**VizAlerts: Data Driven Alerting for Tableau**

**User Guide**

Table of Contents

[What is VizAlerts? 4](#_Toc447637982)

[What can VizAlerts do? 4](#_Toc447637983)

[How do I use it? 4](#_Toc447637984)

[Simple Alerts 6](#_Toc447637985)

[Advanced Alerts 7](#_Toc447637986)

[Important Info About Advanced Alerts 8](#_Toc447637987)

[Advanced Alerts Walkthrough #1 – Send the Trigger Viz 8](#_Toc447637988)

[Advanced Alerts Walkthrough #2 – Send a Custom Viz 13](#_Toc447637989)

[Additional Advanced Alert Features 18](#_Toc447637990)

[Custom Body, Header, and Footer 18](#_Toc447637991)

[Content References 18](#_Toc447637992)

[Custom View References: Getting Almost Any View From Tableau Server 19](#_Toc447637993)

[URL Parameters: ?myParam=myValue 19](#_Toc447637994)

[Custom Filenames for Appended Attachments: |filename 20](#_Toc447637995)

[Hyperlink Text References: VIZ\_LINK() 21](#_Toc447637996)

[Hyperlinked Inline Images: |vizlink 21](#_Toc447637997)

[Merge Multiple PDFs : |mergepdf 21](#_Toc447637998)

[Appended (non-inline) Attachments 22](#_Toc447637999)

[Consolidated Emails 23](#_Toc447638000)

[Sorting Consolidated Emails 24](#_Toc447638001)

[Headers and Footers in Consolidated Emails 24](#_Toc447638002)

[Merged PDFs 24](#_Toc447638003)

[SMS Text Message Alerts 26](#_Toc447638004)

[SMS Alert Configuration 27](#_Toc447638005)

[Supported Mobile Number Formats 28](#_Toc447638006)

[Important Info about SMS Advanced Alerts 28](#_Toc447638007)

[Content References for SMS Advanced Alerts 29](#_Toc447638008)

[Advanced Alert Use Cases 29](#_Toc447638009)

[Tableau Server Monitoring 29](#_Toc447638010)

[Extract Failure to Refresh Notifications 29](#_Toc447638011)

[Bulk Mailing 29](#_Toc447638012)

[Consolidated Emails 30](#_Toc447638013)

[Dashboard with Hidden Embedded Alert 30](#_Toc447638014)

[Testing an Alert 31](#_Toc447638015)

[FAQ / Common Issues / Troubleshooting 32](#_Toc447638016)

[I got a failure email instead of an alert! 32](#_Toc447638017)

[I’m getting Simple Alerts when I should be getting Advanced Alerts. 33](#_Toc447638018)

[What if I don’t get an alert email when I expect one? 33](#_Toc447638019)

[How can I avoid getting continual alerts for the same data? 33](#_Toc447638020)

[I’m getting my Alerts late. 33](#_Toc447638021)

[There is no default footer on my emails. 33](#_Toc447638022)

[My CSV is Bad! 33](#_Toc447638023)

[Wrong worksheet’s data 34](#_Toc447638024)

[Wrong data structure 34](#_Toc447638025)

[My Emails aren’t Consolidating. 34](#_Toc447638026)

[My PDFs aren’t Merging. 35](#_Toc447638027)

[I’m getting too few emails, it looks like they are Over-Consolidating. 35](#_Toc447638028)

[Inline and/or Appended Attachments aren’t showing up in my email. 36](#_Toc447638029)

[Getting VizAlerts Help 36](#_Toc447638030)

[Contributing to VizAlerts 36](#_Toc447638031)

# What is VizAlerts?

Since the dawn of time, humans have brought up the needs that they have for various forms of alerting driven by Tableau Server. Some want to know when extract refreshes fail. Others want to know when they succeed. Salespeople want to know when they hit their quota. Still others simply want to make a few tweaks to the existing Subscription functionality, say, by being able to send a report image to a distribution list that includes recipients who aren’t on Tableau Server.

VizAlerts is an email automation platform intended to seamlessly integrate with Tableau Server. The idea behind it is that anyone should be able to easily build, share, and customize pretty much any email automation based on their own Tableau Server viz data.

