
IIS comes preloaded in Windows 7. Please follow the steps shown below to setup. Sample Screenshots are taken from Windows 7 Professional Edition.

1. Go to Start -> Control Panel -> Programs -> Turn Windows

Features on or off .

2. Turn on Internet Information Services.

3. Under Internet Information Services, turn on the features as in the screen shot:



1. After installing the PrismWeb, Open Computer Management. under Services and Applications select Internet Information Services. In the navigation of the IIS manager, expand your server node. Right click "Sites"?Add Web Site:

- Site name: PrismWeb

- Port: available TCP port (default :80)

- Physical Path:

64-bit version: "C:\Program Files \x86)\Sisense\PrismWeb"

32-bit version: "C:\Program Files \Sisense\PrismWeb"

2. On the left pane select "Application pools"

3. Click "PrismWeb".

4. Select:

“.Net Framework version” : Net Framework v4.0.0319

“Managed pipeline mode” : Integrated

5. Select on the right pane "Advanced Settings".

Note : when using 64 bit installation make sure that under General ? Enable 32-Bit Application is False

Add Web Site

Site name: PrismWeb Application pool: PrismWeb

Content Directory

Physical path: C:\Program Files\Sisense\PrismWeb

Pass-through authentication

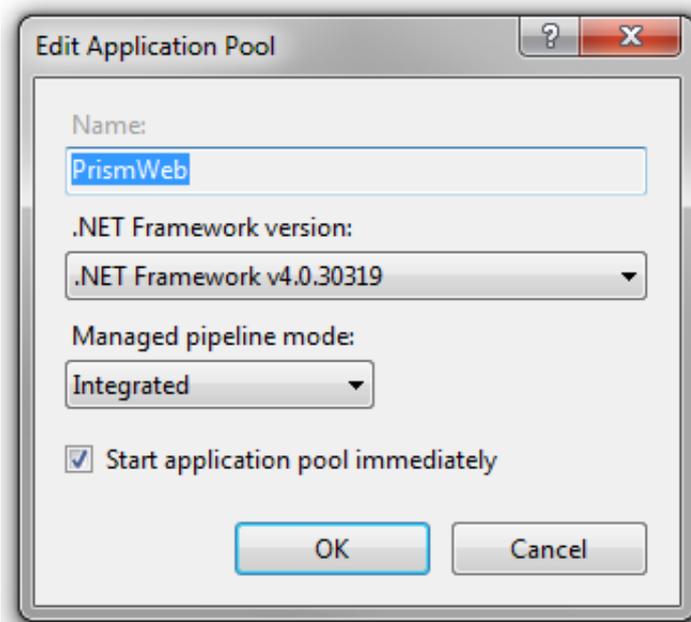
Binding

Type:	IP address:	Port:
http	All Unassigned	80

Host name:

Example: www.contoso.com or marketing.contoso.com

Start Web site immediately

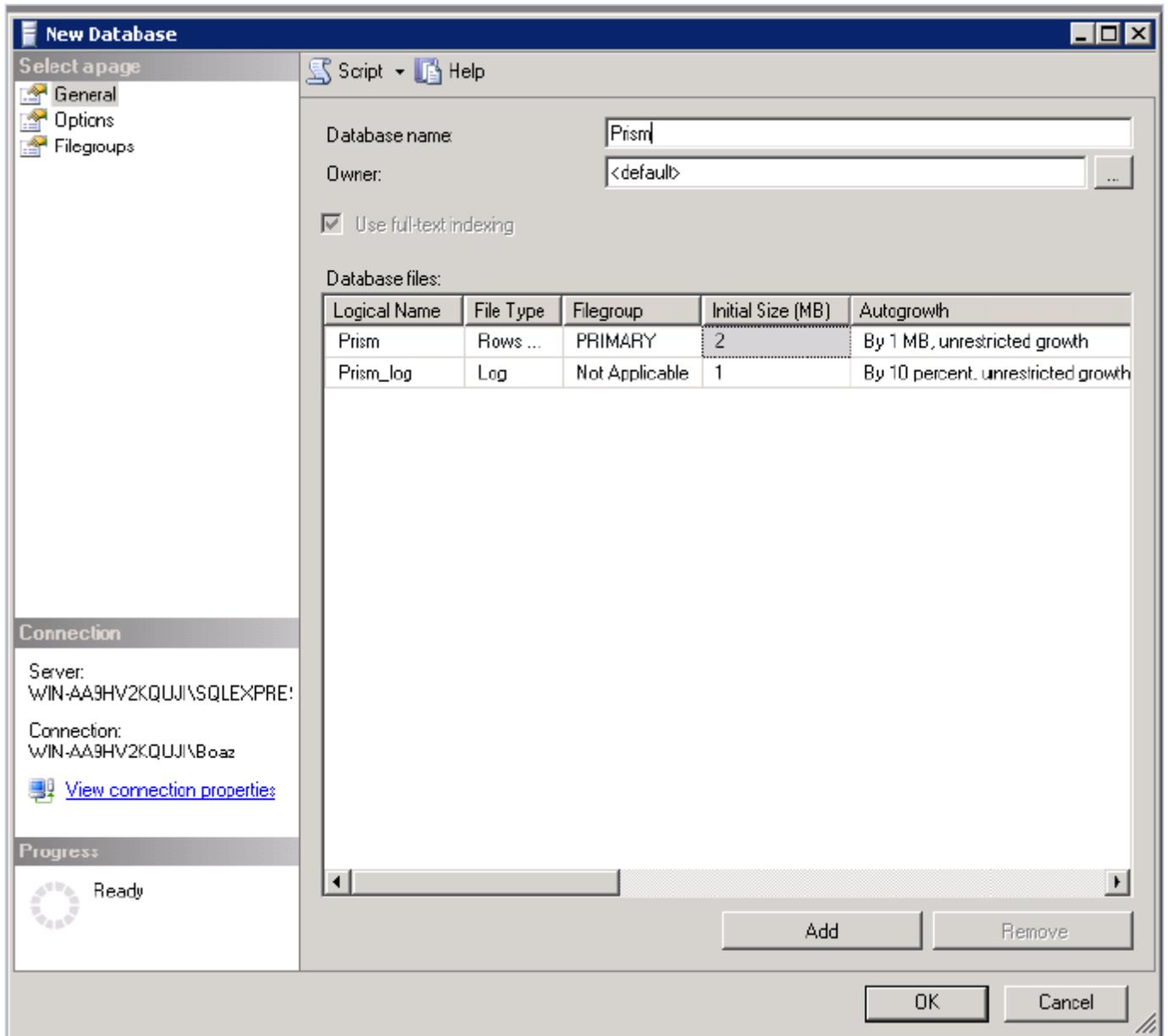


Configuring the Database

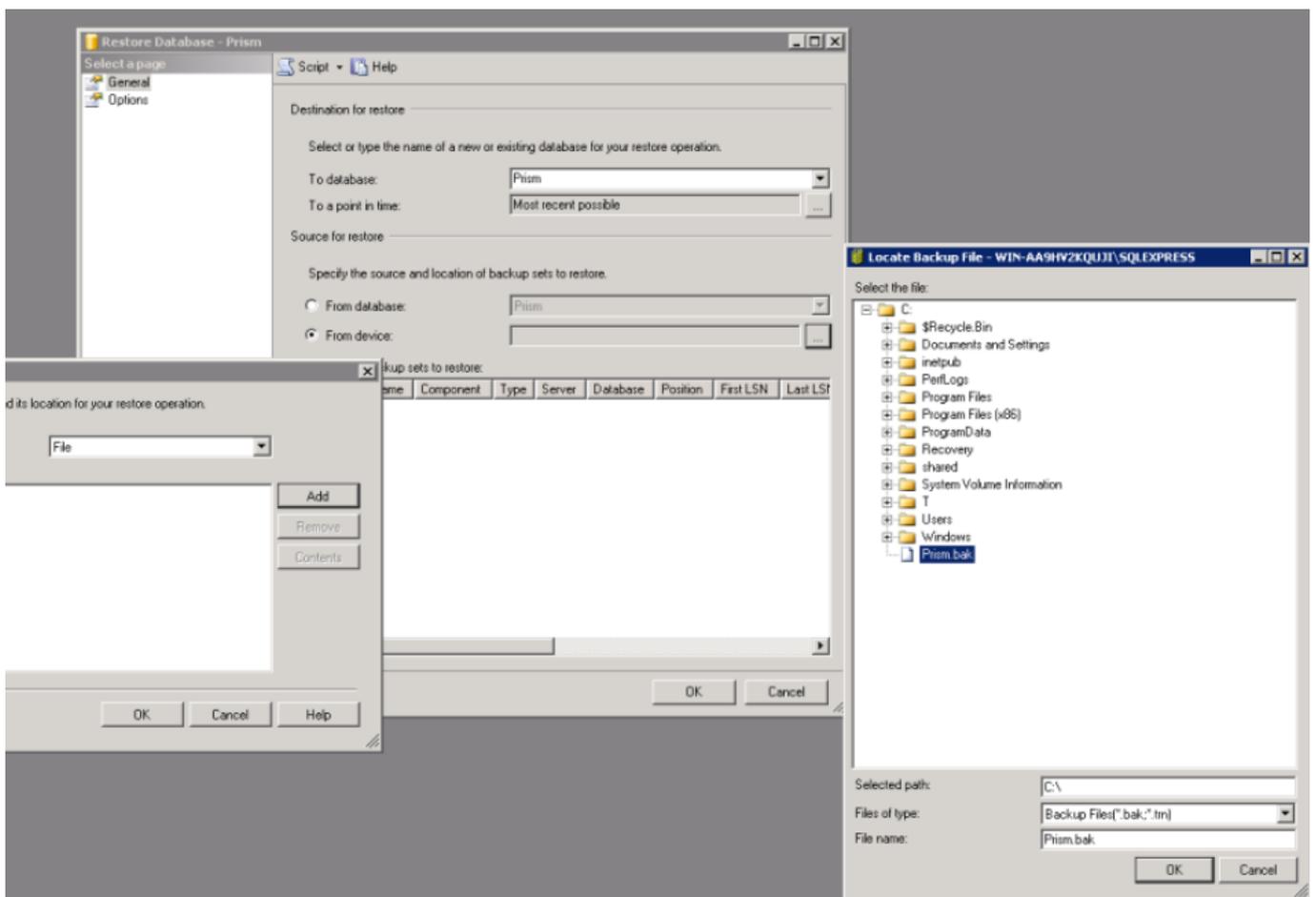
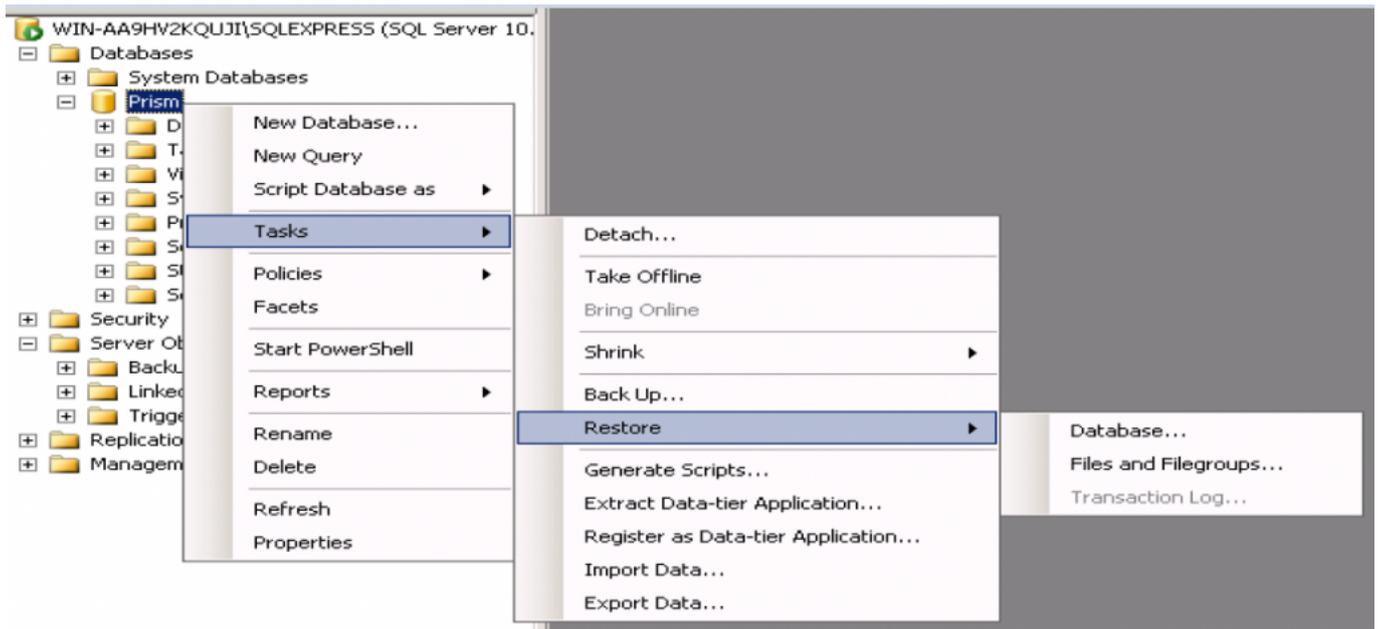
Prism Web platforms repository is stored in MSSQL database. the default database is an embedded version and doesn't require any configurations. To use any other MSSQL version follow the next steps:

The database file is in a backup form (.bak) and located in the Installation folder "\\App_Data\MSSQL\Prism.bak". The .bak file need to be restored to the MSSQL server . In the default configuration, the Prism Web database is set to work with "Windows Authentication". you can change this configuration at any time. Make sure you have running instance of MSSQL Express or MSSQL Server. Restoring Prism Web .bak file:

- 1.Open Microsoft SQL Server Management Studio.
- 2.Right click on the databases folder and click New Database..
- 3.Enter database name : Prism and ok.



4. Select Prism database ?Right click... ?Tasks?Restore?Database ?From device
5. Select the Prism.bak under Installation Folder\\App_Data\\MSSQL\\Prism.bak

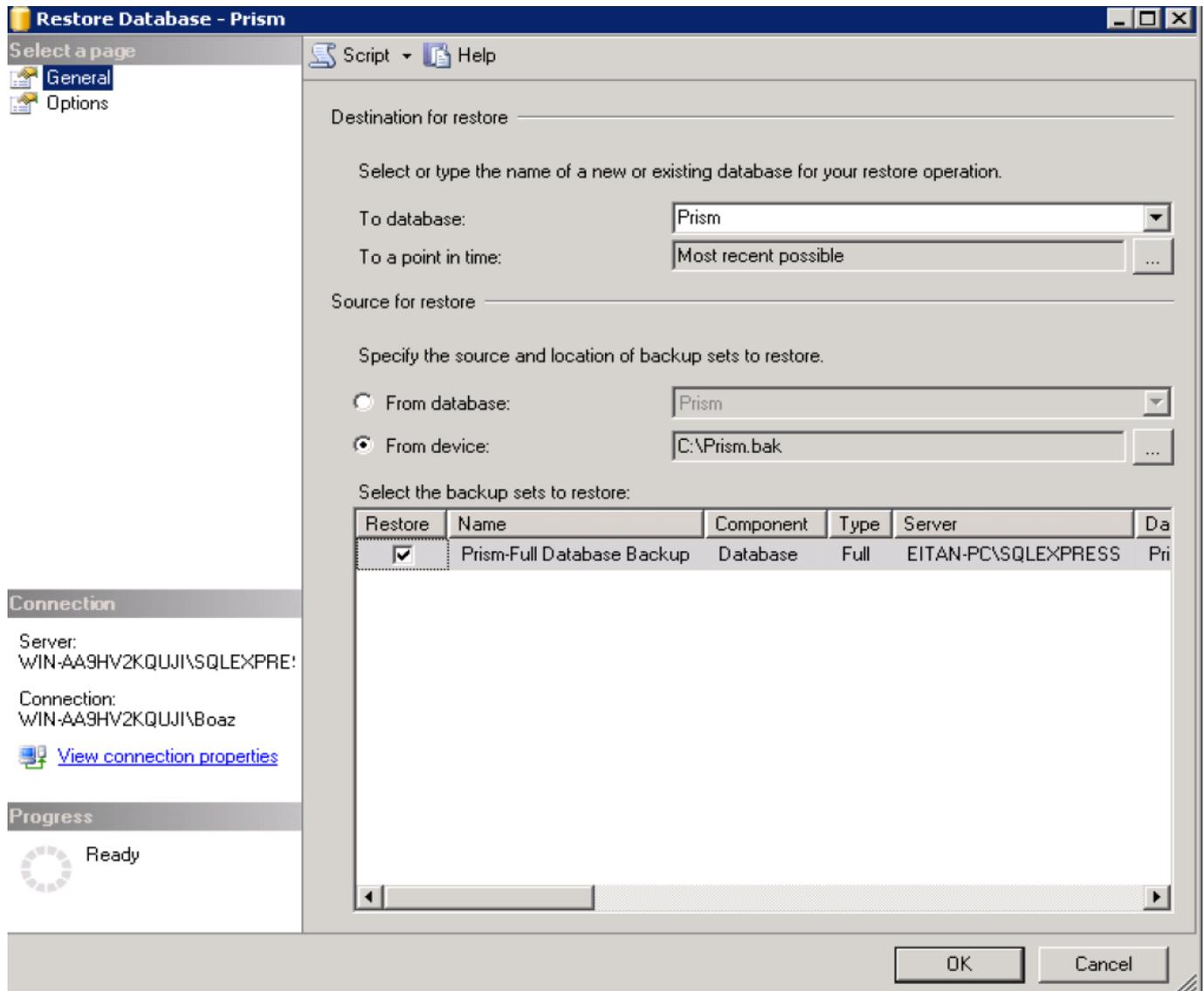


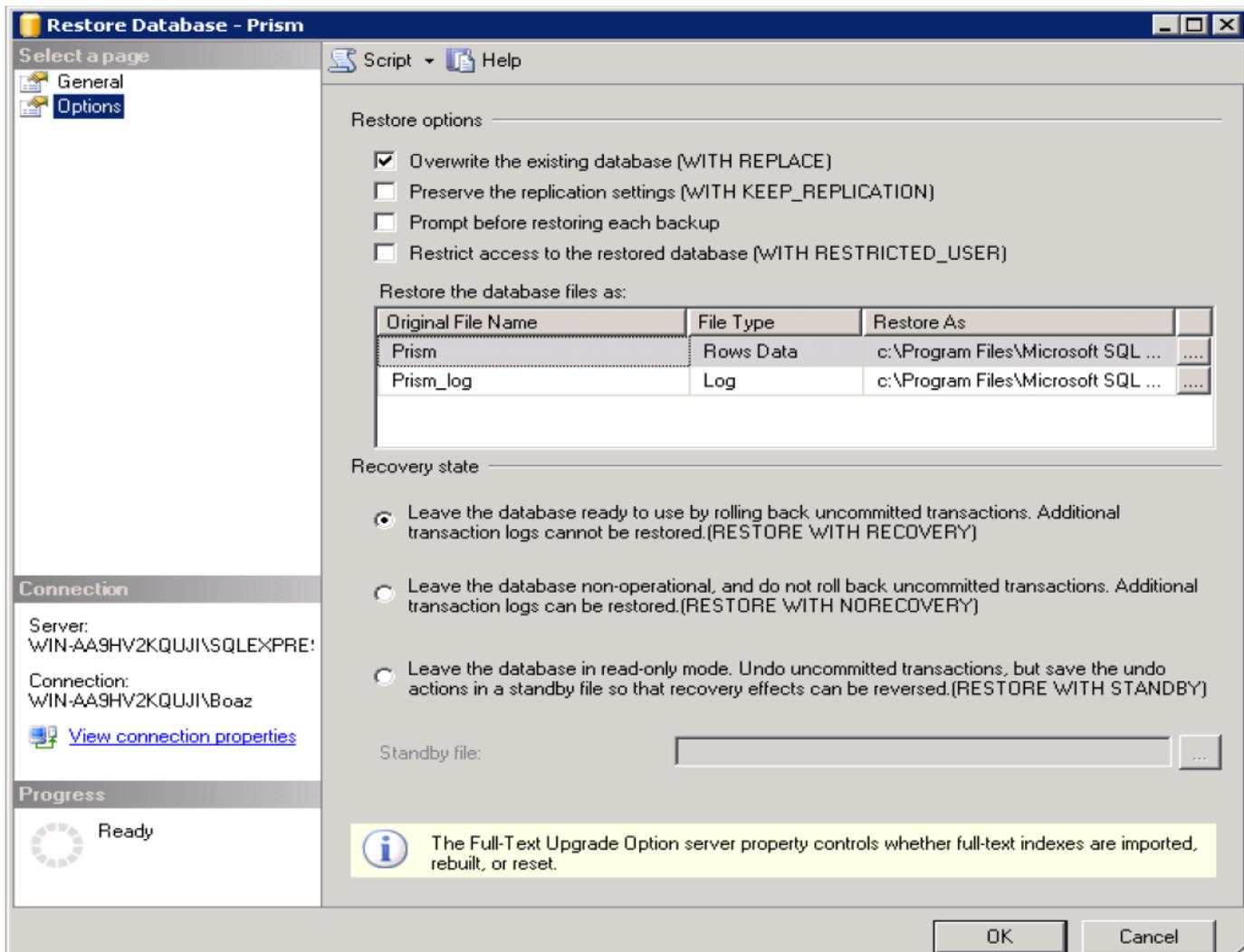
6. Check “Restore”.

7. Under “Options” on the left pane:

Check “overwrite the existing database(WITE REPLACE)”

8. Click OK.





Configuring the Database – Setting Security

Using Windows Authentication Using the predefined MSSQL windows authentication requires to setup the PrismWeb application pool Identity to run under the NetworkService which is a built-in windows identity. This setting is by default only in windows server 2003. note that this is only relevant to the case which the sql server is running on the same machine the IIS is hosted in. in all other cases use “Sql Server mix mode authentication”. To change the application pool identity:

1. In the IIS manager select PrismWeb application pool.
2. On the right pane click “Advanced Settings...”
3. Under Process Model select NetworkService for Identity. After you complete the above steps, configure the PrismWeb database to run under the NetworkService user:

Procedure

1. In the SQL Server management, under Security – Logins

Make sure “NT AUTHORITY\NETWORK SERVICE” exist. If not, create the login user

2. Map the user to the Prism database:

- Under Logins, Select the “NT AUTHORITY\NETWORK SERVICE” and click Properties.

- on the left pane select User Mapping and check Prism.

- on the bottom pane check db_owner and OK.

Login - New

Select a page

- General
- Server Roles
- User Mapping
- Securables
- Status

Script Help

Login name: NT AUTHORITY\NETWORK SERVICE Search...

Windows authentication

SQL Server authentication

Password: []

Confirm password: []

Specify old password

Old password: []

Enforce password policy

Enforce password expiration

User must change password at next login

Mapped to certificate []

Mapped to asymmetric key []

Map to Credential [] Add

Mapped Credentials

Credential	Provider
------------	----------

Remove

Default database: master

Default language: <default>

OK Cancel

Connection

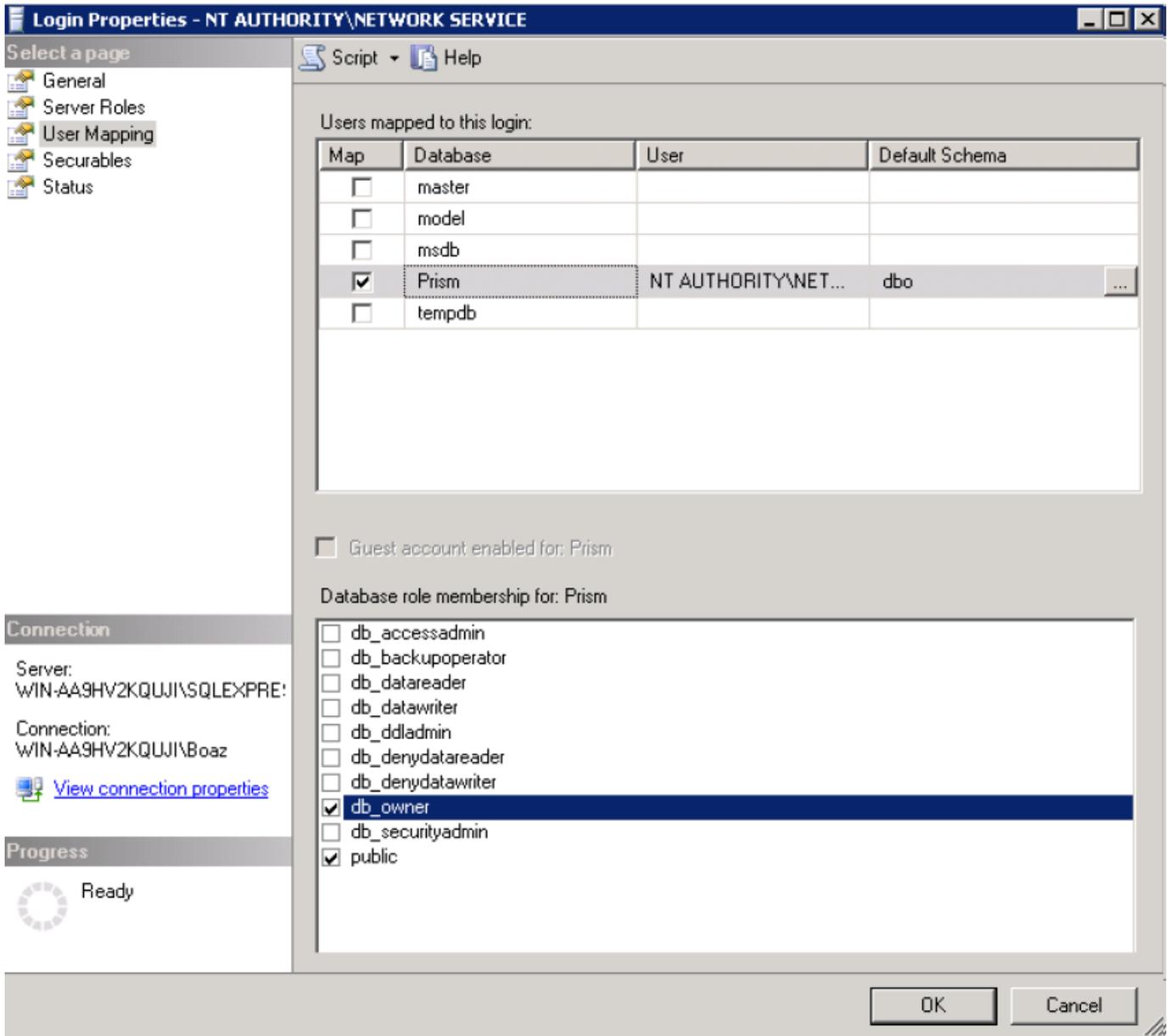
Server: WIN-AA9HV2KQUJI\SQLEXPRES

Connection: WIN-AA9HV2KQUJI\Boaz

[View connection properties](#)

Progress

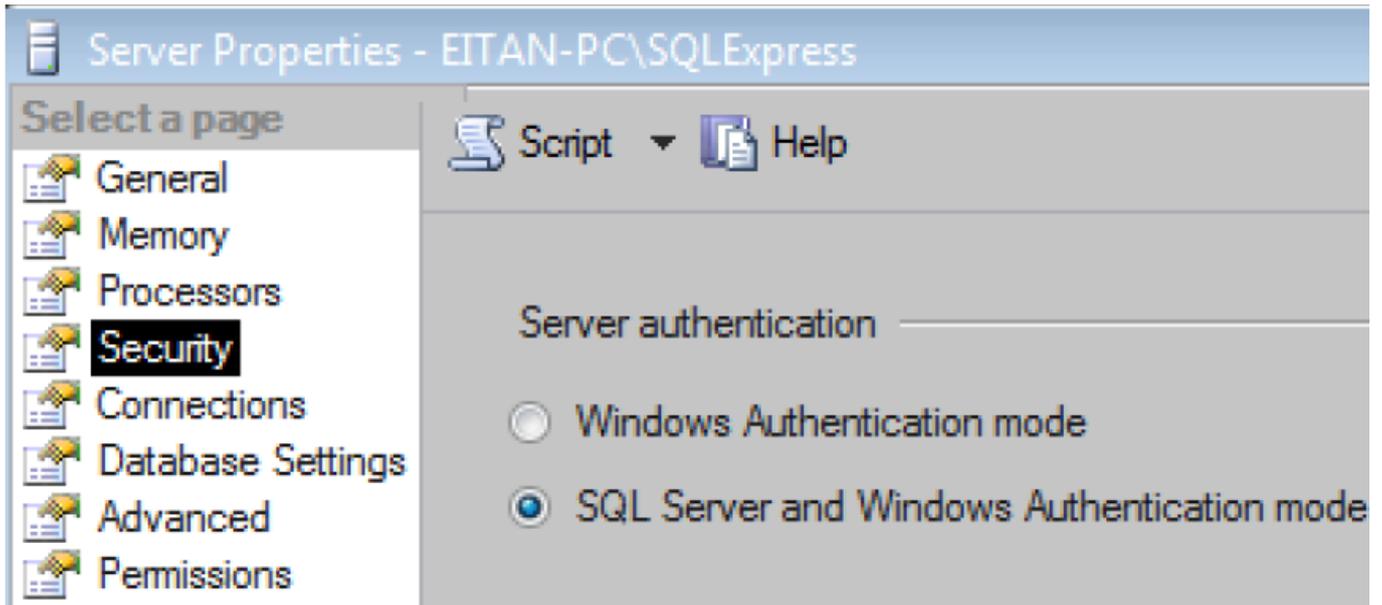
Ready



Using Sql Authentication

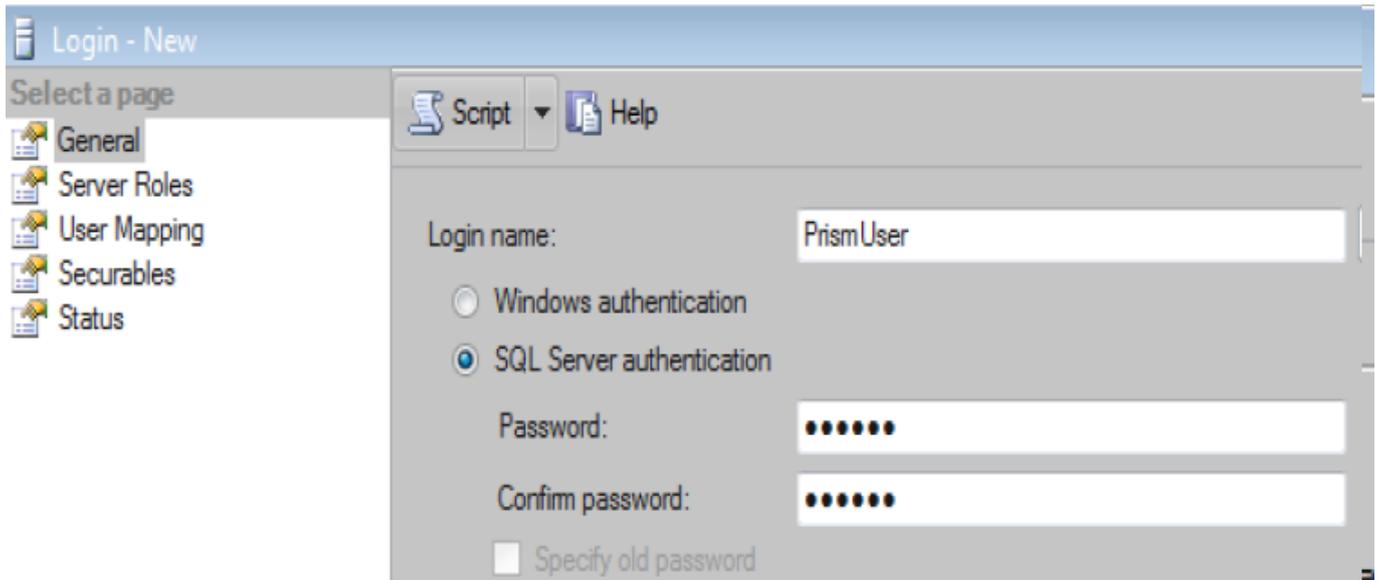
Using sql authentication you configure your database to allow Internet users access to read and (if required) write to your data by user credentials. Make sure the MSSQL is configured to use sql authentication:

- Right click on the Server.
- Click Properties.
- Select Security.



Procedure

1. In the SQL Server Management Tools, Right click on Logins under the Security Node and select New Login.
2. Type the name of your database user in the Login Name box and select SQL Authentication. Set a password and uncheck the Enforce Password Policy, Enforce Password Expiration and User must change password on next login. Click Ok.
3. Under User Mapping check Prism.
4. On the bottom pane check db_owner and OK.



Updating the Connection String

Go to Installation folder\\App_Data\\Configurations\\db.config Make sure DataProviderName = "PrismEntitiesMSSQL" update the provider connection string Data Source with the MSSQL Server name

Switching and Changing Widgets

CHANGING WIDGETS ON THE WEB VIA THE SWITCH

Introduction

Widgets in Prism Web can be switched from one form to another. For example a pivot table can be converted to a bar chart. This enables users greater control over the representation and analysis of data.

This feature is only available in version 4.5 and higher

Switching to a Different Widget

1. Hover the mouse over the top right of the widget till the context menu appears.
2. Click the switch icon till the widget menu appears.
3. Click the appropriate widget option to switch.

