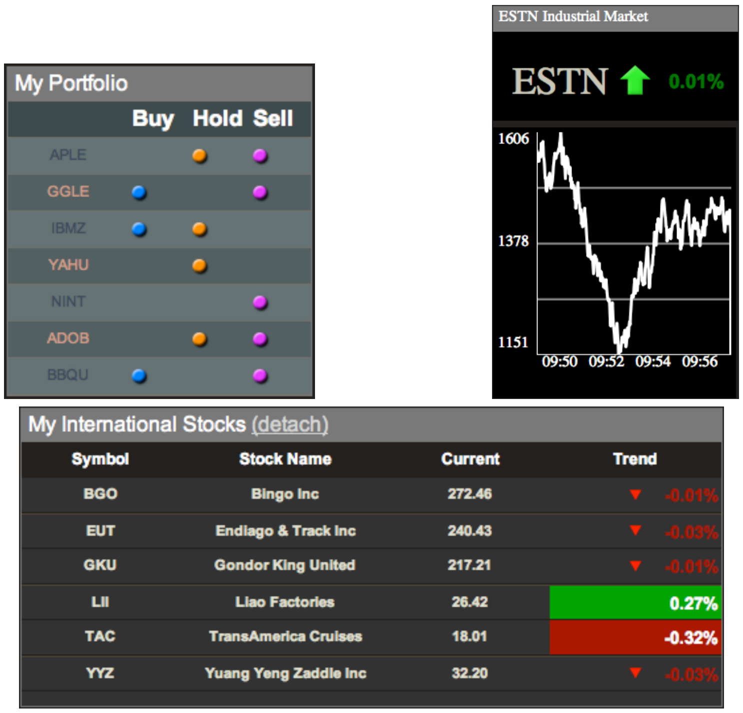
Dashboard handbook

Introduction

Welcome to the Dashboard Handbook, your one-stop source for learning about and creating 4D Dashboards for the desktop. Dashboards are a way of providing visual depictions of real time information and performance indicators on your web page. All the information is pulled from your 4D database and updated in real time, and now you have the power to create them thanks to the 4D Ajax Framework.



My Portfolio

	Buy	Hold	Sell
APLE		●	●
GGLE	●		●
IBMZ	●	●	
YAHU		●	
NINT			●
ADOB		●	●
BBQU	●		●

ESTN Industrial Market

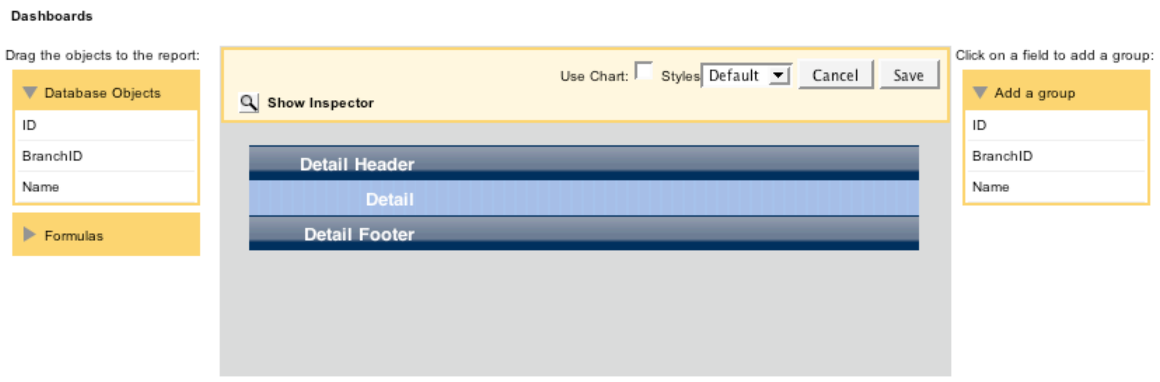
ESTN 0.01%

My International Stocks (detach)

Symbol	Stock Name	Current	Trend
BGO	Bingo Inc	272.46	▼ -0.01%
EUT	Endiago & Track Inc	240.43	▼ -0.03%
GKU	Gondor King United	217.21	▼ -0.01%
LII	Liao Factories	26.42	0.27%
TAC	TransAmerica Cruises	18.01	-0.32%
YYZ	Yuang Yeng Zaddle Inc	32.20	▼ -0.03%

Here are some sample Dashboards.

We have provided a convenient Dashboard Editor as part of the 4D Ajax Framework for easily creating Dashboards on the fly. The graphical user interface supports intuitive drag-and-drop capabilities so that you can quickly begin creating Dashboards of your own.



Here is a look at the Dashboard Editor. This is your blank canvas when creating Dashboards.

Requirements

4D v11 SQL

4D Ajax Framework v11

MoneyExchange.4dbase

The Money Exchange Demo is attached with this document and it comes in one flavor:

- Interpreted Source Database

This interpreted database already has the 4D Ajax Framework v11 installed and the Dashboards created for you. This will allow you access to Developer Hooks and to perform further customization in the back end.

This document provides instructions on how to build each dashboard from scratch as if the HTML page had no JavaScript code and as if there were no dashboards to begin with.

To get started, skip to any chapter that you wish.

- *Chapter 1: Features:* Get an overview of the Dashboard's killer features.
- *Chapter 2: Editor:* Get a quick introduction to the Dashboard Editor's user interface.
- *Chapter 3: Demos:* Follow step-by-step instructions to create Dashboard applications.
  - *Example 1: Simple Dashboards*
    - *Example 1a: Endiago & Track Inc*
    - *Example 1b: Gondor King United*
    - *Example 1c: Liao Factories*
    - *Example 1d: Bingo Inc.*
  - *Example 2: Flashing Indicators*
  - *Example 3: Bonus Indicators*
  - *Example 3: Preset Queries (optional)*
  - *Example 4: Charts*
    - *Example 4a: TAC Chart*
    - *Example 4a: Image URL Chart (optional)*
    - *Example 4b: BGO Chart*
    - *Example 4b: SVG Chart (optional)*
  - *Example 5: Complementary Dashboard*
    - *Example 5a: ESTN Dashboard*
    - *Example 5b: ESTN Chart*
- *Appendix:* Reference section.

Optional examples modify existing examples to showcase alternate features.

Chapter 1: Features

Let's take a look at some of the Dashboard's most notable features.

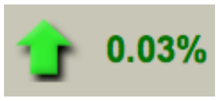
Indicators

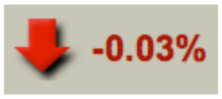
Desktop Dashboards have the Indicators that are available to the iPhone, plus some Bonus Indicators as well.

The iPhone available Indicators are:

- Green
- Yellow
- Red
- Green button
- Yellow button
- Red button

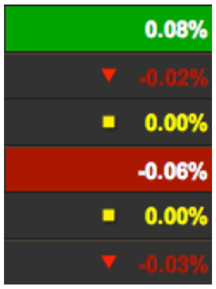
The Bonus Indicators are not currently part of the Dashboard Editor. They are, however, available via Developer Hook *Dax_DevHook_SetIndicators*. This is a list of them:





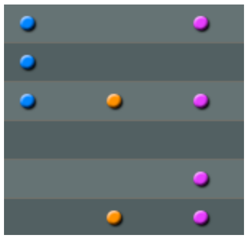
- Arrowup
- Arrowdown
- Noarrowyellow
-

These Indicators work well if you want to display information with large font sizes. You can increase row height via CSS styling.



- Greenstrong
- Redstrong
- Green Triangle
- Red Triangle
- Yellow Square

These Indicators act as an alternative to the iPhone Dashboard Indicators.



- Blue
- Orange
- Purple

This set of Indicators can work well for Boolean type fields.

Note: Indicators "Green button," "Red button," and "Yellow button" are fixed images. Thus they have fixed widths. If you extend your column width beyond the default width of 90 pixels, the text in your field may exceed the size of the image and so the information my look distorted.

Charts

Desktop Dashboard Charts use Developer Hook *Dax_DevHook_DefineChart* to define the values represented in the Chart.

4 potential types of Charts are supported for Desktop Dashboards:

- Canvas
- PNG
- Image URL
- SVG

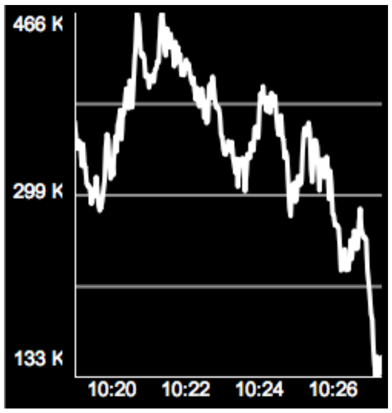
Here is a matrix of supported chart types based on web browser:

	Firefox 3	Safari 4	IE7/8
Canvas Charts	Yes	Yes	Yes
PNG Charts	Yes	Yes	Yes
Image URL Charts	Yes	Yes	Yes
SVG Charts	Yes	Yes	No
Interactive Charts	Yes	Yes	Yes

Charts on the Desktop automatically resize to the spaces allotted to them on the HTML page.

Canvas Chart

Canvas is an HTML element introduced by Apple. It allows the rendering of bitmap images and it is most notably known for powering applications like Apple Dashboard widgets and iPhone applications such as the Stock app. 4D Developers can use this same element to build charts in their 4D Dashboards.



PNG Chart

PNG support allows 4D Developers to append a picture to the dashboard. The pictures will be created into a temporary folder and the accessible URL will be added to the element as a separate link sub-element.

Please note that the original picture format will be converted to PNG and each temporary picture will be removed from the disk once the user session is expired.

There is a bug with 4D v11 SQL and charting for Unicode databases which does not allow charts to be created in 4D. As a result, PNG Charts for iPhone Dashboards cannot be created in this setup. Developers are encouraged to see the Daxipedia for examples on how to create PNG Charts for dashboards:

http://daxipedia.4d.com/index.php/Chart#PNG

Image URL Chart

The Image URL is best used with external images where its URL is composed dynamically, like in the case of Google Charts.

SVG Chart

SVG support allows 4D Developers to an SVG chart to the dashboard. The SVG charts will be created into a temporary folder and the accessible url will be added to the element as separate link sub-element.

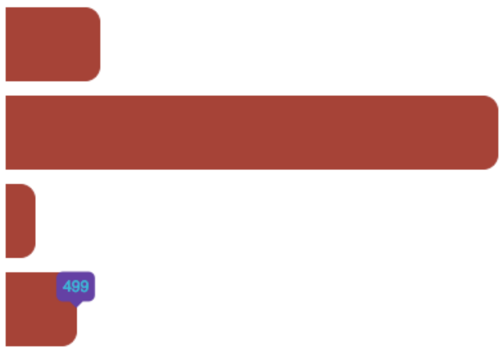
Please note that each temporary svg file will be removed from the disk once the user session is expired.

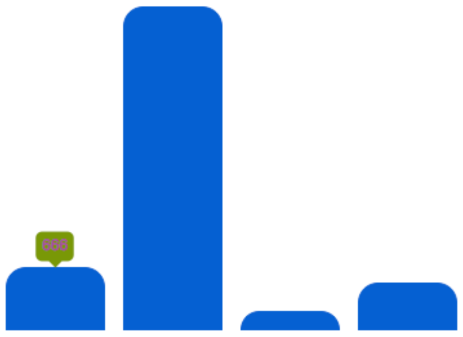
Interactive Charts

4D Ajax Framework v11 Release (11.5) introduced much more interactive and customizable charts. You can hover over the them to interactively see their values, and an expanded set of commands allow for color customization. With more options the new Interactive Charts of the 4D Ajax Framework v11 Release 5 (11.5) give you more power to effectively convey visual information.

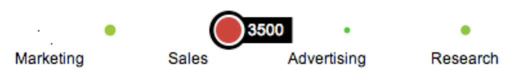
Here is a look at them:

Bar Charts:

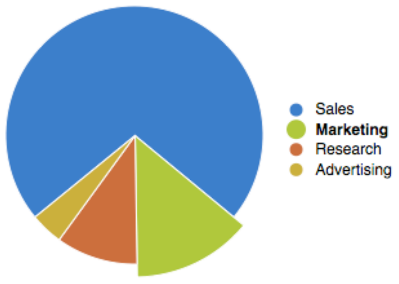




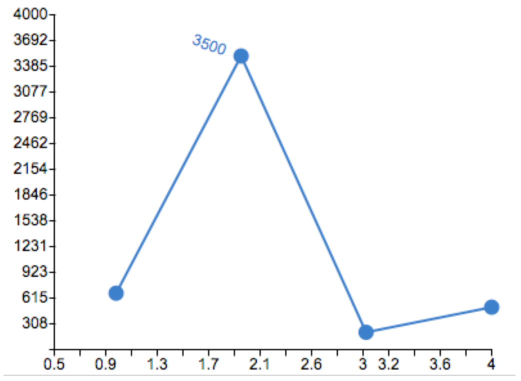
Dot Charts:



Pie Charts:

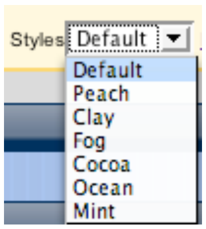


Line Charts:



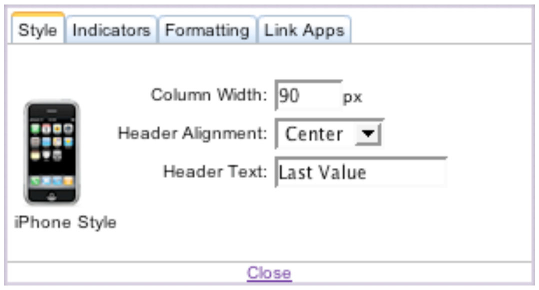
Themes

Desktop Dashboards have their own set of themes to choose from (as opposed to the Blue and Black themes for iPhone Dashboards).



Customization of headers and Detail Rows

We understand the importance of precision down to the very pixel when it comes to creating web pages. Thus, we have provided advanced customization features such as column width (in pixels), header alignment, header text, and detail alignment for Dashboards.