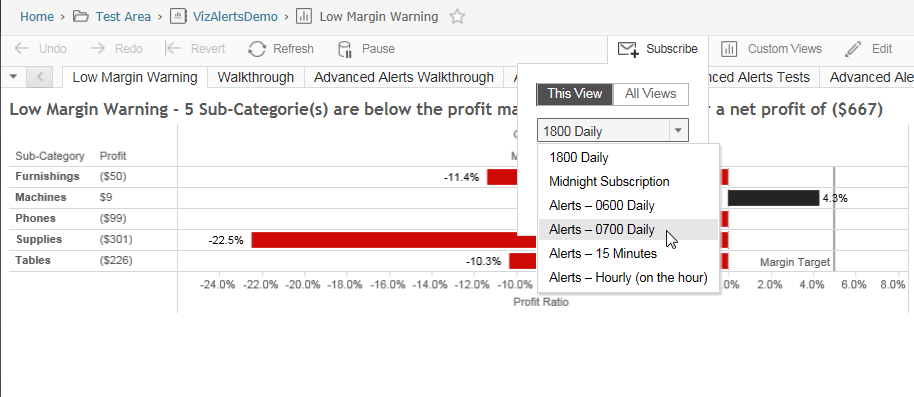
## What can VizAlerts do?

VizAlerts has been designed to support many use cases:

* Sending notifications to subscribers when a condition has been met (or not!) like extract refresh failures, meeting or missing target thresholds, etc.
* Halting emails from being sent to the group unless the data is up to date (while the workbook owner does get an email that the data isn’t up to date).
* Notify data owners when data is corrupt in some way (extraneous values, too many Null values, too large a change, etc.)
* Sending a one line email notification that could be forwarded through an email-to-sms gateway such as [########@txt.att.net](mailto:########@txt.att.net) or a messaging provider such as Twilio.
* Batch reporting to distribution lists of non-Tableau users, for example emailing a weekly operations report to front-line staff who aren’t on Tableau.
* Combining images and text into an HTML email for a more customized, professional look.
* Merge multiple views into a single consolidated PDF, for example sending a company overview and per-region detail.
* Blend views from separate workbooks in the same email, for example a view from the operations dashboard and a view from the finance dashboard.
* Bursting reports, for example sending to a manager a dashboard for each of her direct reports.
* Whatever combinations of the above you can imagine!

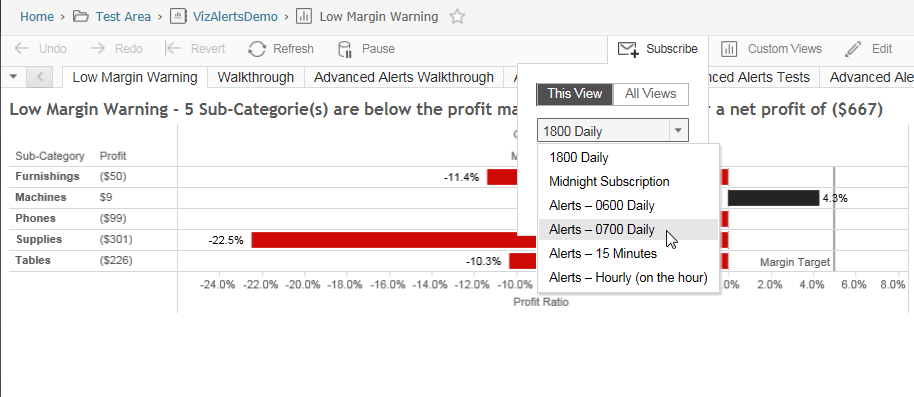
# How do I use it?

An alert is defined by two things: A view published to Tableau Server (called the *trigger view*), and a subscription to that view on a special, *disabled* Subscription schedule. Let’s take a look at a simple example (you can follow along yourself if you publish the included [VizAlerts install forlder]\demo\VizAlertsDemo.twb file to your Tableau Server):



The *view*’s job is to define the criteria for triggering the alert and the content displayed within it. The scheduled *subscription’s* job is to define at what times and how frequently the view is checked to see if an alert should be sent.

VizAlerts functions off of one simple triggering mechanism: **The presence of data in your viz.** If, when VizAlerts is scheduled to check your viz, it finds that **data is present**, it will trigger the alert. If it finds no data, then nothing happens. Thus, the criteria for when you receive an alert depends solely on what data you’ve connected to and how you’ve filtered it. Because of this you have a lot of flexibility in defining your alert.



There’s data so we’ll get an alert

At 7am each day

Okay, we’ve got the basic premise down, now we can get to the fun stuff. Namely, what does an alert look like when it hits your inbox? Who does it get sent to?