City	Fund Status		
town	Funded		
ork	Funded		
ego	Funded		
	Funded		
oines	Funded		
ancisco	Funded		
ork	Acquired		
ancisco	Acquired		
oines	Funded		
	Funded		
	Funded	5,000,000	N/A
to	Funded	N/A	1
minster	Funded	N/A	1
Village	Funded	1,100,000	N/A
ng	Funded	N/A	2
ain View	Funded	N/A	1
ancisco	Funded	N/A	1
ancisco	Funded	N/A	1
RANCISCO	Funded	10,000,000	4
ain View	Funded	N/A	4
ancisco	Funded	4,250,000	1

Rows 1 - 25 of 4253

1	Switch Context Menu
2	Widget Menu
3	Pivot Tables
4	Scatter Charts
5	Polar Charts
6	Maps
7	Line Charts
8	Column Charts
9	Bar Charts
10	Area Charts

Notes about Switching

- Switching between widgets with compatible data structure will be immediate.
- Switching between widgets with incompatible data structure will bring up the 'Edit' window in which you can play around with what goes where.
- You can switch from every widget to every other widget, except the 'Pie' group.
- Pie charts are a separate group and only present when the original widget is a pie. A pie chart can only be switched to different widgets within the pie group

Sharing Switched Widgets

- If a switch a widget and share – you will get it with the switched state.
- If the dashboard owner continues to do switches, then it won't affect the shared users automatically. But when the shared user hits the dashboard 'Restore to original' button – he will see the latest changes the owner made.

Set and Edit a Widget Type in Prism Web

SET OR EDIT A WIDGET TYPE IN PRISM WEB

Introduction

Each widget has its own data layout that determines which dimensions and how dimensions are presented within the widget.

Setting and Editing a Widgets Data Layout

1. Hover the mouse over the top right of the widget till the context menu appears.
2. Click the edit icon (pencil) for the widget data layout menu to appear. Alternatively clicking on the switch icon then selecting the appropriate widget will bring up the data layout menu.
3. Each widget has its own data layout depending on how data is visualized. To understand the components of each widget click on the appropriate widget category in the Prism Web user guide.
4. Right click the top right arrow next to each data element in order to select the appropriate dimension to use. In certain cases multiple values may be selected. Optional data layout components will be presented in small boxes, click the optional item to set the required dimension.

For color elements the dimension referenced must have exception highlighting applied in BI Studio. If the selected dimension does not have Exception Highlighting a default color will be applied.

5. Click 'Apply' to save changes.

City	Fund Status		
town	Funded		
ork	Funded		
ego	Funded		
	Funded		
oines	Funded		
ancisco	Funded		
ork	Acquired		
ancisco	Acquired		
oines	Funded		
	Funded		
	Funded	5,000,000	N/A
to	Funded	N/A	1
minster	Funded	N/A	1
Village	Funded	1,100,000	N/A
ng	Funded	N/A	2
ain View	Funded	N/A	1
ancisco	Funded	N/A	1
ancisco	Funded	N/A	1
RANCISCO	Funded	10,000,000	4
ain View	Funded	N/A	4
ancisco	Funded	4,250,000	1

Rows 1 - 25 of 4253

Example: Data Layout Menu

- 1 Data Layout Component
- 2 Optional Data Layout Components
- 3 Preview Based on Selected Data Dimensions

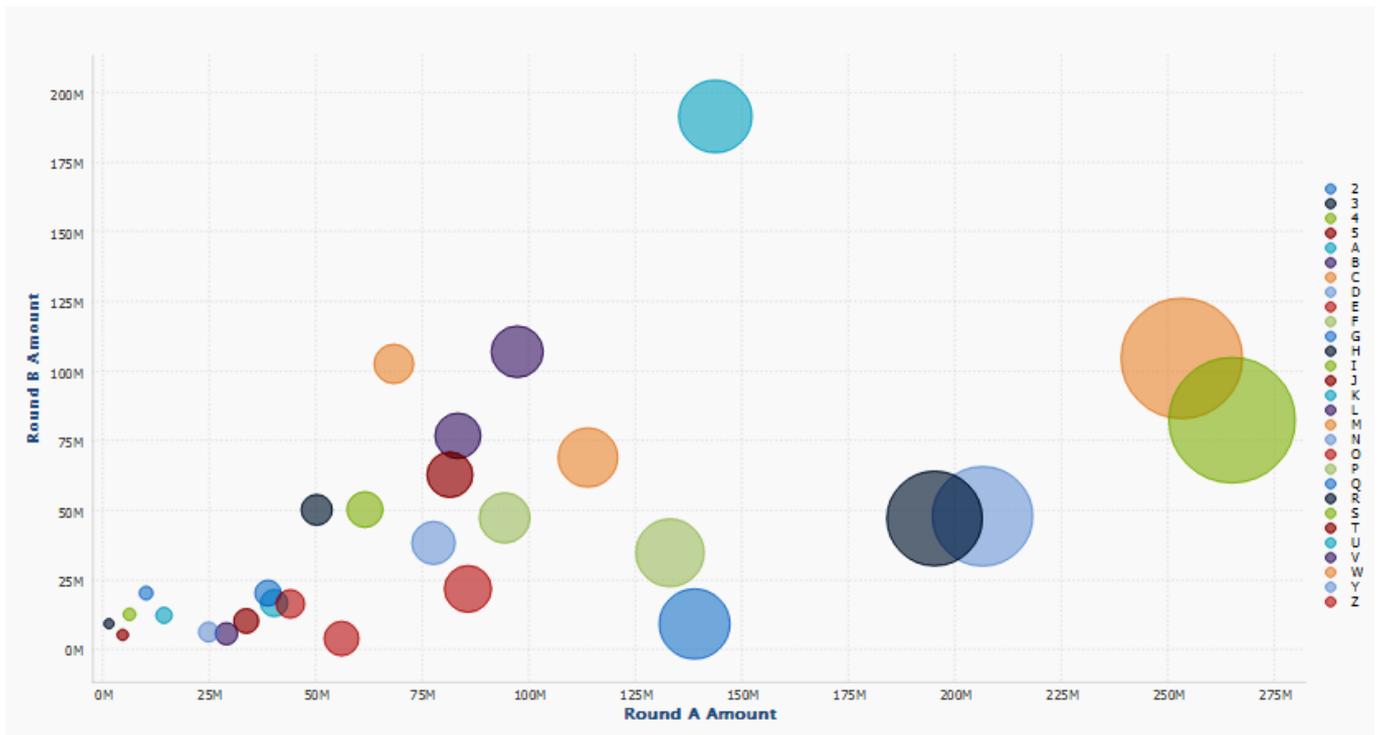
Point and Scatter Charts

SCATTER CHARTS

Introduction

Scatter charts can be enabled in Prism Web by using the widget switch. A scatter is used to plot the relationship between two variables.

There are two types of scatter visualizations, point charts and scatter charts.



Example: Scatter Plot

Point and Scatter Charts Data Layout

Below is a description of the data layout options for point and scatter plots.

Point Chart

- *Axes*

X: Dimension plotted along the x axis.

- *Series*

Color: The color given to data elements - the selected field must be highlighted in BI Studio.

Measures: Value or calculation segmented by x axis.

Filters: Dimension to segment values but not shown in data. For example if a year was placed in the filter data will be restricted to that year however the actual year will not be presented in any place.

Scatter Chart

- *Axes*

X: Dimension plotted along the x axis.

Y: Dimension plotted along the y axis.

- *Series*

Color: The color given to data elements, the selected field must be highlighted in BI Studio.

Details: Additional dimension to segment values

Size: Dimension used to represent the size of point in comparison to others.

Filters: Dimension to segment values but not shown in data. For example if a year was placed in the filter data will be restricted to that year however the actual year will not be presented in any place.

Polar and Radar Charts

POLAR AND RADAR CHARTS

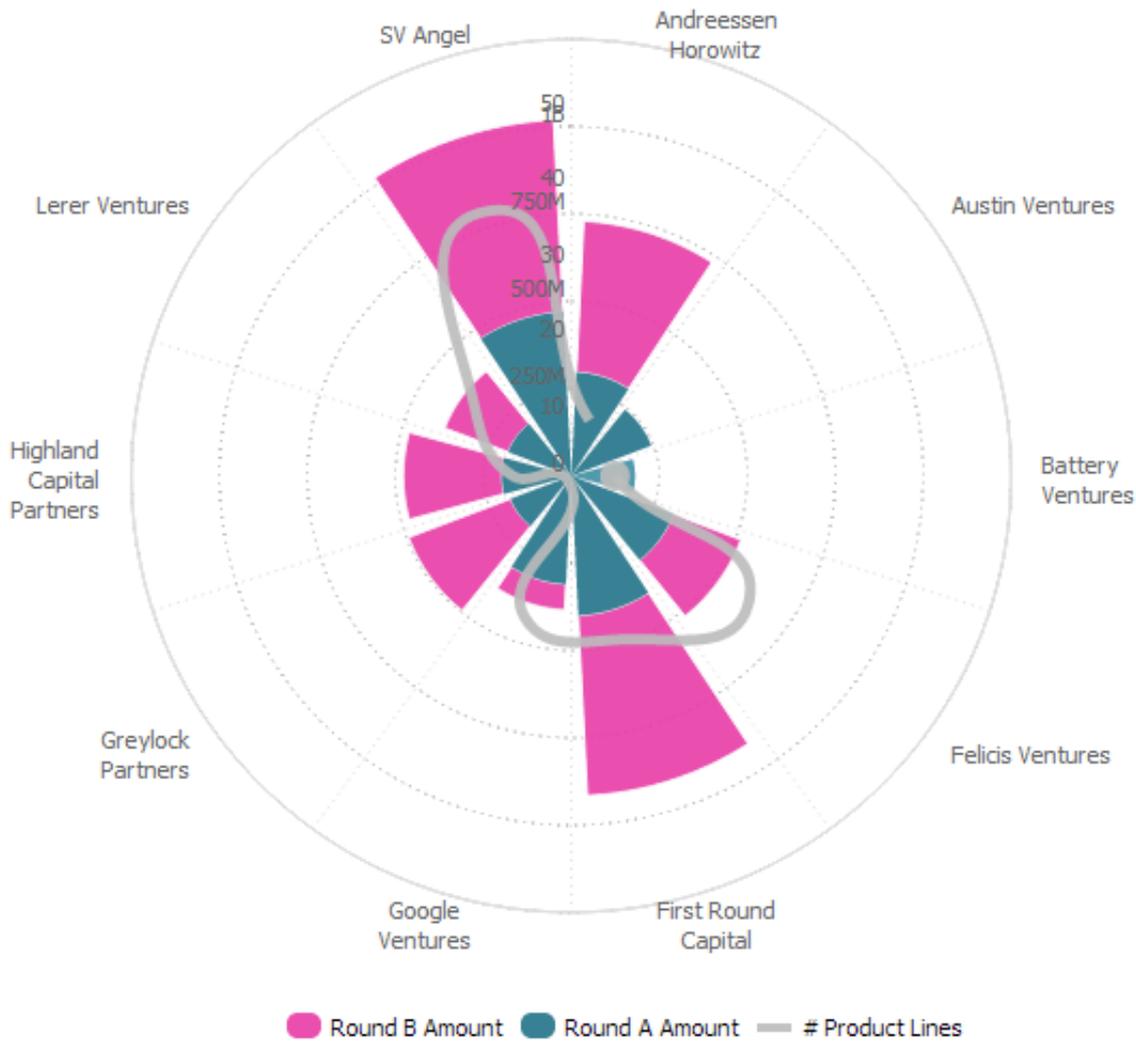
Introduction

Polar and radar charts are especially suited for highlighting outliers and commonality and suitable for plotting small and medium sized multivariate data.

Polar and radar charts enable three or more quantitative dimensions to be represented on an axes starting from the same point.

There are several types of polar visualizations including, windrose, spider, polar and radar charts.





Example: Scatter Plot

Data Layout - Polar \ Radar \ Windrose and Spider Charts

Below is a description of the data layout options for polar and radar charts.

Axes

X: Dimension plotted along the x axis (up to 2 dimensions).

- Series

Color: The color given to data elements - the selected field must be highlighted in BI Studio.

Measures: Value or calculation segmented by x axis.

Filters: Dimension to segment values but not shown in data. For example if a year was placed

in the filter; data will be restricted to that year however the actual year will not be presented in any place.

Scatter and Area Map

SCATTER AND AREA MAPS

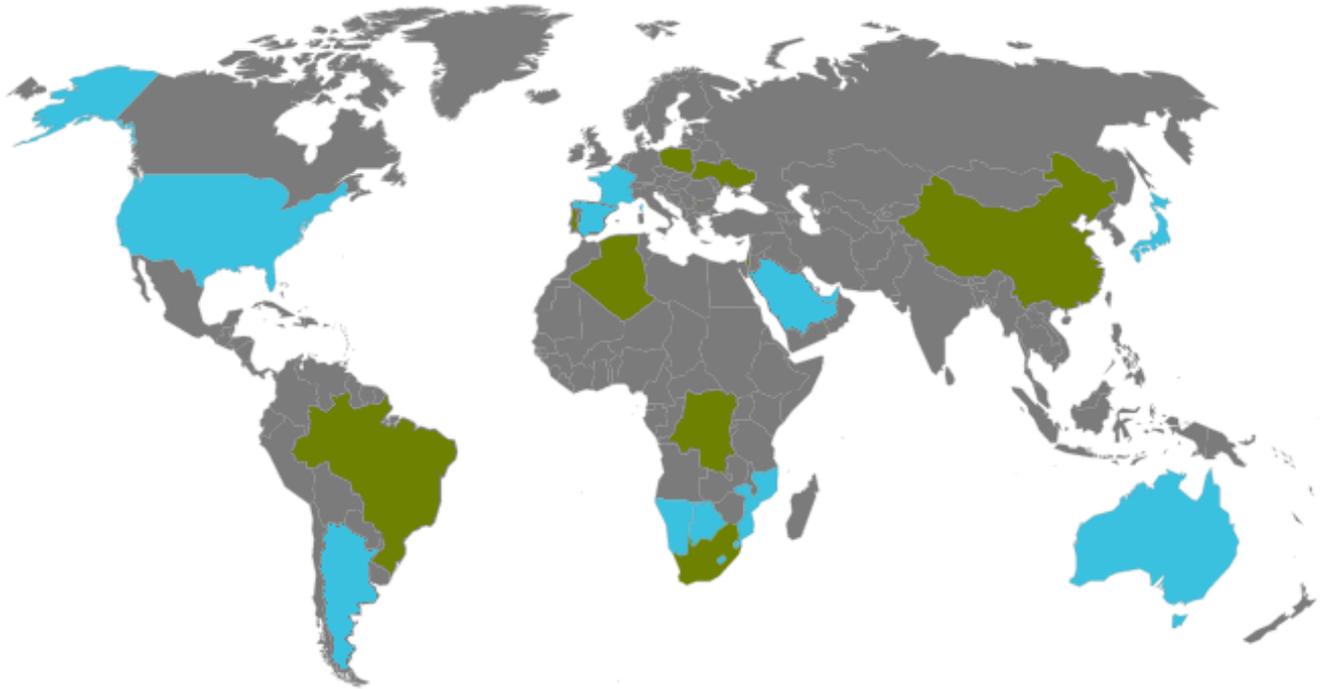
Introduction

Map widgets enable data to be visualized based on country or US State. Mapping may be based on country, city, zip, latitude or longitude.

There are two types of map visualizations, area maps and scatter maps. Area maps highlight countries based on a dimension of interest. Scatter maps are suitable for more accurate data and enable specific geographic dimensions such as zip to be used to visualize points on a map.



Example: Scatter Map



Example: Area Map

Area and Scatter Maps Data Layout

Below is a description of the data layout options for area and scatter maps.

Scatter Map

- *Geo*

Country: Country dimension to base map data.

City: City dimension to base map data.

State: US State dimension to base map data.

Latitude: Latitude value to base map data.

Longitude: Longitude value to base map data.

- *Point*

Color: The color given to the mapping elements - the selected field must be highlighted in BI Studio.

Size: dimension used to determine size of points on map.

Filters: Dimension to segment values but not shown in data. For example if a year was placed in the filter data will be restricted to that year however the actual year will not be presented in any place.

Area Map

- *Geo*

Country: Country dimension to base map data.

State: US State dimension to base map data.

- *Point*

Color: The color given to the mapping elements - the selected field must be highlighted in BI Studio.

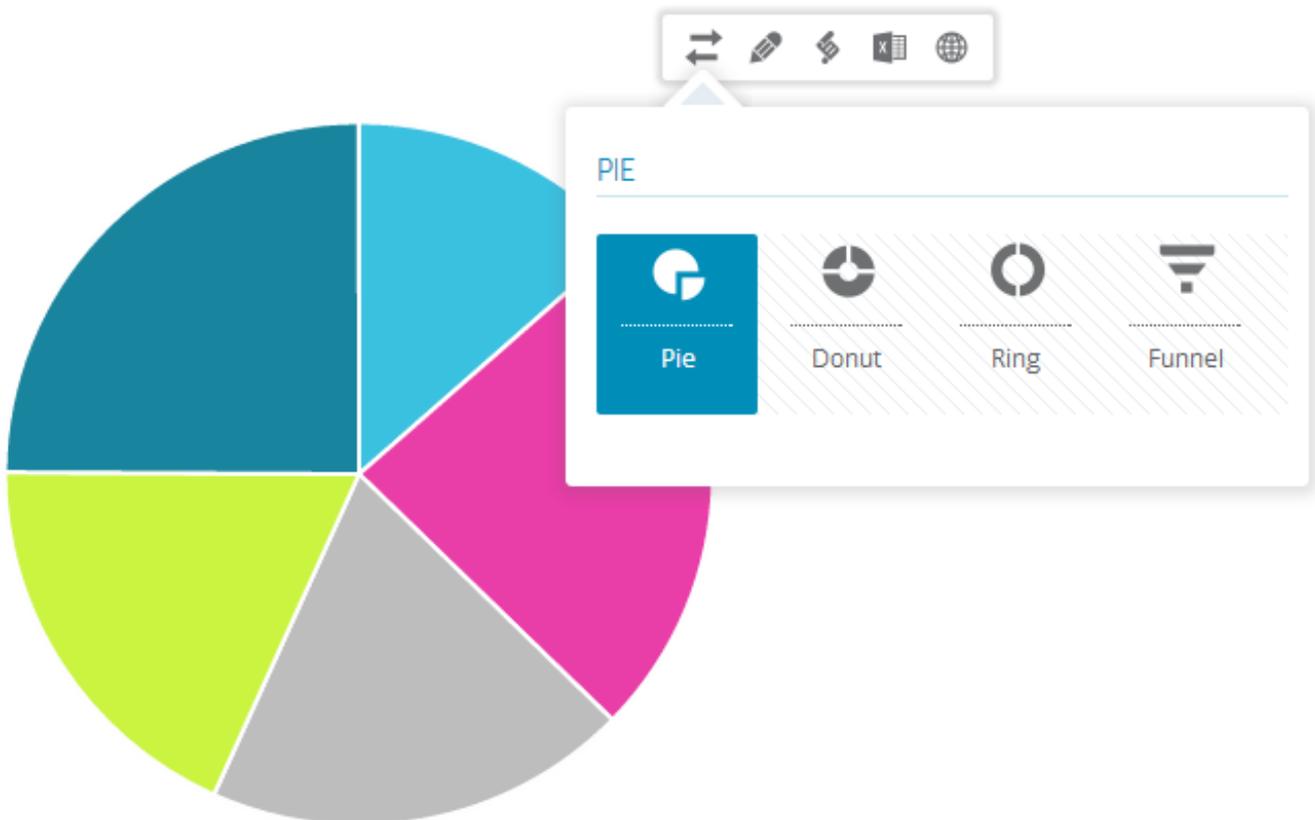
Filters: Dimension to segment values but not shown in data. For example if a year was placed in the filter data will be restricted to that year however the actual year will not be presented in any place.

Pie and Funnel Charts

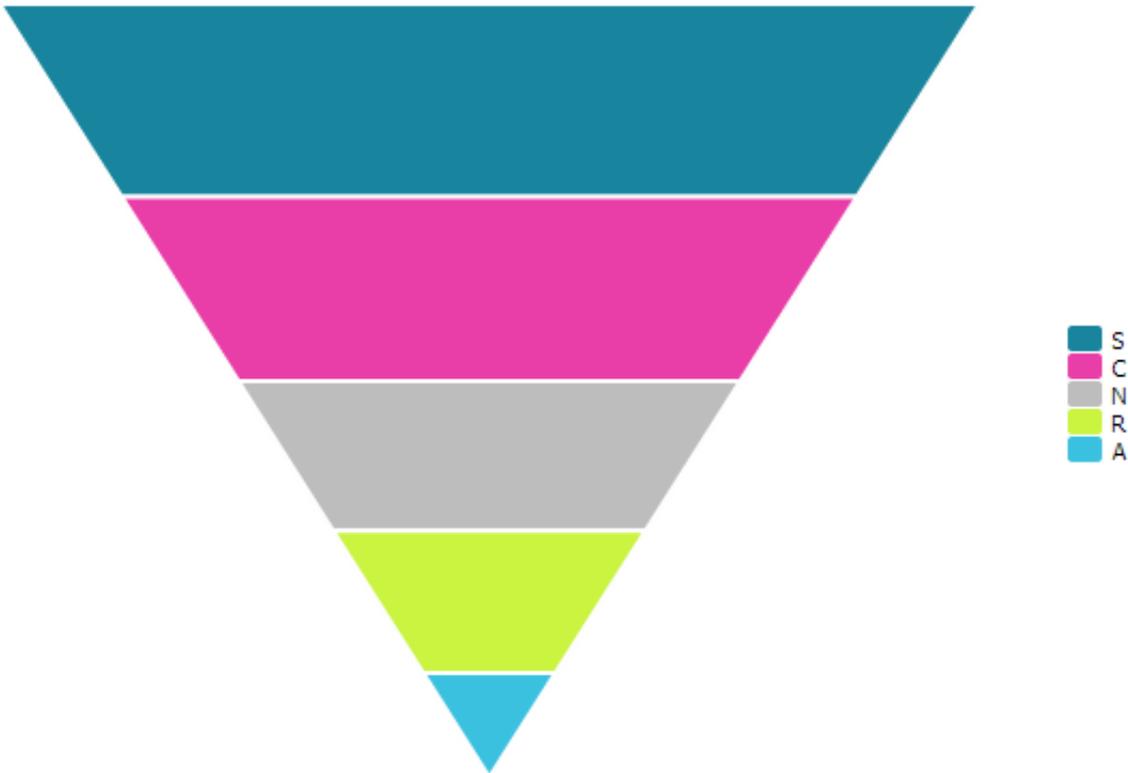
Introduction

Pie and funnel charts are designed to show the relevant contribution of various dimensions. In Prism Web pies are defined as a separate group of widgets. A pie or funnel can only be switched to other widgets within the pie group. To access the pie menu a pie widget must first be created in BI Studio.

There are two types of scatter visualizations, point charts and scatter charts.



Example: Pie Menu



Example: Funnel Chart

Pie and Funnel Charts Data Layout

Below is a description of the data layout options for pie and funnel charts.

Pie, Donut, Ring and Funnel Chart

- *Point*

Color: Dimension used to highlight segment in chart.

Size: Dimension used to represent the size of point in comparison to others.

- *Filter*

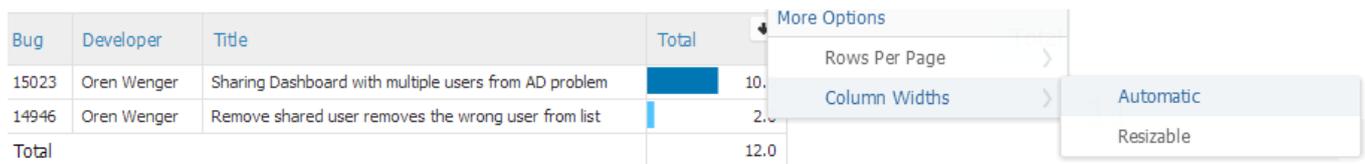
Filters: Dimension to segment values but not shown in data. For example if a year was placed in the filter data will be restricted to that year however the actual year will not be presented in any place.

Adjust Pivot Column Width and Row Numbers

Pivot columns can be moved and resized and the number of rows set dynamically in the web environment.

Edit Pivot Column Width and Row Settings

1. On the top right click the 'Edit' icon.
2. Click for more options.



The image shows a pivot table with columns: Bug, Developer, Title, and Total. The 'Total' column has a blue bar chart. A 'More Options' menu is open over the 'Total' column, showing 'Rows Per Page' and 'Column Widths' options. The 'Column Widths' option is selected, and a sub-menu is open showing 'Automatic' and 'Resizable' options.

Bug	Developer	Title	Total
15023	Oren Wenger	Sharing Dashboard with multiple users from AD problem	10.0
14946	Oren Wenger	Remove shared user removes the wrong user from list	2.0
Total			12.0

Image: Click 'More options' to set column width or row numbers

Adjust Column Width:

Select the 'Column Widths' option then select the relevant setting.

Resizable: This enables a user to manually move and set each column width.

Automatic: This re-sizes columns automatically based on the total width of the pivot.

Adjust Row Numbers:

Select the 'Rows per Page' option then select the relevant setting for the number of rows to display per page from 10 to 200 rows, additional rows will only be accessible via paging at the bottom of the pivot .

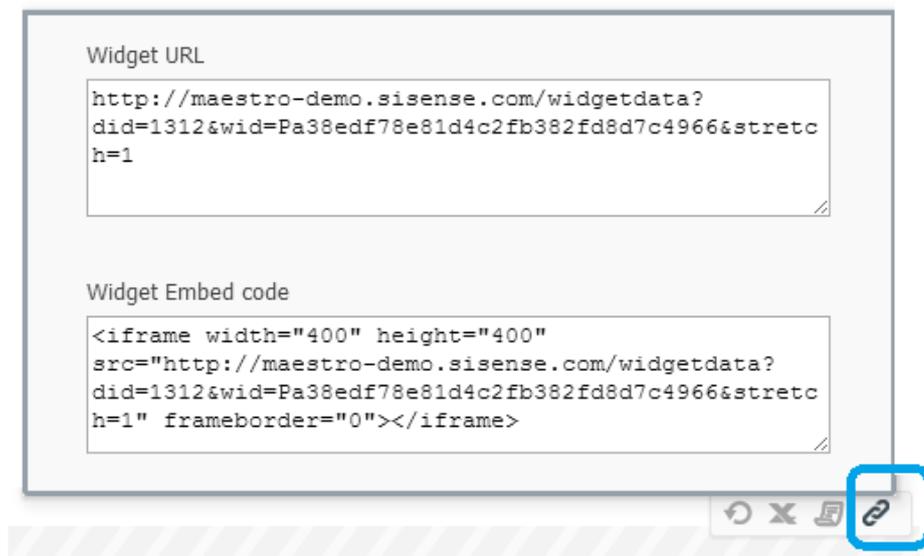
Download, Embed and Share Dashboard Data and Widgets

Introduction

Widgets in Prism Web can easily be shared via a link or iframe. In addition data contained with in the widget can be downloaded in Excel or CSV format via the web.

Share a Widget

Hovering the mouse cursor on the top right of a widget will cause the 'Share Widget' icon to appear. Clicking on this icon will display two separate links to either share a link or embed the widget.



Link Extension - No Stretch

By default a shared widget display in the entire web browser page. To scale the widget to its original size and position the following syntax should be added to the end of link URL - '&stretch=0'.

State	# Disasters	Avg Incident Days	
TX	15	0	●
AR	14	1	●
GA	12	12	●
MS	10	0	●
IL	9	20	●
OK	9	7	●
OH	6	6	●
MI	6	0	●
KS	6	2	●
WI	6	2	●
AL	6	0	●
MO	5	0	●
KY	5	2	●
IN	4	0	●
NE	4	9	●

Page 1 of 2 Rows 1 - 16 of 32

Link Extension - Download as Excel or CSV

Data from a shared widget can be downloaded in Excel or CSV format.

To download the widget data as a *Excel file* the following syntax should be added to the end of link URL - '&format=excel&dl=1'.

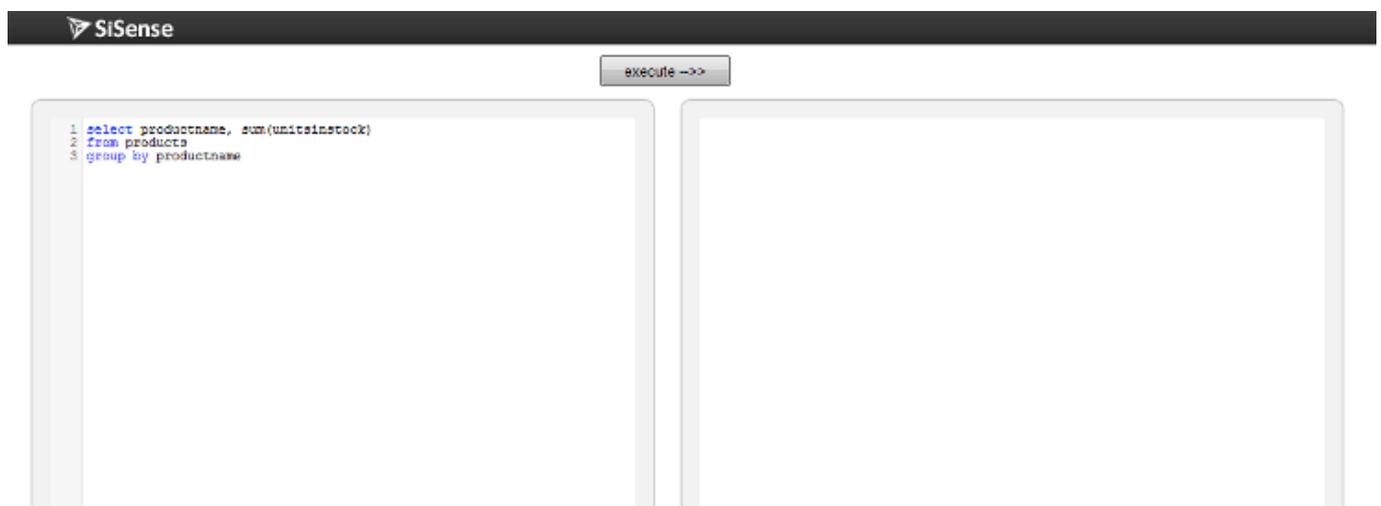
To download the widget data as a *CSV file* the following syntax should be added to the end of link URL - '&format=csv&dl=1'.

State	# Disasters	Avg Incident Days	min # Disasters	max # Disasters	min Avg Incident Days	max Avg Incident Days
TX	15	0	1	15	0	24.4358974358974
AR	14	1	1	15	0	24.4358974358974
GA	12	12	1	15	0	24.4358974358974
MS	10	0	1	15	0	24.4358974358974
IL	9	20	1	15	0	24.4358974358974
OK	9	7	1	15	0	24.4358974358974
OH	6	6	1	15	0	24.4358974358974
MI	6	0	1	15	0	24.4358974358974
KS	6	2	1	15	0	24.4358974358974
WI	6	2	1	15	0	24.4358974358974
AL	6	0	1	15	0	24.4358974358974
MO	5	0	1	15	0	24.4358974358974
KY	5	2	1	15	0	24.4358974358974
IN	4	0	1	15	0	24.4358974358974
NE	4	9	1	15	0	24.4358974358974
IA	4	0	1	15	0	24.4358974358974
NC	4	1	1	15	0	24.4358974358974
LA	4	2	1	15	0	24.4358974358974
TN	4	10	1	15	0	24.4358974358974
MN	3	24	1	15	0	24.4358974358974
ND	3	16	1	15	0	24.4358974358974
WY	2	0	1	15	0	24.4358974358974
CT	2	0	1	15	0	24.4358974358974
PA	2	1	1	15	0	24.4358974358974
MA	2	0	1	15	0	24.4358974358974
NY	1	0	1	15	0	24.4358974358974

SQL API

EXTRACTING DATA FROM THE ELASTICUBE VIA THE SQL API

The SQL Runner enables standard SQL queries to be run on data within an Elasticube and have results returned in JSON or CSV format. This can be used to create web services or run adhoc queries on the Elasticube.



Accessing the SQL Runner in Prism Web

1. Open Prism web in a browser and login.
2. Add the extension 'sqlrunner' at the end of the Prism Web URL. For instance if the URL is localhost:8081 to access the runner enter the URL localhost:8081/sqlrunner
3. Once the runner opens sql queires can be entered on the left panel while results will be returned on the right.