Column Width: 90 px
Header Alignment: Center
Header Text: Last Value

Cross-Browser Compatibility

Desktop Dashboards have been built to look and function in the same manner across all major browsers – Safari 3+, Internet Explorer 7+, and Firefox 2+.

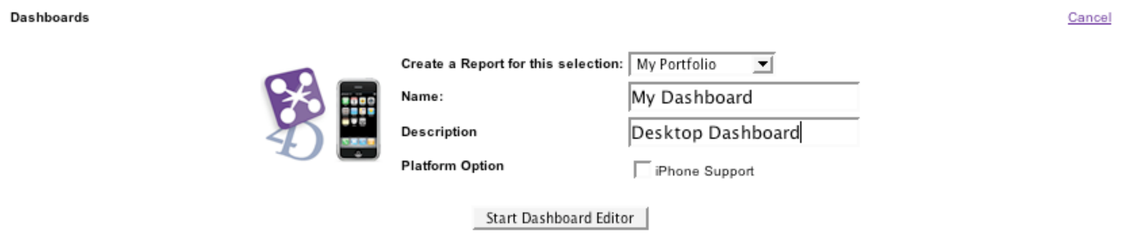
These features, along with a graphical user interface Dashboard Editor, add up to a powerful and convenient toolkit for creating Dashboards on your web pages.

Preset Queries support

Preset Queries are queries defined by the administrator ahead of time to make things easier for end user once they log in. Preset queries are now supported in 4D Ajax Framework v11 Release 1 (11.1).

Chapter 2: Editor

This chapter will get you familiar with the Dashboard Editor. Use the Editor to create Dashboards.

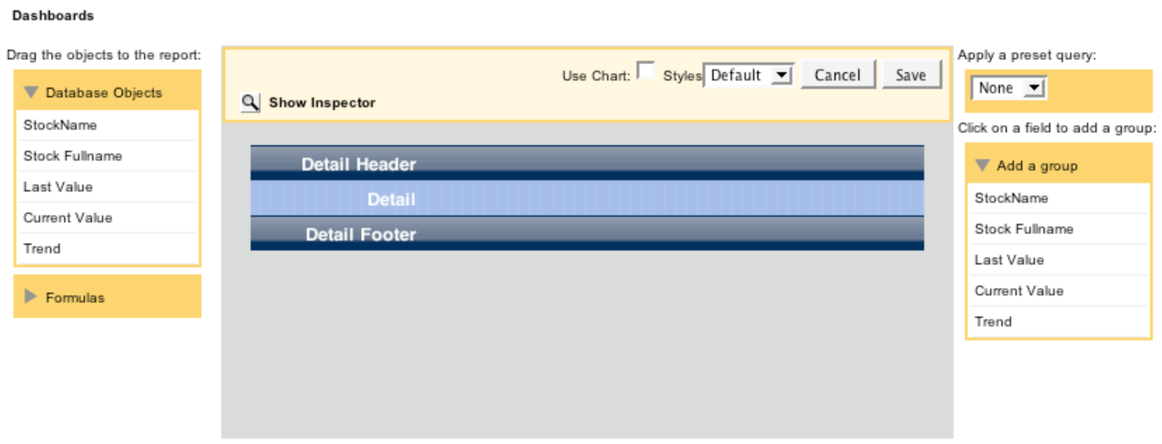


Create a Report for this selection: My Portfolio
Name: My Dashboard
Description: Desktop Dashboard
Platform Option: iPhone Support

Here is the Dashboard Manager tab as part of the Control Panel in the 4D Ajax Framework.

- Choose the Selection (ie. Table, View, DCS) that the Dashboard will be based on.
- Give the Dashboard a name.
- Provide a description.

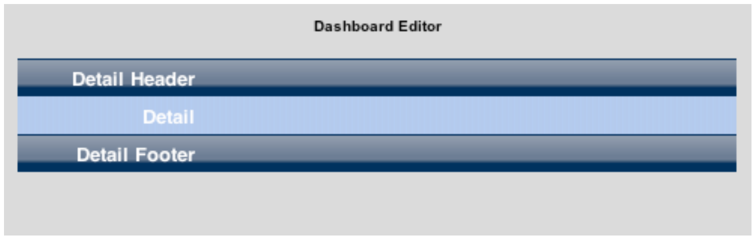
Click the **Start Dashboard Editor** button to begin. You will then be presented with the following.



Styles: Default
Apply a preset query: None

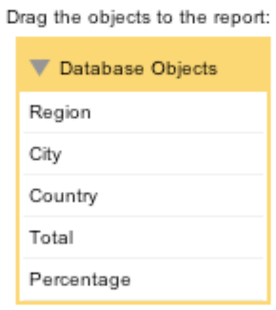
This is the Dashboard Editor. Here you define what the Dashboard application will report on, and how it will be displayed. Let's take a look at the pieces that make up the Dashboard Editor.

Editor



This provides a quick preview of what you are building. Drag fields to this area. Organize information by grouping it. Drag formulas to the Footer areas to perform calculations on specific columns. All these features and more can be done, and this editor outlines how the Dashboard will look.

Database Objects

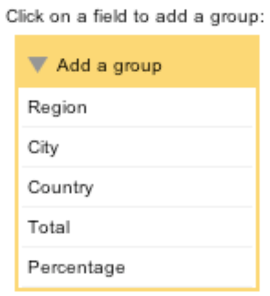

Drag the objects to the report:

Database Objects

- Region
- City
- Country
- Total
- Percentage

These are the fields in your Selection. Drag and drop fields to the Editor area in the way you want them to appear in the Dashboard.

Add a group

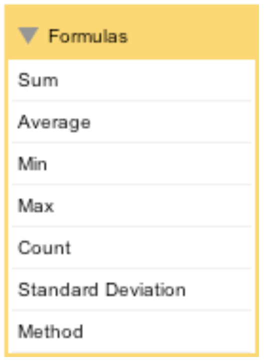

Click on a field to add a group:

Add a group

- Region
- City
- Country
- Total
- Percentage

Group data in the Dashboard by clicking a field from this box.

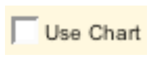
Formulas


Formulas

- Sum
- Average
- Min
- Max
- Count
- Standard Deviation
- Method

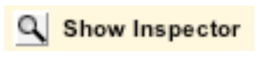
Drag Formulas to the Editor to perform calculations on fields.

Charts

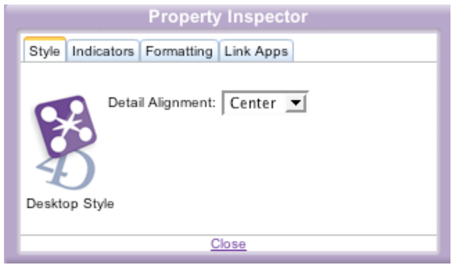


Toggle the ability to use Charts. With this option selected,

Property Inspector



Click this link to load the Property Inspector. If you then select a header or detail cell you can customize look and functionality. The Property Inspector loads with the following tabs:



- **STYLE:**
  For a Desktop Dashboard it should display *Desktop Style.* Here you can alter text alignment for a particular detail cell. You can also customize column width, header alignment, and header text when you select a header cell.

- **INDICATORS:**
  Set conditions for your Indicators here. Priority is given to the conditions listed at the top of the list.

Currently there is no limit to the number of Indicator conditions that you can set. However, please note that the more conditions you set the more processing time it takes.

The order at which these conditions are listed is very important. The conditions can be thought of as Case statements. Thus, the first condition in the list that is evaluated as True exits the entire list and no longer evaluates the remaining conditions.

- **FORMATTING:**
  Select a style from the pull-down menu to format the field.

- **LINK APPS:**
  You can link applications to fields once they are clicked. You can link fields to the following:
  a. Browser: Click a field that has a URL in it and it can load a webpage.
  b. Mail: Click a field that has an email address in it and it can compose an email to that address using your default email client.
  c. Maps: Ciick a field with an address in it and it can go to a Google Maps webpage displaying that address.

Note: Clicking on a Maps link will load the Google Maps page on the same window that contains the Dashboard. Thus, the page that contains the Dashboard will be exited.

PRESET QUERIES SUPPORT

Preset Queries are queries defined by the administrator ahead of time to make things easier for end user once they log in.

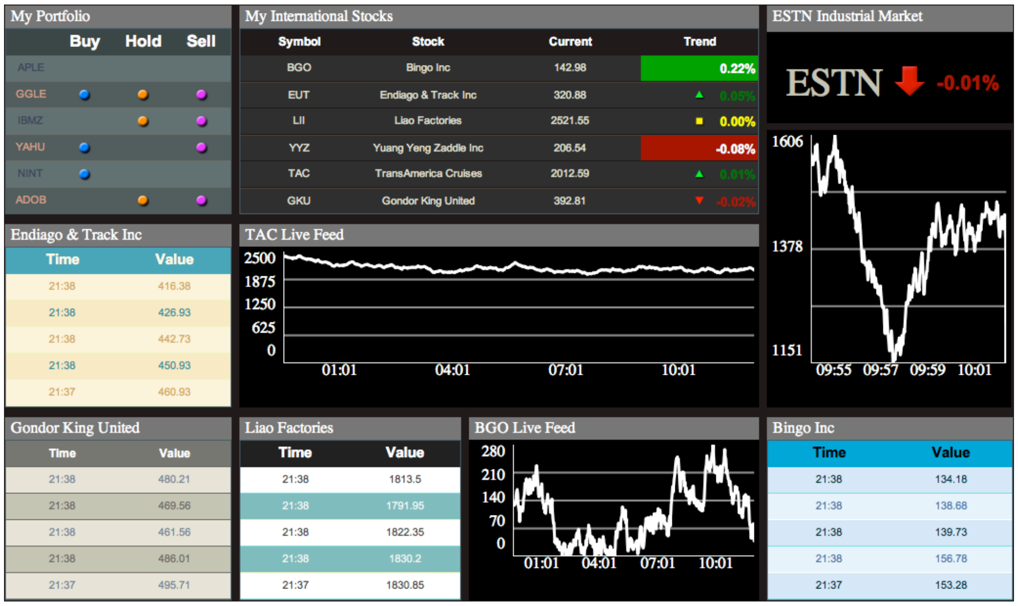
Only Custom Queries and queries created using templates are supported. Dynamic Queries (Data Driven) are not supported.

Chapter 3: Demos

This chapter provides instructions on how to build each dashboard from scratch as if the HTML page had no JavaScript code and as if there were no dashboards to begin with.

Launch the attached Money Exchange Demo database.

Connect to http://localhost:8080/dash.html



My Portfolio

	Buy	Hold	Sell
APLE			
GGLE	●	●	●
IBMZ		●	●
YAHU	●		●
NINT	●		
ADOB		●	●

My International Stocks

Symbol	Stock	Current	Trend
BGO	Bingo Inc	142.98	0.22%
EUT	Endiago & Track Inc	320.88	▲ 0.05%
LII	Liao Factories	2521.55	■ 0.00%
YYZ	Yuang Yeng Zaddle Inc	206.54	-0.08%
TAC	TransAmerica Cruises	2012.59	▲ 0.01%
GKU	Gondor King United	392.81	▼ -0.02%

Endiago & Track Inc

Time	Value
21:38	416.38
21:38	426.93
21:38	442.73
21:38	450.93
21:37	460.93

Gondor King United

Time	Value
21:38	480.21
21:38	469.56
21:38	461.56
21:38	486.01
21:37	495.71

Liao Factories

Time	Value
21:38	1813.5
21:38	1791.95
21:38	1822.35
21:38	1830.2
21:37	1830.85

Bingo Inc

Time	Value
21:38	134.18
21:38	138.68
21:38	139.73
21:38	156.78
21:37	153.28

This is the end product when going over the steps in this chapter to build dashboards. View the examples to see how each dashboard was made.

- Connect to http://localhost:8080/index.html.
- Log in to the framework as *Administrator* (no password).
- Hit the Control Panel button. Go to the Dashboard Manager tab.

Now go through the examples and build the Dashboards.

DASH.HTML

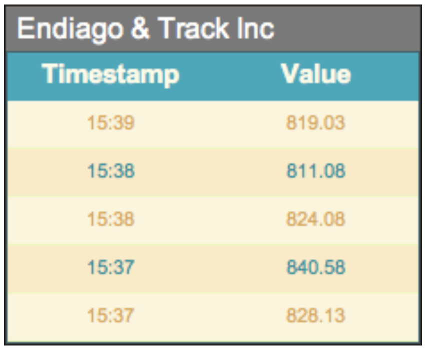
Make sure that you have an HTML editor to edit “dash.html” (found in Webfolder). We will add JavaScript code to embed Dashboards into the page.

Example 1: Simple Dashboards

Here we are going to build simple Dashboards and apply some minor customization to them. The goal here is to get familiar with building Dashboards and then embedding them into an HTML page.

Example 1a: Endiago & Track Inc

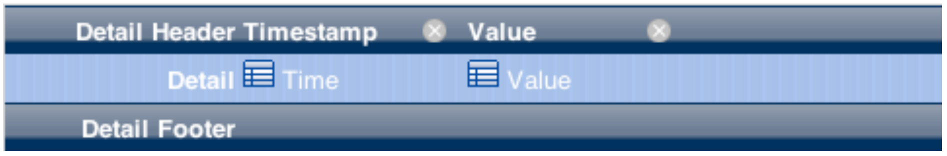
Here we are creating a Dashboard for an imaginary company named “Endiago & Track Inc”. Its data will be periodically updated in real-time.



Endiago & Track Inc	
Timestamp	Value
15:39	819.03
15:38	811.08
15:38	824.08
15:37	840.58
15:37	828.13

To begin, click the *Create* link to create a new Dashboard.