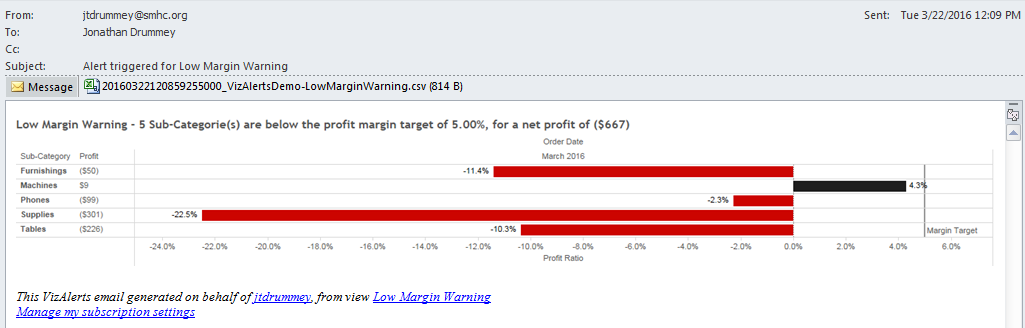
The answer, just like the answer to every good question, is: “It depends!”

There are two modes of email automation supported by VizAlerts: **Simple Alerts** and **Advanced Alerts**. Simple Alerts are basic notifications that will send an image of the viz plus the viz CSV, Advanced Alerts have a huge range of customizations.

# Simple Alerts

True to its name, this is the simplest and easiest form of alerting. Anyone with “view” permissions to a View on Tableau Server can subscribe to it on an Alert schedule, and it will be delivered to them via email if and only if it is triggered. This means that I could send the URL to the view above to all my co-workers and tell them that if they want to get silly text alerts sent to them at 6AM like I get, they only need subscribe to it on the same schedule.

And here is what the Simple Alert they’d get looks like as an email:



The PNG rendering of the View itself is embedded in the email, and the data from the View is attached in CSV format. The image is linked back to the View on Tableau Server, just as in a Subscription. The CSV is simply there to provide additional information that might not be exposed visually, and to provide a static historical record of what caused the alert to be triggered (by the time you open your email and click the link to investigate, the data behind the View may have changed).

**So wait**—what if someone picked a View that wasn’t designed to be an Alert, and subscribed to it on a custom Alerts schedule? What would happen?

The answer is: A Subscription, effectively. Except they’d also get the CSV data attached to the email.

There are a few other important things to know about Simple Alerts:

* VizAlerts will **impersonate** the Subscriber when it checks for viz data from the View they subscribed to. Therefore, any user filters set up that are based on who is viewing the viz will apply as if the Subscriber had pulled the view up manually in a browser. This provides an easy way for one person to author an Alert that will be triggered at different times depending on who subscribed to it, if so desired.
* If a user creates a Customized View from your original Alert view, they can subscribe to it if they like, and VizAlerts will process it just like any other Alert. This provides additional flexibility if individuals want to tweak specific criteria to trigger their alerts.

All this Simple Alert stuff is well and good, but maybe “simple” isn’t a word that describes you. Maybe you’re more the *power user* type. In that case, please proceed to:

# Advanced Alerts

Advanced Alerts are triggered in the same way that Simple Alerts are, but they allow you extremely granular control over **what** you send to **whom**. They work on the following principle:

“For every row of data in the trigger viz, send an email or SMS message[[1]](#footnote-2) with properties and content based on the data in each field”

What tells VizAlerts that given alert is an Advanced Alert (and not a Simple Alert) is the presence of a field named **“ Email Action \*”** in the viz with a value of 1 for Email or 2 for SMS. In addition the properties of an email sent by an Advanced Alert are generated by **specific fields** in the View you subscribe to. These fields can be from raw data or calculated fields that are dynamically generated from your data (ooh, the possibilities!). Here’s the list of fields and values for Email Advanced Alerts:

|  |  |  |
| --- | --- | --- |
| **Field Name (with leading space)** | **Required?** | **Structure & Description** |
| Email Action \* | Yes | Value = 1, tells VizAlerts that this is an Advanced Alert (and not a simple alert) |
| Email To \* | Yes | Email address(es) separated by commas, semicolons, or spaces |
| Email From ~ |  | Email address |
| Email CC ~ |  | Email address(es) |
| Email BCC ~ |  | Email address(es) |
| Email Subject \* | Yes | Text |
| Email Header ~ |  | HTML for optional additional header text & inline images |
| Email Body \* | Yes | HTML for body, can be left blank/Null |
| Email Footer ~ |  | HTML for optional additional footer text & inline images |
| Email Attachment ~ |  | List of content references to appended (non-inline) attachments |
| Email Consolidate ~ |  | No value, just needs to be present in viz to trigger consolidation |
| Email Sort Order ~ |  | Any alphanumeric values, used for sorting consolidated emails |

The naming conventions for the fields are a little strange, but there’s a method to their madness. All the fields are prepended with the word “ Email” (**note the preceding space**), mostly for uniqueness and easy sorting in the Dimensions pane.