4. Click execute to run the query and view results.

SQL API

The syntax below executes an sql query against a datasource, returns json results

site/api/datasources/[datasource-name]/sql?query=[sql-query]

eg. [http://www.sitename.com/api/datasources/laptopsales/sql?query=select sales.city count\(*\) from sales group by sales.city](http://www.sitename.com/api/datasources/laptopsales/sql?query=select sales.city count(*) from sales group by sales.city)

The example returns the city name and count of occurrences from the salestable and would correspond to an SQL statement below

Select city, count()*

From sales

Group By city

Results can also be downloaded in csv format is supported by adding *&format=csv* query parameter.

eg. [http://beta.sisense.com/api/datasources/laptopsales/sql?query=select sales.city count\(*\) from sales group by sales.city&format=csv](http://beta.sisense.com/api/datasources/laptopsales/sql?query=select sales.city count(*) from sales group by sales.city&format=csv)

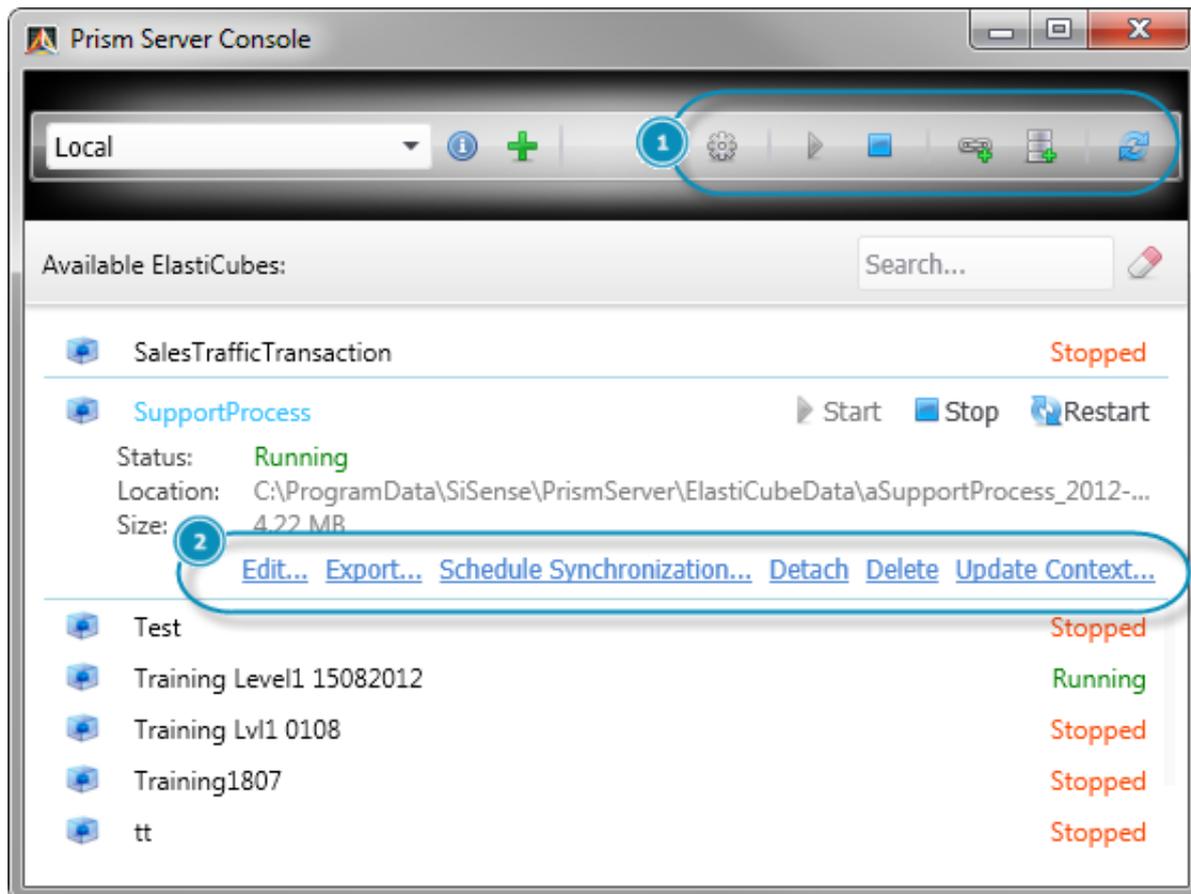
Management & Administration

Introduction to Prism Server Console

Introduction

ElastiCubes are run and maintained in the Prism Server Console. Various administration functions are available to run, edit and manage ElastiCubes in the console.

Prism Server Console Environment



1. Configuration and main menu
2. Manage ElastiCube

Licensing

WEB LICENSING

Activation of Prism Web requires a license file. In order to receive the file follow the steps below.

1. Log into Prism Web .
2. In Prism Web click > settings (icon on the far right)> license settings> switch to another license> then click Don't have a product key? ask us for one
3. Copy and send the key to support@sisense.com
4. Once you receive the license repeat the process in step 1 to activate the license.



Prism Activation

Prism Activation

The Prism activation console provides the ability to view current licensing details, update or reactivate on a different machine. To access the console go to Start>All programs>Sisense>Prism Activation.

This will launch the activation console. Below is a sample of the console and available information.

The screenshot displays the SiSense 'My Account' interface. At the top left is the SiSense logo. The main heading is 'My Account' with a fingerprint icon. Below this is a user profile bar for 'Ternoven Robert' with email 'robi.ter@sisense.com'. Three license cards are shown: '3 YEARS' with an expiration date of '09/09/2016', 'COMMERCIAL LICENSE', and 'WEB' license for 'PUBLIC' access with '1000 users'. At the bottom, there are links for 'Manage Online', 'Sign Out', and 'Contact Support'.

License Type	Duration	Expiration	Access Type	User Count
3 YEARS	3	09/09/2016	PUBLIC	1000 users
COMMERCIAL LICENSE				
WEB			PUBLIC	1000 users

Error and Troubleshooting

TROUBLESHOOTING ELASTICUBE SERVER CONNECTIONS

Symptom

You can connect to the server and preview is showing data, but the build process fails due to connectivity issues.

Cause

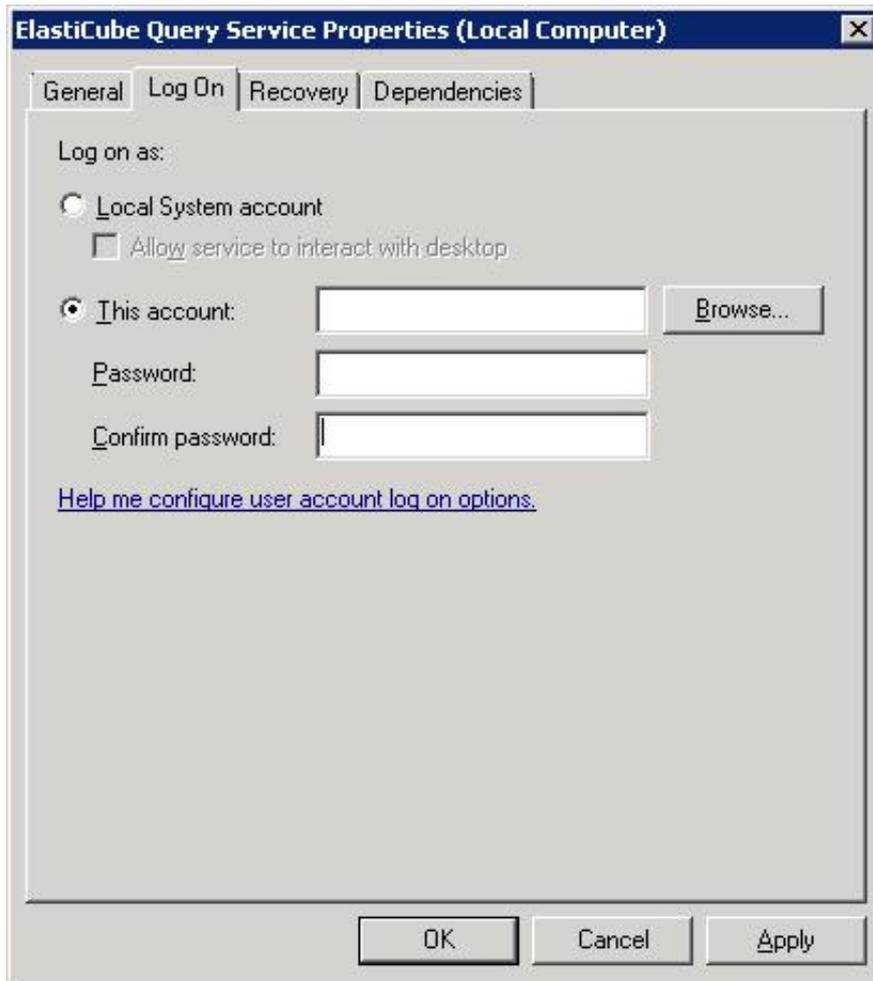
The ElastiCube server service is running on the system account, but not on the logged on user's account.

Solution

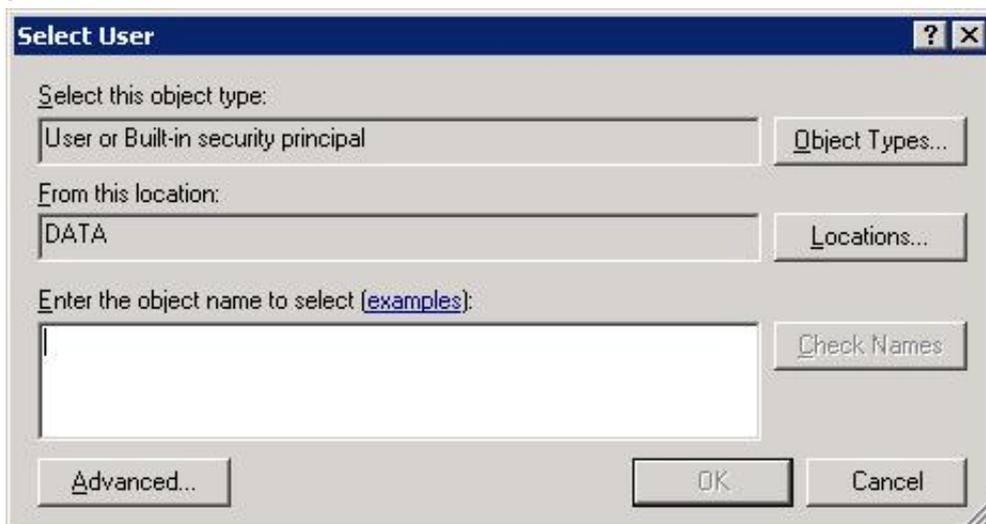
Change the logon properties of the ElastiCube Management Service.

Steps:

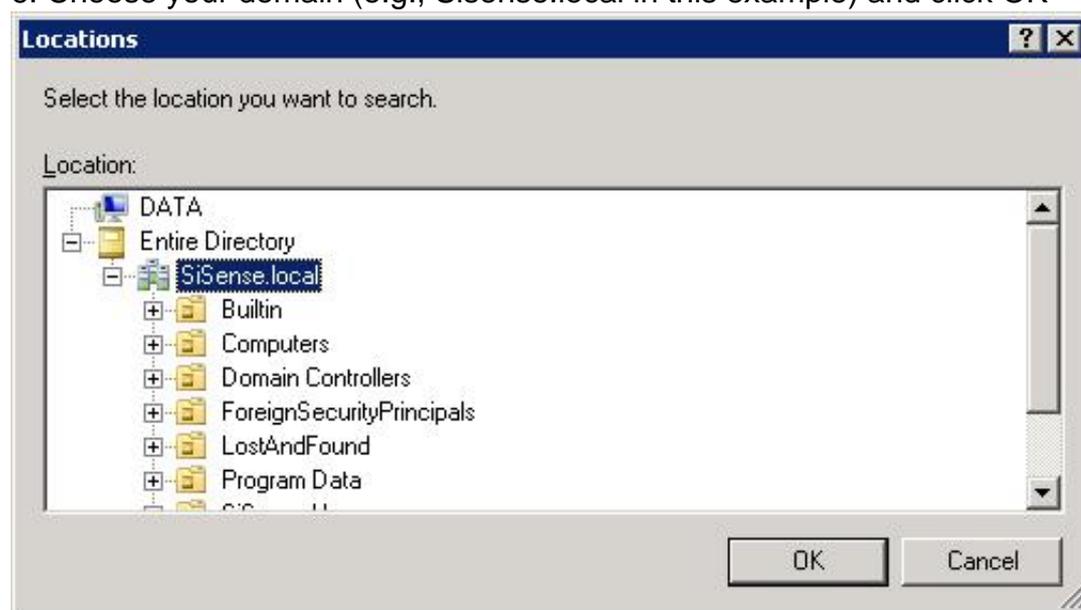
1. Go to Control Panel -> Administrative Tools -> Services
2. Find ElastiCube Management Service, right click and select Stop
3. Find ElastiCube Management Service, right click and select Properties
4. Select the Log On tab, select This Account then click Browse



5. Click Locations..



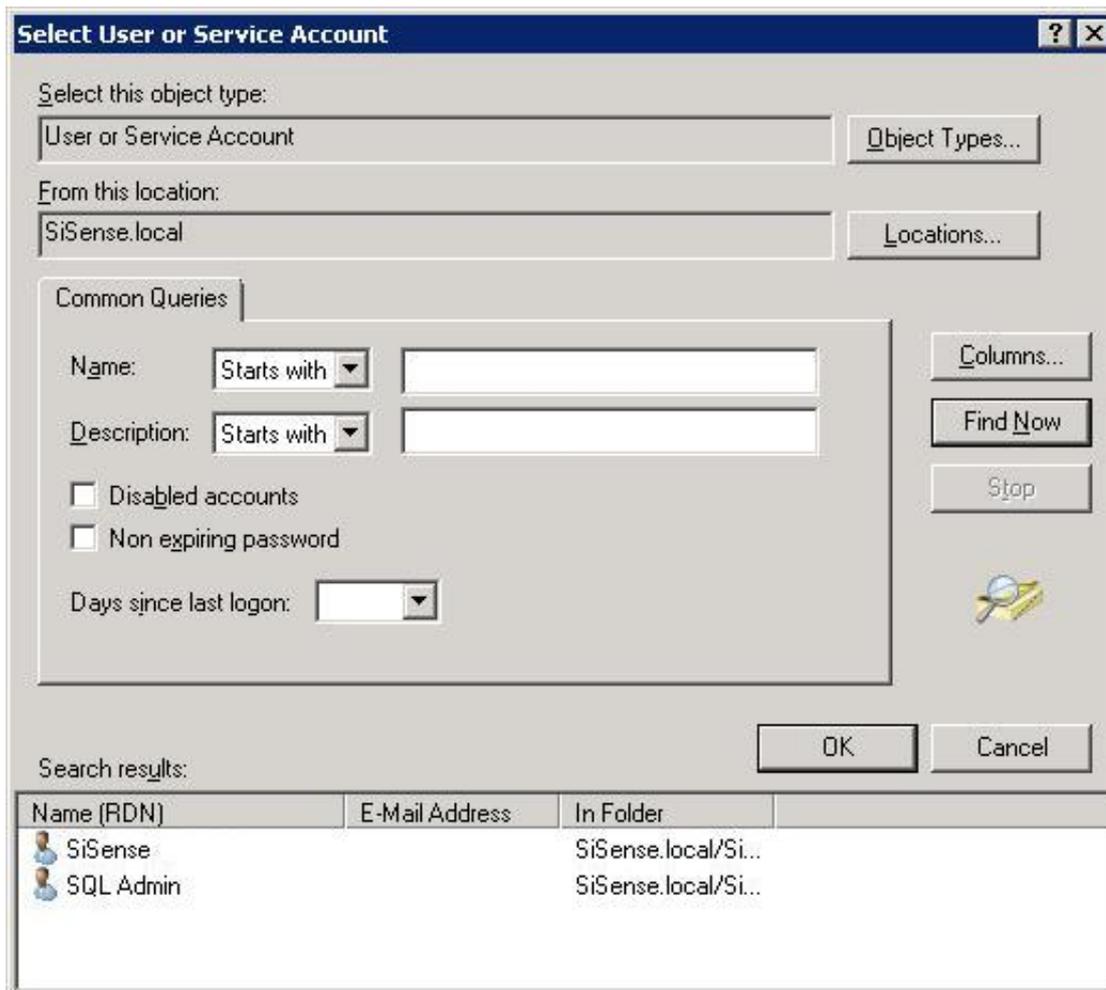
6. Choose your domain (e.g., Sisense.local in this example) and click OK



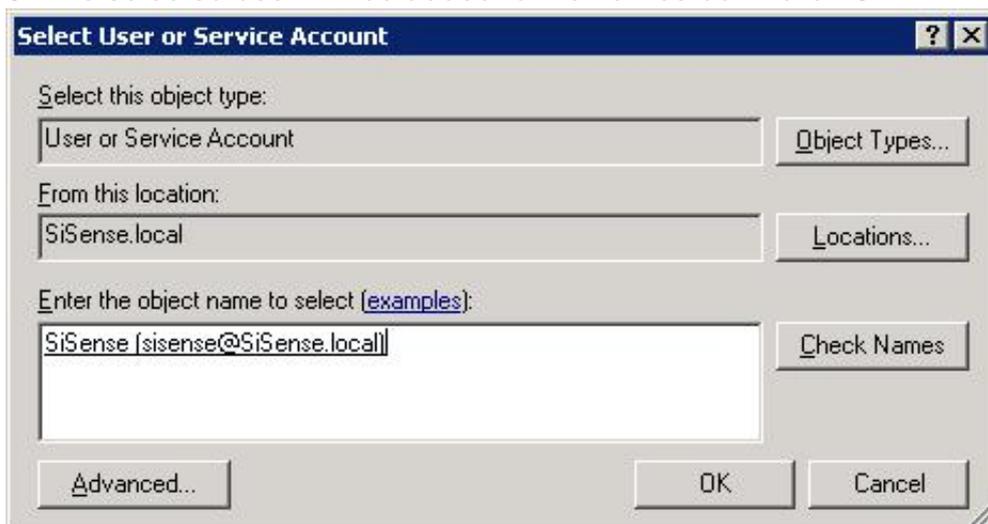
7. Click Advanced...



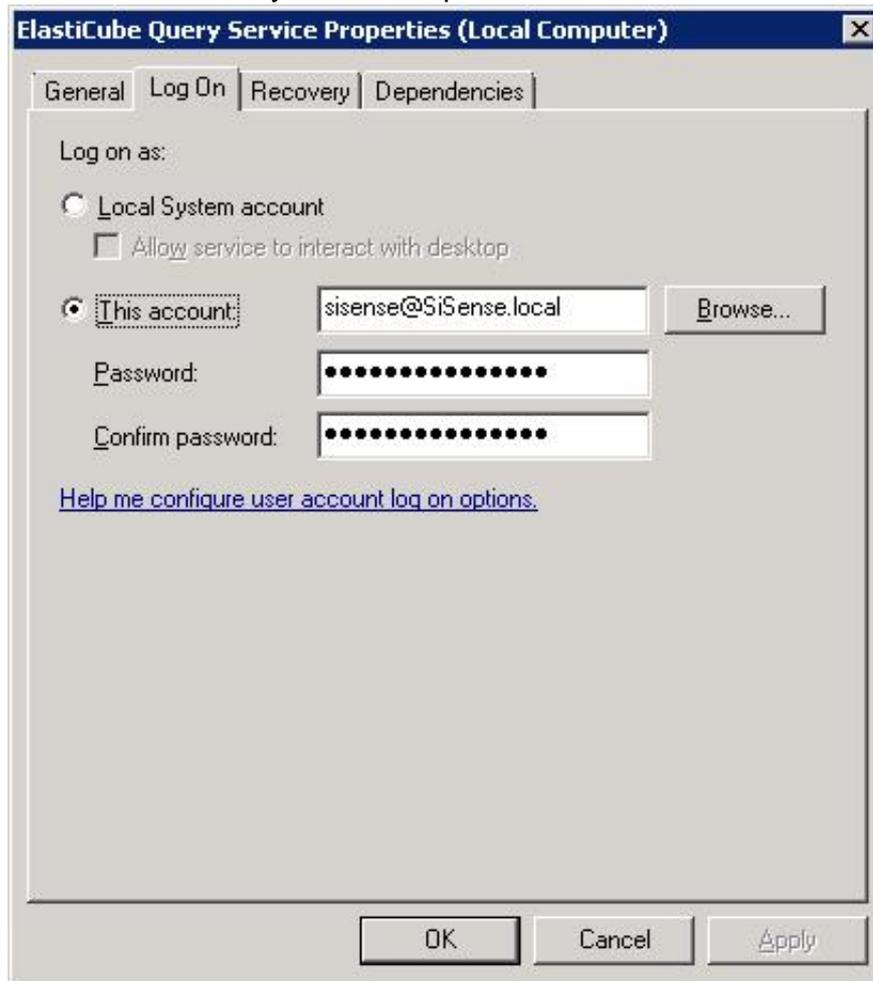
8. Click Find Now, choose the user logged on to the machine and click OK



9. The selected user will be added to the names box - click OK



10. Enter and verify the user's password - click OK

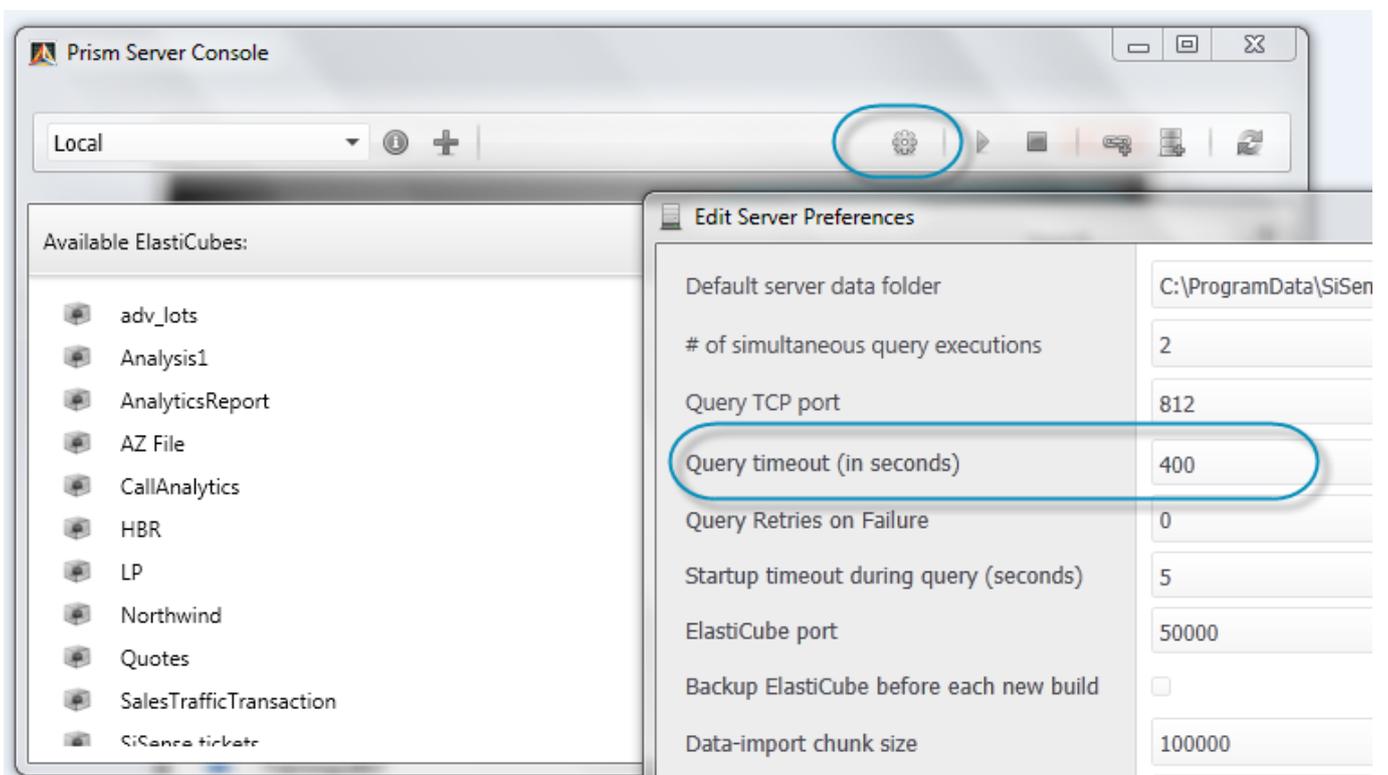


11. Restart the service.

Increase the Timeout Period for Queries

By default queries performed in BI Studio are restricted to a few seconds before timing out. In order to increase this time out period for queries follow the steps below.

1. Open Prism Server Console.
2. In the top right click on the silver cog icon - the first icon in the series ('Click to View/Change server preferences')
3. In the popup window click 'More Settings...'
4. Scroll to the option labelled 'Query Timeout (in seconds)'
5. Enter the time-out period in seconds
6. Click OK



Prism Web Management

PRISM WEB SECURITY SETTINGS

Overview

This article outlines Prism Web security configuration settings. Prism Web security design guidelines addresses all common application vulnerabilities, including:

- Inputs validation
- Authentication
- Authorization
- Session management
- Cryptography
- Exception and configuration management

Although PrismWeb security measures are comprehensive, it is always recommended to use secure transport protocol (SSL) on production servers in order to ensure the safety of sensitive information.

Security Configuration settings

In order to setup and configure Prism Web security, a number of configuration settings are available using the following config file:

%PrismWeb installation folder%\App_Data\Configurations\security.config

Settings	Option	Description	Default
browserPasswordManagerEnabled	True / False	Control the browser password manager options to remember passwords.	true
persistentCookieEnabled	True / False	Show/hide the “remember me” option in log in screen in order to allow persistent session cookie after closing the browser.	true

activationLinkExpirationNumber		Total time in days where the account activation link is valid.	7 days	
captchaEnabled	True / False	Enable "Real Person" (captcha) challenge in login screen.	false	
maxInvalidLoginAttemptsToUseCaptcha	Number	Total failed login attempts before displaying captcha	3	
loginTimeFrame	Number	Total time in minutes where the login screen is active for submitting credentials.	5 minutes	
authCookieName	String	Session cookie name	.prism	
authCookiePath	String	Session cookie location	Default browser cookies path	
authCookieDomain	String			Session
authCookieRequireSsl	True/False			Set if session requires secure connection
authCookieTimeout	Number			Cookie expiration in minutes
requireSSL	True/False			Enforce secure connection for application

--	--	--	--	--

Settings	Option	Description	Default
minRequiredNonalphanumericCharacters	Number	Password strength parameter	0
minRequiredPasswordLength	Number	Password min characters	7
requiresUniqueEmail	True/False	Users email are unique	True
enablePasswordReset	True/False	Enable password reset	True

Configuring SSL on PrismWeb

In order to ensure a secure communication channel between PrismWeb server and clients, it is recommended to configure the hosting IIS to work under secure socket layer certificate (SSL).

A detailed guide covering the steps how to install the certificate in the IIS can be found here:

- [IIS 7 Guide](#)
- [IIS 5-6 Guide](#)

After installing the certificate and applying it to the website, change the following configurations:

C:\Program Files\Sisense\PrismWeb\App_Data\Configurations\security.config

Settings	Option
requireSSL	true
authCookieRequireSsl	true

C:\Program Files\Sisense\PrismWeb\vnnext\config\default.yaml

Settings	Option
requireSSL	True

Manual Configuration for Prism Web



Instruction below outline deployment in the following environments:

- Windows 2003 R2 with IIS6
- Windows Server 2008 with IIS 7, 7.5
- Windows Vista\7

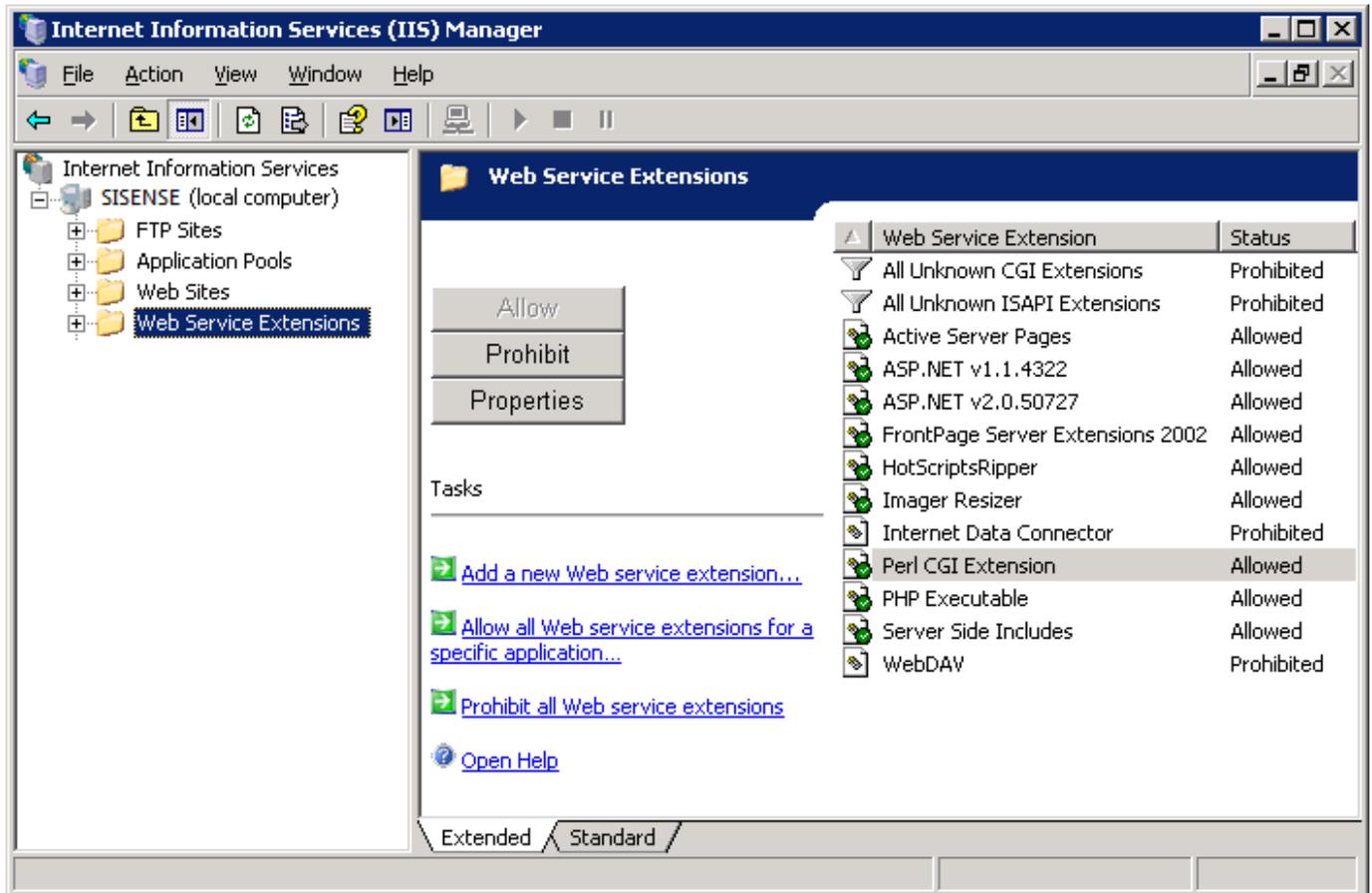
Windows 2003 R2 with IIS6

The following IIS configurations are based on the server operating system:

Windows Server 2003 R2 with IIS 6 Installing IIS in Windows Server 2003 requires your installation CD-ROM or original installation files. Please have the installation files ready before trying to setup IIS. Sample Screen shots are taken from Windows Server 2003 Standard Edition. Please follow the steps shown below to setup

To open Internet Information Services Manager in your start menu click Run and type inetmgr. Expand your server node and click Web Service Extensions. In the right pane, make sure ASP.NET v4.0 extension is in status Allowed.





Creating the Website

1. After installing PrismWeb, create a new website named "PrismWeb" using the Web Site Creation Wizard. - IP: All Unassigned

- port: available TCP port (default :80)

- Host header: leave blank or host name

- Path:

64-bit version : "C:\Program Files \(\x86\)Sisense\PrismWeb"

32-bit version : "C:\Program Files \Sisense\PrismWeb"

- Access Permissions: Read, Execute

Click Application Pools and create a new one called "PrismWeb".

Web Site Creation Wizard [X]

Web Site Description
Describe the Web site to help administrators identify it.

Type a description of the Web site.

Description:

< Back **Next >** Cancel

Web Site Creation Wizard [X]

IP Address and Port Settings
Specify an IP address, port setting, and host header for the new Web site.

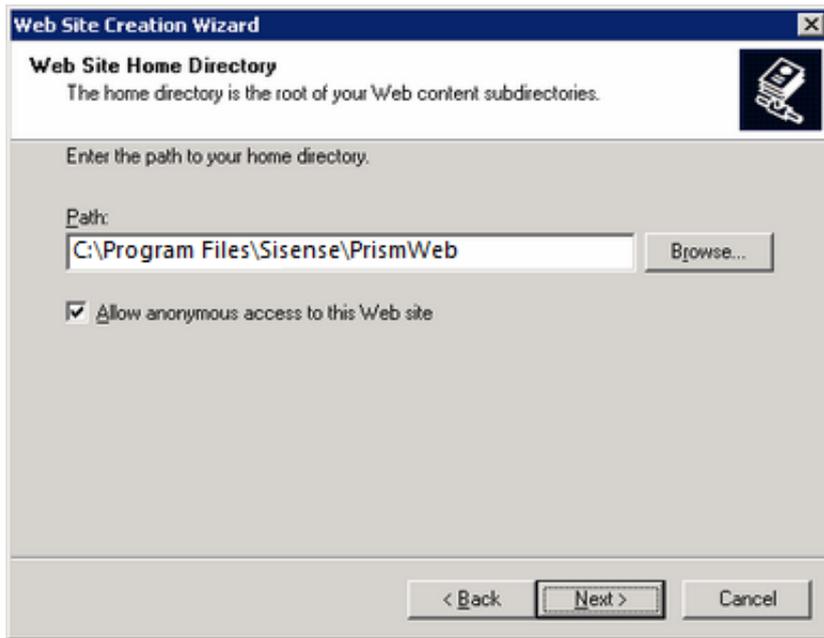
Enter the IP address to use for this Web site:

TCP port this Web site should use (Default: 80):

Host header for this Web site (Default: None):

For more information, read the IIS product documentation.