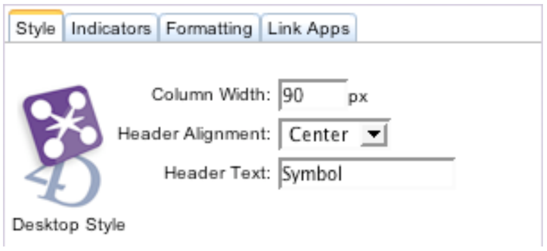
- Choose “DCS_EUT” as the selection.
- Name the Dashboard “EUT”.
- Set the Description as “Endiago & Track Inc”.
- Do **not** select ‘iPhone Support’ for Platform Option. Click *Start Dashboard Editor* to begin.



In the Editor:

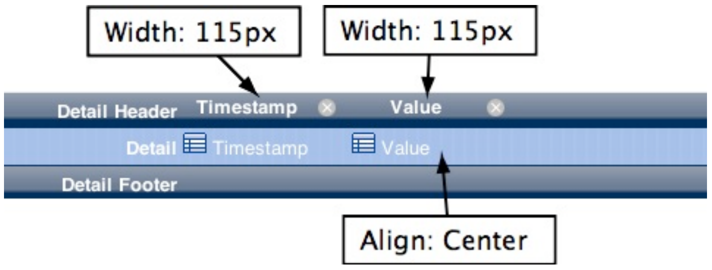
- Drag *Time,* and *Value* to the Editor.

Now select the *Timestamp* detail header cell. It will load the Property Inspector.



Column Width: 90 px
Header Alignment: Center
Header Text: Symbol

This is the *Property Inspector*. Under the *Style Tab* it displays different ways the Header can be modified.



Select the *Timestamp* header cell.

- Set Column Width: 115px

Select the *Value* header cell.

- Set Column Width: 115px

Select the *Value* detail cell.

- Align: Center

Set the CSS Style to *Peach*. Save the Dashboard.

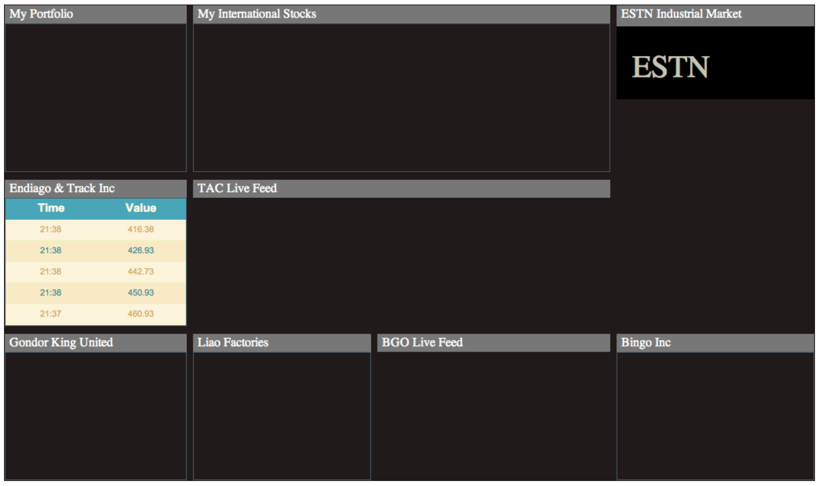
Dash.html

Add the following code to dash.html within *function onAfterInit()* in the <script> element:

```
EUT = new dashboardViewer("EUT", $("dashboardEUT"));
EUT.setRefreshInterval(12000);
EUT.viewport.showVerScrollbar(false);
```

This code will embed the Dashboard "EUT" within the HTML page in the <div> "dashboardEUT".

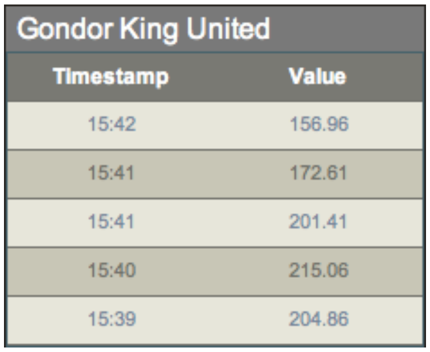
Reconnect to http://localhost:8080/dash.html. It should now look like:



Time	Value
21:38	416.38
21:38	426.93
21:38	442.73
21:38	450.93
21:37	460.93

Example 1b: Gondor King United

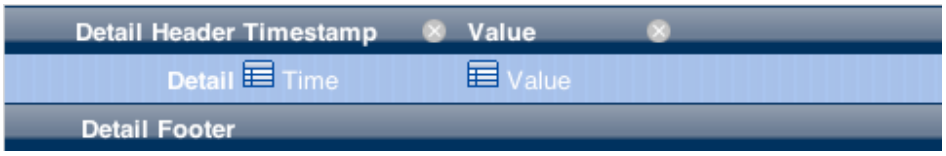
Here we are creating a Dashboard for an imaginary company named "Gondor King United". Its data will be periodically updated in real-time.


Gondor King United

Timestamp	Value
15:42	156.96
15:41	172.61
15:41	201.41
15:40	215.06
15:39	204.86

To begin, click the *Create* link to create a new Dashboard.

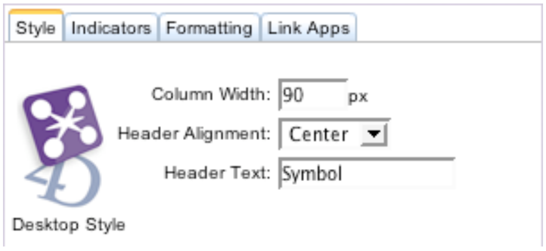
- Choose "DCS_GKU" as the selection.
- Name the Dashboard "GKU".
- Set the Description as "Gondor King United".
- Do **not** select 'iPhone Support' for Platform Option. Click *Start Dashboard Editor* to begin.



In the Editor:

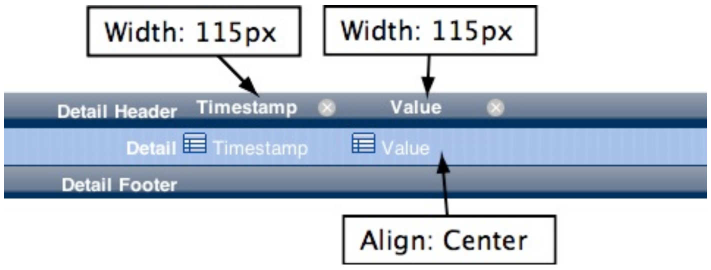
- Drag *Time,* and *Value* to the Editor.

Now select the *Timestamp* detail header cell. It will load the Property Inspector.



Column Width: 90 px
Header Alignment: Center
Header Text: Symbol

This is the *Property Inspector*. Under the *Style Tab* it displays different ways the Header can be modified.



Select the *Timestamp* header cell.

- Set Column Width: 115px

Select the *Value* header cell.

- Set Column Width: 115px

Select the *Value* detail cell.

- Align: Center

Set the CSS Style to *Clay*. Save the Dashboard.

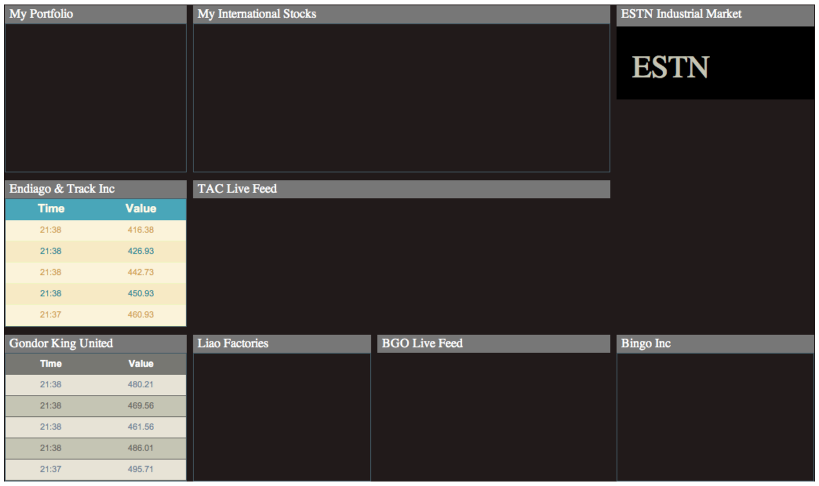
Dash.html

Add the following code to dash.html within *function onAfterInit()* in the <script> element:

```
GKU = new dashboardViewer("GKU", $("dashboardGKU"));
GKU.setRefreshInterval(16000);
GKU.viewport.showVerScrollbar(false);
```

This code will embed the Dashboard “GKU” within the HTML page in the <div> “dashboardGKU”.

Reconnect to http://localhost:8080/dash.html. It should now look like:



Endiago & Track Inc

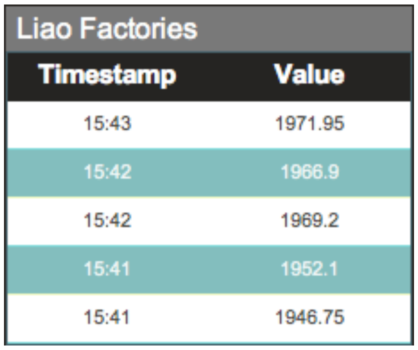
Time	Value
21:38	416.38
21:38	426.93
21:38	442.73
21:38	450.93
21:37	460.93

Gondor King United

Time	Value
21:38	480.21
21:38	469.56
21:38	461.56
21:38	486.01
21:37	495.71

Example 1c: Liao Factories

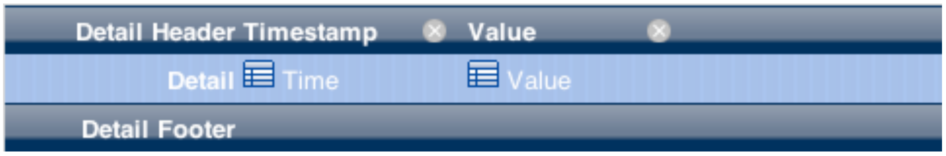
Here we are creating a Dashboard for an imaginary company named “Liao Factories”. Its data will be periodically updated in real-time.


Liao Factories

Timestamp	Value
15:43	1971.95
15:42	1966.9
15:42	1969.2
15:41	1952.1
15:41	1946.75

To begin, click the *Create* link to create a new Dashboard.

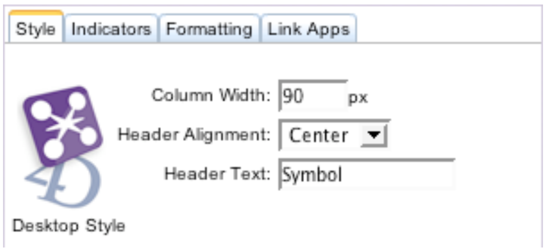
- Choose “DCS_LII” as the selection.
- Name the Dashboard “LII”.
- Set the Description as “Liao Factories”.
- Do **not** select ‘iPhone Support’ for Platform Option. Click *Start Dashboard Editor* to begin.



In the Editor:

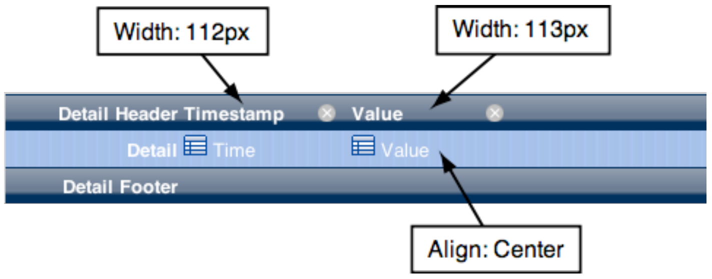
- Drag *Time,* and *Value* to the Editor.

Now select the *Timestamp* detail header cell. It will load the Property Inspector.



Column Width: 90 px
Header Alignment: Center
Header Text: Symbol

This is the *Property Inspector*. Under the *Style Tab* it displays different ways the Header can be modified.



Select the *Timestamp* header cell.

- Set Column Width: 115px

Select the *Value* header cell.

- Set Column Width: 115px

Select the *Value* detail cell.

- Align: Center

Set the CSS Style to *Mint*. Save the Dashboard.

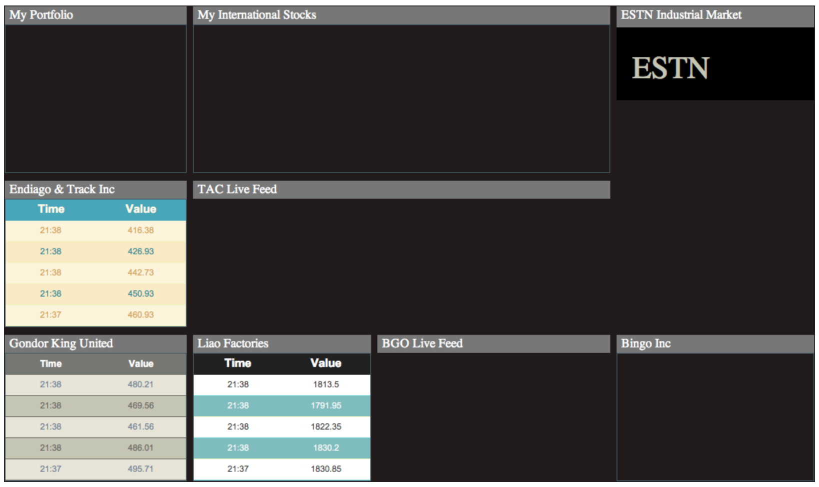
Dash.html

Add the following code to dash.html within *function onAfterInit()* in the <script> element:

```
LII = new dashboardViewer("LII", $("dashboardLII"));
LII.setRefreshInterval(15000);
LII.viewport.showVerScrollbar(false);
```

This code will embed the Dashboard “LII” within the HTML page in the <div> “dashboardLII”.