* Asterisks (\*) represent fields **required** for an Advanced Alert,
* Tildes (~) represent **optional** fields.

But the most important part to know about setting up the fields for Advanced Alerts is that the field names **must** **match these exactly** to be used for email properties. All other fields with different names in your trigger viz will be ignored by the Advanced Alert.

### Important Info About Advanced Alerts

Here are several things to keep in mind about Advanced Alerts:

* If you are building Advanced Alerts using an Excel data source with the “ Email…” fields defined in an Excel sheet and Tableau’s native Excel connector then Tableau’s default behavior is to strip the leading spaces. You’ll need to rename the fields in Tableau back to have the leading spaces.
* While VizAlerts supports Unicode characters in general, it cannot send mail to email addresses containing them. If any are detected, an error email will be sent instead.
* Though an Advanced Alert email can contain views from entirely different workbooks all of those workbooks must reside in the same Tableau Site. You can’t combine views from multiple Sites into the same email.

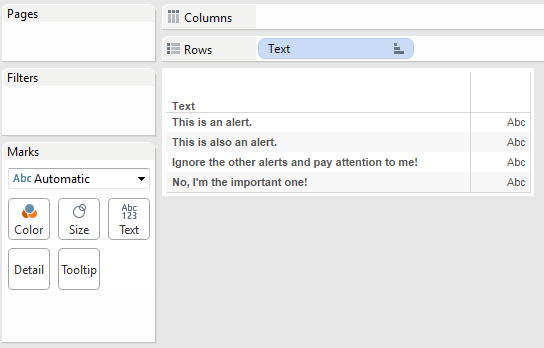
An Advanced Alert is executed via Subscription, just as a Simple Alert is, but with a couple of key differences:

* **Single Subscriber**: Only the **author** of the Advanced Alert can set up a Subscription to that Advanced Alert. Any other user’s subscription to it will be disregarded. To avoid confusion, it may be useful to limit View permissions on them, so other users cannot subscribe at all.
* **Impersonation**: VizAlerts will use the **author’s** Tableau Server user to generate the content for the Advanced Alert. This means that the author is solely responsible for ensuring that any content being emailed out is appropriate to the intended audience.

## Advanced Alerts Walkthrough #1 – Send the Trigger Viz

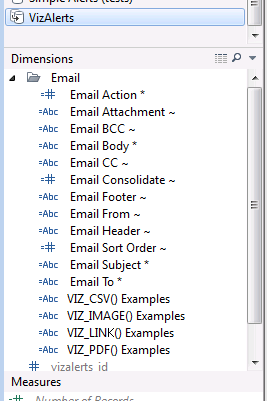
Okay, so let’s say we don’t want to have to tell people to subscribe to the simple alert from the previous example. Instead, we want to just deliver it to their inbox automatically at 7AM every day! Let’s build a trigger view for an Advanced Alert to support that.

First, open the VizAlertsDemo workbook in Tableau Desktop to the “AdvAlert Walkthrough #1 – 1” worksheet:

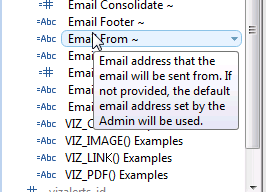


Okay. We have a field called “Text” in there. That’s not useful since it doesn’t map to any email properties. We need the fields to have the proper conventions, just as we saw in the table above. But it’s kind of a pain to have to build all those calculations with the proper names. Thankfully, there’s a faster solution:

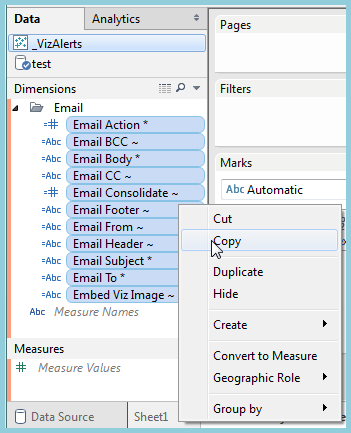
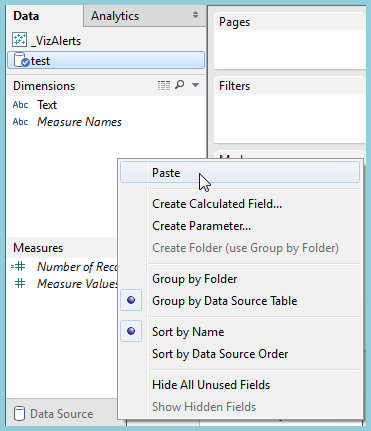
Open a connection to the VizAlerts.tdsx file in your VizAlerts\demo install folder (If you plan on using this a lot you might publish this source to Tableau Server):