< Back **Next >** Cancel



Creating the Website

2. Right-click on the PrismWeb Website and Properties:

Tab: ASP.NET

ASP.NET version: 4.0.30319

Tab: Home Directory

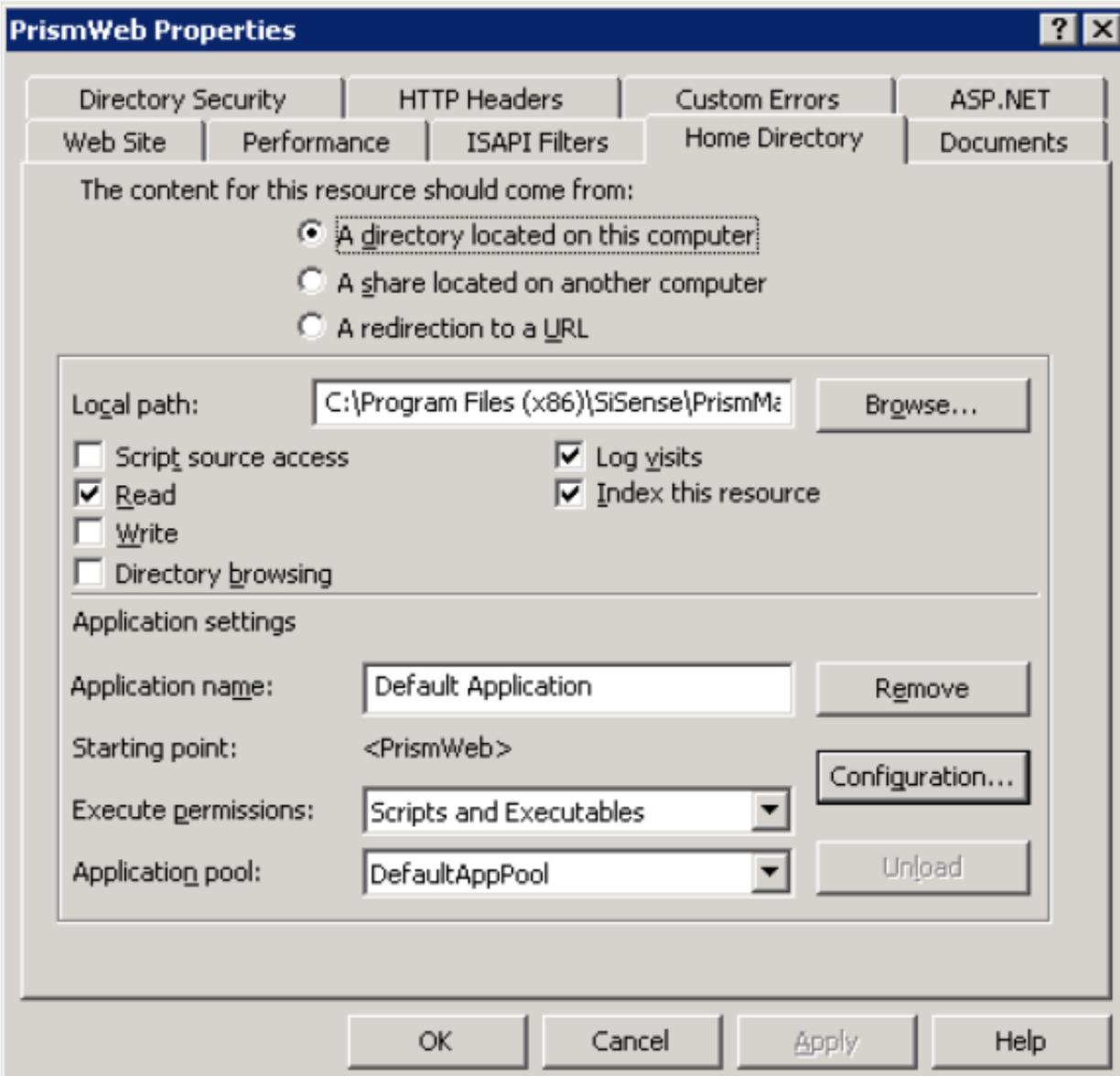
1. Click Configuration... button.

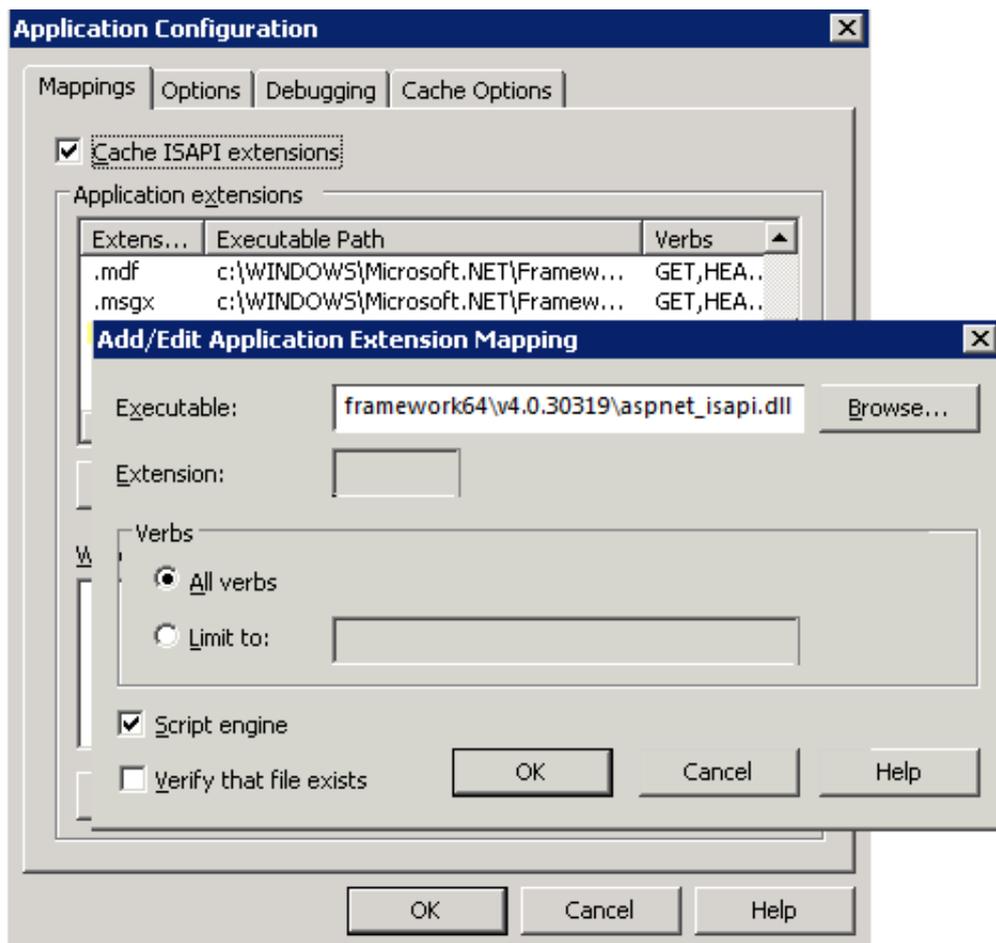
2. On Mappings tab, make sure you have mappings to the .svc extension and make sure their Executable Path is set to ~\aspne_isapi.dll. Click Edit... and copy the path.

3. Click Insert on the buttons below.

4. Paste the executable path and uncheck the "Verify if file exists".

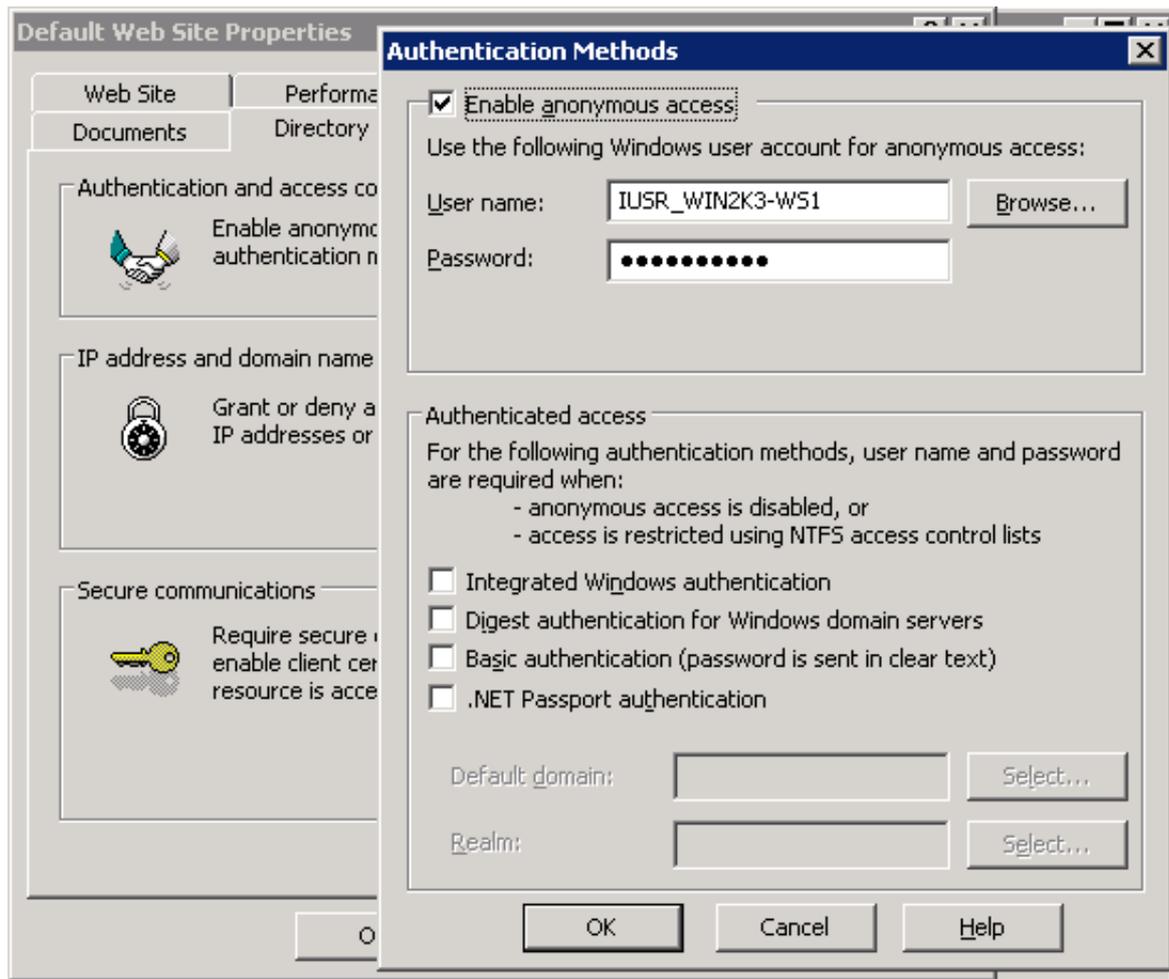
5. On Application pool, select PrismWeb





Tab: Directory Security

1. In Authentication and access control, click Edit. The Authentication Methods window appears.
2. Make sure that Enable anonymous access checkbox is selected and that none of the checkboxes in Authenticated access is selected.



Windows Server 2008 with IIS 7, 7.5

IIS comes preloaded in Windows 2008 Server. Please follow the steps shown below to setup. Sample Screenshots are taken from Windows 2008 Server R2 Enterprise Edition.

1. Go to Start -> Control Panel -> Programs -> Turn Windows Features on or off

2. Select and expand Roles tab.

3. If you do not see Web Server (IIS) role, you must add it using the Add Roles button:

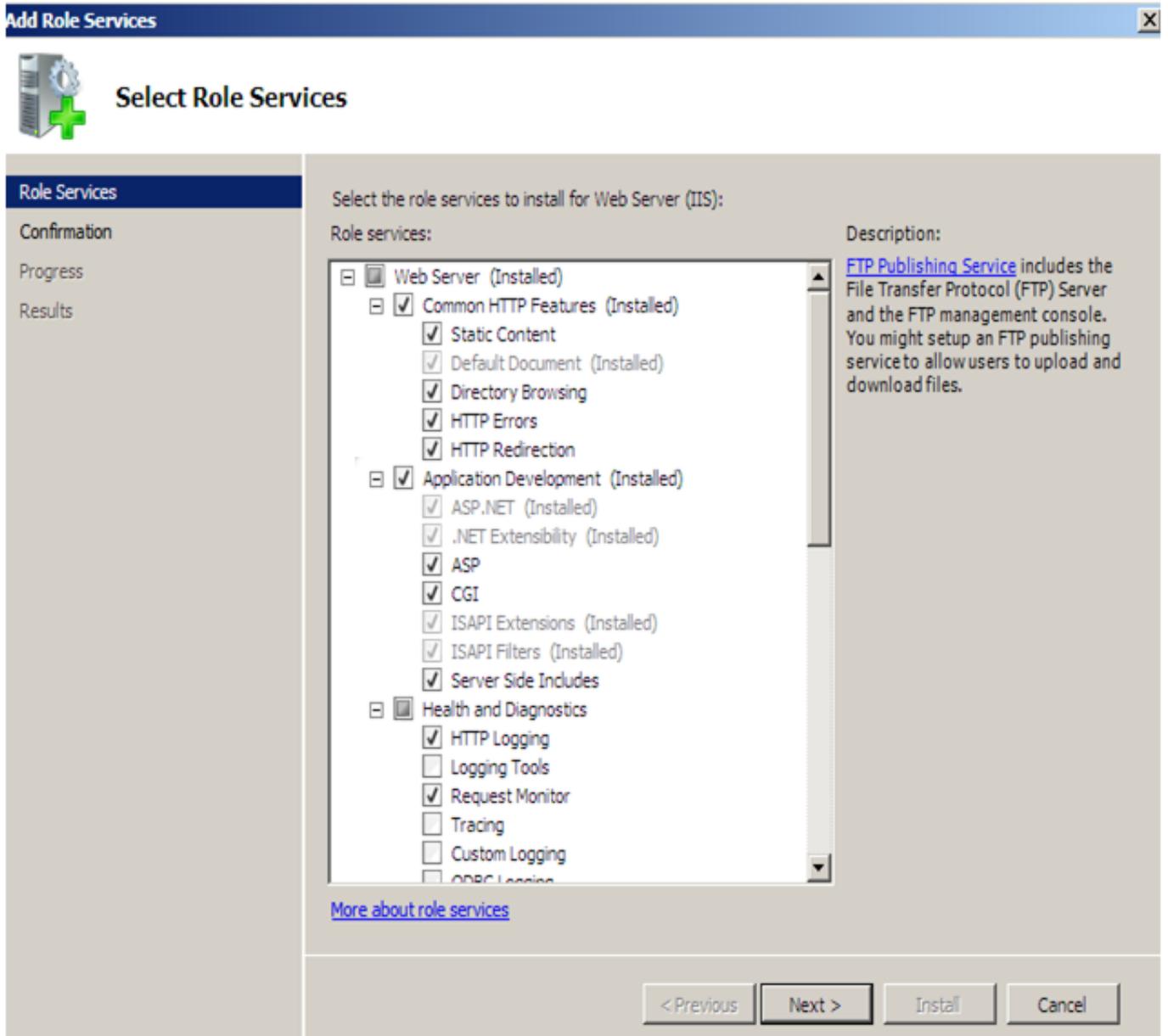
- Click Web Server (IIS) and in the right pane, expand Role Services.
- Make sure the following services are installed:
 - Common HTTP Features? Static Content.
 - Application Development? ASP.NET, .NET Extensibility, ISAPI Extensions and ISAPI Filters.

- Management Tools ?IIS Management Console, IIS Management Scripts and Tools, and Management Service.

4. In the navigation tree on the left, click Features.

5. Make sure the following feature added:

.NET Framework 3.5.1 Features ?WCF Activation? HTTP Activation.



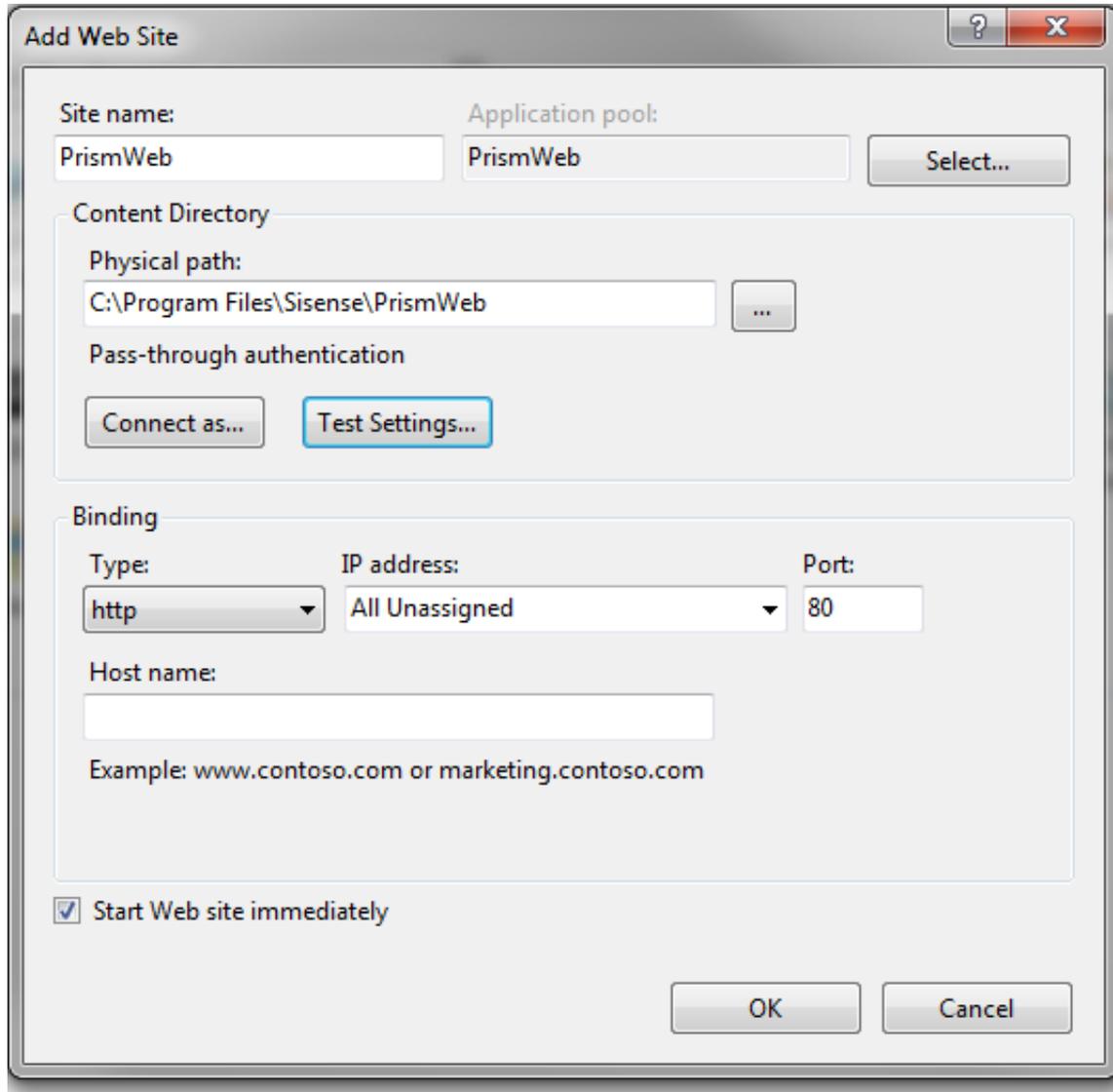
Creating the Website

1. Expand your server node. Right click "Sites"?Add Web Site. - Site name: PrismWeb - Port: available TCP port (default :80)

- Physical Path:

64-bit version: "C:\Program Files (x86)\Sisense\PrismWeb"

32-bit version: "C:\Program Files\Sisense\PrismWeb"



2. On the left pane select "Application pools"

3. Click "PrismWeb".

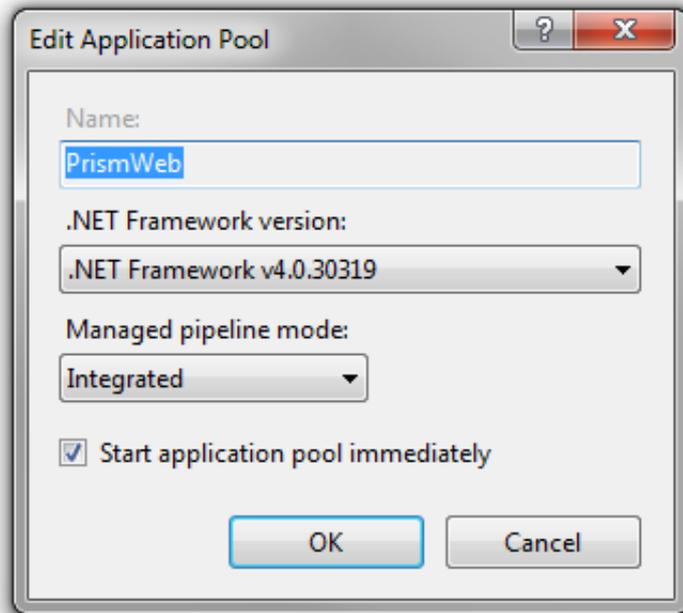
4. Select:

“.Net Framework version” : Net Framework v4.0.0319

“Managed pipeline mode” : Integrated

5. Select on the right pane “Advanced Settings”.

Note : when using 64 bit installation make sure that under General ? Enable 32-Bit Application is False



6. Expand Sites node and click PrismWeb.

7. In section IIS on the right, double-click Authentication.

8. Make sure Anonymous Authentication and Forms Authentication are enabled and all other authentications are disabled.

Authentication

Name	Status	Response Type
Anonymous Authentication	Enabled	
ASP.NET Impersonation	Disabled	
Basic Authentication	Disabled	HTTP 401 Challenge
Digest Authentication	Disabled	HTTP 401 Challenge
Forms Authentication	Enabled	HTTP 302 Login/Redirect
Windows Authentication	Disabled	HTTP 401 Challenge

Windows 7

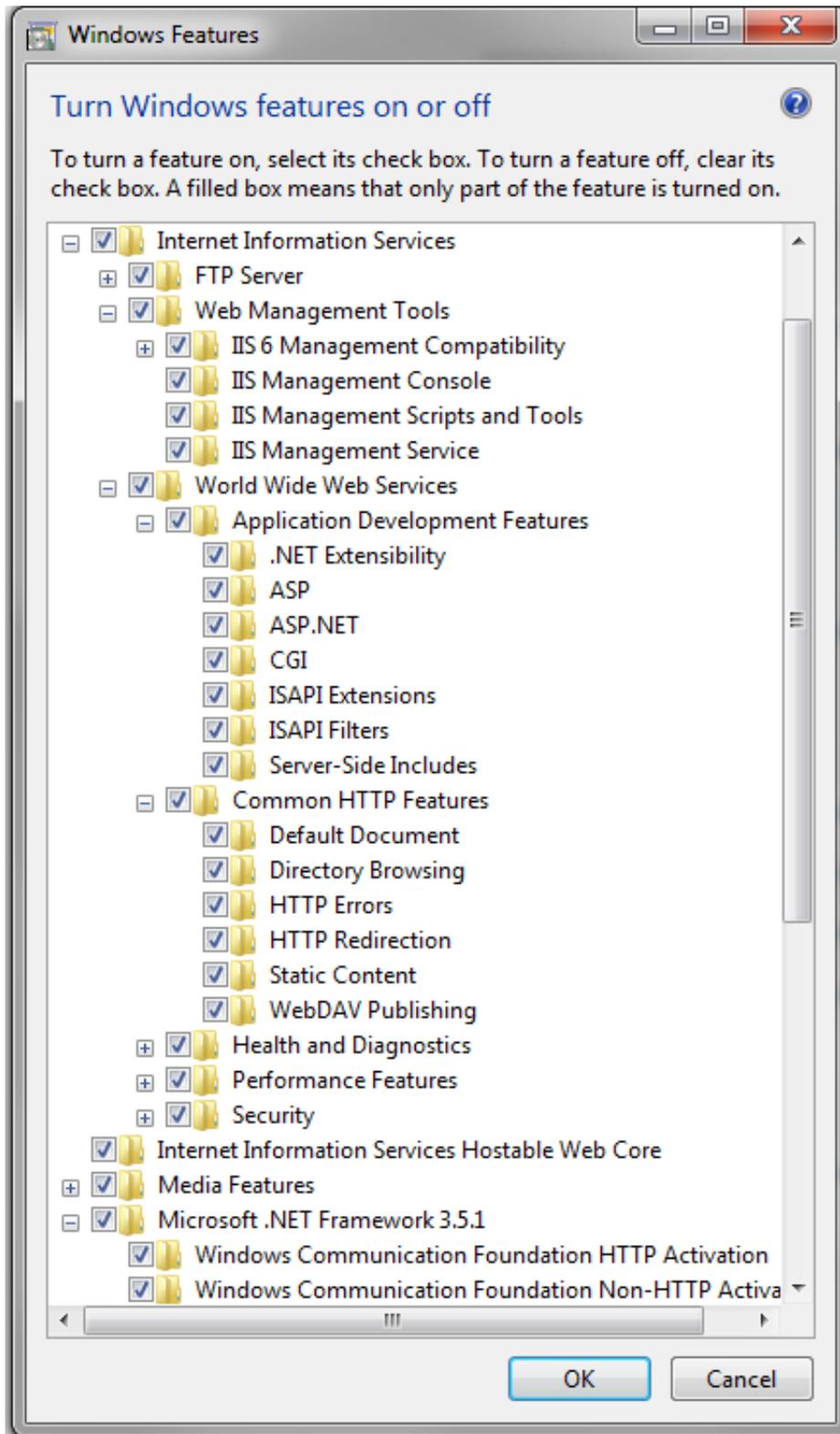
IIS comes preloaded in Windows 7. Please follow the steps shown below to setup. Sample Screenshots are taken from Windows 7 Professional Edition.

1. Go to Start -> Control Panel -> Programs -> Turn Windows

Features on or off .

2. Turn on Internet Information Services.

3. Under Internet Information Services, turn on the features as in the screen shot:



1. After installing the PrismWeb, Open Computer Management. under Services and Applications select Internet Information Services. In the navigation of the IIS manager, expand your server node. Right click "Sites"?Add Web Site:

- Site name: PrismWeb

- Port: available TCP port (default :80)

- Physical Path:

64-bit version: "C:\Program Files \x86)\Sisense\PrismWeb"

32-bit version: "C:\Program Files \Sisense\PrismWeb"

2. On the left pane select "Application pools"

3. Click "PrismWeb".

4. Select:

“.Net Framework version” : Net Framework v4.0.0319

“Managed pipeline mode” : Integrated

5. Select on the right pane "Advanced Settings".

Note : when using 64 bit installation make sure that under General ? Enable 32-Bit Application is False

Add Web Site

Site name: PrismWeb Application pool: PrismWeb

Content Directory

Physical path: C:\Program Files\Sisense\PrismWeb

Pass-through authentication

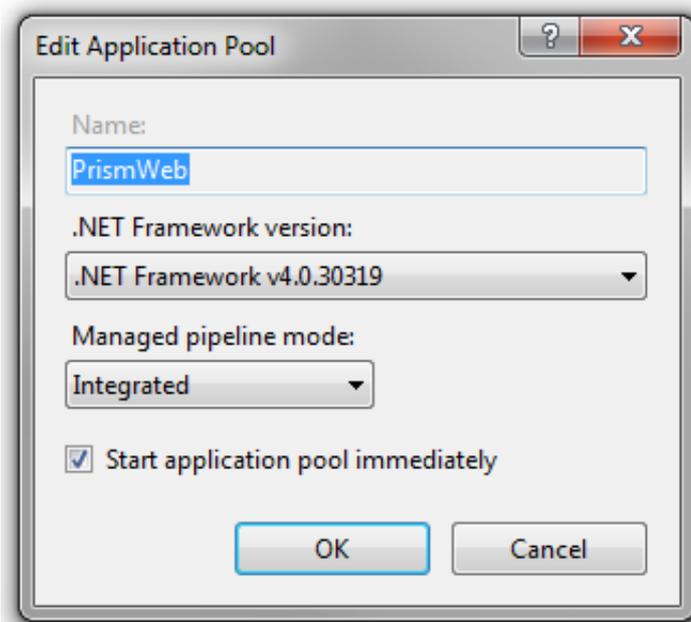
Binding

Type:	IP address:	Port:
http	All Unassigned	80

Host name:

Example: www.contoso.com or marketing.contoso.com

Start Web site immediately

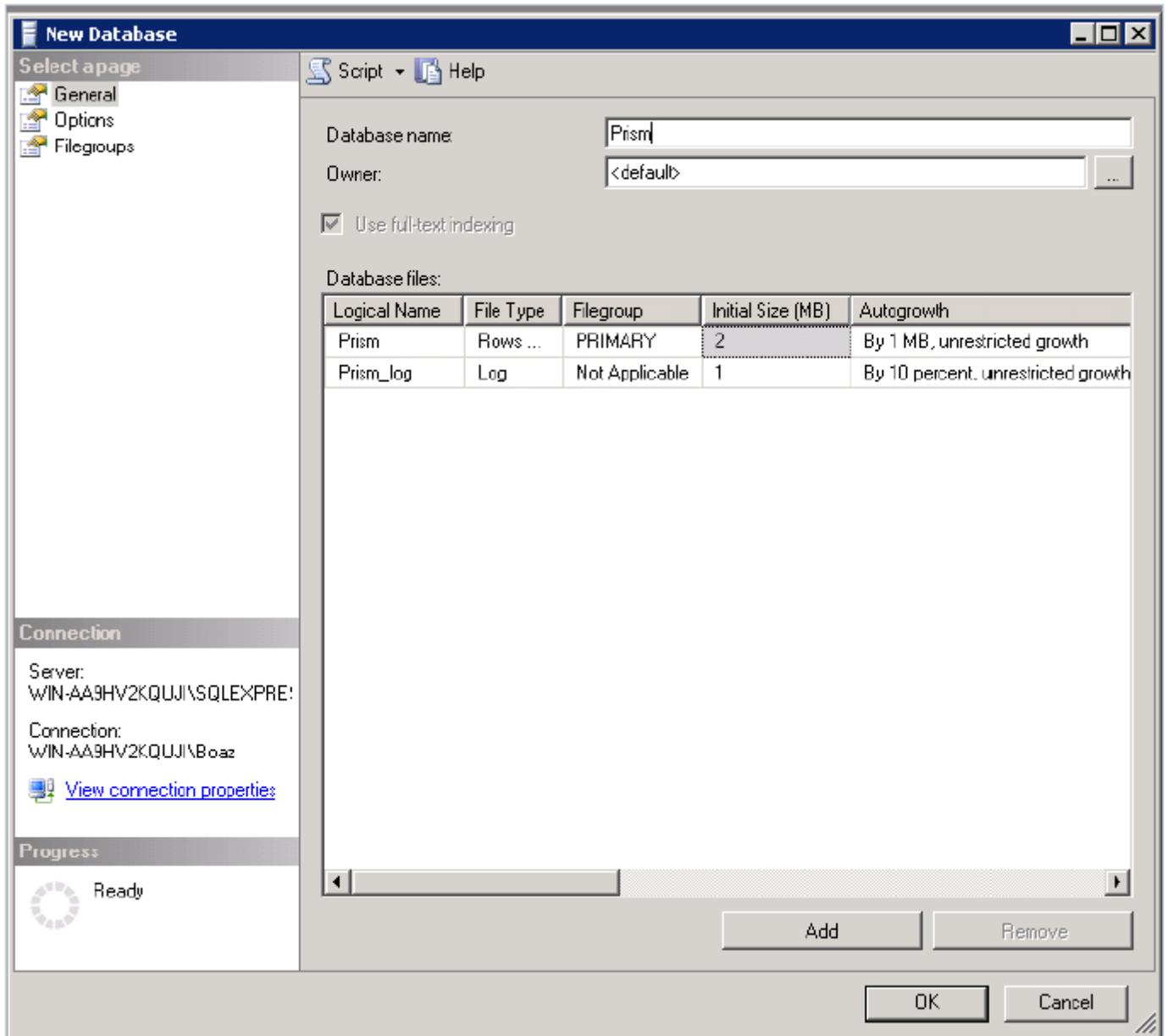


Configuring the Database

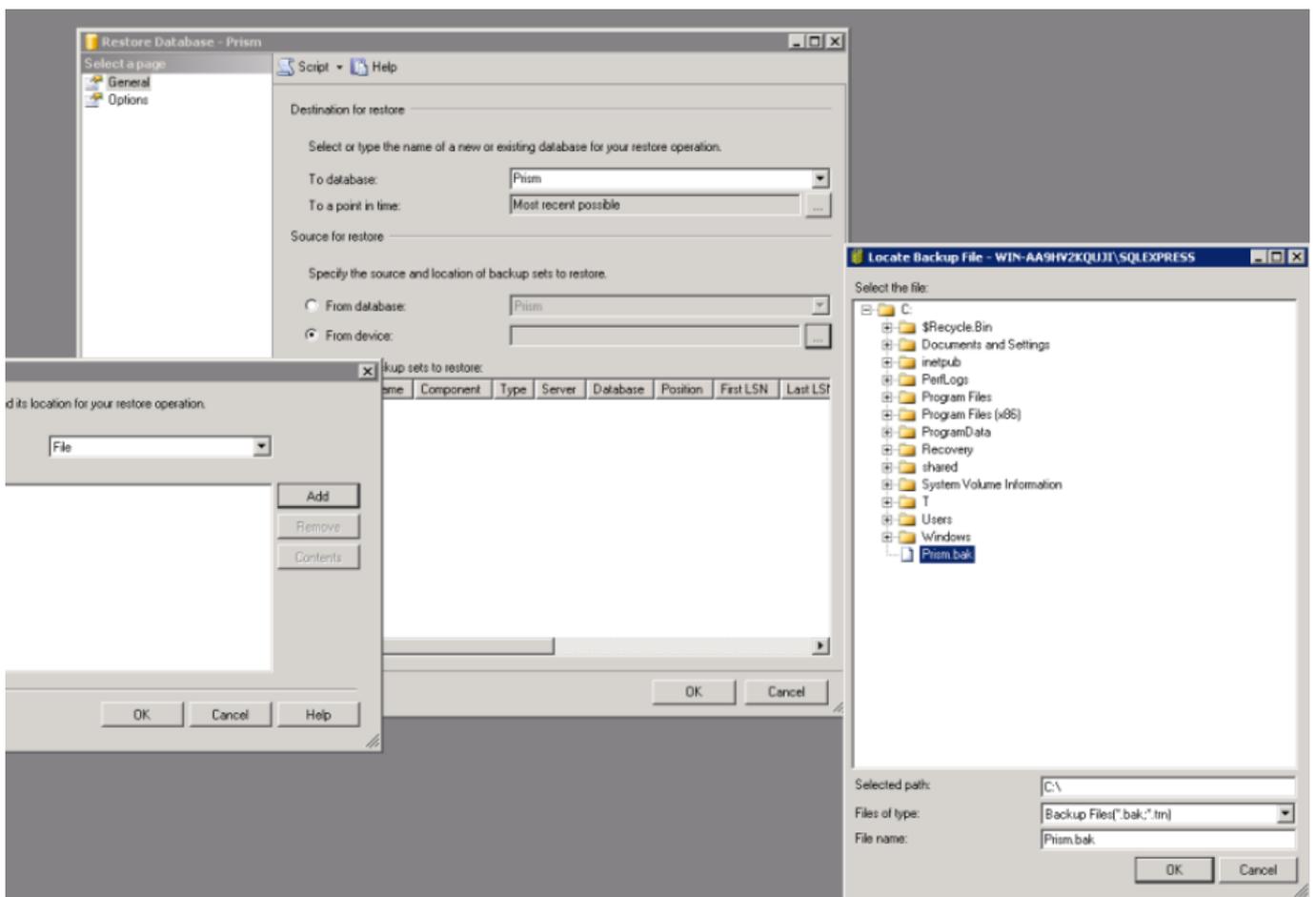
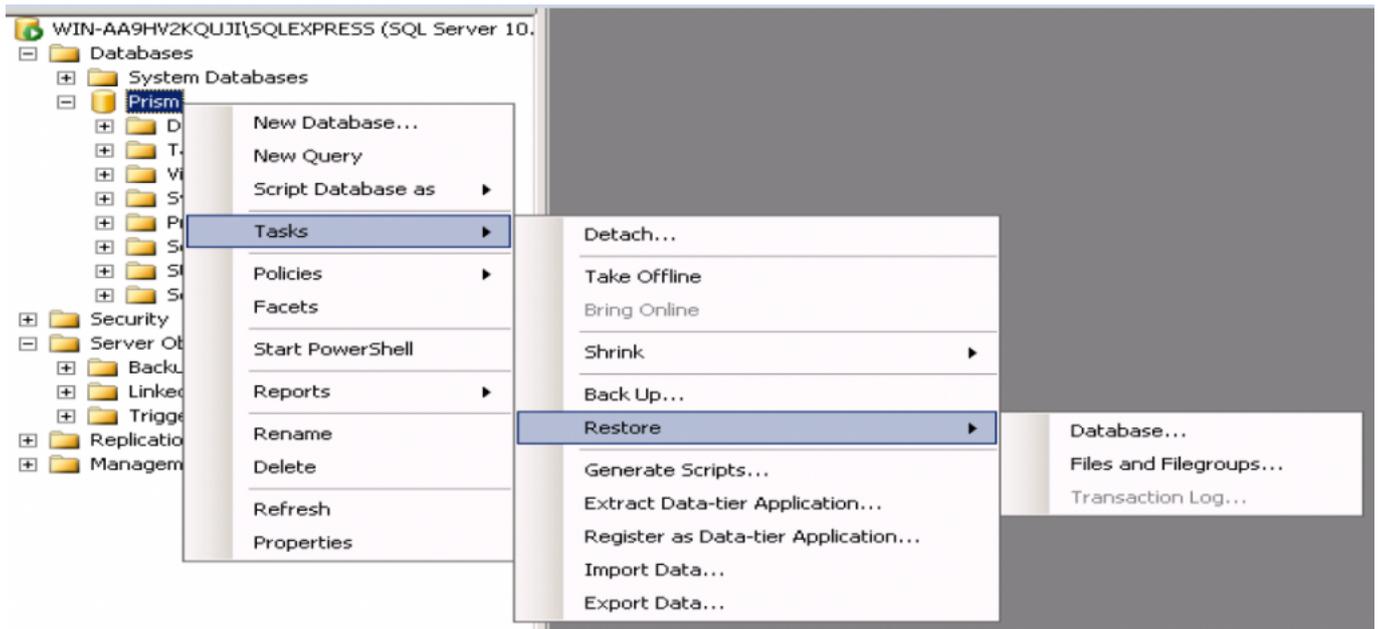
Prism Web platforms repository is stored in MSSQL database. the default database is an embedded version and doesn't require any configurations. To use any other MSSQL version follow the next steps:

The database file is in a backup form (.bak) and located in the Installation folder "\\App_Data\MSSQL\Prism.bak". The .bak file need to be restored to the MSSQL server . In the default configuration, the Prism Web database is set to work with "Windows Authentication". you can change this configuration at any time. Make sure you have running instance of MSSQL Express or MSSQL Server. Restoring Prism Web .bak file:

- 1.Open Microsoft SQL Server Management Studio.
- 2.Right click on the databases folder and click New Database..
- 3.Enter database name : Prism and ok.



4. Select Prism database ?Right click... ?Tasks?Restore?Database ?From device
5. Select the Prism.bak under Installation Folder\App_Data\MSSQL\Prism.bak

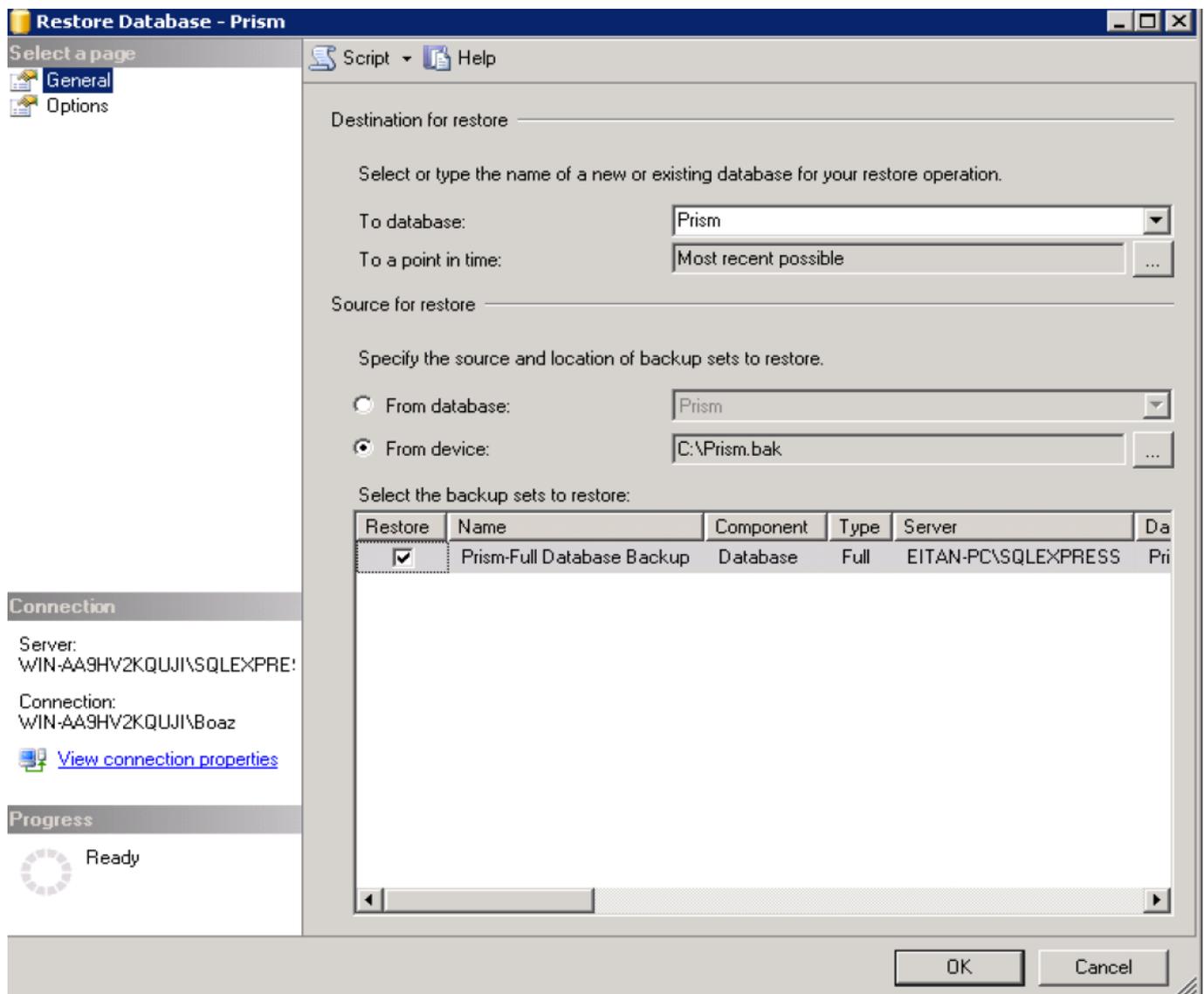


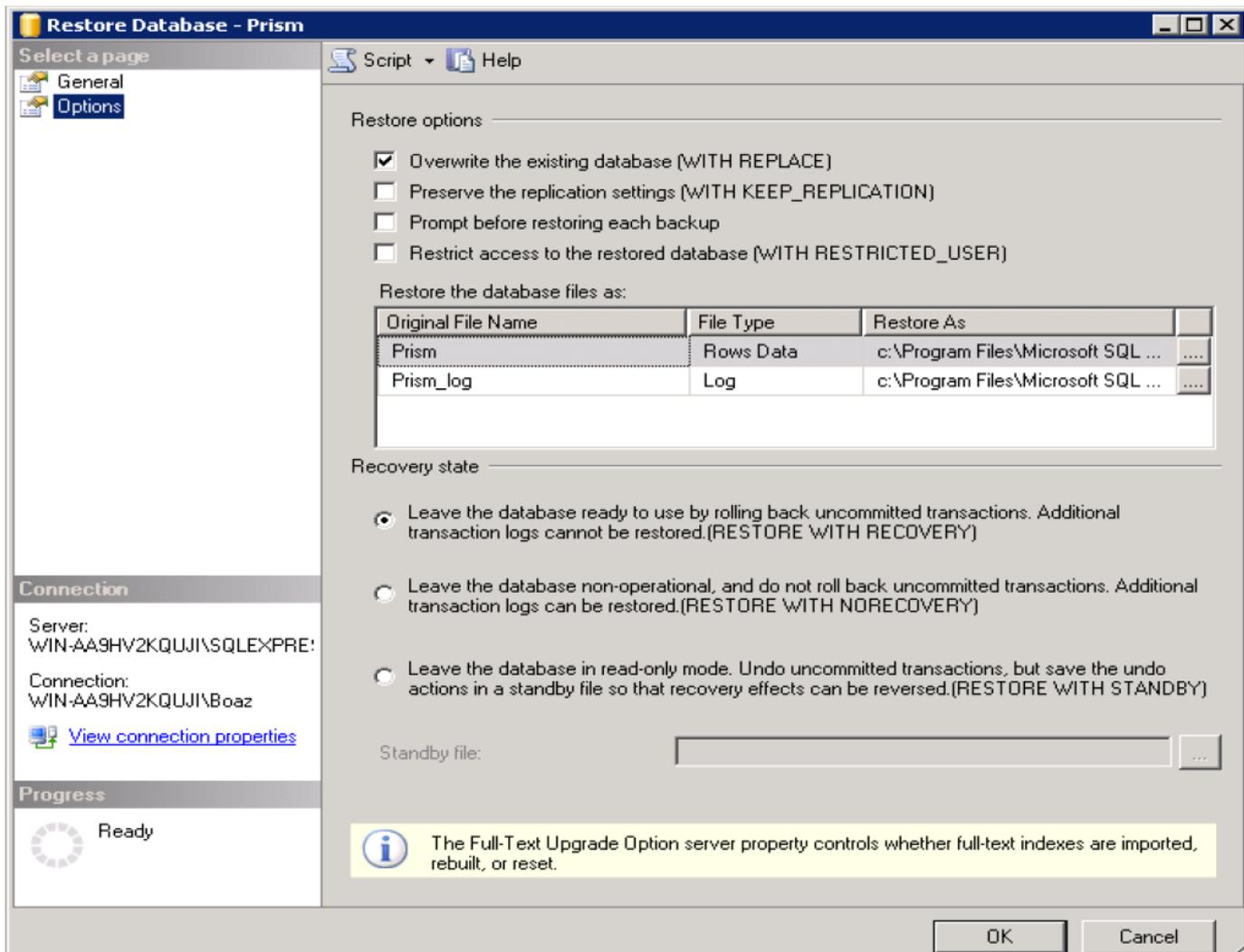
6. Check “Restore”.

7. Under “Options” on the left pane:

Check “overwrite the existing database(WITE REPLACE)”

8. Click OK.





Configuring the Database – Setting Security

Using Windows Authentication Using the predefined MSSQL windows authentication requires to setup the PrismWeb application pool Identity to run under the NetworkService which is a built-in windows identity. This setting is by default only in windows server 2003. note that this is only relevant to the case which the sql server is running on the same machine the IIS is hosted in. in all other cases use “Sql Server mix mode authentication”. To change the application pool identity:

1. In the IIS manager select PrismWeb application pool.
2. On the right pane click “Advanced Settings...”
3. Under Process Model select NetworkService for Identity. After you complete the above steps, configure the PrismWeb database to run under the NetworkService user:

Procedure

1. In the SQL Server management, under Security – Logins

Make sure “NT AUTHORITY\NETWORK SERVICE” exist. If not, create the login user

2. Map the user to the Prism database:

- Under Logins, Select the “NT AUTHORITY\NETWORK SERVICE” and click Properties.

- on the left pane select User Mapping and check Prism.

- on the bottom pane check db_owner and OK.

Login - New

Select a page

- General
- Server Roles
- User Mapping
- Securables
- Status

Script Help

Login name: Search...

Windows authentication

SQL Server authentication

Password:

Confirm password:

Specify old password

Old password:

Enforce password policy

Enforce password expiration

User must change password at next login

Mapped to certificate

Mapped to asymmetric key

Map to Credential Add

Mapped Credentials

Credential	Provider
------------	----------

Remove

Default database:

Default language:

OK Cancel

Connection

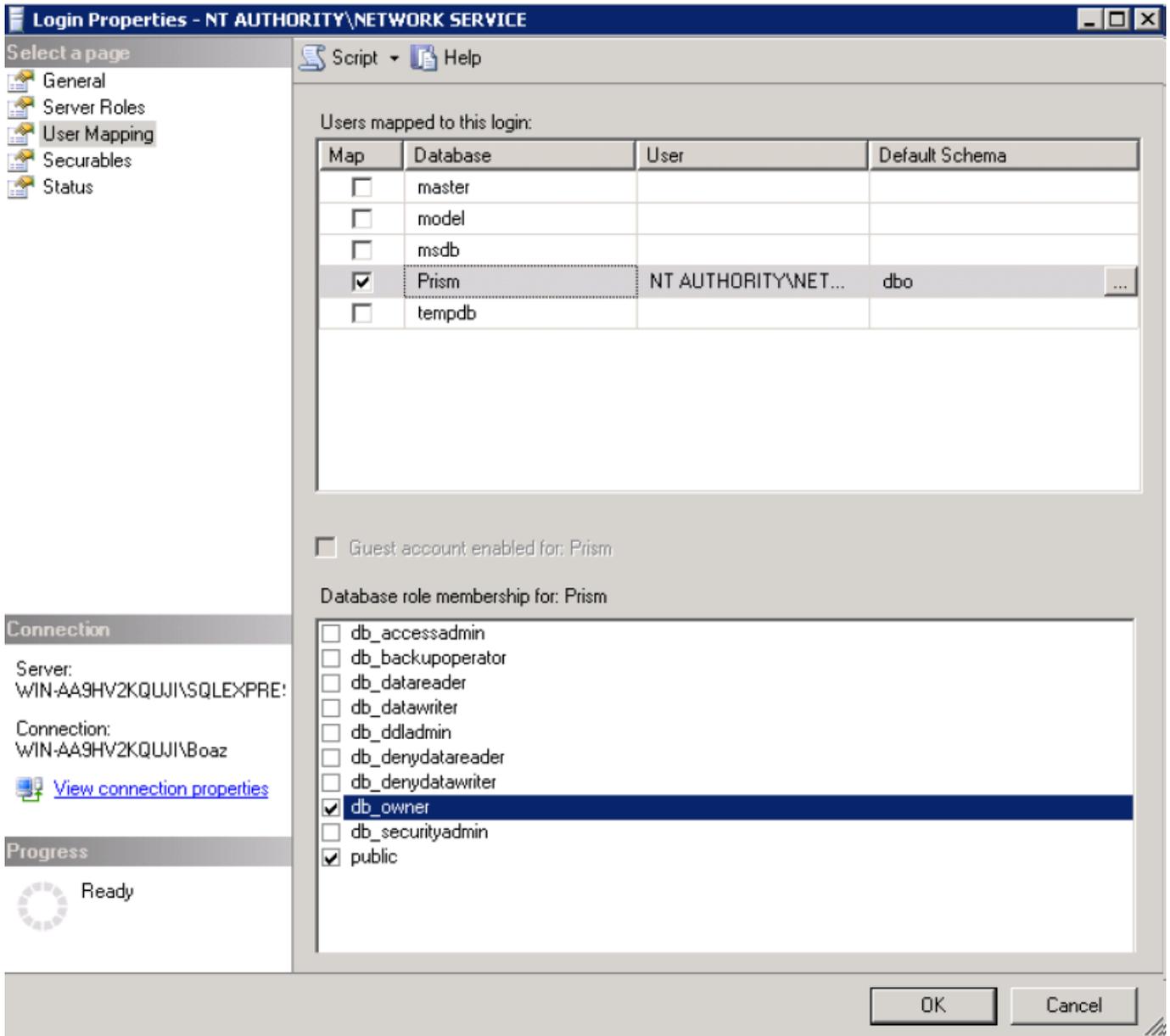
Server: WIN-AA9HV2KQUJI\SQLEXPRES

Connection: WIN-AA9HV2KQUJI\Boaz

[View connection properties](#)

Progress

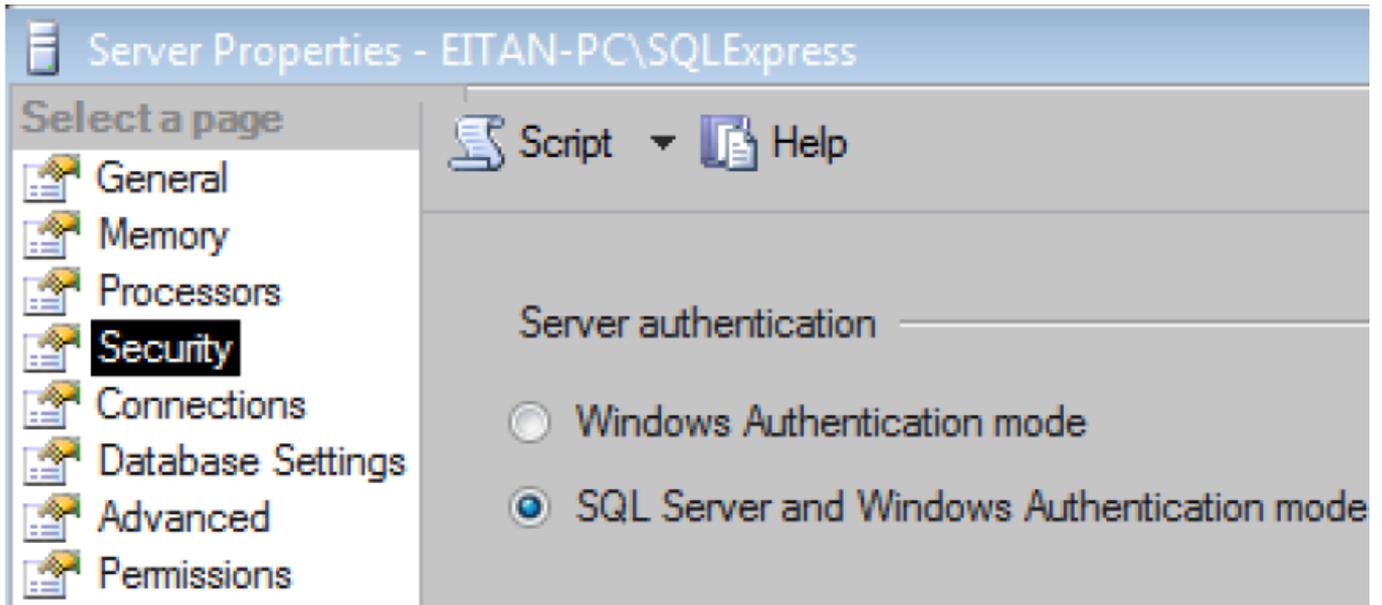
Ready



Using Sql Authentication

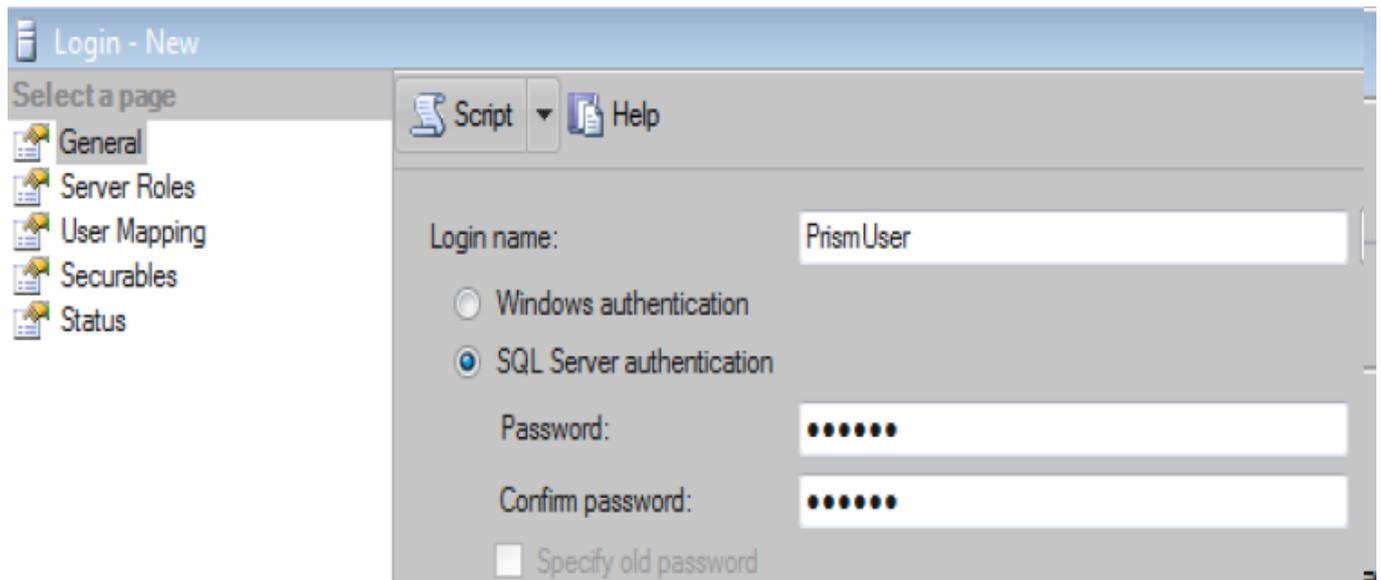
Using sql authentication you configure your database to allow Internet users access to read and (if required) write to your data by user credentials. Make sure the MSSQL is configured to use sql authentication:

- Right click on the Server.
- Click Properties.
- Select Security.



Procedure

1. In the SQL Server Management Tools, Right click on Logins under the Security Node and select New Login.
2. Type the name of your database user in the Login Name box and select SQL Authentication. Set a password and uncheck the Enforce Password Policy, Enforce Password Expiration and User must change password on next login. Click Ok.
3. Under User Mapping check Prism.
4. On the bottom pane check db_owner and OK.



Updating the Connection String

Go to Installation folder\\App_Data\\Configurations\\db.config Make sure DataProviderName = "PrismEntitiesMSSQL" update the provider connection string Data Source with the MSSQL Server name

Embed a Dashboard into Another Application

Prism Web dashboards can be embedded into other web application using iframes. In order to avoid authenticating with the PrismWeb application, first use the login REST API and then address the dashboard.

Embedding a Dashboard

1. Copy the dashboard address from the web browser.



2. Remove the “s” from “dashboards” and the “#” as well as the “=” sign and following numbers.

For Example...

Original URL: `http://192.192.192.10/dashboards#f=101&d=314`

IFrame URL Format: `http://192.192.192.10/dashboard?d=314`

3. Set the iframe url src to the new address.

Example

Making a Dashboard Public

PUBLIC DASHBOARDS

Making a dashboard enables anyone, regardless of authentication, to access the dashboard on the web using a link.

Your license must allow you to enable public dashboards, you can confirm this with your Sales Executive. Once you have confirmed branding rights, please send a new activation key to support@sisense.com

Enabling a Public Dashboard

1. To enable a public dashboard in Prism Web follow the steps below:
2. Go to C:\Program Files\Sisense\PrismWeb\App_Data\Configurations.

OR for older versions C:\Program Files
(x86)\Sisense\PrismWeb\App_Data\Configurations.

3. Click on security.config or open it with notepad.
4. In the security file change `allowAnonymousAccess="false"` to `allowAnonymousAccess="true"`
5. Save the file after making any changes.
6. Load your prism web site and press Shift + F5
7. Select the relevant dashboard.
8. Click 'Share' and then click 'Make public'
9. Copy the link and click 'Save Changes'

Single Sign-on (SSO)

The Prism Web environment supports Single Sign-on (SSO) within a second level domain, meaning the issued authentication cookie can apply to the main domain and all corresponding sub-domains.

For Example -

Main Application: [http:// www.myportal.com](http://www.myportal.com)

PrismWeb: <http://prismweb.myportal.com>

Enabling SSO

In order to enable the shared authentication cookie follow the steps below.

1. Under PrismWeb\ App_Data\ Configurations\security.config in **securityConfiguration**, change the "**authCookieDomain**" property to your domain (**myportal.com**)

Example:

2. Perform an http post request to remote login REST API with json parameters to address: **<http://yourprismweb/account/sso>**

Parameters:

```
{"username\":\"username\",\"password\":\"password\"}
```

Response body contract:

```
{
  "authentication": {
    "ticket": "token info",
    "cookiename": "cookie name ",
    "domain": "cookie domain",
    "isauthenticated": true/false
  }
}
```

C# Example

```
class Program
{
    static void Main(string[] args)
    {
```

```

        var webRequest = GetPreparedUploadWebRequest();
        StreamWriter requestWriter;
        using (requestWriter = new StreamWriter(webRequest.GetRe
questStream()))
        {
            requestWriter.Write("{\"username\": \"un@sisense.com\", \"
password\": \"pass\"}");
        }
        // wait for response from server.
        using (WebResponse wresp = webRequest.GetResponse())
        {
            // set resulting response message.
            string responseFromServer = string.Empty;
            // get response message.
            using (Stream stream2 = wresp.GetResponseStream())
            using (StreamReader reader2 = new StreamReader(stream2))
            responseFromServer = reader2.ReadToEnd();
        }
    }
    private static HttpWebRequest GetPreparedUploadWebRequest()
    {
        // set address.
        string url = "http://yourprismweb/account/sso";
        // prepare request.
        HttpWebRequest wr = (HttpWebRequest)WebRequest.Create(ur
l);

        wr.ContentType = "application/json";
        wr.Method = "POST";
        wr.KeepAlive = false;
        //wr.AutomaticDecompression = DecompressionMethods.GZip;
        //wr.SendChunked = true;
        wr.Timeout = (int)TimeSpan.FromMinutes(30).TotalMillise
conds;

        wr.Credentials = System.Net.CredentialCache.DefaultCrede
ntials;

        return wr;
    }
}

```

3. {
 - the authentication cookie with the response data:
 - "authentication": {
 - "ticket": "token info",
 - "cookiename": "cookie name ",
 - "domain": "cookie domain",
 - "isauthenticated": true/false

```
}  
}
```

PHP Example

Sign Out

Perform an http post request to remote logout

<http://yourprismweb/logout>

Response contract:

response.success

C# Example

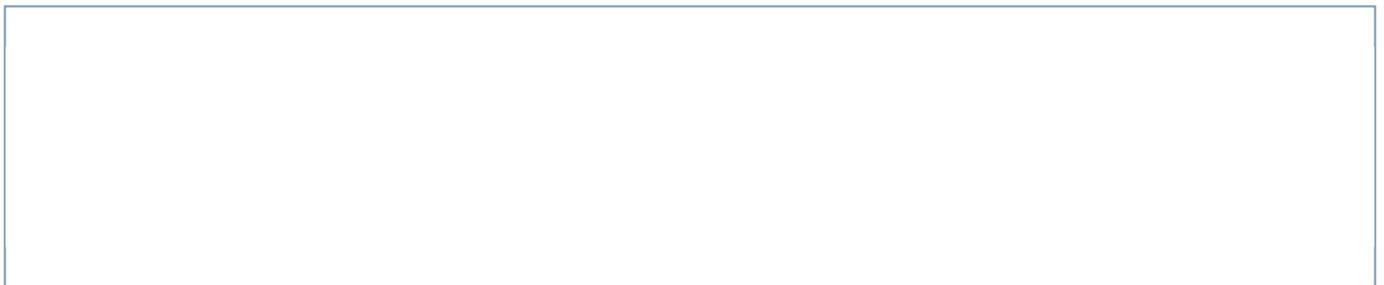
```
class Program  
{  
    static void Main(string[] args)  
    {  
        var webRequest = GetPreparedUploadWebRequest();  
        // wait for response from server.  
    }  
}
```

```

using (WebResponse wresp = webRequest.GetResponse())
{
    // set resulting response message.
    string responseFromServer = string.Empty;
    // get response message.
    using (Stream stream2 = wresp.GetResponseStream())
    using (StreamReader reader2 = new StreamReader(stream2))
    responseFromServer = reader2.ReadToEnd();
}
}
private static HttpWebRequest GetPreparedUploadWebRequest()
{
    // set address.
    string url = "http://yourprismweb/logout";
    // prepare request.
    HttpWebRequest wr = (HttpWebRequest)WebRequest.Create(url);
    wr.ContentType = "application/json";
    wr.Method = "POST";
    wr.KeepAlive = false;
    //wr.AutomaticDecompression = DecompressionMethods.GZip;
    //wr.SendChunked = true;
    wr.Timeout = (int)TimeSpan.FromMinutes(30).TotalMilliseconds;
    wr.Credentials = System.Net.CredentialCache.DefaultCredentials;
    return wr;
}
}

```

PHP Example



Adding a User

Below are code examples to add a user.

PHP Example

Deleting a User

Below are code examples to delete a user.

PHP Example

Re-Branding Prism Web

Re-branding enables you to change the logo, website redirect, favicon, product and company details in Prism Web.

Your license must allow you to perform re-branding, you can confirm this with your Sales Executive. Once you have confirmed branding rights, please send a new activation key to support@sisense.com

Re-Branding Process

To re-brand Prism Web follow the steps below:

1. Create a logo sized 100x27px and transparent, save it as a PNG file.
2. Create a folder C:\Program Files\Sisense\PrismWeb\Prismlogo and save the logo in this folder as logo.png
3. Go to C:\Program Files\Sisense\PrismWeb\App_Data\Configurations. OR in older versions C:\Program Files (x86)\Sisense\PrismWeb\App_Data\Configurations.
4. Click on branding.config or open it with notepad.
5. In the file ensure that enabled="true".
6. Edit the line starting logolmageSrc, and edit it to read -
logolmageSrc="C:\prismlogo\logo.png". You can also update details for the following fields:
 - Product details: product=""
 - Company details: company=""
 - Favicon Image: favIconSrc="DRIVE:\FOLDER\favicon.ico"
 - Website to direct on clicking logo: websiteUrl=
7. Save the file after making any changes.
8. Load your prism web site and press Shift + F5

Adding Google Analytics Tracking to a Sisense Dashboard

Google Analytics is a popular web service for tracking web visitors and actions. The steps below describe integrating Google Analytics into a Prism Web Dashboard.