Reconnect to http://localhost:8080/dash.html. It should now look like:



Endiago & Track Inc

Time	Value
21:38	416.38
21:38	426.93
21:38	442.73
21:38	450.93
21:37	460.93

Gondor King United

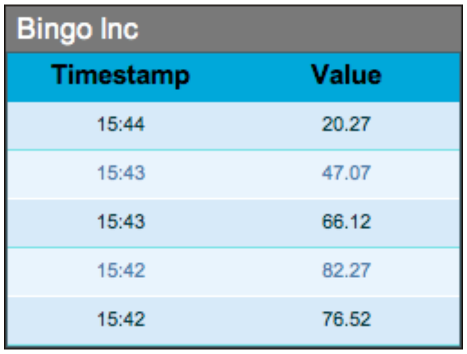
Time	Value
21:38	480.21
21:38	469.56
21:38	461.56
21:38	486.01
21:37	495.71

Liao Factories

Time	Value
21:38	1813.5
21:38	1791.95
21:38	1822.35
21:38	1830.2
21:37	1830.85

Example 1D: Bingo Inc.

Here we are creating a Dashboard for an imaginary company named “Bingo Inc”. Its data will be periodically updated in real-time.

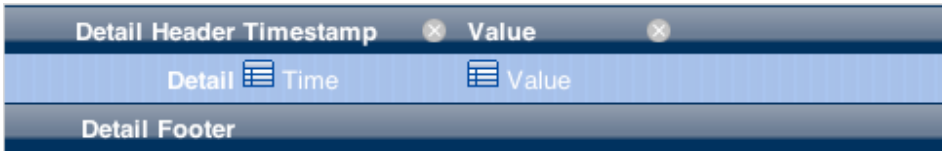


Bingo Inc

Timestamp	Value
15:44	20.27
15:43	47.07
15:43	66.12
15:42	82.27
15:42	76.52

To begin, click the *Create* link to create a new Dashboard.

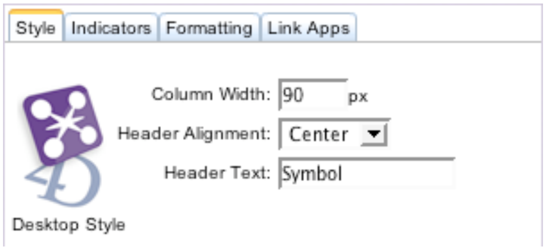
- Choose “DCS_BGO” as the selection.
- Name the Dashboard “BGO”.
- Set the Description as “BGO Inc.”.
- Do **not** select ‘iPhone Support’ for Platform Option. Click *Start Dashboard Editor* to begin.



In the Editor:

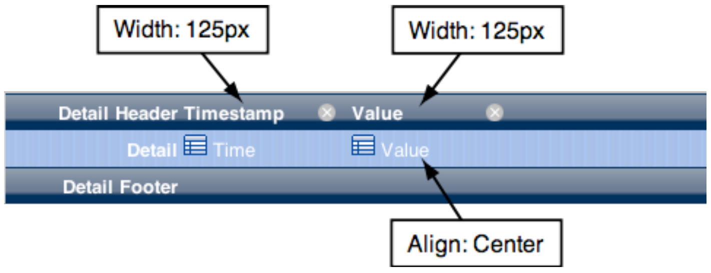
- Drag *Time,* and *Value* to the Editor.

Now select the *Timestamp* detail header cell. It will load the Property Inspector.



Column Width: 90 px
Header Alignment: Center
Header Text: Symbol

This is the *Property Inspector*. Under the *Style Tab* it displays different ways the Header can be modified.



Select the *Timestamp* header cell.

- Set Column Width: 115px

Select the *Value* header cell.

- Set Column Width: 115px

Select the *Value* detail cell.

- Align: Center

Set the CSS Style to *Fog*. Save the Dashboard.

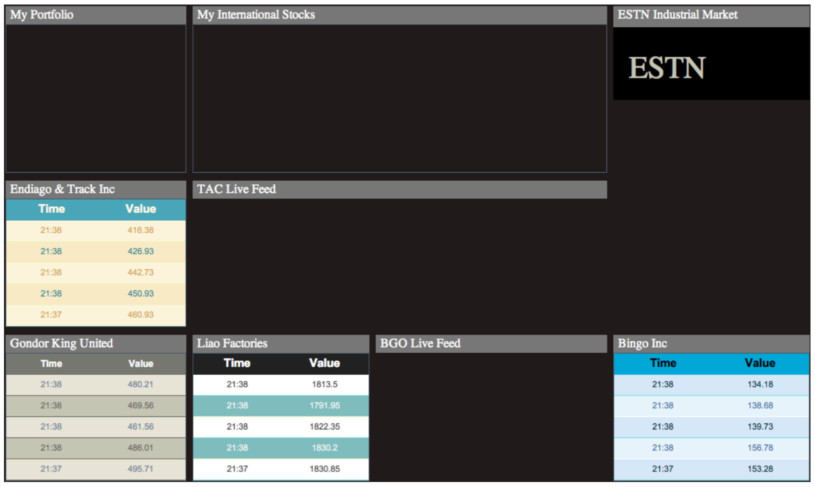
Dash.html

Add the following code to dash.html within *function onAfterInit()* in the <script> element:

```
BGO = new dashboardViewer("BGO", $("dashboardBGO"));
BGO.setRefreshInterval(14000);
BGO.viewport.showVerScrollbar(false);
```

This code will embed the Dashboard "BGO" within the HTML page in the <div> "dashboardLBGO".

Reconnect to http://localhost:8080/dash.html. It should now look like:



Endiago & Track Inc

Time	Value
21:38	416.38
21:38	426.93
21:38	442.73
21:38	450.93
21:37	460.93

Gondor King United

Time	Value
21:38	480.21
21:38	469.56
21:38	461.56
21:38	486.01
21:37	495.71

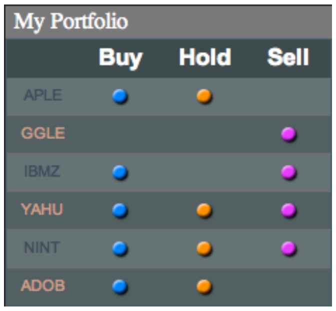
Liao Factories

Time	Value
21:38	1813.5
21:38	1791.95
21:38	1822.35
21:38	1830.2
21:37	1830.85

Bingo Inc

Time	Value
21:38	134.18
21:38	138.68
21:38	139.73
21:38	156.78
21:37	153.28

Example 2: Flashing Indicators



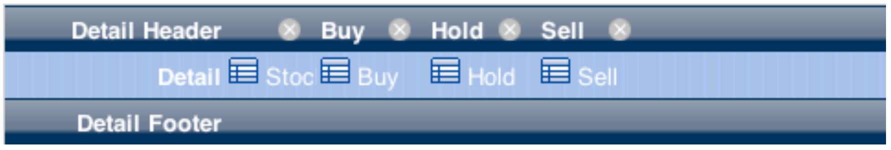
My Portfolio

	Buy	Hold	Sell
APLE	●	●	
GGLE			●
IBMZ	●		●
YAHU	●	●	●
NINT	●	●	●
ADOB	●	●	

Here we are going to build a Dashboard using the *orange, purple, blue* indicators. They can work well with Boolean type fields. In this example, we'll actually randomize values to demonstrate the visual effect of flashing these indicators in a blinking manner.

To begin, click the *Create* link to create a new Dashboard.

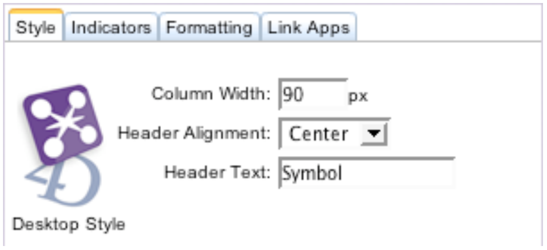
- Choose “My Portfolio” as the selection.
- Name the Dashboard “My Portfolio”.
- Set the Description as “Flashing Indicators”.
- Do **not** select ‘iPhone Support’ for Platform Option. Click *Start Dashboard Editor* to begin.



In the Editor:

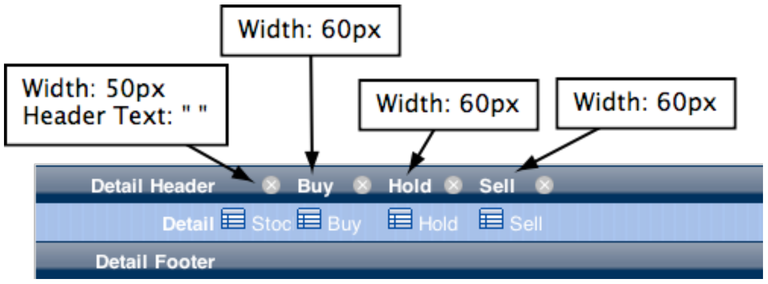
- Drag *Stock, Buy, Hold* and *Sell* to the Editor.

Now select the *Stock* detail header cell. It will load the Property Inspector.



Column Width: 90 px
Header Alignment: Center
Header Text: Symbol

This is the *Property Inspector*. Under the *Style Tab* it displays different ways the Header can be modified.



Customizing the Headers: Style

Select the *Stock* header cell.
- Set Column Width: 50px
- Set Header Text: “ “

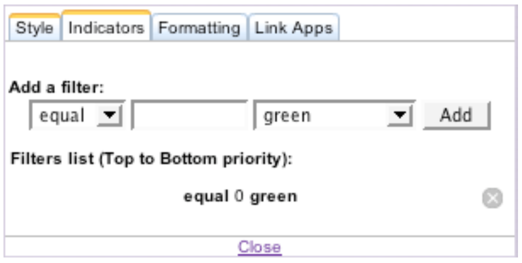
Select the *Buy* header cell.
- Set Column Width: 60px

Select the *Hold* header cell.
- Set Column Width: 60px

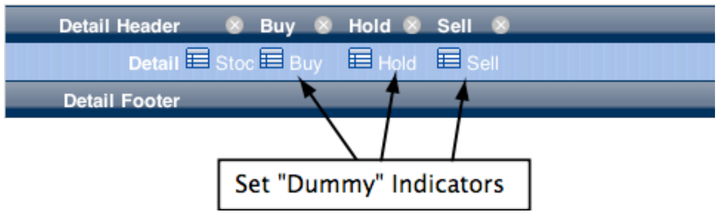
Select the *Sell* detail cell.
- Set Column Width: 60px

Customizing the Detail: Indicators

Here we want to add Indicators. We can do so by clicking a detail cell, which will then load the Property Inspector.



This is the *Property Inspector*. Go to the *Indicators* tab to add Indicators.



Select the *Buy* detail cell.

- Under *Indicators*, add a filter where “equal” to “0” is “green”.
- Hit the *Add* button so that it gets populated in the list.
- Do the same for the *Hold* and *Sell* detail cell.

When we say Dummy Indicators we mean that these conditions will be overruled by the conditions you set in Developer Hook *Dax_DevHook_SetIndicators.* The Dummy Indicators can be thought of placeholders that trigger the Developer Hook. As long as your data values are evaluated by the dummy conditions, *Dax_DevHook_SetIndicators* will then get triggered.

Set the CSS Style to *Ocean*. Save the report.

Add the following Case Statement to *Dax_DevHook_SetIndicators:*

```
: ($ReportName_t="My Portfolio")
C_REAL($randomValue)
$randomValue:=Random%(2)
If ($randomValue=1)
       Case of
       : ($fieldnames_atp->{$fieldnum_l}="Buy")
              $0:="blue"
       : ($fieldnames_atp->{$fieldnum_l}="Hold")
              $0:="orange"
       Else
              $0:="purple"
       End case
Else
       $0:=""
End if
```

With the Dummy Indicators in place, *Dax_DevHook_SetIndicators* will get triggered.

The code in *Dax_DevHook_SetIndicators* is displayed and is further explained in the *Appendix* section.

Dash.html

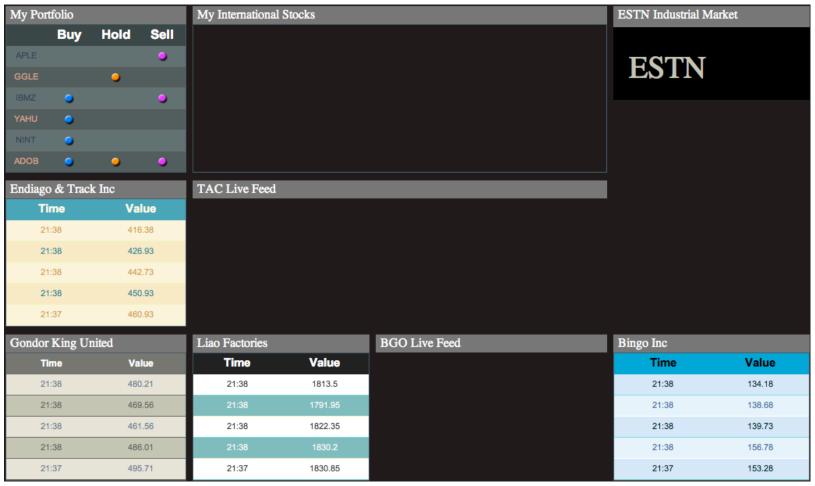
Add the following code to dash.html within *function onAfterInit()* in the <script> element:

```
MyPortfolio = new dashboardViewer("My Portfolio", $("dashboardMyPortfolio"));
```

```
MyPortfolio.setRefreshInterval(13000);
MyPortfolio.viewport.showVerScrollbar(false);
```

This code will embed the Dashboard "My Portfolio" within the HTML page in the <div> "dashboardMyPortfolio".