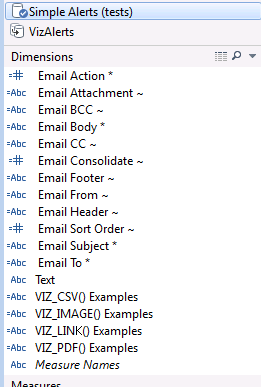
Hey, look at that. It happens to have all the proper fields in it! And each has a comment describing its own usage:



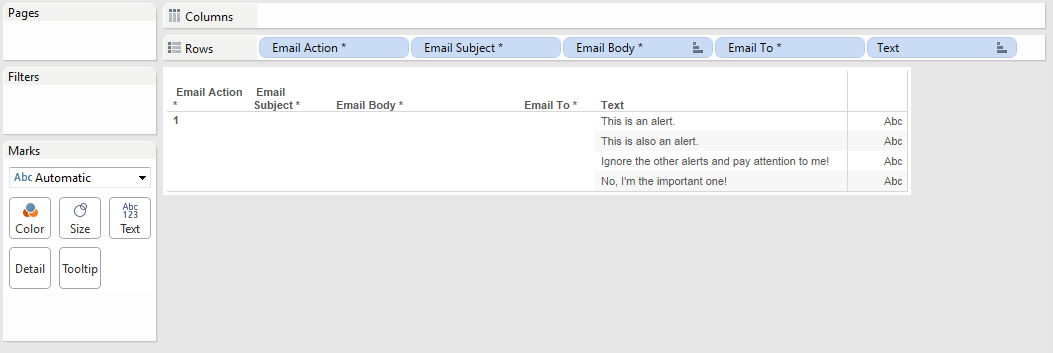
Now, this is just a dummy datasource with one row in it. We *could* get the fields into our viz by blending the data, but that would take a bit of work. A far easier way is to simply copy and paste them into our original data connection. Like so:

And if we create a nice little folder for them to live in, you get this lovely little setup:

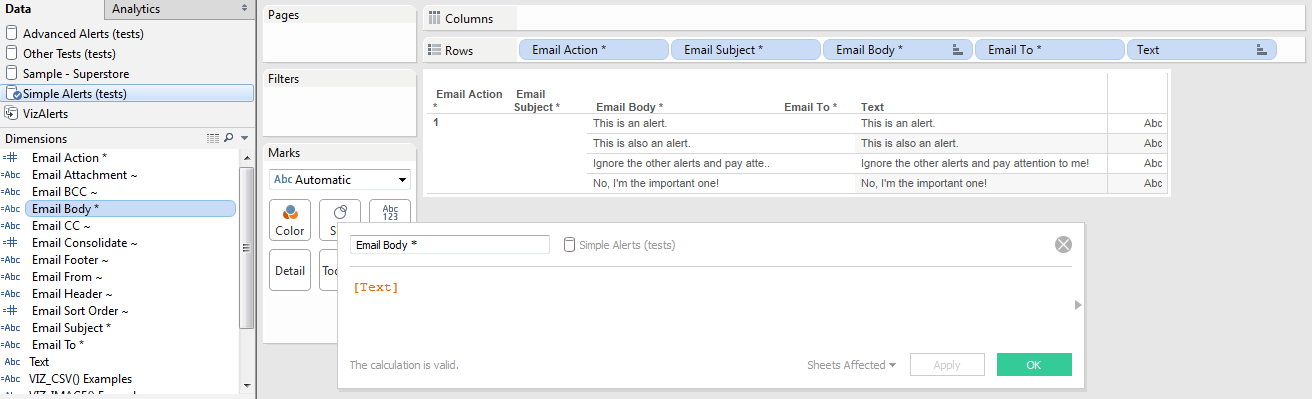


Now, let’s Control+Click and drag all the **required** fields (everything with an \* at the end) out to the Rows shelf to make sure they’re present in our trigger viz. (These have been in logical order):