1. Place the config file below into the following folder C:\Program Files\Sisense\PrismWeb\App_Data\Configurations.
Download [config file \(change file extension from .txt to .config\)](#)
2. Place the js file below into the following folder: C:\Program Files\Sisense\PrismWeb\Resources.
Download [js file](#)
3. Edit the js file and and replace the stars (*) in the second line with your Google Analytics account details
(`_gaq.push(['_setAccount', '**-*****-*']);`)

This will enable Google Analytics tracking on all Prism Web dashboards.

Active Directory Management

Introduction

Prism Web enables sharing dashboards across multiple users by adding them to the shared users list via the sharing panel.

Those users may be picked from the current registered users list, or could be newly added by typing their email. The Active Directory Manager adds a new capability for seeking and adding users directly from the organization's Active Directory settings.

Setting the Active Directory (Quick Steps)

1. From the Settings menu, click the Active Directory Settings item.
2. Fill in your Active Directory connectivity credentials: address, domain, user name and password, then click 'Connect'.
3. Navigate, expand and drill to lower level folders, then check folders from which users can be imported.
4. Finally click 'Save Settings'.
5. You're done. The next time you type a search term in the sharing panel, it will additionally suggest you with users from the folders you have checked.

Active Directory Settings

Opening the Manager

Click the cog icon (settings) in the upper right corner.



Click the Active Directory Settings menu item.

Connecting to your Active Directory

1. Fill in the four input boxes:
 - Server Address
 - Domain
 - User Name
 - Password
2. Click Connect.

Welcome to Active Directory Settings

Fill your Active Directory login credentials, set the server address, the domain, the user name and the password and click the Connect button. Once connected, you'll be able to choose out of which folders users can be imported and be shared with your dashboards.

For example

Welcome to Active Directory Settings

Fill your Active Directory login credentials, set the server address, the domain, the user name and the password and click the Connect button. Once connected, you'll be able to choose out of which folders users can be imported and be shared with your dashboards.

Setting Folders Once connected, the first folders' level is displayed. Each folder supports the following actions:

Un/Check

Checking a folder sets it and any descendant folder of it (regardless of its check state) as a potential users' origin. Click the left hand check box in order to un/check the folder.

Expand/Collapse

Expanding or collapsing a folder can be achieved by clicking the expand/collapse thumb, double clicking the folder or pressing the keyboard's right or left arrow navigation keys. Finally, to commit changes, click Save Settings.

Manage your Active Directory Settings at dev.sisense.local

You're currently editing out of which folders from dev.sisense.local users can be imported and be shared with your dashboards. Click the Disconnect button in order to connect and set a different Active Directory.

Disconnect

- Computers (no sub-folders)
- Domain Controllers (no sub-folders)
- ForeignSecurityPrincipals (no sub-folders)
- HelpDesk50k
 - NestRoot
 - level1
 - level2 (no sub-folders)
- Managed Service Accounts (no sub-folders)
- Marketing10k (no sub-folders)
- OneLevelSearch
- Program Data
 - Sales30k (no sub-folders)
- System
- Users (no sub-folders)

Save Settings

Resetting last Settings

When entering the Active Directory Manager, after a previous valid setting was saved, two options are available – Connect and Reset.

By clicking Reset, any saved settings (login, folders) will be immediately removed.

Connect to your Active Directory at dev.sisense.local

Click the Connect button in order to manage out of which folders users can be imported and be shared with your dashboards. Click the Reset button in order to reset your Active Directory settings and start-up a new Active Directory connection.

Reset

Connect

Setting different Active Directory

In order to set a different Active Directory, you should reset your last settings by clicking Reset. If you are already connected, click Disconnect and then Reset.

Extending Prism

INTRODUCTION TO JAVASCRIPT EXTENSION

Introduction

Sisense's Javascript extendibility feature gives you a new layer of control over your dashboards. It lets you extend the abilities of your dashboard according to your needs.

You can use your own Javascript code to do the following:

- Manipulate specific widgets
- Manipulate the entire dashboard
- Access the underlying widget data
- Create custom visualizations and new widgets

There are two entry points for your Javascript code:

- Widget Scripts
- The Dashboard Script

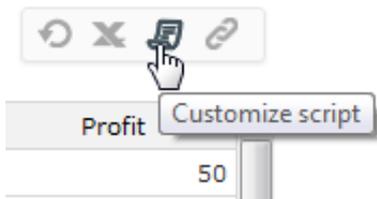
Extending Widgets

Widget Scripts

The widget script gives you Javascript entry points for a particular widget. The supported widgets for direct Javascript access are:

- Pivot
- Chart
- 'Custom Web Script' button

To open a widget script, hover over the Pivot/Chart/Button in Prism Web, and click the Customize script button in the toolbar menu.



Note

Clicking the button opens a new window showing you the initial widget script:

```

1/*
2 * to use online script you can use jquery getScript plugin as follows:
3 *
4 * $.getScript('http://abc.com/abc.js', function(data, textStatus)
5 * {
6 *     // your code which requires abc.js here
7 * });
8 *
9 * // more info at: http://api.jquery.com/jquery.getScript/
10 *
11 */
12
13
14/*
15 * executed after the dashboard was first initialized, useful when applying custom behavior to the dashboard
16 * @param (args) wraps all function arguments
17 */
18this.initializing = function (args) {
19
20};
21
22/*
23 * executed after all dashboard widget models were initialized. useful for binding with other widget events.
24 * @param (args) wraps all function arguments
25 */
26this.initialized = function (args) {
27};
28
29/*
30 * executed after refresh (single/multiple widgets) was started
31 * @param (args) wraps all function arguments
32 */
33this.refreshStarted = function (args) {
34};
35
36/*
37 * executed before widget results are loaded to the widget. allows single dashboard-level widget query result manipulation.
38 * @param (args) wraps all function arguments
39 */
40this.beforeWidgetResultLoaded = function (args) {
41
42};
43
44/*
45 * executed after refresh (single/multiple widgets) was ended
46 * @param (args) wraps all function arguments
47 */
48this.refreshEnded = function (args) {
49};
50

```

This script is editable, and changes you do can be saved by clicking the 'Save Changes' button on the upper right corner, or by hitting Ctrl+S on your keyboard.

The 'isCustom' property

The 'isCustom' property at the top of the script determines if the widget will be rendered or not. If you want to replace the widget entirely set 'isCustom' to false. This also affects the 'args' object which is explained in detail below.

Script Methods

There are 5 methods in the script you can choose as entry points for your code:

1. 'init' - Executes after the widget initializes, before query results are available. Useful when entirely overriding the rendering of the widget.
2. 'resized' - Executes after the widget is resized.
3. 'beforeResultLoaded' - Executes when the query results are available, before they are

loaded into the DOM. Allows canceling such behavior and replace it with custom widget implementation. To cancel the default population just set 'args.cancel = true' and the widget default population will be ignored.

4. 'afterResultLoaded' - Executes after the widget results are added to the DOM. Useful for applying DOM tweaks after the query has been completed and rendered.
5. 'handleDashboardInitialized' - Executes after all dashboard widget models are initialized. Useful for binding to other widget events.

The 'args' object

The 'args' object is available in each method to give you access to various objects and elements of the widget, and is different for each widget.

It contains the following properties:

- 'controller' – Represents the widget controller object.
- 'element' - The HTML element that contains the Widget in the DOM.
- 'result' – Contains the query result data behind the widget. (For button?)
- 'widget' – The Widget's object model.

Adding resources

You can use your custom Javascript code to access your own resource files, like images and other Javascript files. To do that, physically copy these files to your Prism Web installation folder.

Resource files will then be accessible by using a relative path in your code, i.e. '\myImage.png' or '\myResourceFolder\myImage.png'.

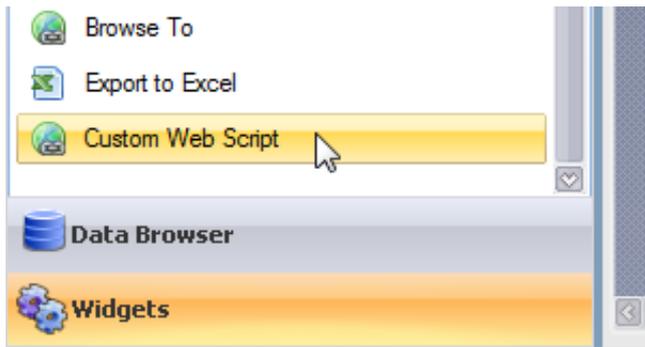
Dynamically loading local or remote Javascript files can be done as follows:

```
$.getScript('\abc.js', function(data, textStatus) {  
// your code which requires abc.js here });
```

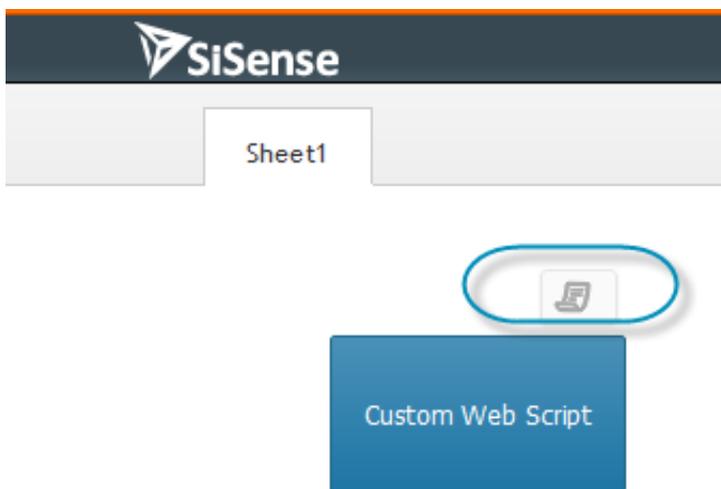
Running Javascript with the 'Custom Web Script' Button

Using the 'Custom Web Script' button you can create a button that executes your own custom Javascript code on click.

Start by adding a 'Custom Web Script' widget from the Widgets Explorer in BIStudio to your dashboard and publish it.



In Prism Web click the 'Customize Script' button to open the script editing window. Use the 'execute' method to add your own Javascript functionality. This code will be fired whenever the button is clicked.



Extending the Pivot Widget

Adding Functionality

If the 'isCustom' property at the top of the script remains 'false', the Pivot widget will be rendered to the DOM, but you can use the script methods to add various functionalities.

When 'isCustom = false' the args.result object contains the following properties:

- '\$stable' – The HTML table element that holds the Pivot results in the DOM.
- 'metadata' – An object that represents the metadata of the widget.

Using the args object and the supplied script methods you can manipulate the Pivot easily in various scenarios.

Accessing the Pivot Data

The Pivot's Custom Javascript Script enables you to access the pivot data directly, without accessing the DOM. In order for the data to be accessible through the script you need to set 'isCustom=true' at the top of the widget script. This will disable the default rendering of the Pivot and make the Pivot data accessible through the 'args.result' object.

When 'isCustom = true' the args.result object contains the following:

- Properties:
 - 'data' – A two dimensional array containing the Pivot data.
 - 'metadata' – An object that represents the metadata of the widget.
- Methods:
 - 'getCell (row, column)' – Gets the cell object for the given coordinate.
 - 'getCellValue (row, column)' – Gets the cell's actual value for the given coordinate.
 - 'getCellValues (column)' – Gets all cell values at the given column.
 - 'getCells (column)' – Gets all cells at the given column.
 - 'getFieldMetadata (r)' – Gets the metadata for the given index/name.
 - 'getRow (index)' – Gets a row by an index.
 - 'getRowCount' – Gets the row count.
 - 'getRows' - Gets all rows.

The 'cell' object that is returned by the various methods above and represents a single Pivot cell contains the following properties:

- 'data' – Contains the actual value behind the cell.
- 'value' – Contains the formatted string value that is presented to the user.

Using these objects and methods you can replace the Pivot with your own custom Javascript visualizations and widgets. See the 'Custom Widgets' section for examples.

Example – Adding Custom Images to the Pivot

This example shows how to add custom images into your Pivot. In this case we are inserting a green up pointing arrow image for positive values in the Pivot, and a red down pointing array for negative values.

Years	Category Name	Growth	sum Freight
1998	Beverages	-22.33 %	14,682
	Condiments	-40.76 %	6,524
	Confections	-45.00 %	9,789
	Dairy Products	-26.50 %	12,812
	Grains/Cereals	-27.89 %	6,615
	Meat/Poultry	-49.88 %	5,142
	Produce	-55.54 %	3,595
	Seafood	-34.30 %	9,283
1997	Beverages	199.97 %	18,903
	Condiments	335.61 %	11,014
	Confections	740.10 %	17,708

Page 1 of 3 Rows 1 - 11 of 24

To achieve the final result we did the following steps:

1. Put our images in a folder named 'arrows' in the Prism Web installation folder.
2. Edited the 'afterResultLoaded' method in the Pivot's script:

The script:

```

this.afterResultLoaded = function (args) {
// 1. Set options      var options = {
    imageColumn: 'Growth',
    replaceValues: false    };
// 2. Define function that decides which image to show for a given value
    var getImage = function (numericValue) {
        if (numericValue > 0)      return '/arrows/green.png';    };
// 3. Get the spans from the DOM that represent the cells of the column
    var spans = $($args.result.$table).find(
        '[field="' + options.imageColumn + '" ] span'),
        colText,    colValue,
        imgName;
// 4. Iterate over the spans and insert the image
    for (var i = 1; i < spans.length; i++) {
        if (options.replaceValues)    spans[i].replaceWith(imgDiv);
        else    spans[i].insertBefore(imgDiv);
    }
};

```

Extending the Chart Widget

The chart can be extended just like the Pivot. The script methods and args object are similar. The 'args.result' object in the chart contains the same methods as in the Pivot, and the following properties:

- 'dataset' – Contains the data behind the chart.
- 'metadata' - An object that represents the metadata of the widget.
- 'selection' – An array containing entries of the selected dimensions in the chart.

The underlying data of the chart can be accessed using the same cell methods as described in the Pivot section above. Also, each data series of the chart has an entry in the 'args.result.dataset.series' object.

Custom Widgets

Custom Widgets

You can use Javascript to create your own widgets and visualization, using your preferred 3rd party libraries or self-generated code.

To add a custom widget follow these steps:

1. Create a Pivot that holds the data you need for your widget.
2. Publish your dashboard with the Pivot. Position the Pivot at the location in which you want your new widget to appear.
3. Click the 'Customize Script' button of the Pivot.
4. Set the 'isCustom' property at the top of the script to 'true'.
5. Edit the 'beforeResultLoaded' method. Use the 'args' object and the various data retrieval methods described above to iterate over the pivot data and use it with your own Javascript visualization.

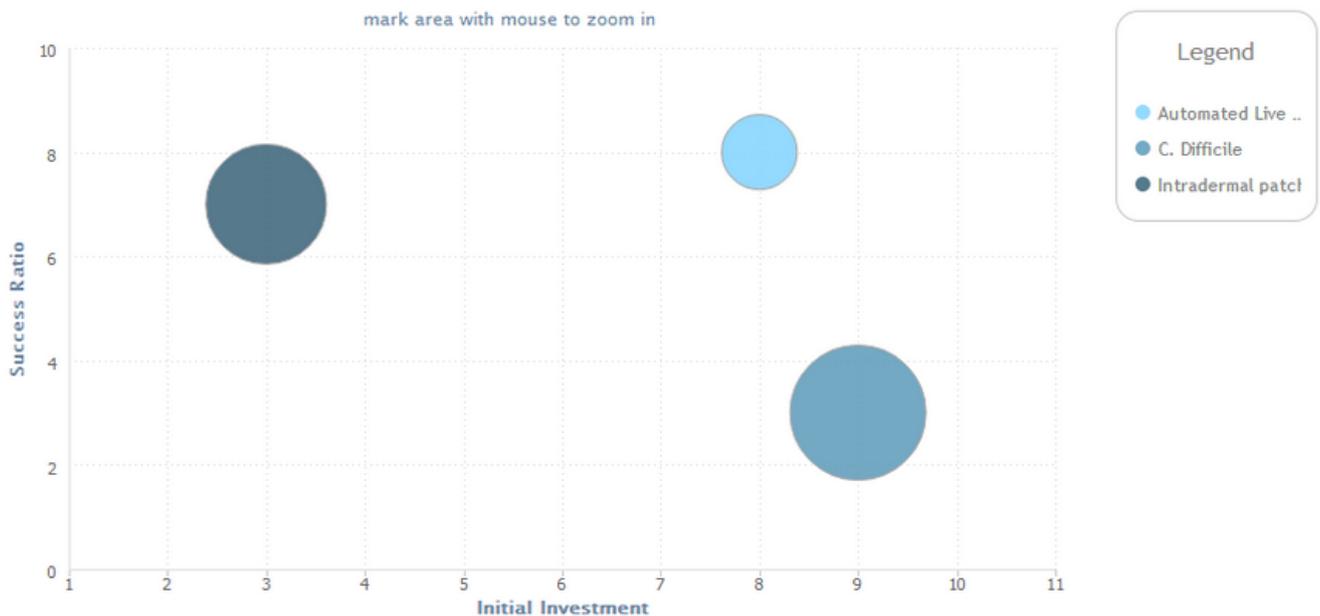
A good practice for easier reuse of your widgets is to create them in self-contained Javascript files and use them as a local resource. Then use the 'beforeResultLoaded' method to load and run them.

Custom Widget Example - Create a Scatter Chart

Introduction

This is a quick walk-through on how to turn your Pivot to a Scatter Chart using the built-in JavaScript extend-ability support.

This article only serves as an example of JavaScript extensions.



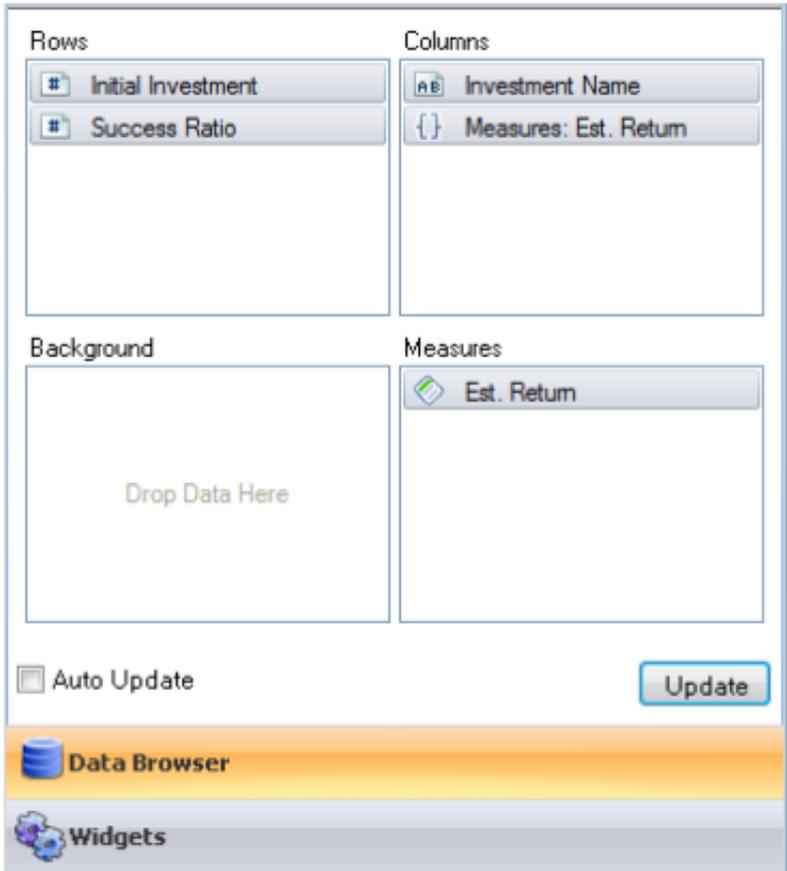
Step 1 – Get the Scatter Chart Extension

Download this [zip file](#) and unzip the ScatterChartExtension folder into your Prism Web installation folder.

The default location for this folder is C:\Program Files\Sisense\PrismWeb

Step 2 – Create the Pivot

Create a Pivot table in BI Studio. Add two rows, one measure and one column to it. For example:



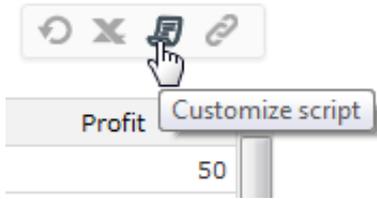
First publish your dashboard with the pivot table to Prism Web and open it in your browser. This is how the pivot looks like in the browser:

Initial Investment	Success Ratio	Intradermal patch	Automated Live Virus Platform Screening	C. Difficile
		Est. Return	Est. Return	Est. Return
3	7.69230769230769	80		
8	8.33333333333334		50	
9	3.33333333333333			90

Now the pivot is ready to be transformed into a scatter chart.

Step 3– Creating the Scatter Chart in Prism Web

Click the 'Customize Script' button at the top right of the pivot.



This will open a new window in which you can customize the widget with JavaScript.

Look for 'this.isCustom = false;' and replace it with 'this.isCustom = true;'.

Make the following changes in this window.

Copy and paste the following script to replace the init function, so it looks like

```
    this.init = function (args) {  
  
$.getScript('/scatterChartExtension/js/scatterChartExtension.js', function(data, textStatus)  
{  
SCATTER_CHART_EXTENSION.init();  
});  
};
```

Copy & Paste the following script to replace the beforeResultLoaded function, so it looks like

```
    this.beforeResultLoaded = function (args) {  
  
$.getScript('/scatterChartExtension/js/scatterChartExtension.js', function(data, textStatus)  
{ var indexMap = {x_axis:"Initial Investment", y_axis:"Success Ratio", size:"Est. Return",  
name:"Investment Name"}; SCATTER_CHART_EXTENSION.createScatterChart(args.result,  
args.element, indexMap); }); };
```

Note

Now click the save button on the upper right to save the script.
Go back to the Pivot and refresh the page.

Your Pivot is now a Scatter Chart. Hovering over bubbles shows their measured value in the tool tip.

Step 4– Configuring the Scatter Chart

It is possible to configure the Scatter Chart using a custom 'Options' object.

For example, let's say we want to give a title to our chart, make our bubbles bigger, and color them in reddish tones instead of blue.

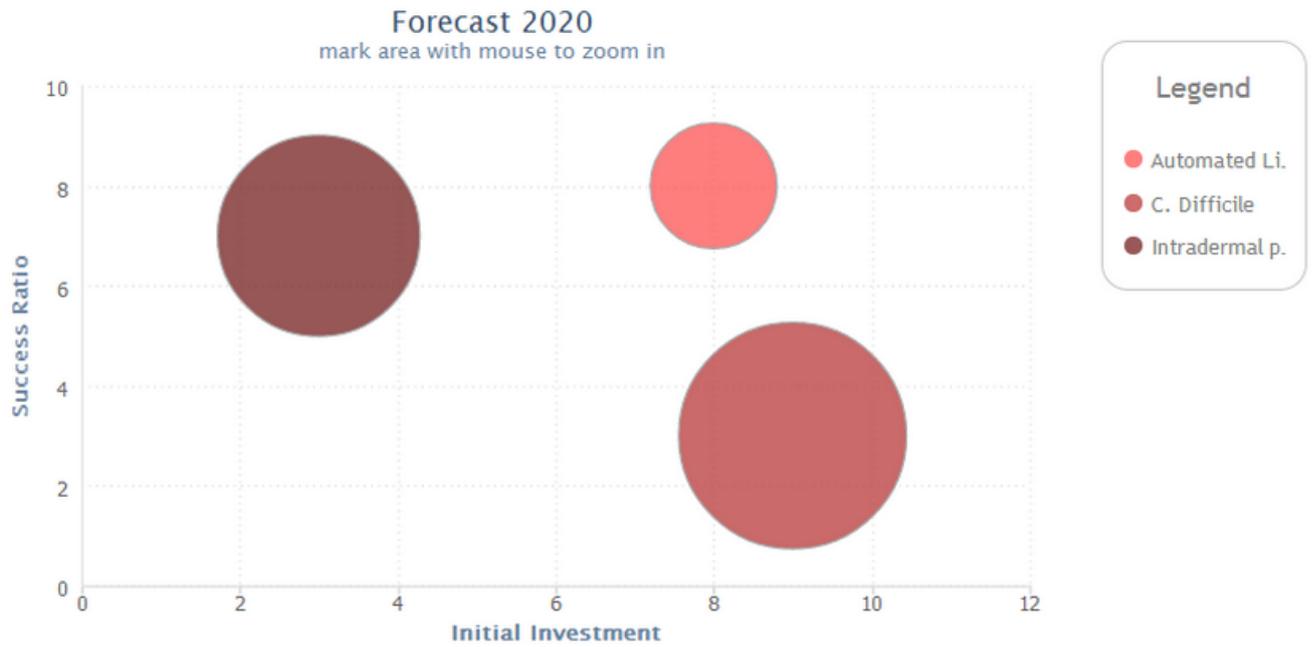
We need to create an options object and pass it as an additional parameter to the createScatterChart function in the following manner:

Copy and paste the following script to replace the beforeResultLoaded function, so it looks like

```
this.beforeResultLoaded = function (args) {  
  
$.getScript('/scatterChartExtension/js/scatterChartExtension.js', function(data, textStatus)  
  
{ var options = { title: { text: 'Forecast 2020' }, RADIUS_CONSTANT: 7, series: [ { color:  
'rgba(223, 83, 83, 0.8)' //transparent red } ]  
}; var indexMap = {x_axis:"Initial Investment", y_axis:"Success Ratio", size:"Est. Return",  
name:"Investment Name"}; SCATTER_CHART_EXTENSION.createScatterChart(args.result,  
args.element, indexMap, options); }); });
```

Note

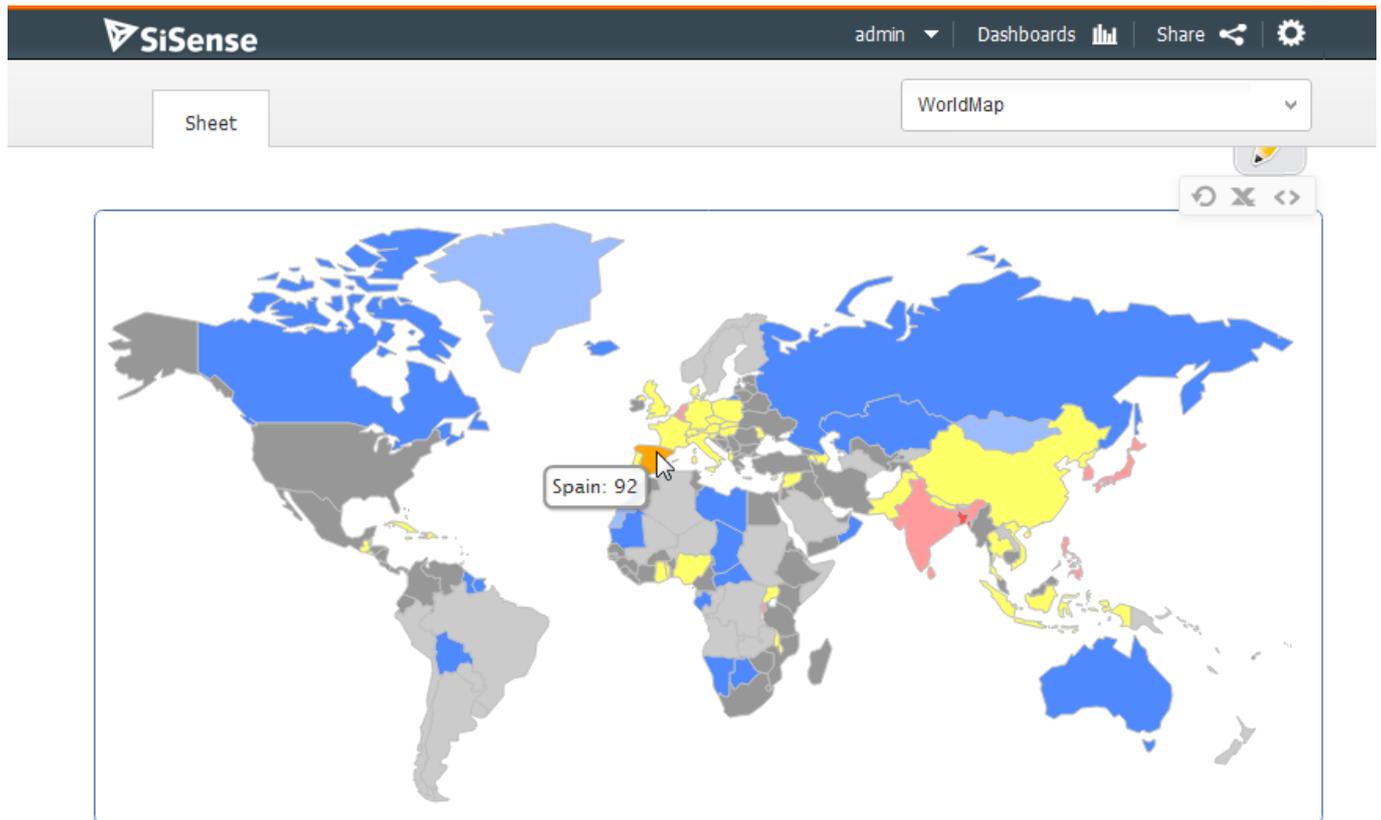
Our modified chart will now look like this:



Custom Widget Example - World Map Widget

Introduction

This example demonstrates how you can use a Pivot into a world map.



This is a quick walk-through on how to turn your Pivot to a World Map using the built-in Javascript extensibility support.

Step 1 - Get the World Map Extension

Download [mapcomponent.js](#) and place it in your Prism Web installation folder.

The default location for this folder is in 'C:\Program Files\Sisense\PrismWeb' or in older versions 'C:\Program Files (x86)\Sisense\PrismWeb'.

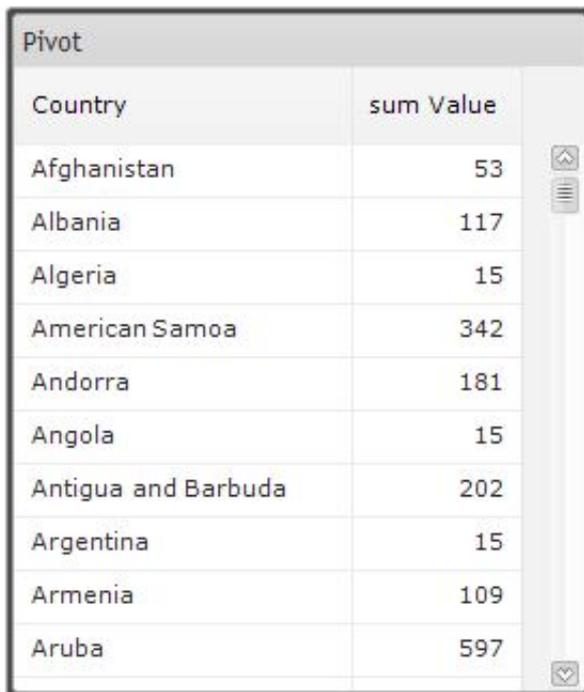
In order for the Map widget to work, you'll need a Pivot with country names. You can see the list of supported countries in the beginning of 'mapcomponent.js'. Only country names that are in this list will work. If you use different names or spelling for some countries, you need to edit 'mapcomponent.js' and replace the existing country names with the ones you're using. Do not

make any changes to the two letter country code each country is mapped to.

Step 2 - Create the Pivot

Create a Pivot table in BI Studio with a dimension of countries, and a measure you're interested in.

For example:



The image shows a screenshot of a Pivot table in BI Studio. The table has two columns: 'Country' and 'sum Value'. The data is as follows:

Country	sum Value
Afghanistan	53
Albania	117
Algeria	15
American Samoa	342
Andorra	181
Angola	15
Antigua and Barbuda	202
Argentina	15
Armenia	109
Aruba	597

Hide the title of the Pivot by disabling 'Show Title' in the Pivot's menu.

Country	sum Value
Afghanistan	53
Albania	117
Algeria	15
American Samoa	342
Andorra	181
Angola	15
Antigua and Barbuda	202
Argentina	15
Armenia	109
Aruba	597

Quick Shortcuts

Data

- Edit Data Hierarchies...
- Switch Axes
- Widget Interconnectivity ▶

Actions

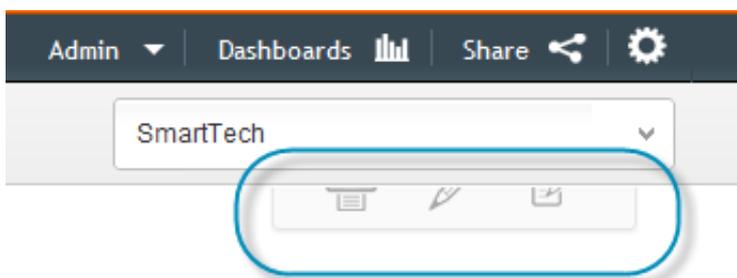
- Show the Data Browser... F4
- Create Linked Chart
- Copy Data-Behind
- Copy as Image
- Save to Repository
- Export to Excel ▶

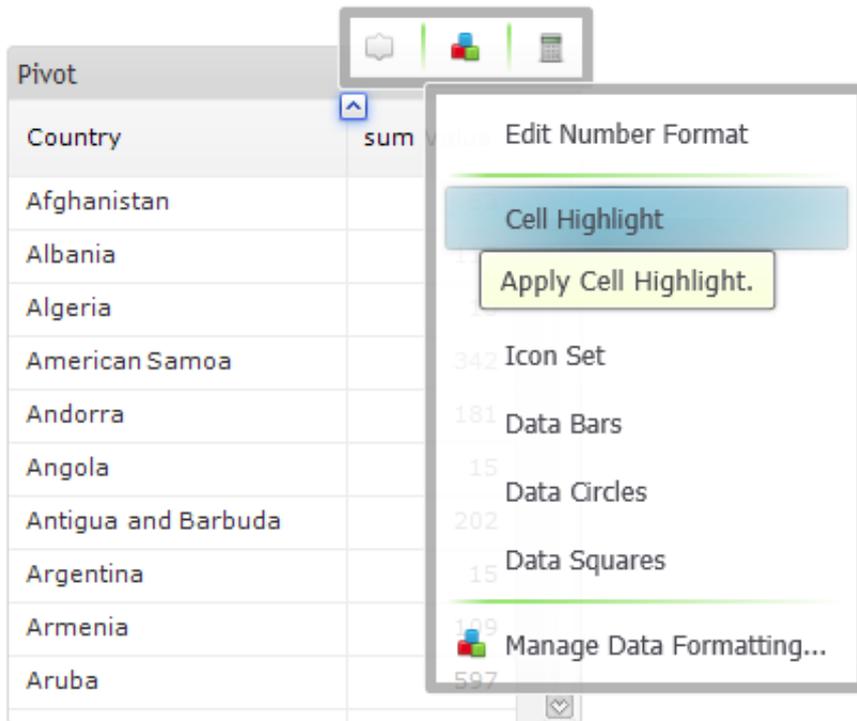
Widget Settings

- Show Title
- Widget Size ▶
- Manage Data Formatting...
- Fixed Column Width ▶

Now let's define colors for our measured values.