Reconnect to http://localhost:8080/dash.html. It should now look like:



My Portfolio

	Buy	Hold	Sell
APLE			●
GGLE		●	
IBMZ	●		●
YAHU	●		
NINT	●		
ADOB	●	●	●

Endiago & Track Inc

Time	Value
21:38	416.38
21:38	426.93
21:38	442.73
21:38	450.93
21:37	460.93

Gondor King United

Time	Value
21:38	480.21
21:38	469.56
21:38	461.56
21:38	486.01
21:37	495.71

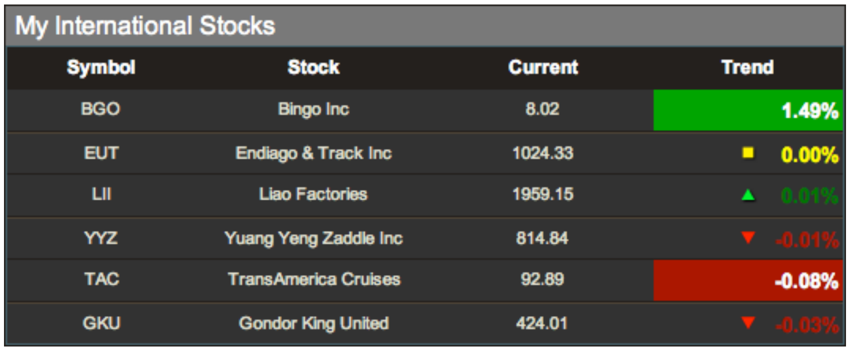
Liao Factories

Time	Value
21:38	1813.5
21:38	1791.95
21:38	1822.35
21:38	1830.2
21:37	1830.85

Bingo Inc

Time	Value
21:38	134.18
21:38	138.68
21:38	139.73
21:38	156.78
21:37	153.28

Example 3: Bonus Indicators



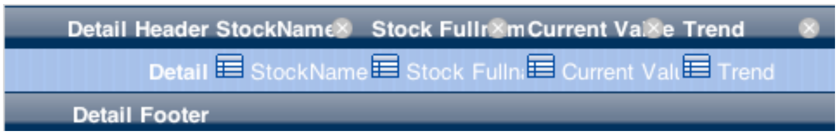
My International Stocks

Symbol	Stock	Current	Trend
BGO	Bingo Inc	8.02	1.49%
EUT	Endiago & Track Inc	1024.33	■ 0.00%
LII	Liao Factories	1959.15	▲ 0.01%
YYZ	Yuang Yeng Zaddle Inc	814.84	▼ -0.01%
TAC	TransAmerica Cruises	92.89	-0.08%
GKU	Gondor King United	424.01	▼ -0.03%

Here we are going to build a Dashboard that lists some stocks along with their Current value and Trend (% of increase or decrease from previous value). One feature we would like to highlight in this example is a few of the Bonus Indicators via Developer Hook Dax_DevHook_Setindicators.

To begin, click the *Create* link to create a new Dashboard.

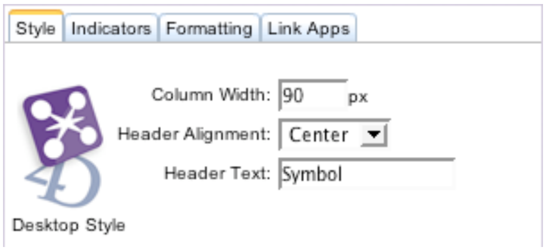
- Choose "Stocks" as the selection.
- Name the Dashboard "My International Stocks"
- Set the description as "Bonus Indicators"
- Do **not** select 'iPhone Support' for Platform Option. Click *Start Dashboard Editor* to begin.



- Drag *StockName, Stock Fullname, Current Value,* and *Trend* to the Editor.

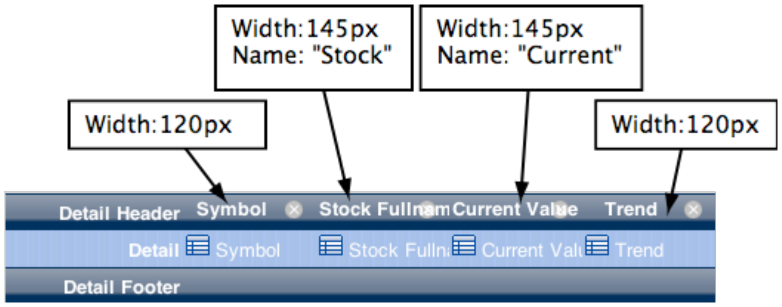
Customizing the Headers: Style

Here we will select a header cell. Then it will load the Property Inspector.



Column Width: 90 px
Header Alignment: Center
Header Text: Symbol

This is the *Property Inspector*. Under the *Style Tab* it displays different ways the Header can be modified.



Select the *StockName* header cell.
- Set Column Width: 120px
- Set Header Text: “Symbol”

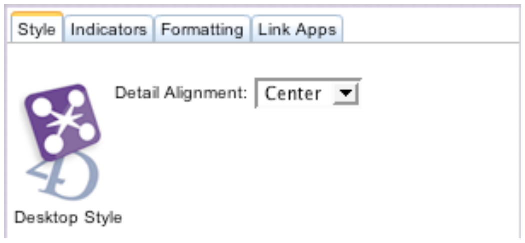
Select the *Stock Fullname* header cell.
- Set Column Width: 145px
- Set Header Text: “Stock”

Select the *Current Value* header cell.
- Set Column Width: 145px
- Set Header Text: “Current”

Select the *Trend* header cell.
- Set Column Width: 120px

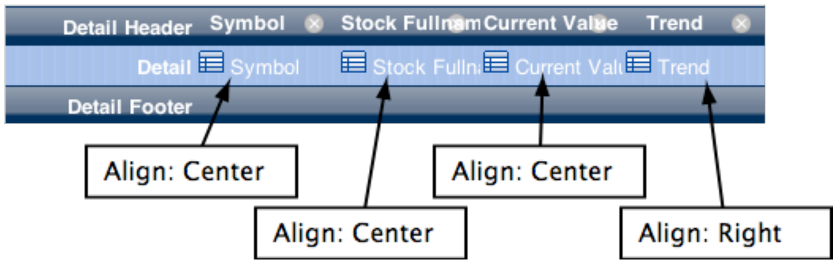
Customizing the Detail: Style

Select a Detail cell. Then it will load the Property Inspector.



Detail Alignment: Center

This is the *Property Inspector*. Under the *Style Tab* it displays different ways the Detail can be modified.



Select the *Symbol* detail cell.

- Set Detail Alignment: Center

Select the *Stock Fullname* detail cell.

- Set Detail Alignment: Center

Select the *Current Value* detail cell.

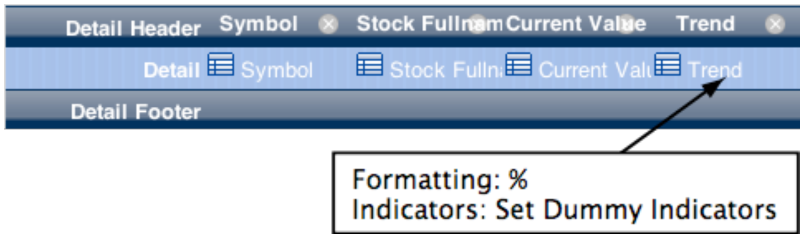
- Set Detail Alignment: Center

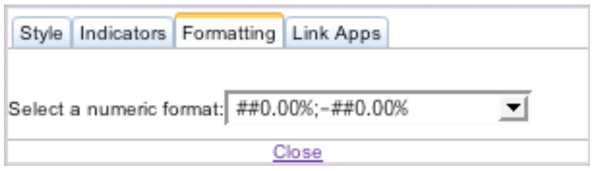
Select the *Trend* detail cell.

- Set Detail Alignment: Right

Customizing the Detail: Indicators and Formatting

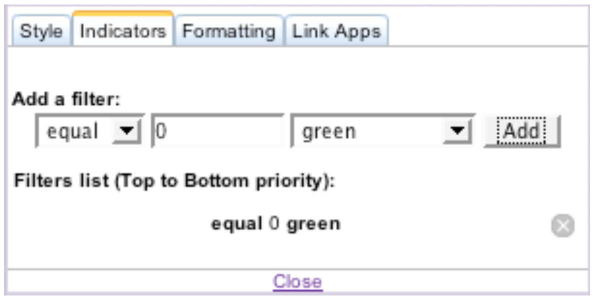
Here we continue to further customize the *Trend* cell using the *Property Inspector.* Hit the *Indicators* and the *Formatting* tabs to make the necessary changes.





Select a numeric format: ##0.00%;-##0.00%

- Under *Formatting,* set the format for %, as seen to the left.



Add a filter: equal 0 green

Filters list (Top to Bottom priority):

equal 0 green

- Under *Indicators*, add a filter where "equal" to "0" is "green".
- Hit the *Add* button so that it gets populated in the list.

This Indicator "equal 0 green" really has no meaning to us. Thus, it is a Dummy Indicator. We created this Indicator to flag the 4D database that the Indicators in this Dashboard can be overridden by the conditions we set in *Dax_DevHook_SetIndicators*.

Set the CSS Style to *Cocoa*. Save the Dashboard.

Add the following Case to the Case Statement in *Dax_DevHook_SetIndicators:*

```
: ($ReportName_t="My International Stocks")
```

```
C_LONGINT($foundat_l)
C_REAL($budget_r;$spent_r;$trend_r)
$foundat_l:=Find in array($fieldnames_atp->;"Trend")
       If ($foundat_l#-1)
              $trend_r:=Num($fieldvalue_atp->{$foundat_l})
              Case of
              : ($trend_r<-0.05)
                     $0:="redstrong"
              : ($trend_r>0.05)
                     $0:="greenstrong"
              : ($trend_r<0)
                     $0:="redtriangle"
              : ($trend_r>0)
                     $0:="greentriangle"
              Else
                     $0:="yellowsquare"
              End case
       End if
```

With the Dummy Indicators in place, *Dax_DevHook_SetIndicators* will get triggered.

The code in *Dax_DevHook_SetIndicators* is displayed and is further explained in the *Appendix* section.

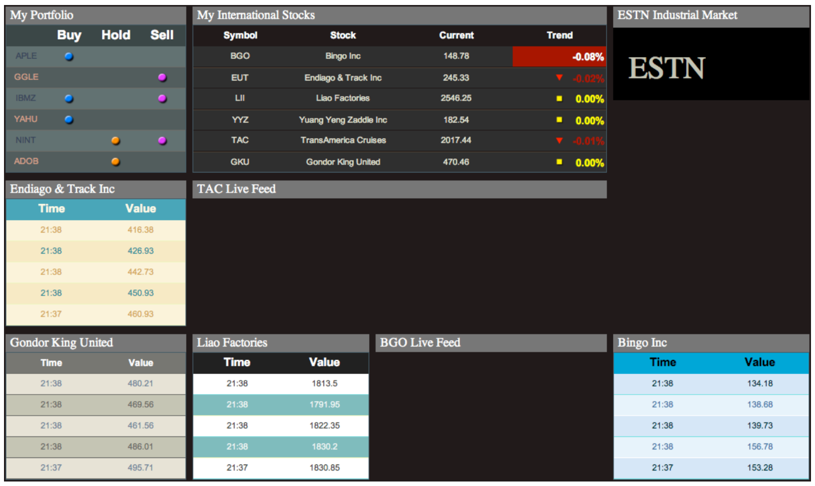
Dash.html

Add the following code to dash.html within *function onAfterInit()* in the <script> element:

```
InternationalStocks = new dashboardViewer("My International Stocks",
$("dashboardInternational"));
InternationalStocks.setRefreshInterval(10000);
InternationalStocks.viewport.showVerScrollbar(false);
```

This code will embed the Dashboard “My International Stocks” within the HTML page in the <div> “dashboardInternational”.

Reconnect to http://localhost:8080/dash.html. It should now look like:



My Portfolio

	Buy	Hold	Sell
APLE	●		
GGLE			●
IBMZ	●		●
YAHU	●		
NINT		●	●
ADOB		●	

My International Stocks

Symbol	Stock	Current	Trend
BGO	Bingo Inc	148.78	-0.08%
EUT	Endiago & Track Inc	245.33	▼ -0.02%
LII	Liao Factories	2546.25	■ 0.00%
YYZ	Yuang Yeng Zaddle Inc	182.54	■ 0.00%
TAC	TransAmerica Cruises	2017.44	▼ -0.01%
GKU	Gondor King United	470.46	■ 0.00%

ESTN Industrial Market

ESTN

Endiago & Track Inc

Time	Value
21:38	416.38
21:38	426.93
21:38	442.73
21:38	450.93
21:37	460.93

TAC Live Feed

Gondor King United

Time	Value
21:38	480.21
21:38	469.56
21:38	461.56
21:38	486.01
21:37	495.71

Liao Factories

Time	Value
21:38	1813.5
21:38	1791.95
21:38	1822.35
21:38	1830.2
21:37	1830.85

BGO Live Feed

Bingo Inc

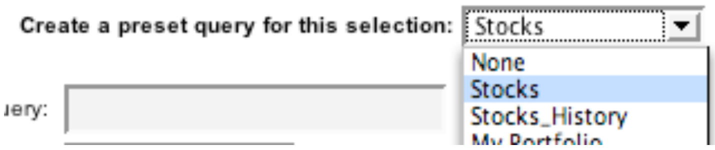
Time	Value
21:38	134.18
21:38	138.68
21:38	139.73
21:38	156.78
21:37	153.28

Example 3: Preset Queries (optional)

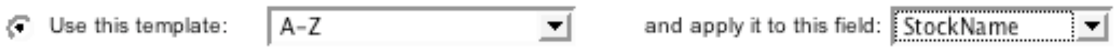
Here we will see how Preset Queries can be applied to dashboards. First, let’s create a Preset Query using the Query Manager.

Create the Preset Query

In the Control Panel, go to the Query Manager tab. Hit the Create link.