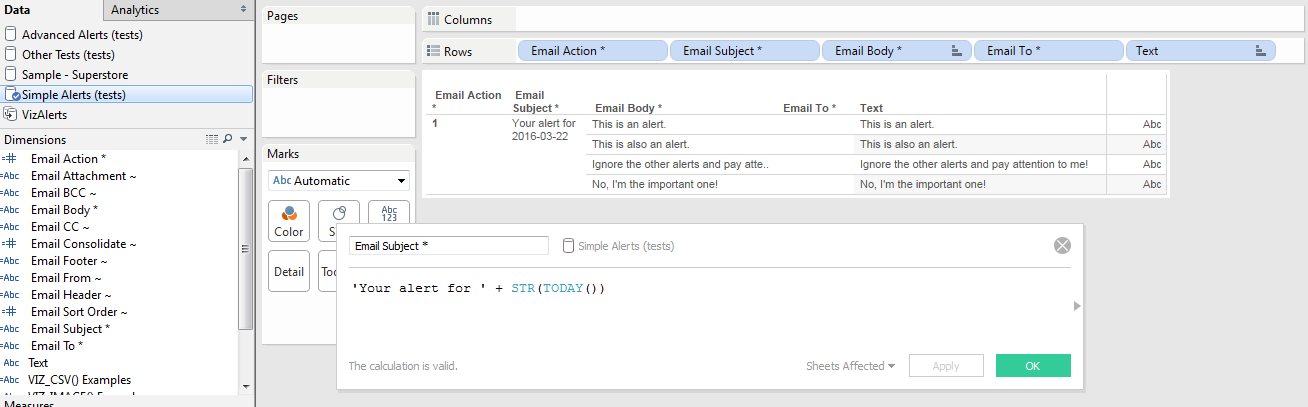


Wait, what’s this “ Email Action \*” field about? That doesn’t actually do anything more than tell VizAlerts that this is an Advanced Alert.

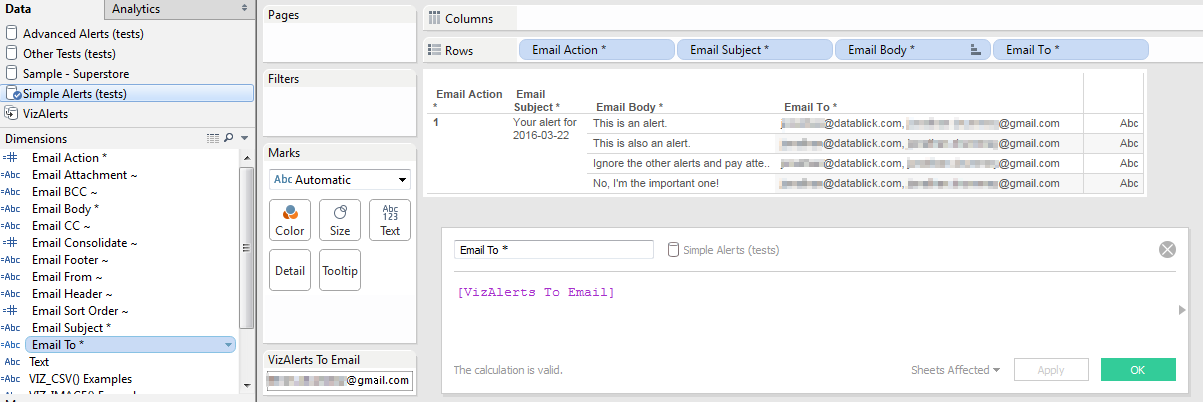
These calculated fields are all intentionally left blank so that you can fill them in with your own data. To do so, all we need to do is edit them. For this example, we’ll add static string values for the To and Subject fields, but we could just as easily derive those from our data if we had values in there. For the Body, we’ll actually just reference the “Text” field directly:



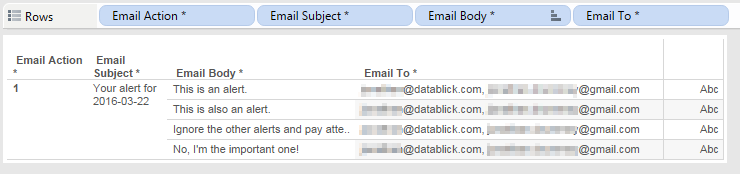
For the Email Subject \* field let’s create a calculation and put the date into it:



And for the Email To \* let’s make it a parameter so we can change it:



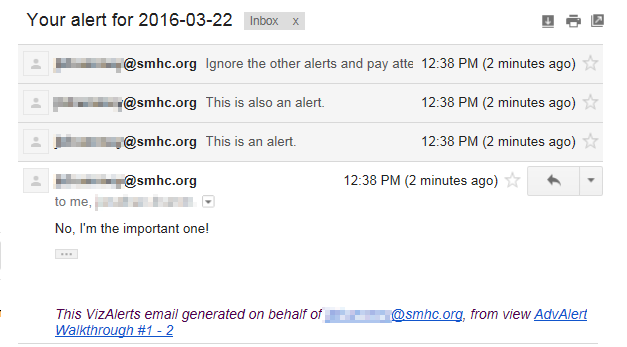
Okay, so after all that work, here’s what we have once we remove the redundant and ignored “Text” field from the viz (remember, VizAlerts ignores anything not starting with “ Email”:



What this means to VizAlerts is: **For every row of data, send one email** to someone @datablick.com and someone else @gmail.com with the Subject “Your alert for [date]” and the Body being the Email Body. Using the above as an example, four emails will be sent to each of the two examples. (The above worksheet is VizAlertsDemo\AdvAlert Walkthrough #1 – 2).