Hover over your measured column and from its menu choose "Cell Highlight".





In the window that is opened, assign colors to values as demonstrated in the following picture, according to your rules.

Add

Where

Enabled Default [Remove](#)

Where

Enabled Default [Remove](#)

Where

Enabled Default [Remove](#)

Where

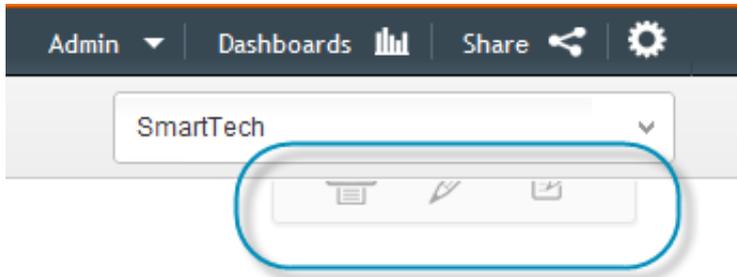
Enabled Default [Remove](#)

As a result your cells will now be colored.

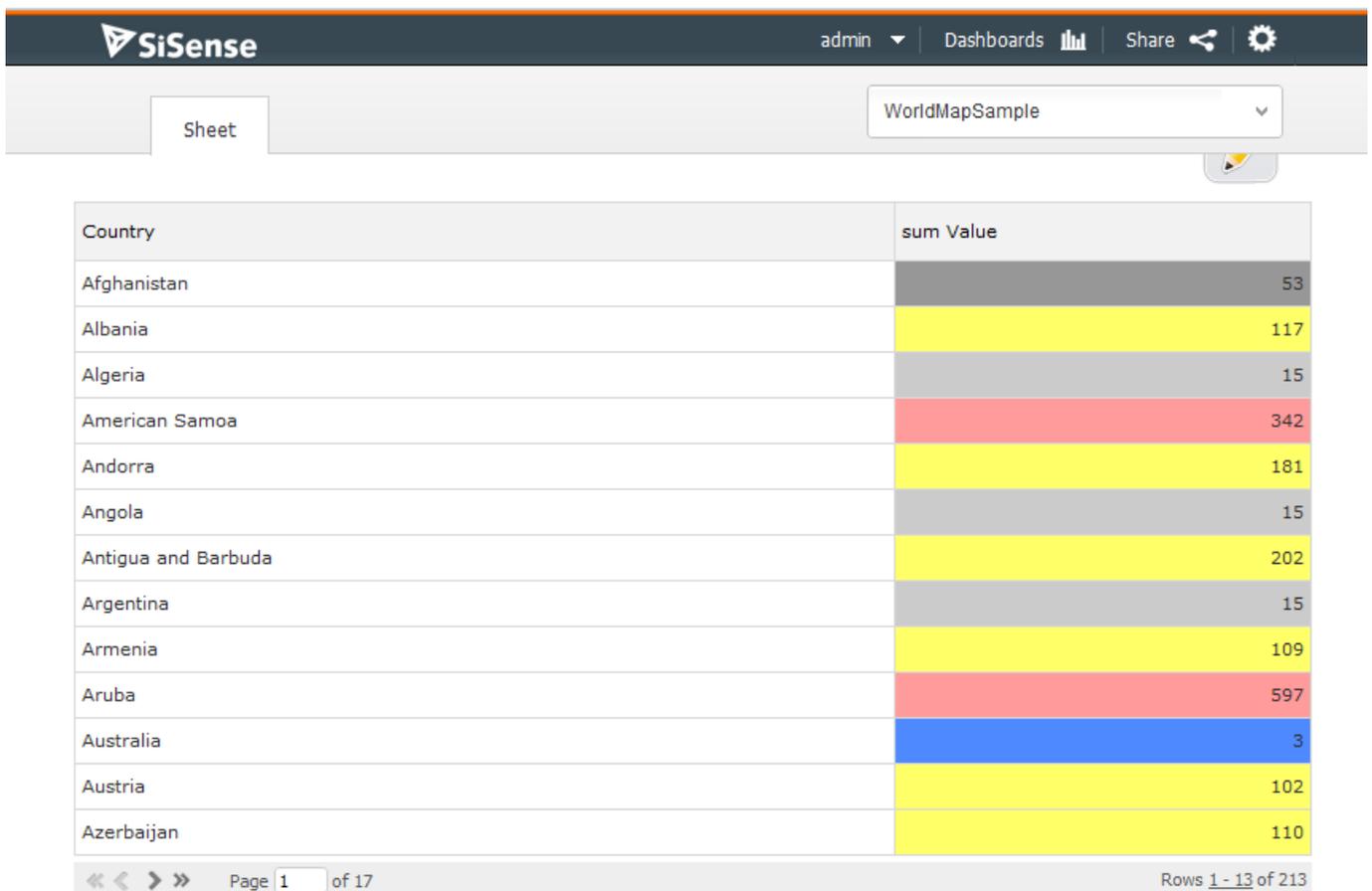
Country	sum Value
Afghanistan	53
Albania	117
Algeria	15
American Samoa	342
Andorra	181
Angola	15
Antigua and Barbuda	202
Argentina	15
Armenia	109
Aruba	597
Australia	3

Step 3 – Creating the Map in Prism Web

First publish your dashboard to Prism Web and open it in your browser. In Prism Web, click the 'Edit' button at the top right to position and re-size the pivot to the desired size of the map.



This is how our Pivot looks like after being re-sized:

A screenshot of the SiSense dashboard. The top navigation bar includes the SiSense logo, 'admin', 'Dashboards', 'Share', and a gear icon. Below the navigation bar, there is a 'Sheet' tab and a dropdown menu showing 'WorldMapSample'. The main content area displays a pivot table with the following data:

Country	sum Value
Afghanistan	53
Albania	117
Algeria	15
American Samoa	342
Andorra	181
Angola	15
Antigua and Barbuda	202
Argentina	15
Armenia	109
Aruba	597
Australia	3
Austria	102
Azerbaijan	110

At the bottom of the table, there is a pagination bar showing 'Page 1 of 17' and 'Rows 1 - 13 of 213'.

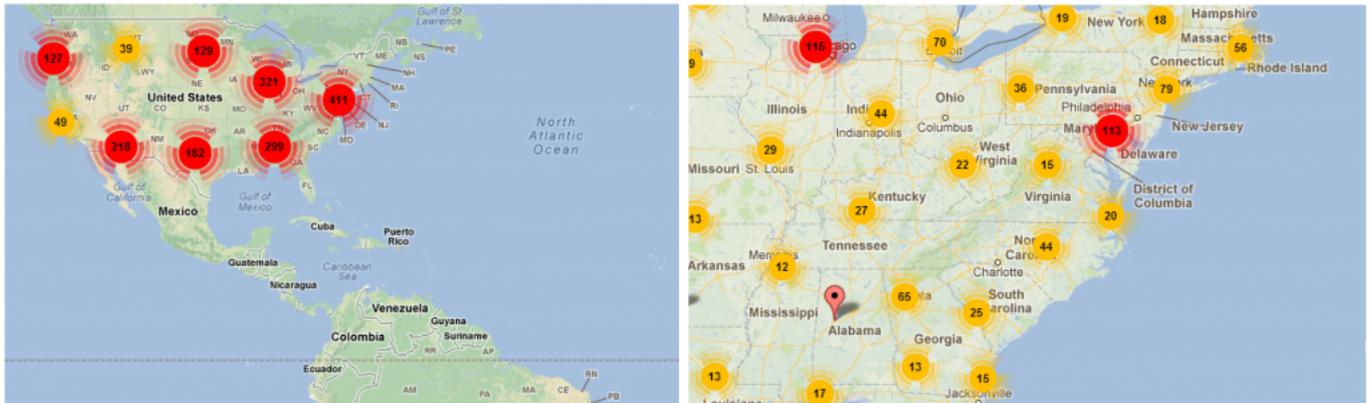
Now your pivot is ready to be transformed to a map.

Click the 'Customize Script' button at the top right of the pivot:

Custom Widget Example - Google Heat Map

Introduction

This example demonstrates how you can use a Pivot into a world heat map.



This is a quick walk-through on how to turn your Pivot to a Heat Map using the built-in Javascript extensibility support.

Step 1 - Get the World Map Extension

Download [m.js](#) and place it in your Prism Web installation folder.

The default location for this folder is in 'C:\Program Files\Sisense\PrismWeb'

or in older versions 'C:\Program Files (x86)\Sisense\PrismWeb'.

In order for the Map widget to work, you'll need a Pivot with city names, longitude, latitude and a measure. You can see an example of the pivot below

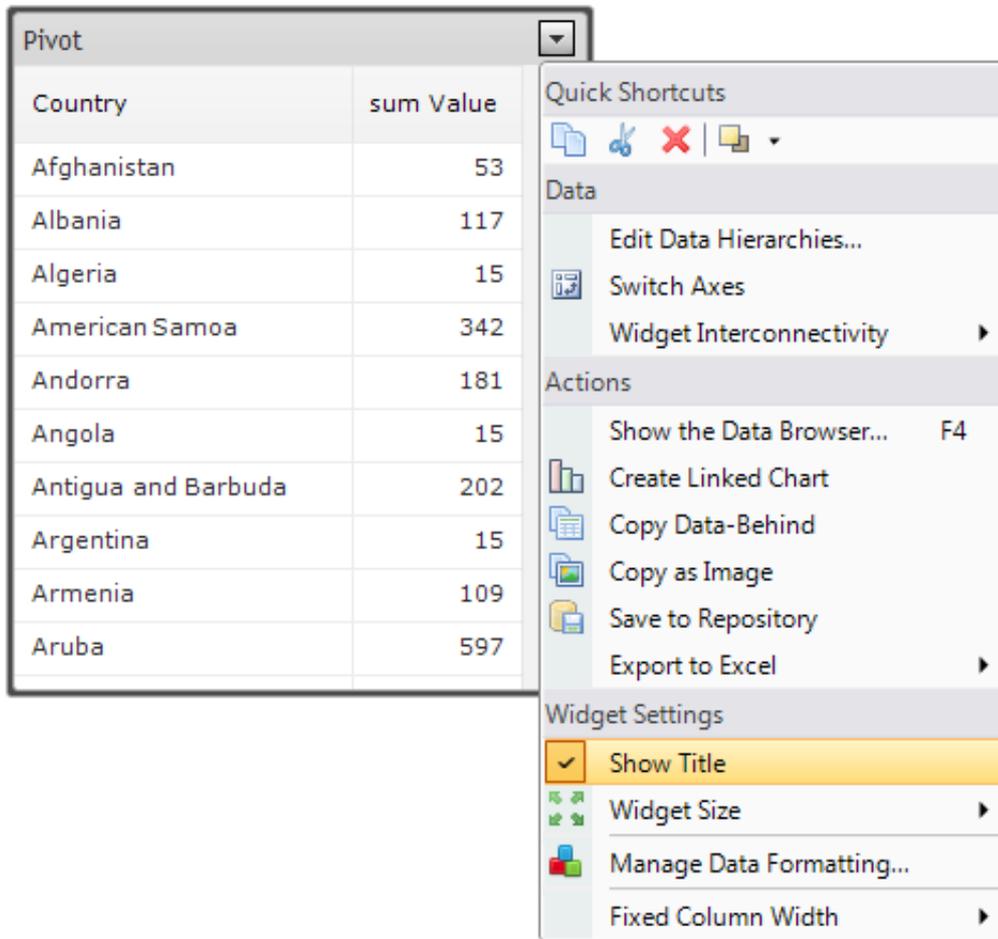
Step 2 - Create the Pivot

Create a Pivot table in BI Studio with a dimension of city the associated longitude and latitude and a measure you're interested in.

For example:

Location Name	Latitude	Longitude	0
Aberdeen	39.5213	-76.178	0
	45.4575	-98.44	0
Ablene	32.4023	-99.763	0
Abingdon	39.4603	-76.3167	0
Abington	42.1106	-70.9279	0
Abington Township	40.1246	-75.116	0
Acworth	34.0398	-84.6835	0
Addison	32.9514	-96.8537	0
Ahwatukee	33.3214	-111.981	0
Aiken	33.5044	-81.7112	0
Alabaster	33.2265	-86.8058	0
Alamo Heights	29.489462	-98.447581	0

Hide the title of the Pivot by disabling 'Show Title' in the Pivot's menu.



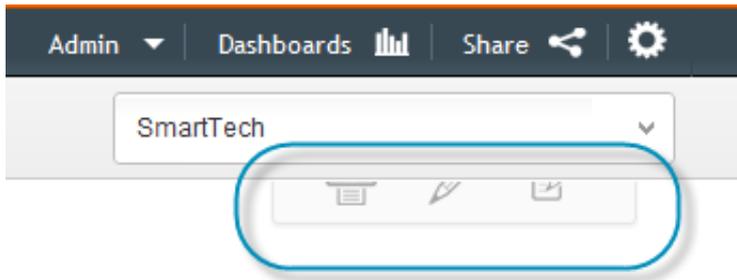
The image shows a Pivot widget with a dropdown menu open. The widget displays a table with two columns: 'Country' and 'sum Value'. The data rows are: Afghanistan (53), Albania (117), Algeria (15), American Samoa (342), Andorra (181), Angola (15), Antigua and Barbuda (202), Argentina (15), Armenia (109), and Aruba (597). The dropdown menu is divided into sections: 'Quick Shortcuts' (with icons for copy, paste, delete, and refresh), 'Data' (with options: Edit Data Hierarchies..., Switch Axes, and Widget Interconnectivity), 'Actions' (with options: Show the Data Browser... F4, Create Linked Chart, Copy Data-Behind, Copy as Image, Save to Repository, and Export to Excel), and 'Widget Settings' (with options: Show Title (checked), Widget Size, Manage Data Formatting..., and Fixed Column Width).

Country	sum Value
Afghanistan	53
Albania	117
Algeria	15
American Samoa	342
Andorra	181
Angola	15
Antigua and Barbuda	202
Argentina	15
Armenia	109
Aruba	597

Step 3 – Creating the Map in Prism Web

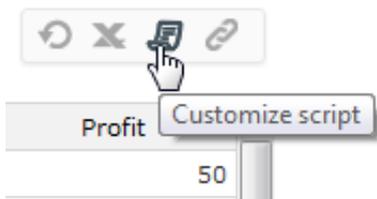
First publish your dashboard to Prism Web and open it in your browser.

In Prism Web, click the 'Edit' button at the top right to position and re-size the pivot to the desired size of the map.



Now your pivot is ready to be transformed to a map.

Click the 'Customize Script' button at the top right of the pivot:



Look for 'this.isCustom = false;' and replace it with 'this.isCustom = true;'.

Copy & Paste the following script to replace it so it looks like this:

```
// you can use the api below to request custom resource to be available in the the current context, all api calls will be deferred until all resources are loaded//
```

```
// require('/a.js', 'b.js', 'c.js');
```

```
// requireCSS('/styles.css');/*
```

```
* set to true when replacing widget rendering engine.
```

```
*/
```

```
this.isCustom = true;
```

```
/*
```

```
* Triggered after the widget was first initialized, useful when overriding entire widget rendering, i/e replacing the button with a whole different button ux
```

```
* @param {args} wraps all function arguments, including widget, dashboard and widget controller instances.
```

```

*/

this.init = function (args) {

};
/*

* Triggered after the widget was resized.

* @param {args} wraps all function arguments, including query result, widget, dashboard and
widget controller instances.

*/

this.resized = function(args) {

};
/*

* Triggered when the widget selection was changed

* @param {args} wraps all function arguments, including new selection, widget, dashboard and
widget controller instances.

*/

this.selectionChanged = function (args) {

};
/*

* Triggered before the widget query result is loaded to the DOM and allows canceling such
behavior and replace it with custom widget implementation.

* to cancel the default population just set 'args.cancel = true' and the widget default population
will be ignored.

* @param {args} wraps all function arguments, including query result, widget, dashboard and
widget controller instances.

*/

this.beforeResultLoaded = function (args) {

};
/*

```

* Triggered after the widget result were added to the DOM. useful to apply DOM tweaks after query result population.

* @param {args} wraps all function arguments, including query result, widget, dashboard and widget controller instances.

```
*/  
  
this.afterResultLoaded = function (args) {  
  
element = args.element;  
  
qresult = args.result;  
$.ajax({  
  
type: "GET",  
  
url: 'http://maps.googleapis.com/maps/api/js?v=3&sensor=true&key=AlzaSyDmYbMPyWlvxy-  
COhENlvLfyPUKeP86tPM&callback=onGoogleMapLoaded',  
  
dataType: "script",  
  
cache: true  
  
});  
  
}  
  
/*  
  
*  
  
*/  
  
window.onGoogleMapLoaded = function()  
  
{  
$.getScript('http://google-maps-utility-library-  
v3.googlecode.com/svn/trunk/markerclustererplus/src/markerclusterer.js', function(data,  
textStatus)  
  
{  
$.getScript('http://google-maps-utility-library-  
v3.googlecode.com/svn/trunk/infobox/src/infobox.js', function(data, textStatus)  
  
{
```

```

function formatWithCommas(x) {
var parts = x.toString().split(".");
return parts[0].replace(/\B(?=(\d{3})+(?=$))/g, ",") + (parts[1] ? "." + parts[1] : "");
};

function makeInfoWindowEvent(map, infowindow, contentString, marker) {
google.maps.event.addListener(marker, 'click', function() {
infowindow.setContent(contentString);
infowindow.open(map, marker);
});
}
// initialize map

var myLatLng = new google.maps.LatLng(37.9797,-100.843);

var myOptions = {
zoom: 3,
center: myLatLng,
mapTypeId: google.maps.MapTypeId.ROADMAP
}

var map = new google.maps.Map(element, myOptions);

var infowindow = new google.maps.InfoWindow();
// create markers for the map

var i = 0, l = qresult.data.length, j, m = qresult.md.byIndex.length;

var markers = [], curRes = [], markerText;
for (; i

markerText = "+ qresult.data\[i\]\[0\]\[\"text\"\] + '";

```

```

j = 3;

for (; j

markerText += '
' + qresult.md.byIndex[j]["title"] + ': : ' + qresult.data[i][j]["text"] + '
';

}

markerText += ";

var marker = new google.maps.Marker({

map: map,

position: new google.maps.LatLng(qresult.data[i][1]["data"], qresult.data[i][2]["data"]),

raiseOnDrag: false,

visible: true,

draggable: false,

title: qresult.data[i][3]["text"]

});

makeInfoWindowEvent(map, infowindow, markerText, marker);
markers.push(marker);

}
// creates the clusters markers

var markerCluster = new MarkerClusterer(map, markers,{ averageCenter: true });
var infowindow = new InfoBox({boxStyle: { background: '#ffffff', border: "2px dashed blue",
padding: '4px',textAlign: "center", fontSize: "8pt" }, closeBoxURL: "" });
google.maps.event.addListener(markerCluster, 'mouseover', function(cluster) {

var sum = 0;

_.each(cluster["markers_"], function (item) {

sum += parseInt((item["title"]).replace(", ", ""));

});

```

```
infowindow.setContent('sum: ' +formatWithCommas(sum));

infowindow.open(map);

infowindow.setPosition(cluster.getCenter());

});

google.maps.event.addListener(markerCluster, 'mouseout', function(cluster) {

infowindow.close();

});

google.maps.event.addListener(markerCluster, 'click', function(cluster) {

infowindow.close();

});
});
});
}
```

Now Push the save button on the upper right to save the script.

Go back to the Pivot and refresh the page.

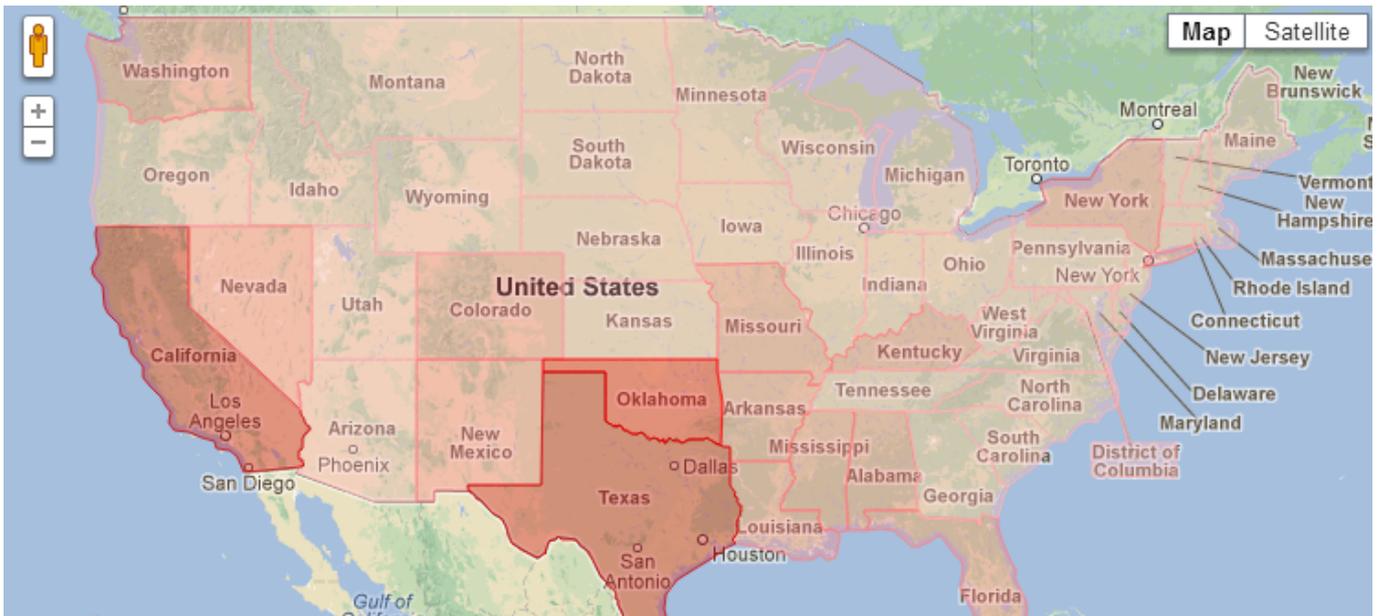
Congratulations! Your Pivot is now a map.

Double clicking on the segments show a narrower segment of data on the map.

Custom Widget Example - States Map

Introduction

This example demonstrates how you can change a Pivot into a heat map of states in the US.



This is a quick walk-through on how to turn your Pivot to a heat map of states in the US using the built-in Javascript extend-ability support.

Before Starting

Unzip and install the two files below in the Prism Web folder located in 'Program Files\Sisense\PrismWeb'.

- [US-States XML](#)
- [Map Component](#)

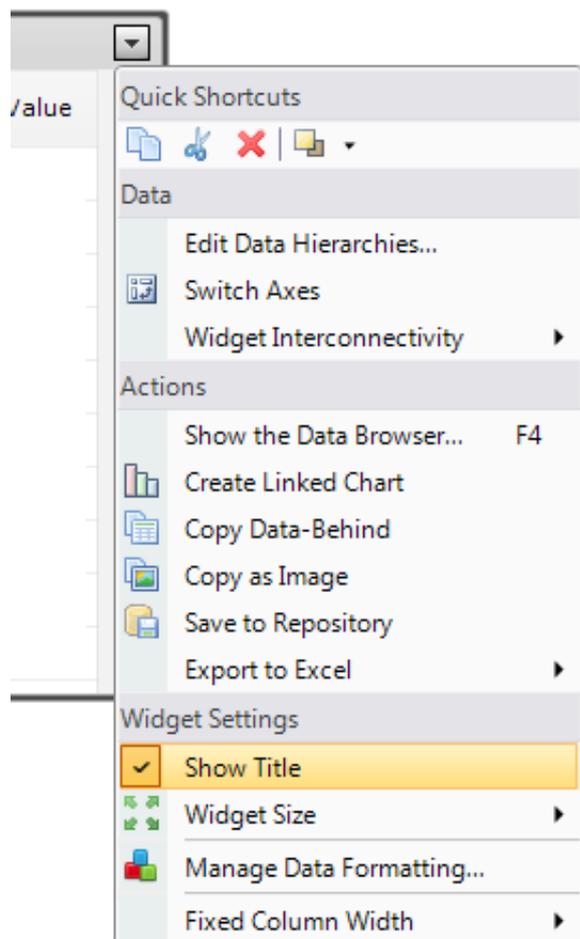
Step 1 - Create the Pivot

Create a Pivot table in BI Studio with a dimension of states, and a measure you're interested in. Note, ensure there are no spaces before or after the state names.

For example:

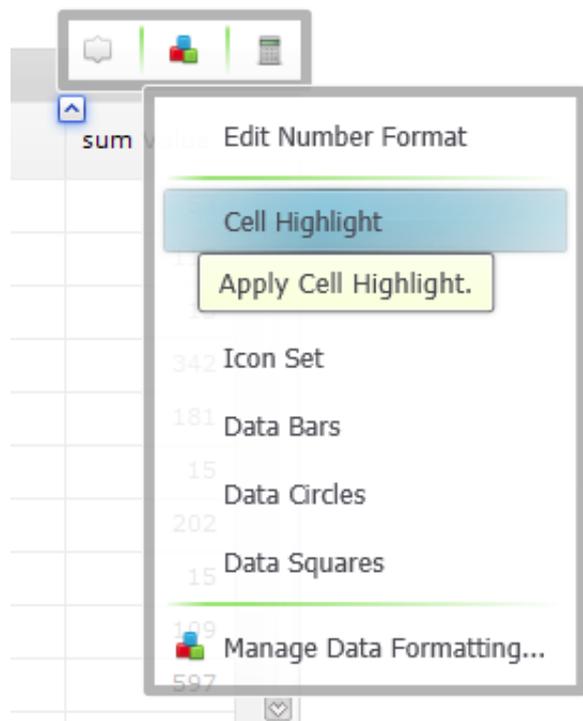
State or Territory	count Number
Alabama	77
Alaska	46
American Samoa	11
Americas	0
Arizona	56
Arkansas	62
California	208
Colorado	67
Connecticut	26
Delaware	18

Hide the title of the Pivot by disabling 'Show Title' in the Pivot's menu.



Now let's define colors for our measured values.

Hover over your measured column and from its menu choose "Cell Highlight".



In the window that is opened, assign colors to values as demonstrated in the following picture, according to your rules.

Add

Where [sum Value] < 3
 Enabled Default Remove

Where [sum Value] < 10
 Enabled Default Remove

Where [sum Value] < 30
 Enabled Default Remove

Where [sum Value] < 100
 Enabled Default Remove

OK Cancel

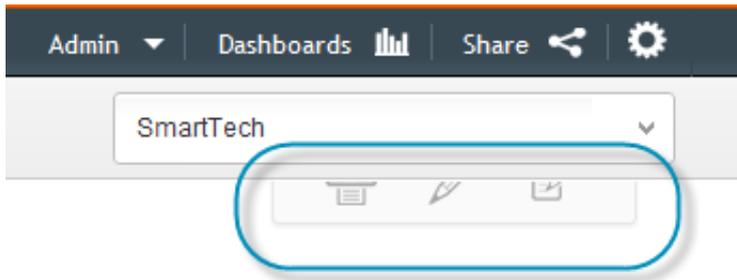
As a result your cells will now be colored.

State or Territory	count	Number
Alabama		77
Alaska		46
American Samoa		11
Americas		0
Arizona		56
Arkansas		62
California		203
Colorado		67
Connecticut		26
Delaware		18

Step 3 – Creating the Map in Prism Web

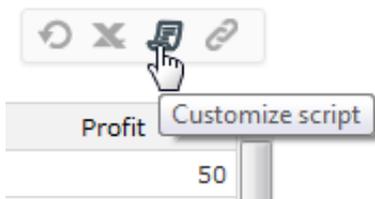
First publish your dashboard to Prism Web and open it in your browser.

In Prism Web, click the 'Edit' button at the top right to position and re-size the pivot to the desired size of the map.



Now your pivot is ready to be transformed to a map.

Click the 'Customize Script' button at the top right of the pivot:



Copy & Paste the following script to **replace all the existing script** so it looks like this:

```
/** to use online script you can use jquery getScript plugin as follows:** $.getScript('/abc.js',  
function(data, textStatus)
```

```
* {  
  
* // your code which requires abc.js here  
  
* });  
  
*  
  
* // more info at: http://api.jquery.com/jQuery.getScript/  
  
*/  
/*  
  
* set to true when replacing widget rendering engine.  
  
*/  
  
this.isCustom = true;
```

```
var cData,cContanier;
```

```
var mapLoaded = false;
```

```
var rowcount;
```

```
var cLong;
```

```
var cLat;
```

```
var me;
```

```
var hashNames = new Object();
```

```
var poly;
```

```
/*  
* exected after the widget controller was initialized, useful when overriding entire widget in  
order to instantiate entirely different implementation.
```

```
*  
* @param {args} wraps all function arguments, including widget, dashboard and widget  
controller instances.
```

```
*
```

```
*/
```

```
this.init = function (args)
```

```
{
```

```
me = this;
```

```
}
```

```
/*
```

```
* executed before the widget query result is loaded to the DOM and allows canceling such  
beavior and replace it with custom widget implementation.
```

```
* to cancel the default population just set 'args.cancel = true' and the widget default population  
will be ignored.
```

```
*
```

* @param {args} wraps all function arguments, including query result, widget, dashboard and widget controller instances.

*

*/

```
this.beforeResultLoaded = function (args)
```

```
{
```

```
}
```

```
/*
```

* executed after the widget result were added to the DOM. useful to apply DOM tweaks after query result population.

*

* @param {args} wraps all function arguments, including query result, widget, dashboard and widget controller instances.

*

*/

```
this.afterResultLoaded = function (args)
```

```
{
```

```
cContanier = args.element;
```

```
cData = args.result.data;
```

```
rowcount = args.result.data.length;
```

```
hashNames = new Object();
```

```
$.ajax({
```

```
type: "GET",
```

```
url: 'http://maps.googleapis.com/maps/api/js?sensor=true&key=AlzaSyDmYbMPyWlvxy-COhENlvLfyPUKeP86tPM&callback=loadMaps',
```

```
dataType: "script",
```

```
cache: true
```

```

});
//var script = document.createElement("script");

//script.src = "http://maps.googleapis.com/maps/api/js?sensor=true&key=ABQIAAAAhOMilxP5
hMji3MEPbkSBBTfxlvSK0b4z-
FtBdvaPKlPltTKwBShLLVhDkkV1gLCzJKBm5HluA_q9Q&callback=loadMaps";

//script.type = "text/javascript";

//document.getElementsByTagName("head")[0].appendChild(script);
}
window.loadMaps = function()

{

var myLatLng = new google.maps.LatLng(38,-96);

var myOptions = {

zoom: 4,

center: myLatLng,

mapTypeId: google.maps.MapTypeId.ROADMAP

}

var map = new google.maps.Map(cContanier, myOptions);

var infowindow = new google.maps.InfoWindow();
//download this file to your server first http://data.sisense.com/Temp/Countries borders.xml

$.ajax({

type: "GET",

url: "/us-states.xml",

dataType: "xml",
success: function(xml) {

for(var i = 0; i

{

$(xml).find('state').each(function()

```

```
{  
  
var name = $(this).attr('name');  
  
debugger  
  
if(name === cData[i][0].trim())  
  
{  
  
var n = 0;  
  
var countryCoords = [];  
  
$(this).find('point').each(function()  
  
{  
  
var lat = $(this).attr('lat');  
  
var lng = $(this).attr('lng');  
  
countryCoords[n++] = new google.maps.LatLng(lat,lng);  
  
});  
  
var polyx = new google.maps.Polygon({  
  
paths: countryCoords,  
  
strokeColor: cData[i][1].style['background-color'],  
  
strokeOpacity: 0.8,  
  
strokeWeight: 1,  
  
fillColor: cData[i][1].style['background-color'],  
  
fillOpacity: 0.35  
  
});  
  
polyx.setMap(map);  
  
var obj = {
```

```

'name':cData[i][0],
'value':cData[i][1].data.toString()
};

polyx.objInfo = obj;

google.maps.event.addListener(polyx, 'click', function(event)
{
var celVal = polyx.objInfo['value'].substring(0,4);

infowindow.setContent(this.objInfo['name'] + ' ' + this.objInfo['value'] + ' Disasters');

infowindow.setPosition(event.latLng);

infowindow.open(this.map);

});

}

});

}

});

}

loaded = function()

{

me.showMap();

}

this.showMap = function()

{
}
}
/*

```

* executed after all dashboard widget models were initialized. useful for binding with other widget events.