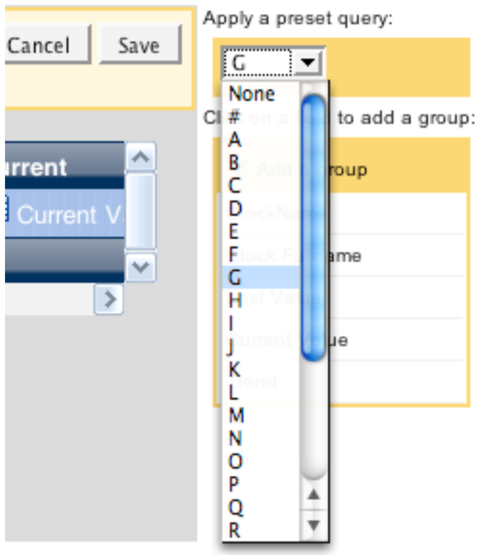
Choose the “Stocks” table as the selection.


Use this template: A-Z and apply it to this field: StockName

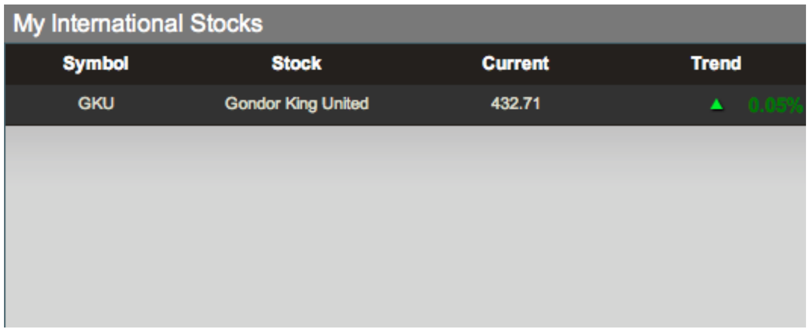
Choose the “A-Z” query template. Apply it to the “StockName” field. Hit the *Next* button to save the Preset Query. A Preset Query has now been created for the “StockName” field in the “Stocks” table. The “A-Z” template performs a query for each letter of the alphabet on the “StockName” field.

Apply the Query to the Dashboard

Open up the Dashboard created in Example 3.



Go to the Preset Query area and select “G”. This query will look up all stocks that begin with the letter ‘G’. Hit the *Save* button.


My International Stocks

Symbol	Stock	Current	Trend
GKU	Gondor King United	432.71	▲ 0.05%

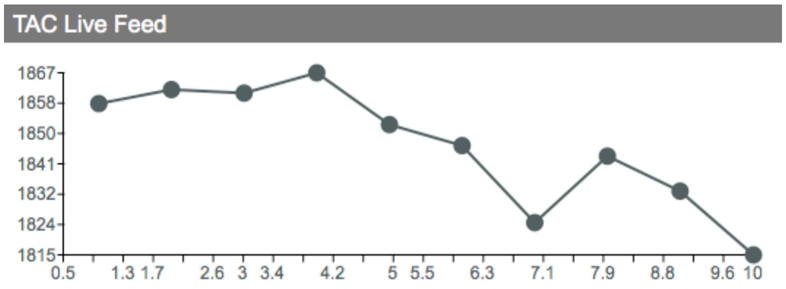
Reload the custom HTML page (dash.html) and the Dashboard for International Stocks should now only display the stock "Gondor King United".

The blank area is intentionally displayed to show the missing area that is usually occupied by the other stocks.

Example 4: Charts

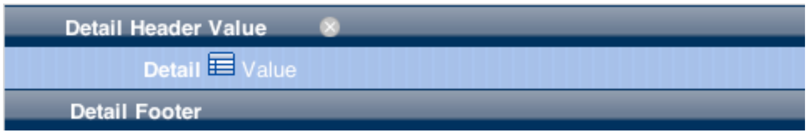
Here we will build charts. For Desktop Dashboards we must create a Dummy Dashboard that will trigger Developer Hook *Dax_DevHook_DefineChart*. It is called a Dummy Dashboard because the contents of the Dashboard have no effect on the Chart at all, but it must be created so that the 4D database can send Chart values and labels to the browser.

Example 4a: TAC Chart



To begin, click the *Create* link to create a new Dashboard.

- Choose "DCS_TAC" as the selection.
- Name the Dashboard "TAC Chart"
- Set the description as "TAC Chart"
- Do **not** select 'iPhone Support' for Platform Option. Click *Start Dashboard Editor* to begin.



- Drag *Value* to the Editor.

Make sure the checkbox for "Use Chart" is selected. Save the Dashboard.

Add the following Case to the Case Statement in *Dax_DevHook_DefineChart:*

```
:($SelectionName_t="DCS_TAC") & ($ReportName_t="TAC Chart")
$StockName_t:=Substring($SelectionName_t;5;3200)
$i:=Find in array(◊at_Stock;$StockName_t)


` Declare the arrays
ARRAY REAL(totals_ar;0)

QUERY([Stocks_History];[Stocks_History]StockName="TAC")
ORDER BY([Stocks_History];[Stocks_History]KeyTime;<)
If (Records in selection([Stocks_History])>500)
       REDUCE SELECTION([Stocks_History];10)
End if

SELECTION TO ARRAY([Stocks_History]CurrentValue;totals_ar)

vMax:=GetMaxValue (->totals_ar)

C_REAL($minYvalue_r;$maxYvalue_r)
ARRAY TEXT(xlabel_at;15)  `*** Label on X-Axis (Must use process_array_variable
- TEXT)
ARRAY TEXT(ylabel_at;5)  `*** Label on Y-Axis (Must use process_array_variable
- TEXT)

xlabel_at{1}:=String(Current time-†09:00:00†;HH MM )
xlabel_at{2}:=String(Current time-†06:00:00†;HH MM )
xlabel_at{3}:=String(Current time-†03:00:00†;HH MM )
xlabel_at{4}:=String(Current time;HH MM )

ylabel_at{1}:="0"
ylabel_at{2}:=String(Int(vMax*0.25))
ylabel_at{3}:=String(Int(vMax*0.5))
ylabel_at{4}:=String(Int(vMax*0.75))
ylabel_at{5}:=String(vMax)
$minYvalue_r:=0  `*** It should correspond to the minimum value in ylabel_at
$maxYvalue_r:=vMax  `*** It should correspond to the maximum value in ylabel_at
```

To reiterate, the selection and the contents of the Dashboard have no effect on the Chart that we are going to build. The most important thing to keep in mind is to validate the name of the Dashboard (in this case "TAC Chart") in order to trigger *Dax_DevHook_DefineChart,* which will populate and define the Chart.

Even though the Dummy Dashboard has no effect on the Chart, it is best to keep the Dashboard simple. The Dashboard is still sent with the Chart to the browser, thus it is best to keep the traffic light by making sure the selection is as small as possible.

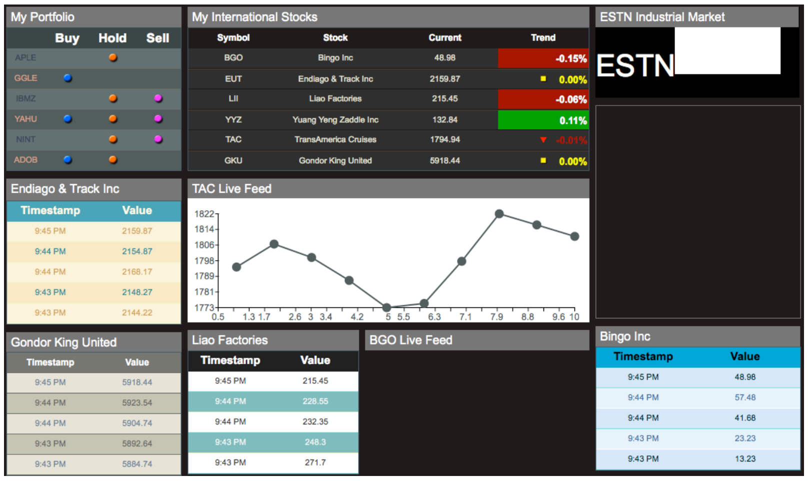
Dash.html

Add the following code to dash.html within *function dax_loginSuccess()* in the <script> element:

```
var Chart_TAC = new dax_chartViewer('TAC Chart', $('dashboardTACChart'), {
type   : dax_chartViewer.TYPE_LINE,

/* optional   */
color                        : '#414E4F',
hoverBackgroundColor         : '#ccffcc',
hoverTextColor               : '#cdef01',
labelColor                   : '#414E4F'
});
Chart_TAC.setRefreshInterval(13);
```

This code will embed the Dashboard “TAC Chart” within the HTML page in the <div> “dashboardTACChart”.

Reconnect to http://localhost:8080/dash.html. It should now look like:



My Portfolio

	Buy	Hold	Sell
APLE		●	
GGLE	●		
IBMZ		●	●
YAHU	●	●	●
NINT		●	●
ADOB	●	●	

My International Stocks

Symbol	Stock	Current	Trend
BGO	Bingo Inc	48.98	-0.15%
EUT	Endiago & Track Inc	2159.87	■ 0.00%
LII	Liao Factories	215.45	-0.06%
YYZ	Yuang Yeng Zaddle Inc	132.84	0.11%
TAC	TransAmerica Cruises	1794.94	▼ -0.01%
GKU	Gondor King United	5918.44	■ 0.00%

ESTN Industrial Market

ESTN

Endiago & Track Inc

Timestamp	Value
9:45 PM	2159.87
9:44 PM	2154.87
9:44 PM	2168.17
9:43 PM	2148.27
9:43 PM	2144.22

TAC Live Feed

Gondor King United

Timestamp	Value
9:45 PM	5918.44
9:44 PM	5923.54
9:44 PM	5904.74
9:43 PM	5892.64
9:43 PM	5884.74

Liao Factories

Timestamp	Value
9:45 PM	215.45
9:44 PM	228.55
9:44 PM	232.35
9:43 PM	248.3
9:43 PM	271.7

BGO Live Feed

Bingo Inc

Timestamp	Value
9:45 PM	48.98
9:44 PM	57.48
9:44 PM	41.68
9:43 PM	23.23
9:43 PM	13.23

Example 4a: Image URL Chart (Optional)

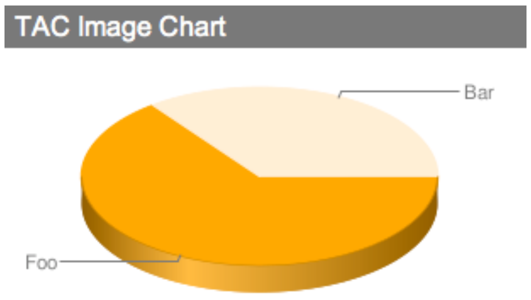
Alternatively, you can modify the case statement in *Dax_DevHook_DefineChart* to show an Image URL chart instead of the Interactive chart created in “Example 4a: TAC Chart.” Replace the case statement with the following:

```
: ($SelectionName_t="DCS_TAC") & ($ReportName_t="TAC Chart")

C_TEXT($url)
ARRAY TEXT($urlArray;2)

$urlArray{1}:="http://chart.apis.google.com/chart?cht=p3&chd=t:90,49&chs=300x15
0&chl=Foo|Bar"
DAX_Dev_SetDashboardImageURL (->$urlArray)
```

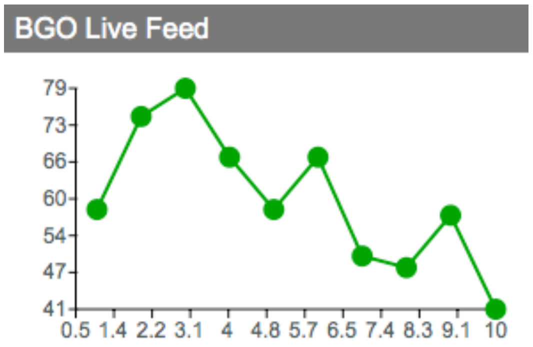
Then replace the JavaScript code used in “Example 4a: TAC Chart” with:

```
Chart_TAC = new chartViewer("TAC Chart", $('dashboardTACChart'));
Chart_TAC.setRefreshInterval(5200);
```

This is a proof of concept way of seeing how an image URL can be displayed in a dashboard.

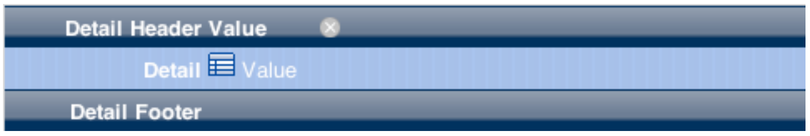


Example 4b: BGO Chart



To begin, click the *Create* link to create a new Dashboard.

- Choose “DCS_BGO” as the selection.
- Name the Dashboard “BGO Chart”
- Set the description as “BGO Chart”
- Do **not** select ‘iPhone Support’ for Platform Option. Click *Start Dashboard Editor* to begin.



- Drag *Value* to the Editor.

Make sure the checkbox for “Use Chart” is selected. Save the Dashboard.