Now that we’re ready to have this alert run, we need to publish it to Tableau Server and subscribe to it on one of the Alerts schedules, just as we did for the Simple Alert.

The results of the alert we set up hit the datablick.com inbox like so:



And the same set of emails went to the Gmail address at the same time.

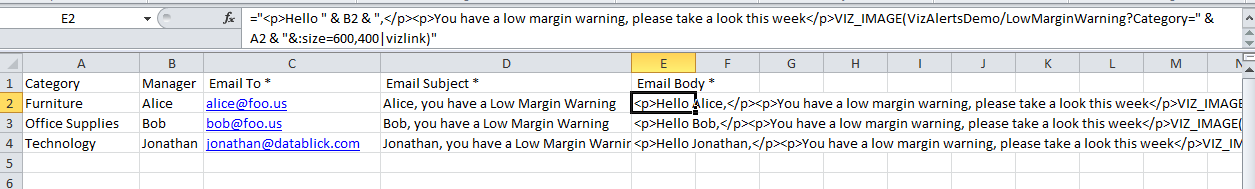
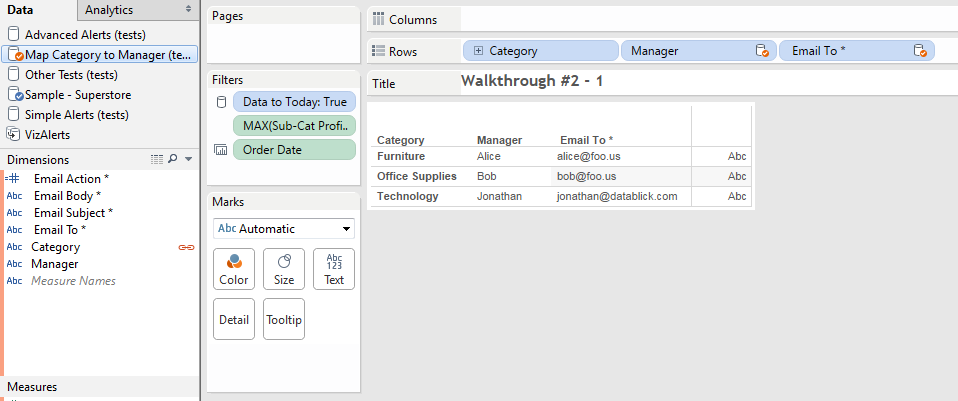
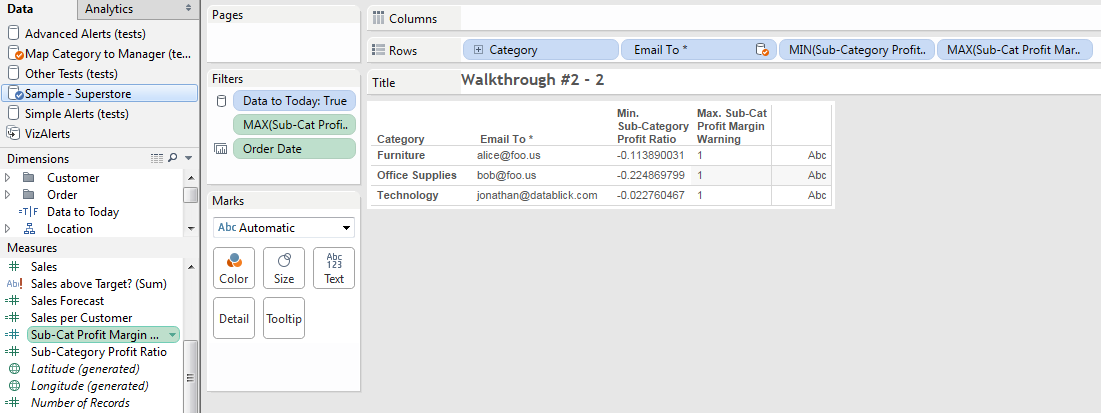
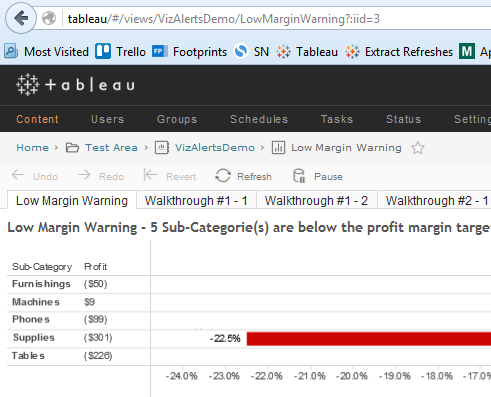