*

* @param {args} wraps all function arguments, including widget, dashboard and widget controller instances.

*

*/

```
this.handleDashboardInitialized = function (args)
```

```
{
```

```
}
```

Now Push the save button on the upper right to save the script.

Go back to the Pivot and refresh the page.

Congratulations! Your Pivot is now a heat map of states.

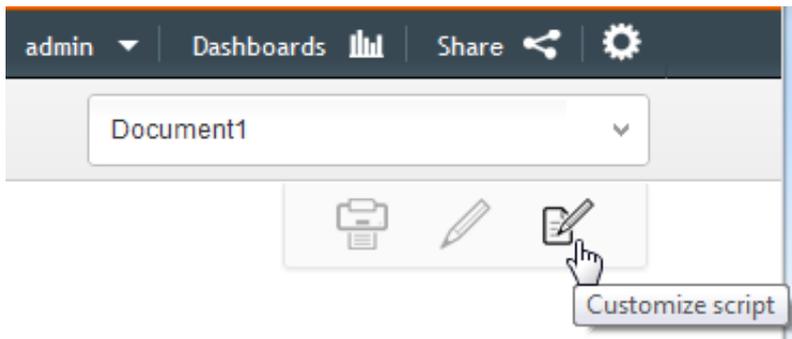
Hovering over states shows their measured value in the tool tip.

Extending the Dashboard

Extending the Dashboard

Introduction

Prism Web's Javascript extendibility feature enables you to run your own Javascript code inside your dashboard. To access the Dashboard script click the 'Customize script' button at the upper right corner of the dashboard.



Note

The dashboard script will be opened in a new window:

```

1/*
2 * to use online script you can use jquery getScript plugin as follows:
3 *
4 * $.getScript('http://abc.com/abc.js', function(data, textStatus)
5 * {
6 *     // your code which requires abc.js here
7 * });
8 *
9 * // more info at: http://api.jquery.com/jquery.getScript/
10 *
11 */
12
13
14/*
15 * executed after the dashboard was first initialized, useful when applying custom behavior to the dashboard
16 * @param (args) wraps all function arguments
17 */
18this.initializing = function (args) {
19
20};
21
22/*
23 * executed after all dashboard widget models were initialized. useful for binding with other widget events.
24 * @param (args) wraps all function arguments
25 */
26this.initialized = function (args) {
27};
28
29/*
30 * executed after refresh (single/multiple widgets) was started
31 * @param (args) wraps all function arguments
32 */
33this.refreshStarted = function (args) {
34};
35
36/*
37 * executed before widget results are loaded to the widget. allows single dashboard-level widget query result manipulation.
38 * @param (args) wraps all function arguments
39 */
40this.beforeWidgetResultLoaded = function (args) {
41
42};
43
44/*
45 * executed after refresh (single/multiple widgets) was ended
46 * @param (args) wraps all function arguments
47 */
48this.refreshEnded = function (args) {
49};
50

```

This script is editable, and changes you do can be saved by clicking the 'Save Changes' button on the upper right corner, or by hitting Ctrl+S on your keyboard.

Script Methods

There are 5 methods in the script you can choose as entry points for your code:

1. 'initializing' - Executes on dashboard initialization. Useful when applying custom behavior to the dashboard.
2. 'initialized' - Executes after all dashboard widget models are initialized. Useful for binding to other widget events.
3. 'refreshStarted' - Executes after a refresh (single/multiple widgets) has started.
4. 'beforeWidgetResultLoaded' - Executes before widget results are loaded into the widget. Allows single dashboard-level widget query result manipulation.
5. 'refreshEnded' - Executes after a refresh (single/multiple widgets) has ended.

Align Dashboard to the Center of the Browser

EXAMPLE - ALIGNING THE DASHBOARD IN THE CENTER OF THE BROWSER

The script below aligns the dashboard contents so that it appears in the center of the browser as opposed to the default align left.

Copy and past the script by clicking the 'Customize script' button at the upper right corner of the dashboard.

```
/*      A private function to center the dashboard      */  var _centerD
ashboard = function() {      // Preparing      var docks = $(''.R
adDock'),      dock,      len = docks.length,
      i = 0, l,      left = 9999,      right = -9999;

// Collecting left/right most positions
for (; i dock = $(docks[i]);
l = dock.position().left; dock.attr('oleft', l);
left = Math.min(l, left);
right = Math.max(l + dock.width() , right);
}

// Calculating new shift
var dashboardwidth = right-left,
leftpad = ($(document).width() - dashboardwidth) / 2,
i = 0, l = docks.length;

if (leftpad

// Do the Shift
for (; i dock = $(docks[i]);
dock.css('left', leftpad + parseInt(dock.attr('oleft')) - left + 'px');
}
};

// replace the empty initialized function with this one
this.initialized = function (args) {

_centerDashboard();
```

};

Automatically Refresh a Dashboard

EXAMPLE - AUTOMATICALLY REFRESH A DASHBOARD

Overview

In certain case a dashboard in Prism Web may need to be refreshed on a specified time frame. For example if data is being updated on a frequent basis an automatic refresh will reflect these data changes in the web dashboard.

The line below, needs to be added to the refreshEnded function of the dashboard script. Click the 'Customize script' button at the upper right corner of the dashboard and add the script below.

The code will refresh the dashboard 5 seconds after the previous refresh has ended.

```
setTimeout(function() { args.dashboard.refresh(); }, 5000);
```

Widget Refresh

A single widget can also be refreshed on a specified time period.

Add the line below to the afterResultLoaded

```
setTimeout(function() { prism.WidgetClient.refresh(args.widget); }, 5000);
```

Add a Welcome Screen with your Company Logo

Example – Adding a Welcome Screen with your Company Logo

In this example we'll use the dashboard script to create a welcome screen with a company logo and custom text when the dashboard loads. Clicking on it will remove it.

To do this, open the dashboard script page, and replace the 'initializing' method with the following Javascript code:

```
this.initializing = function (args) {  
  // Create an overlay div      var $overlayDiv = $(''  
  ');      $overlayDiv.css( {          "position": "absolute",          "  
  background-color": "black",          "opacity": "0.8",          "left": "  
  0px",          "top": "0px",          "width": "100%",          "height": "  
  100%",          "text-align": "center",          "font-size": "26px",  
          "z-index": "1000"      });  
  // Create a div for the company logo & text
```

```
  var $innerOverlayDiv = $(''
```

Your custom text goes here!

Click to continue

```
  ');
```

```
$innerOverlayDiv.css( {  
  "margin": "100px auto",  
  "text-align": "center"  
});  
$overlayDiv.append($innerOverlayDiv);  
// Add the overlay div to the body  
$(document.body).prepend($overlayDiv);  
// Bind an event to remove it on click  
$(document.body).one('click', function () {
```

```
$('#mainOverlay').remove();  
}  
};
```



After saving the script, the next time you refresh your dashboard you will see the new welcome screen:

The screenshot shows the SiSense dashboard interface. At the top, there is a navigation bar with the SiSense logo, an 'Admin' dropdown, 'Dashboards' with a bar chart icon, 'Share' with a share icon, and a settings gear icon. Below the navigation bar, there is a 'Sheet' tab and a dropdown menu showing 'DashboardScriptExample'. The main content area features a large, semi-transparent dark overlay with the following text: 'Dashboard Script Example', 'Click the 'Customize Script' button on the upper right to see the', 'Your custom text goes here!', and 'Click to continue'. A large, colorful 3D pyramid graphic is overlaid on the dashboard. In the background, several data visualizations are visible: a table with columns 'Years', 'Category Name', 'Growth', and 'sum Freight'; a bar chart titled 'Yearly avg. Unit Price'; and a donut chart titled 'Top 4 Categories by Freight' showing segments for 'Seafood' (20%) and 'Beverages' (29%).

Years	Category Name	Growth	sum Freight
1998	Beverages	-22.33 %	14,682
	Condiments	-40.76 %	6,524
	Confections	-45.00 %	9,789
	Dairy Products	-26.50 %	12,812
1997	Beverages	-27.89 %	6,615
	Condiments	-29.88 %	5,142
	Confections	-55.54 %	3,595
	Dairy Products	-34.00 %	7,283

Adjusting Column Width in a Pivot

Overview

In certain cases a dashboard in Prism Web may need to adjust the column width in a pivot table

The line below, needs to be added to the `afterResultLoaded` function in the pivot script. The code runs on all the columns from column `i`, you'll need to change the "var i" variable to set the starting column. The 'width' syntax refers to the adjusted size of the column eg. `css('width', '50px')`.

```
this.afterResultLoaded = function (args) {  
  
var headercells = $('thead td', args.element);  
var i = 3, l = headercells.length;  
  
for (; i  
  
headercells.eq(i).css('width', '50px');  
}
```

Prism Studio Model

Introduction

Prism Studio Model is a dashboard API for .NET applications. With it you can write your own applications that programmatically connect to your dashboards. Using the API you can retrieve widget data, change selections, manipulate your dashboard, and even export it to a pdf file.



Prerequisites for your .NET project:

1. Add a reference to 'Prism.StudioModel.dll' in your project.
2. Your .Net application's target framework has to be set to '.NET Framework 4'.
3. Your application must be configured to use a single threaded apartment (STA). You can do this by adding the [STAThread] attribute to your program's entry point.
4. You must copy all the dlls from Prism's installation folder into your application's runtime folder.

You can continue and read about the API, or go straight to the example.

The API

The 'PSM' class

This is a static class that contains only one method:

- Open – Connects to dashboard (psm file) and returns a 'PrismDocument' object.

Here's an example for how to connect to a psm file:

```
PrismDocument doc = PSM.Open(@"C:\MyDashboard.psm");
```

The 'PrismDocument' Class

Represents a Prism dashboard document – a .psm file.

Properties:

- Pages – An enumerable collection of 'PrismPage' objects.

Pre & Post Build Plugins

Introduction

Plugins enable you to run your own .NET code at the beginning and/or end of an ElastiCube build. This gives you the flexibility to enhance Prism with your own functionalities.

Examples of use cases:

1. Copy csv files from a remote location to your local machine before running an ElastiCube build.
2. Manipulate and rename csv or Excel files before importing them into the ElastiCube.
3. Send emails after a build finishes
4. Refresh your dashboard and export to pdf using the Studio Model after each build.

Creating a Plugin

Plugins can be written in any .NET language.

Setting up your .NET project:

1. Add a reference to 'ElastiCube.PluginBase.dll' in your project.
2. Add a reference to the .NET dll 'System.ComponentModel.Composition' in your project.
3. Add an 'ElastiCubePart' attribute to your main Class:

```
[ElastiCubePart("MyPlugin", ExecutionTypes.ExecuteAlways, false)]
```

The first parameter is the plugin name as it will be used in the ElastiCube Manager. The second parameter determines if the build should fail if the plugin fails.

4. Your main class must implement the 'IElastiCubePart' interface (referenced in 'ElastiCube.PluginBase.dll').
5. Your final plugin dll must be copied into the 'plugins' folder in the Prism installation directory. If the 'plugins' directory doesn't exist there, create it. The default location of the folder is: ['C:\Program Files\Sisense\Prism\plugins'](#)

Plugin example

```
using System; using System.Collections.Generic; using System.Linq; using System.Text; using ElastiCube.PluginBase; namespace BuildPluginSample { [ElastiCubePart("MyPlugin", ExecutionTypes.ExecuteAlways, false)] public class PluginClass:
```

```

IElastiCubePart      {

    #region IElastiCubePart Implementation
        public IElastiCubePartResult Run(
IElastiCubePartArgs args)      {
// 0. Prepare the result object
        Elast
iCubePartResult eCubePartresult = new
    ElastiCubePartResult();
// 1. Get ElastiCube name          string ecubeName = args.
ElastiCubeXml.Elements().First().Attribute("ID").Value;

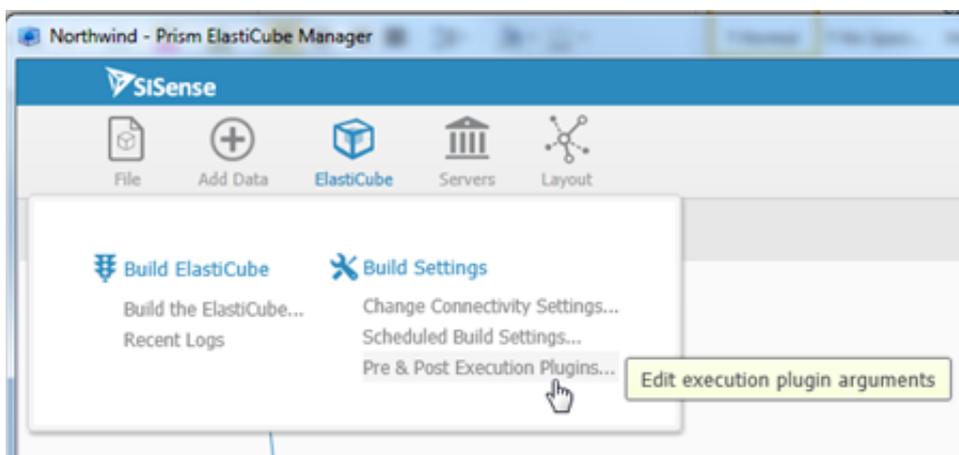
// 2. Extract plugin arguments          int myInt = Int32.
Parse(args.Args.Attribute("MyInt").Value);          if (
myInt == -1)          {
// Setting IsTerminatingProcess = true terminates
the build after the plugin execu
tion.

        eCubePartresult.
IsTerminatingProcess = true;          }
        // 3. Log operation
        args.EventLogger.WriteLog("myInt: " + myInt, "
MyPlugin on ecube " + ecubeName);          return
    eCubePartresult;          }          #endregion      } }

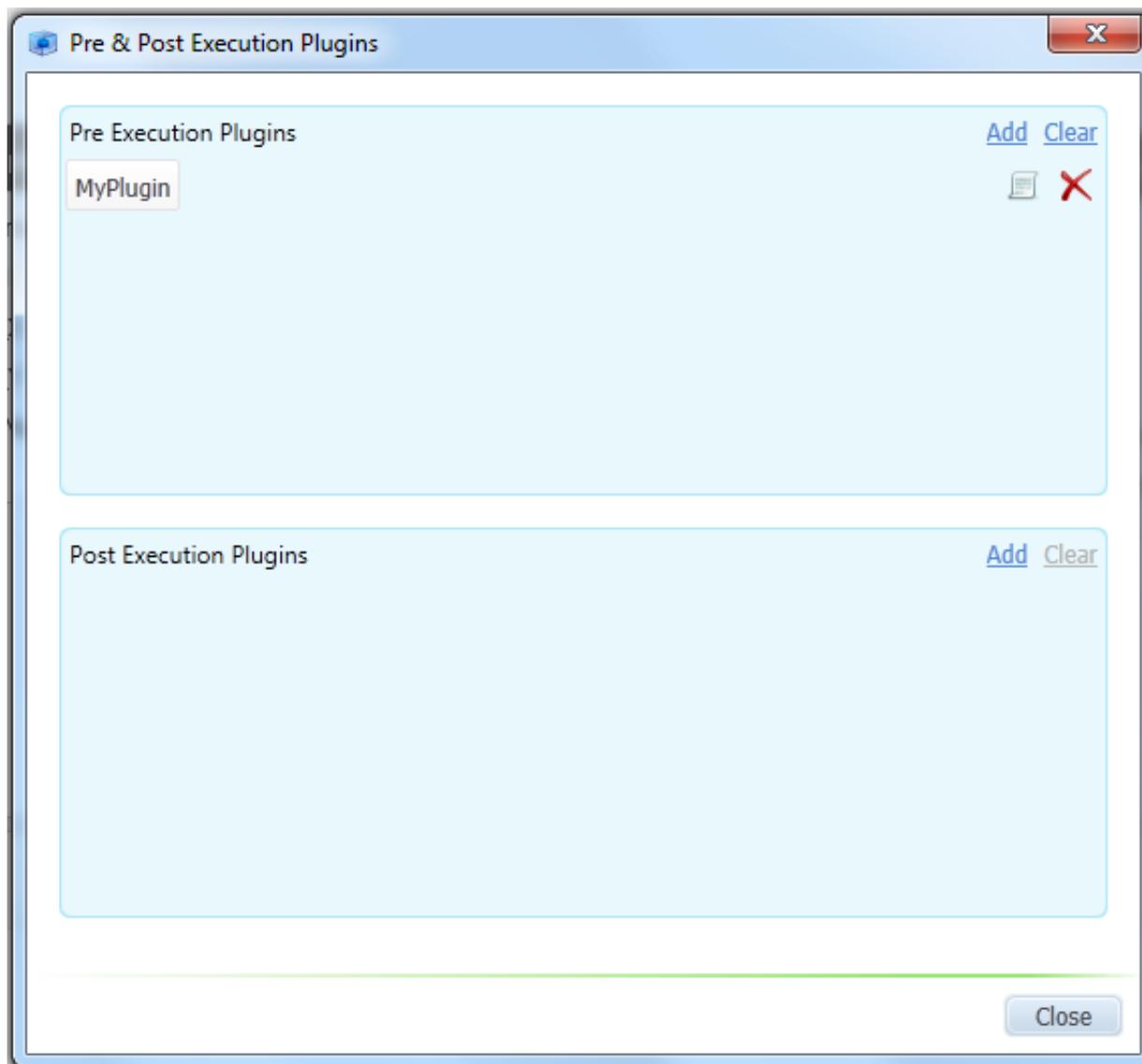
```

Using the Plugin

To use your plugins click the 'Pre & Post Execution Plugins' button under the 'ElastiCube' menu in the ElastiCube Manager.



This will open the Plugin window. Click the 'Add' button to add your plugin, and rename it to match the name you used in the Class attribute of your .NET project.



You send arguments to your plugin in XML form by clicking the 'Edit execution plugin arguments' button on the right side of the window.

An example of how the arguments should be formatted:

You can then access these attributes from your .NET application using the `args.Args.Attribute(name)` method. Here's an example in C#:

```
int myInt = Int32.Parse(args.Args.Attribute("MyInt").Value);
```

You can use the same plugins for Pre & Post execution.

Example: Send Build Status Email

The post plugin below sends emails to a list of email addresses with the build status.



Note



[Complete Code DLL](#)

```
using System; using System.Collections.Generic; using System.Linq;
using System.Text; using Elasticube.PluginBase; using System.Net.M
ail; using System.Net; using System.Xml.Linq; namespace PluginSam
ple { [ElasticubePart("MyPlugin", ExecutionTypes.ExecuteAlways,
false)] public class Plugin: IElasticubePart { #reg
ion IElasticubePart Implementation public IElasticubePartRe
sult Run(IElasticubePartArgs args) { // 0. Pre
pare the result object ElasticubePartResult eCubePartresu
lt = new ElasticubePartResult(); // 1. Get Elasticube I
D string ecubeID = args.ElasticubeXml.Elements().First().
Attribute("ID").Value; //2. Get Elasticube name
string elasticubeName = args.ElasticubeName; //3.
Get the "From" email address string fromEmailAddress = a
rgs.Args.Attribute("FromEmailAddress").Value; //4. Get
```

```

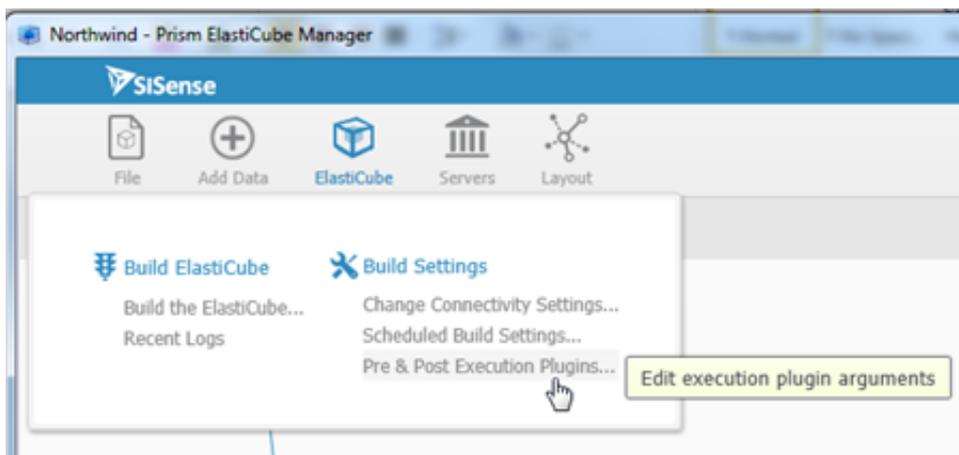
the "From" name          string fromPassword = args.Args.Attribute
("FromEmailPassword").Value;          //5. Get the "To" name
          string[] toEmailAddresses = args.Args.Attribute("ToEmailAddr
ess").Value.Split(",").ToCharArray();          //6. Get the emai
l subject          string emailSubject = args.Args.Attribute("Emai
lSubject").Value;          //7. Get the email body
string emailBody = args.Args.Attribute("EmailBody").Value;          //
Update the email subject with the build status          if (args.I
sTerminating)          {          emailSubject = emailSubj
ect + " - Build Failed ";          }          else
{          emailSubject = emailSubject + " - Build Finished
Successfully ";          }          //Iterating over the "To
Emails" and send each of them an email.          for (int i = 0; i

```

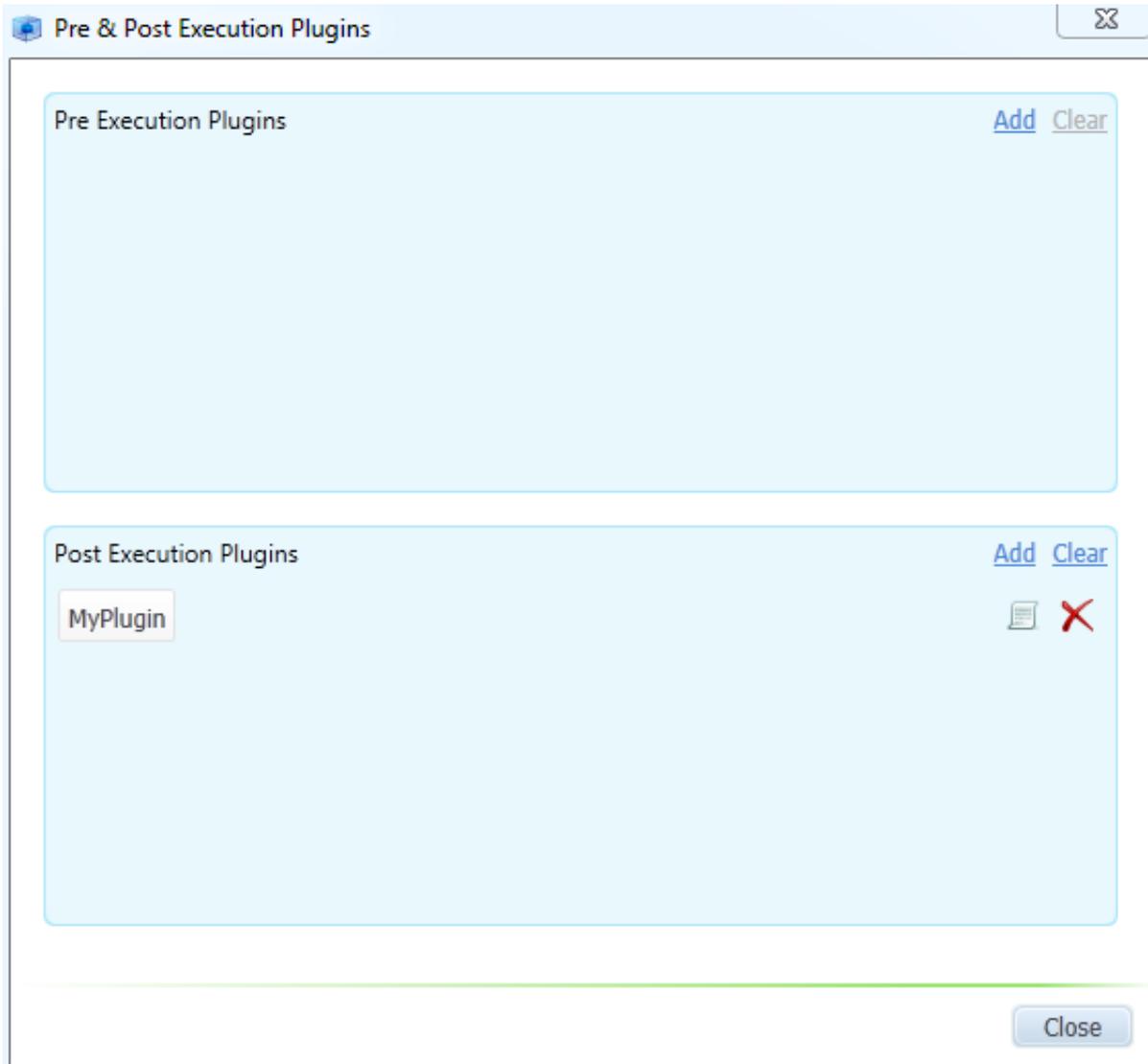
Using the Plugin

Your final plugin dll must be copied into the 'plugins' folder in the Prism installation directory. If the 'plugins' directory doesn't exist there, create it. The default location of the folder is: 'C:\Program Files\Sisense\Prism\plugins'

To use your plugins click the 'Pre & Post Execution Plugins' button under the 'Elasticube' menu in the Elasticube Manager.



This will open the Plugin window. Click the 'Add' button to add your plugin, and rename it to match the name you used in the Class attribute of your .NET project.



You send arguments to your plugin in XML form by clicking the 'Edit execution plugin arguments' button on the right side of the window. An example of how the arguments should be formatted:

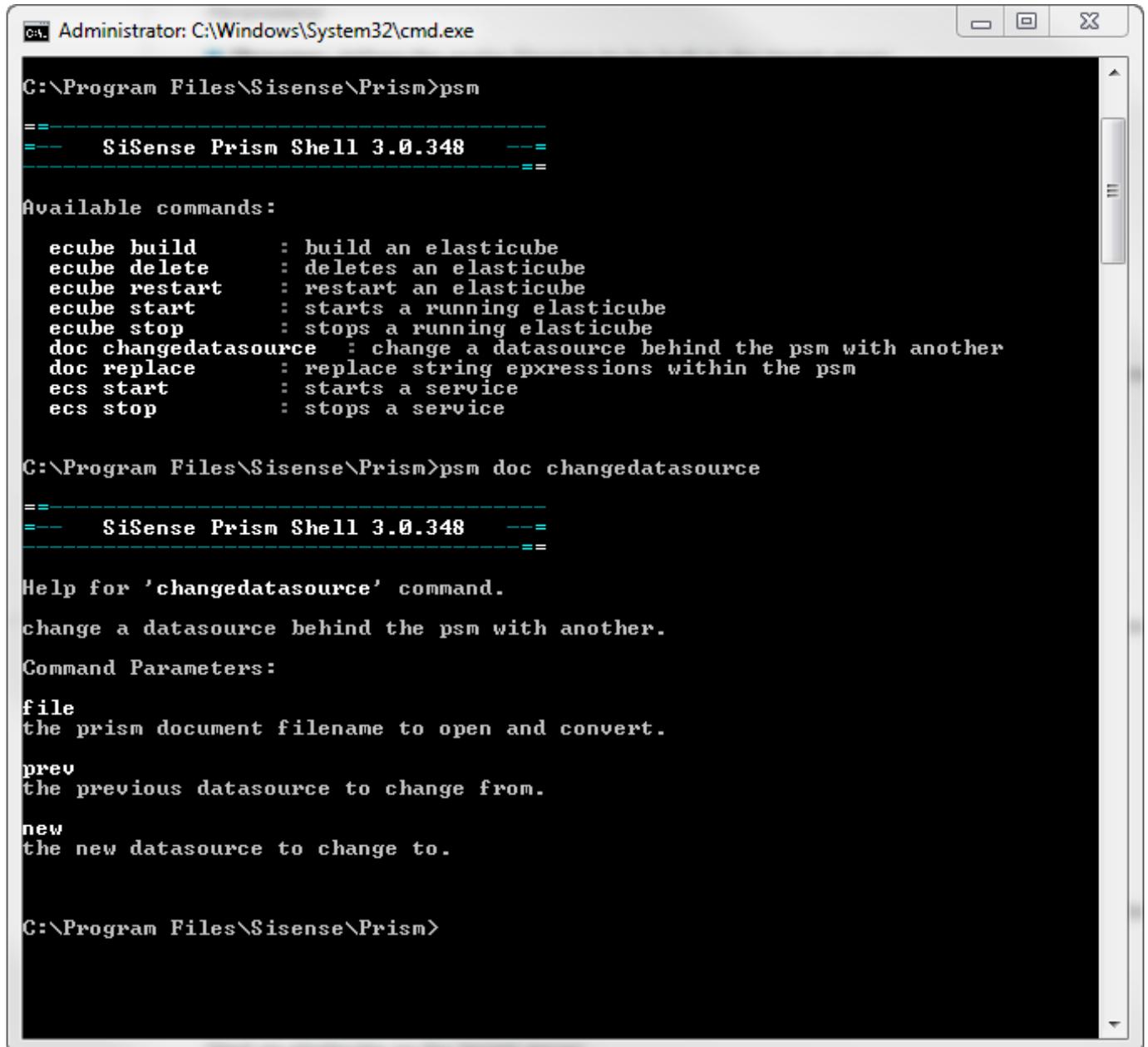
XML Sample:

Note

In this example we used smtp.google.com to send emails.

Sisense Shell (PSM.EXE)

Sisense supports executing ElastiCube related commands using the Windows command line.



```
Administrator: C:\Windows\System32\cmd.exe
C:\Program Files\Sisense\Prism>psm
=====
---  SiSense Prism Shell 3.0.348  ---
=====

Available commands:

ecube build      : build an elasticube
ecube delete    : deletes an elasticube
ecube restart   : restart an elasticube
ecube start     : starts a running elasticube
ecube stop      : stops a running elasticube
doc changedatasource : change a datasource behind the psm with another
doc replace     : replace string expressions within the psm
ecs start       : starts a service
ecs stop        : stops a service

C:\Program Files\Sisense\Prism>psm doc changedatasource
=====
---  SiSense Prism Shell 3.0.348  ---
=====

Help for 'changedatasource' command.
change a datasource behind the psm with another.
Command Parameters:
file
the prism document filename to open and convert.
prev
the previous datasource to change from.
new
the new datasource to change to.

C:\Program Files\Sisense\Prism>
```

Available Commands

Build

Build an ecube file or server hosted elasticube

Parameters:

- **filename:** defines the ecube filename to be built to the target server.
- **name:** defines the name of the elasticube which should be built on the target server.
- **serverAddress:** defines the target server address on which the elasticube should be built.
- **mode:** defines the mode in which the elasticube should be built, available modes:
 1. **mode=restart:** Rebuild the ElastiCube from scratch.
 2. **mode=full:** Rebuild the ElastiCube and accumulate data for tables marked as accumulative. This mode should only be used for accumulative builds.
 3. **mode=changes:** Rebuild from scratch tables that have changed in the ElastiCube schema.
 4. **mode=metadataonly:** Update the ElastiCube server with the ElastiCube schema, without building.

Changedatasource

Change the data source behind a psm to another data source.

Parameters:

- **file:** the prism document filename to open and convert.
- **prev:** the previous data source to change from.
- **new:** the new data source to change to.

Delete

Delete an elasticube from the target server

Parameters:

- **name:** defines the name of the elasticube which should be removed from the target server.
- **serverAddress:** defines the target server address from which the elasticube should be removed.

Restart

Restart an elasticube on the target server.

Parameters:

- **name:** defines the name of the elasticube which should be restarted.
- **serverAddress:** defines the target server address on which the elasticube should be

restarted.

Example:

```
psm ecube build filename="c:\mycube.ecube" serverAddress="localhost" mode=restart
```

Start

Start an elasticube on the target server.

Parameters:

- **name:** defines the name of the elasticube which should be started.
- **serverAddress:** defines the target server address on which the elasticube should be started.

Stop

Stop a running elasticube on the target server.

Parameters:

- **name:** defines the name of the elasticube which should be stopped.
- **serverAddress:** defines the target server address on which the elasticube should be stopped.

Attach

Attaches an ElastiCube folder to the local server.

Parameters:

- **path:** defines the full path of the folder you want to attach.

```
Administrator: Command Prompt
psm.exe ecube attach path=C:\ProgramData\Sisense\PrismServer\ElastiCubeData\ars_2015-1-8_16-0-51_
```



```
Administrator: Command Prompt
psm.exe ecube attach path=C:\ProgramData\Sisense\PrismServer\ElastiCubeData\ars_2015-1-8_16-0-51
=====
SiSense SiSense Shell 4.7.2900.21
=====
| attaching C:\ProgramData\Sisense\PrismServer\ElastiCubeData\ars_2015-1-8_16-0-51... done.
```

Detach

Detaches an ElastiCube from the local server.

Parameters:

- **name:** defines the name of the ElastiCube you want to detach.

```
Administrator: Command Prompt
psm.exe ecube detach name=rs_
```



```
Administrator: Command Prompt
psm.exe ecube detach name=rs
=====
SiSense SiSense Shell 4.7.2900.21
=====
| detaching rs... done.
```