Add the following Case to the Case Statement in *Dax_DevHook_DefineChart:*

```
: ($SelectionName_t="DCS_BGO") & ($ReportName_t="BGO Chart")
$StockName_t:=Substring($SelectionName_t;5;3200)
$i:=Find in array(◊at_Stock;$StockName_t)

` Declare the arrays
ARRAY REAL(totals_ar;0)

QUERY([Stocks_History];[Stocks_History]StockName="BGO")
ORDER BY([Stocks_History];[Stocks_History]KeyTime;<)
If (Records in selection([Stocks_History])>500)
      REDUCE SELECTION([Stocks_History];10)
End if
SELECTION TO ARRAY([Stocks_History]CurrentValue;totals_ar)

vMax:=GetMaxValue (->totals_ar)

C_REAL($minYvalue_r;$maxYvalue_r)
ARRAY TEXT(xlabel_at;10)  `*** Label on X-Axis (Must use process_array_variable
- TEXT)
ARRAY TEXT(ylabel_at;5)  `*** Label on Y-Axis (Must use process_array_variable
- TEXT)

xlabel_at{1}:=String(Current time-†24:00:00†;HH MM )
xlabel_at{2}:=String(Current time-†16:00:00†;HH MM )
xlabel_at{3}:=String(Current time-†08:00:00†;HH MM )
```

```
xlabel_at{4}:=String(Current time;HH MM )

xlabel_at{1}:=String(Current time-†09:00:00†;HH MM )
xlabel_at{2}:=String(Current time-†06:00:00†;HH MM )
xlabel_at{3}:=String(Current time-†03:00:00†;HH MM )
xlabel_at{4}:=String(Current time;HH MM )

ylabel_at{1}:="0"
ylabel_at{2}:=String(Int(vMax*0.25))
ylabel_at{3}:=String(Int(vMax*0.5))
ylabel_at{4}:=String(Int(vMax*0.75))
ylabel_at{5}:=String(vMax)
$minYvalue_r:=0  `*** It should correspond to the minimum value in ylabel_at
$maxYvalue_r:=vMax  `*** It should correspond to the maximum value in ylabel_at

DAX_Dev_SetDashboardChart (->xlabel_at;->ylabel_at;-
>totals_ar;$minYvalue_r;$maxYvalue_r)
```

To reiterate, the selection and the contents of the Dashboard have no effect on the Chart that we are going to build. The most important thing to keep in mind is to validate the name of the Dashboard (in this case “BGO Chart”) in order to trigger *Dax_DevHook_DefineChart,* which will populate and define the Chart.

Even though the Dummy Dashboard has no effect on the Chart, it is best to keep the Dashboard simple. The Dashboard is still sent with the Chart to the browser, thus it is best to keep the traffic light by making sure the selection is as small as possible.

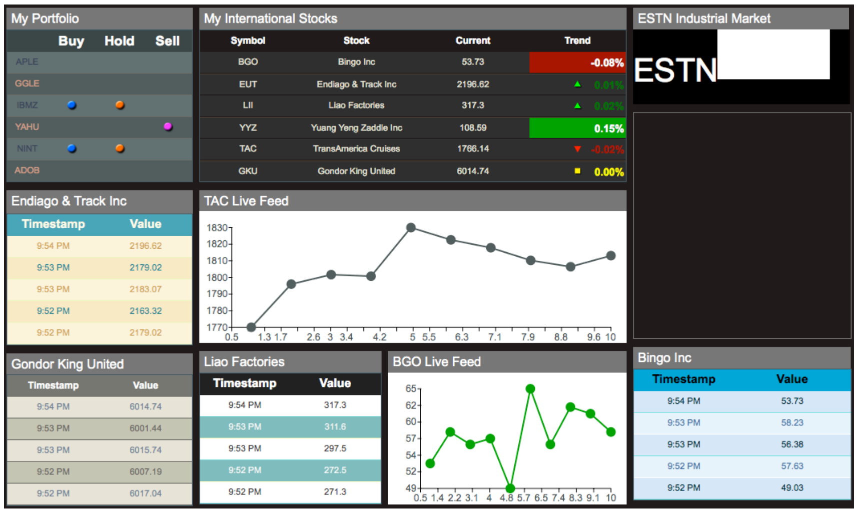
Dash.html

Add the following code to dash.html within *function dax_loginSuccess()* in the <script> element:

```
var Chart_BGO = new dax_chartViewer('BGO Chart', $('dashboardBGOChart'), {
type   : dax_chartViewer.TYPE_LINE,

color                        : '#009900',
hoverBackgroundColor         : '#ccffcc',
hoverTextColor               : '#cdef01',
labelColor                   : '#414E4F'

});
Chart_BGO.setRefreshInterval(11);
```

This code will embed the Dashboard “BGO Chart” within the HTML page in the <div> “dashboardBGOChart”.

Reconnect to http://localhost:8080/dash.html. It should now look like:


My Portfolio

	Buy	Hold	Sell
APLE			
GGLE			
IBMZ	●	●	
YAHU			●
NINT	●	●	
ADOB			

My International Stocks

Symbol	Stock	Current	Trend
BGO	Bingo Inc	53.73	-0.08%
EUT	Endiago & Track Inc	2196.62	▲ 0.01%
LII	Liao Factories	317.3	▲ 0.02%
YYZ	Yuang Yeng Zaddle Inc	108.59	0.15%
TAC	TransAmerica Cruises	1766.14	▼ -0.02%
GKU	Gondor King United	6014.74	■ 0.00%

ESTN Industrial Market

ESTN

Endiago & Track Inc

Timestamp	Value
9:54 PM	2196.62
9:53 PM	2179.02
9:53 PM	2183.07
9:52 PM	2163.32
9:52 PM	2179.02

TAC Live Feed

Gondor King United

Timestamp	Value
9:54 PM	6014.74
9:53 PM	6001.44
9:53 PM	6015.74
9:52 PM	6007.19
9:52 PM	6017.04

Liao Factories

Timestamp	Value
9:54 PM	317.3
9:53 PM	311.6
9:53 PM	297.5
9:52 PM	272.5
9:52 PM	271.3

BGO Live Feed

Bingo Inc

Timestamp	Value
9:54 PM	53.73
9:53 PM	58.23
9:53 PM	56.38
9:52 PM	57.63
9:52 PM	49.03

Example 4b: SVG Chart (Optional)

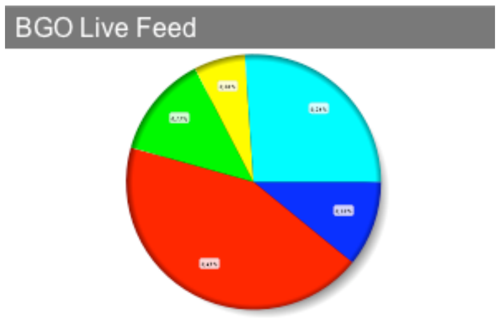
Modify the case statement in *Dax_DevHook_DefineChart* to show an SVG URL chart instead of the Canvas chart created in “Example 4b: BGO Chart.” Replace the case statement with the following:

```
: ($SelectionName_t="DCS_BGO") & ($ReportName_t="BGO Chart")

C_BLOB($svgblob1)
C_LONGINT($chartarea_l;$i)
C_PICTURE($picture)

ARRAY STRING(40;$label_at;1)
ARRAY REAL($value_ar1;1)
ARRAY REAL($value_ar2;1)
ARRAY REAL($value_ar3;1)
ARRAY REAL($value_ar4;1)
ARRAY REAL($value_ar5;1)

$value_ar1{1}:=50
$value_ar2{1}:=200
$value_ar3{1}:=60
$value_ar4{1}:=30
$value_ar5{1}:=120

GRAPH($chart_l;7;$label_at;$value_ar1;$value_ar2;$value_ar3;$value_ar4;$value_a
r5)
GRAPH SETTINGS($chart_l;0;0;0;2;True;False;False;"";"BGO Inc";" Endiago & Track
Inc";"Liao Factories";"TransAmerica Cruises";"Gondor King United")
WRITE PICTURE FILE("mysvg.svg";$picture;".svg")
DOCUMENT TO BLOB(document;$svgblob1)

DAX_Dev_SetDashboardSVG (->$svgblob1)
```

Then replace the JavaScript code used in “Example 4b: BGO Chart.”” with:

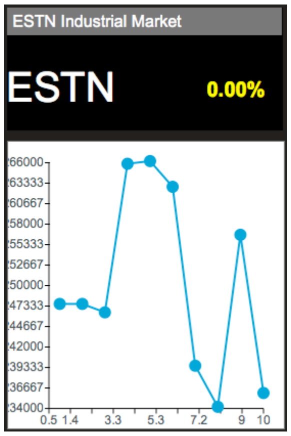
```
Chart_BGO = new chartViewer("BGO Chart", $('dashboardBGOChart'));
Chart_BGO.setRefreshInterval(5010);
```

This is a proof of concept way of seeing how an SVG URL can be displayed in a dashboard.



Example 5: Complementary Dashboard

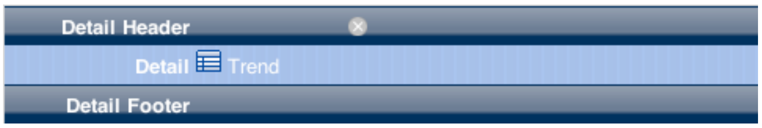
Here we are going to build two Dashboards that complement each other on the HTML page. They are a Dashboard and a Chart and they will display similar information for the same imaginary stock market, the "ESTN Industrial Market."



Example 5a: ESTN Dashboard

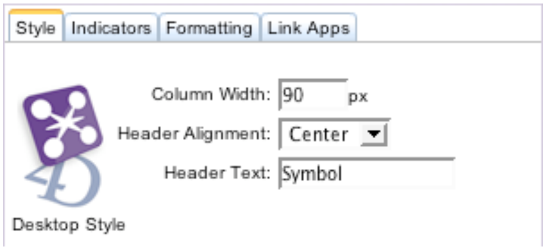
To begin, click the *Create* link to create a new Dashboard.

- Choose "Markets" as the selection.
- Name the Dashboard "ESTN Trend"
- Set the description as "ESTN Dashboard"
- Do **not** select 'iPhone Support' for Platform Option. Click *Start Dashboard Editor* to begin.



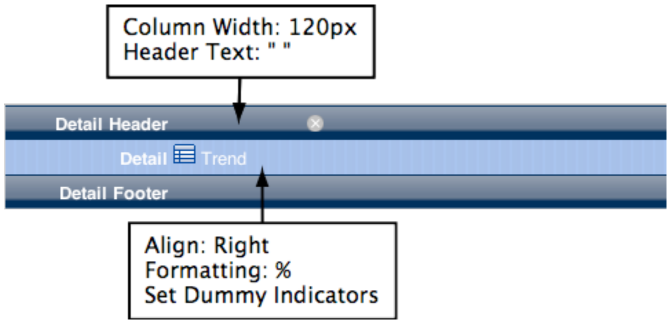
- Drag *Trend* to the Editor.

Now select the *Trend* detail header cell. It will load the Property Inspector.



Column Width: 90 px
Header Alignment: Center
Header Text: Symbol

This is the *Property Inspector*. Under the *Style Tab* it displays different ways the Header can be modified.

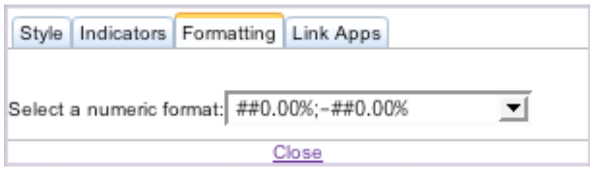


Select the *Trend* header cell.

- Set Column Width: 120px
- Header Text: " "

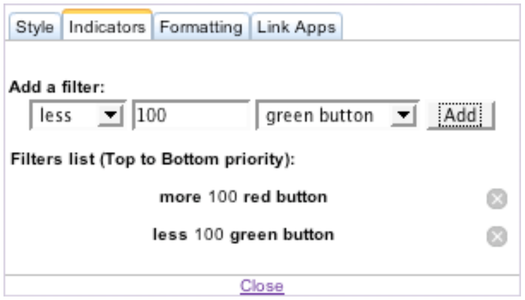
Select the *Trend* detail cell.

- Align: Right



Select a numeric format: ##0.00%;-##0.00%

- Under *Formatting,* set the format for %, as seen to the left.



Add a filter:
less 100 green button

Filters list (Top to Bottom priority):
more 100 red button
less 100 green button

- Under *Indicators*, add a filter where:
  - More 100 red button
  - Less 100 green button
- Hit the *Add* button so that it gets populated in the list.

This Indicator "equal 0 green" really has no meaning to us. Thus, it is a Dummy Indicator. We created this Indicator to flag the 4D database that the Indicators in this Dashboard can be overridden by the conditions we set in *Dax_DevHook_SetIndicators*.

Save the Dashboard.

Add the following Case to the Case Statement in *Dax_DevHook_SetIndicators:*

```
: ($ReportName_t="ESTN Trend")
C_LONGINT($foundat_l)
C_REAL($budget_r;$spent_r;$trend_r)
$foundat_l:=Find in array($fieldnames_atp->;"Trend")
```

```
If ($foundat_l#-1)
        $trend_r:=Num($fieldvalue_atp->{$foundat_l})
        Case of
        : ($trend_r<0)
                $0:="arrowdown"
        : ($trend_r>0)
                $0:="arrowup"
        Else
                $0:="noarrowyellow"
        End case
End if
```

With the Dummy Indicators in place, *Dax_DevHook_SetIndicators* will get triggered.

The code in *Dax_DevHook_SetIndicators* is displayed and is further explained in the *Appendix* section.

Dash.html

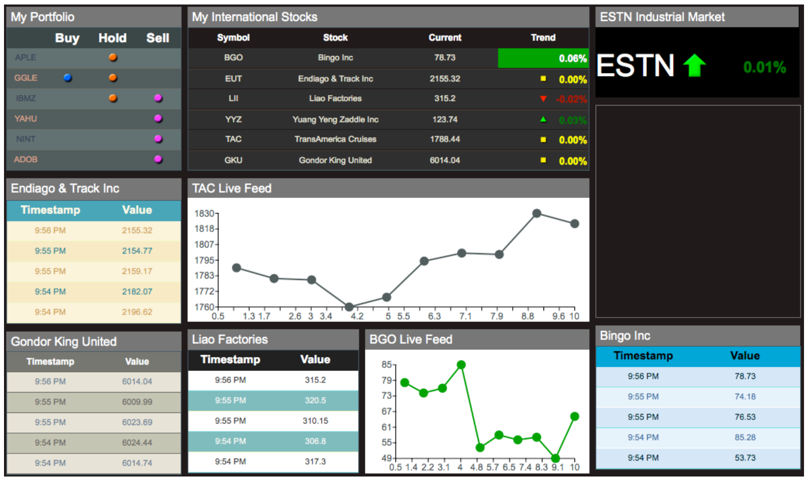
Add the following code to dash.html within *function dax_loginSuccess()* in the <script> element:

```
ESTN_Trend = new dashboardViewer("ESTN Trend", $("dashboardESTN"));
ESTN_Trend.viewport.showVerScrollbar(false);
ESTN_Trend.setRefreshInterval(20000);
ESTN_Trend.setRowHeightInPx(35);                          // set large row height
```

Note: Row height was increased so that font can be displayed larger. Also, the Indicators used for this example are large as well and would also benefit from the height increase.

This code will embed the Dashboard "ESTN Trend" within the HTML page in the <div> "dashboardESTN".

Reconnect to http://localhost:8080/dash.html. It should now look like:



My Portfolio

	Buy	Hold	Sell
APLE		●	
GGLE	●	●	
IBMZ		●	●
YAHU			●
NINT			●
ADOB			●

My International Stocks

Symbol	Stock	Current	Trend
BGO	Bingo Inc	78.73	0.06%
EUT	Endiago & Track Inc	2155.32	■ 0.00%
LII	Liao Factories	315.2	▼ -0.02%
YYZ	Yuang Yeng Zaddle Inc	123.74	▲ 0.03%
TAC	TransAmerica Cruises	1788.44	■ 0.00%
GKU	Gondor King United	6014.04	■ 0.00%

ESTN Industrial Market

ESTN ⬆ 0.01%

Endiago & Track Inc

Timestamp	Value
9:56 PM	2155.32
9:55 PM	2154.77
9:55 PM	2159.17
9:54 PM	2182.07
9:54 PM	2196.62

TAC Live Feed

Gondor King United

Timestamp	Value
9:56 PM	6014.04
9:55 PM	6009.99
9:55 PM	6023.69
9:54 PM	6024.44
9:54 PM	6014.74

Liao Factories

Timestamp	Value
9:56 PM	315.2
9:55 PM	320.5
9:55 PM	310.15
9:54 PM	306.8
9:54 PM	317.3

BGO Live Feed

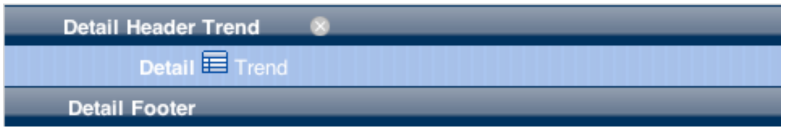
Bingo Inc

Timestamp	Value
9:56 PM	78.73
9:55 PM	74.18
9:55 PM	76.53
9:54 PM	85.28
9:54 PM	53.73

Example 5b: ESTN Chart

To begin, click the *Create* link to create a new Dashboard.

- Choose "Markets" as the selection.
- Name the Dashboard "ESTN Chart"
- Set the description as "ESTN Chart"
- Do **not** select 'iPhone Support' for Platform Option. Click *Start Dashboard Editor* to begin.



- Drag *Trend* to the Editor.

Make sure the checkbox for “Use Chart” is selected. Save the Dashboard.

Add the following Case to the Case Statement in *Dax_DevHook_DefineChart:*

```
: ($SelectionName_t="View_2") & ($ReportName_t="ESTN Chart")

` Declare the arrays
ARRAY REAL(totals_ar;0)

QUERY([Markets_History];[Markets_History]MarketName="ESTN")
$att:=String(Current date-1;7)+":"+String(Current time-†00:08:00†;HH MM SS )
QUERY SELECTION([Markets_History];[Markets_History]KeyTime>=$att)
ORDER BY([Markets_History];[Markets_History]KeyTime;<)
If (Records in selection([Markets_History])>500)
       REDUCE SELECTION([Markets_History];10)
End if
SELECTION TO ARRAY([Markets_History]CurrentValue;totals_ar)

$minYvalue_r:=Min([Markets_History]CurrentValue)
$maxYvalue_r:=Max([Markets_History]CurrentValue)

C_REAL($minYvalue_r;$maxYvalue_r)
ARRAY TEXT(xlabel_at;10)  `*** Label on X-Axis (Must use process_array_variable
- TEXT)
ARRAY TEXT(ylabel_at;5)  `*** Label on Y-Axis (Must use process_array_variable
- TEXT)

xlabel_at{1}:=String(Current time-†00:06:00†;HH MM )
xlabel_at{2}:=String(Current time-†00:04:00†;HH MM )
xlabel_at{3}:=String(Current time-†00:02:00†;HH MM )
xlabel_at{4}:=String(Current time;HH MM )

ylabel_at{1}:=String(Round(Int($minYvalue_r)/1000;0))+"K"
ylabel_at{2}:=""
ylabel_at{3}:=String(Round(Int(($maxYvalue_r+$minYvalue_r)*0.5)/1000;0))+"K"
ylabel_at{4}:=""
ylabel_at{5}:=String(Round(Int($maxYvalue_r)/1000;0))+"K"
`*** It should correspond to the minimum value in ylabel_at
`*** It should correspond to the maximum value in ylabel_at

DAX_Dev_SetDashboardChart (->xlabel_at;->ylabel_at;-
>totals_ar;$minYvalue_r;$maxYvalue_r)
```

To reiterate, the selection and the contents of the Dashboard have no effect on the Chart that we are going to build. The most important thing to keep in mind is to validate the name of the Dashboard (in this case “ESTN Chart”) in order to trigger *Dax_DevHook_DefineChart,* which will populate and define the Chart.

Even though the Dummy Dashboard has no effect on the Chart, it is best to keep the Dashboard simple. The Dashboard is still sent with the Chart to the browser, thus it is best to keep the traffic light by making sure the selection is as small as possible.

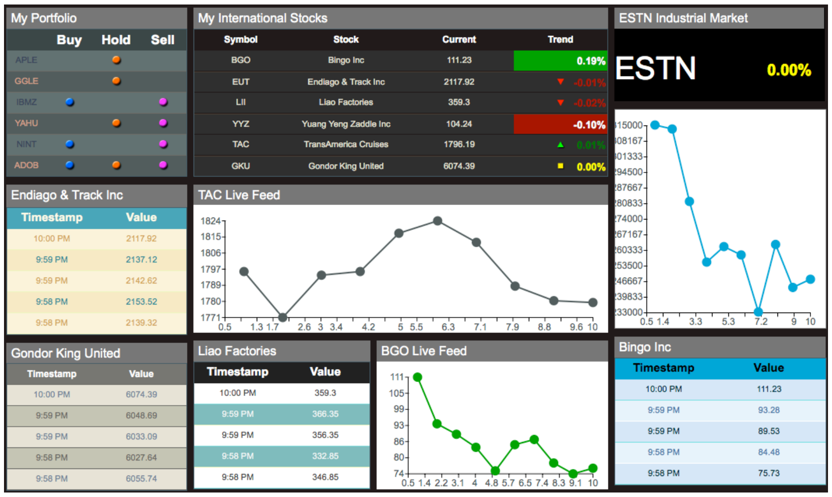
Dash.html

Add the following code to dash.html within *function dax_loginSuccess()* in the <script> element:

```
var Chart_ESTN = new dax_chartViewer('ESTN Chart', $('dashboardESTNChart'), {
type   : dax_chartViewer.TYPE_LINE,

color                      : '#0097D1',
hoverBackgroundColor       : '#ccffcc',
hoverTextColor             : '#cdef01',
labelColor                 : '#414E4F'

});
Chart_ESTN.setRefreshInterval(12);
```

This code will embed the Dashboard "ESTN Chart" within the HTML page in the <div> "dashboardESTNChart".

Reconnect to http://localhost:8080/dash.html. It should now look like:



My Portfolio

	Buy	Hold	Sell
APLE		●	
GGLE		●	
IBMZ	●		●
YAHU		●	●
NINT	●		●
ADOB	●	●	●

My International Stocks

Symbol	Stock	Current	Trend
BGO	Bingo Inc	111.23	0.19%
EUT	Endiago & Track Inc	2117.92	▼ -0.01%
LII	Liao Factories	359.3	▼ -0.02%
YYZ	Yuang Yeng Zaddle Inc	104.24	-0.10%
TAC	TransAmerica Cruises	1796.19	▲ 0.01%
GKU	Gondor King United	6074.39	■ 0.00%

ESTN Industrial Market

ESTN 0.00%

Endiago & Track Inc

Timestamp	Value
10:00 PM	2117.92
9:59 PM	2137.12
9:59 PM	2142.62
9:58 PM	2153.52
9:58 PM	2139.32

Gondor King United

Timestamp	Value
10:00 PM	6074.39
9:59 PM	6048.69
9:59 PM	6033.09
9:58 PM	6027.64
9:58 PM	6055.74

Liao Factories

Timestamp	Value
10:00 PM	359.3
9:59 PM	366.35
9:59 PM	356.35
9:58 PM	332.85
9:58 PM	346.85

Bingo Inc

Timestamp	Value
10:00 PM	111.23
9:59 PM	93.28
9:59 PM	89.53
9:58 PM	84.48
9:58 PM	75.73

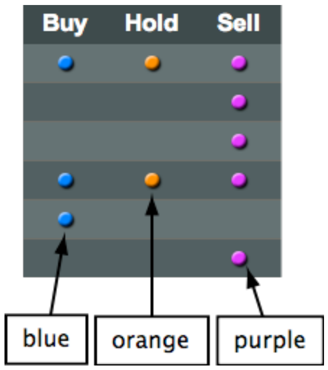
Limitations

Charts are limited to 500 array values for Desktop Dashboards. This is done to keep optimal performance and size for the XML files. Sending more than 500 values would comprise the integrity of the Chart.

Example 2: Flashing Indicators

For reference this is the code in *Dax_DevHook_SetIndicators* for *Example 2: Flashing Indicators*:

```
: ($ReportName_t="My Portfolio")
C_REAL($randomValue)
$randomValue:=Random%(2)
If ($randomValue=1)
       Case of
       : ($fieldnames_atp->{$fieldnum_l}="Buy")
              $0:="blue"
       : ($fieldnames_atp->{$fieldnum_l}="Hold")
              $0:="orange"
       Else
              $0:="purple"
       End case
Else
       $0:=""
End if
```



This block of code is called for each detail cell ofnthe table. A random value of “0” or “1” is generated. If it is “1”, then that cell will be populated with an Indicator. The color of that Indicator is determined by the column that cell is in.

Random values were generated intentionally for the purpose of demonstrating the look of flashing Indicators.

The order at which these conditions are listed is very important. The conditions are treated as Case statements. Thus, the first condition in the list that is evaluated as True exits the entire list and no longer evaluates the remaining conditions.

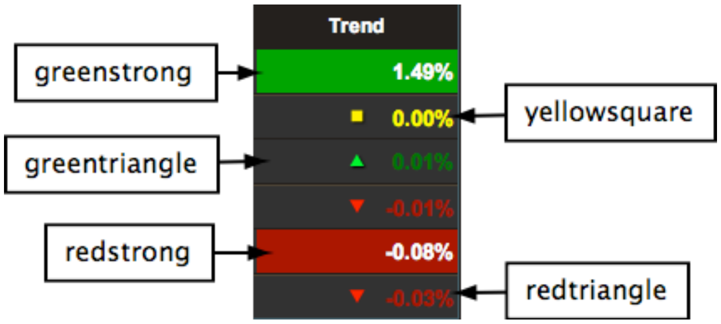
Example 3: Bonus Indicators

For reference this is the code in *Dax_DevHook_SetIndicators* for *Example 3: Bonus Indicators*:

```
: ($ReportName_t="My International Stocks")

C_LONGINT($foundat_l)
C_REAL($budget_r;$spent_r;$trend_r)
```

```
$foundat_l:=Find in array($fieldnames_atp->;"Trend")
        If ($foundat_l#-1)
                $trend_r:=Num($fieldvalue_atp->{$foundat_l})
                Case of
                : ($trend_r<-0.05)
                        $0:="redstrong"
                : ($trend_r>0.05)
                        $0:="greenstrong"
                : ($trend_r<0)
                        $0:="redtriangle"
                : ($trend_r>0)
                        $0:="greentriangle"
                Else
                        $0:="yellowsquare"
                End case
        End if
```

Please note that we use a case statement to trap when this specific Dashboard is called. Thus, it is very important that you name the Dashboard “My International Stocks” as mentioned so that this code can take effect.



We used case statements to indicate different types of values for *Trend*.

- Where *Trend* < -0.05% use “redstrong”.
- Where *Trend* > 0.05% use “greenstrong”.
- Where *Trend* < 0% use “redtriangle”.
- Where *Trend* < 0% use “greentriangle”.
- Else use “yellowsquare”.

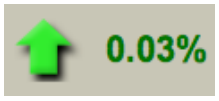
The order at which these conditions are listed is very important. The conditions are treated as Case statements. Thus, the first condition in the list that is evaluated as True exits the entire list and no longer evaluates the remaining conditions.

Example 5a: ESTN Dashboard

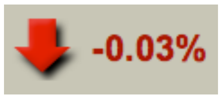
For reference this is the code in *Dax_DevHook_SetIndicators* for *Example 5a: ESTN Dashboard*:

```
: ($ReportName_t="ESTN Trend")
C_LONGINT($foundat_l)
C_REAL($budget_r;$spent_r;$trend_r)
$foundat_l:=Find in array($fieldnames_atp->;"Trend")
If ($foundat_l#-1)
        $trend_r:=Num($fieldvalue_atp->{$foundat_l})
        Case of
        : ($trend_r<0)
                $0:="arrowdown"
        : ($trend_r>0)
                $0:="arrowup"
        Else
                $0:="noarrowyellow"
        End case
End if
```

Please note that we use a case statement to trap when this specific Dashboard is called. Thus, it is very important that you name the Dashboard “ESTN Trend” as mentioned so that this code can take effect.



arrowup



arrowdown

We set case statements in the following manner:

- When the *Trend* value is < “0”, use “arrowdown”
- When the Trend value is > “0”, use “arrowup”
- Else, use “noarrowyellow”

The order at which these conditions are listed is very important. The conditions are treated as Case statements. Thus, the first condition in the list that is evaluated as True exits the entire list and no longer evaluates the remaining conditions.