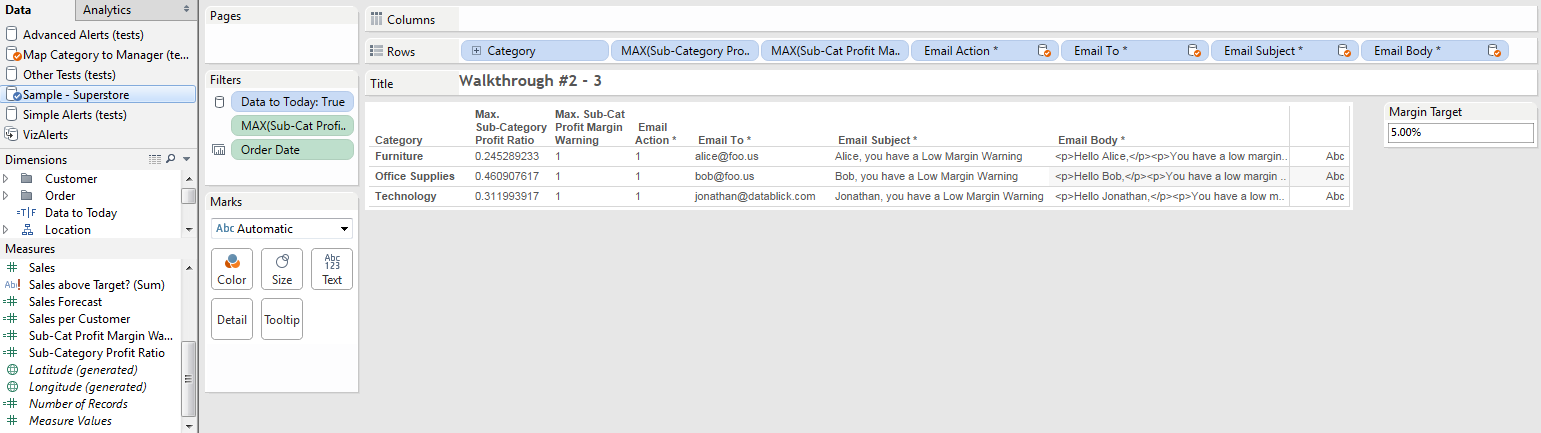
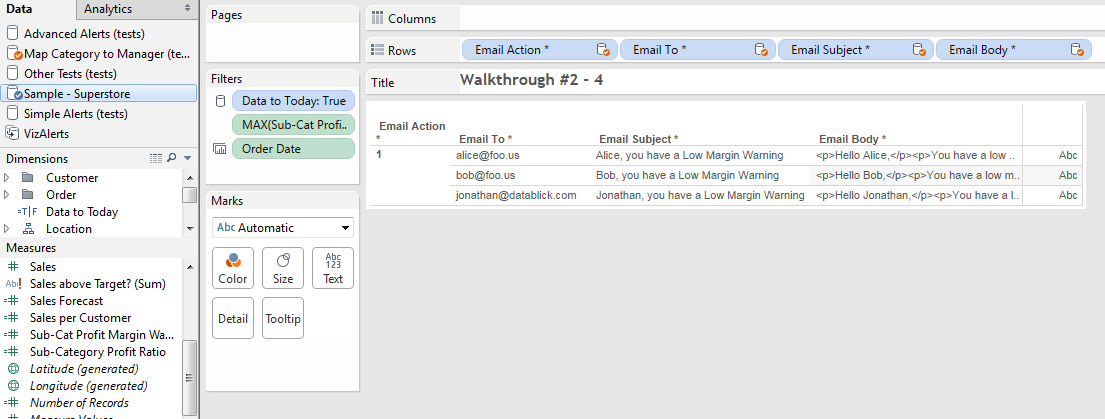
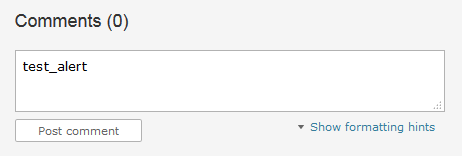
As we’ve seen the email properties are fields in Tableau. We made these calculated field, they could just as easily be set up in the data source. You can control for each row who gets what email. This allows you to use your data to drive the logic behind who is sent an email and what it contains, and more!

## Advanced Alerts Walkthrough #2 – Send a Custom Viz

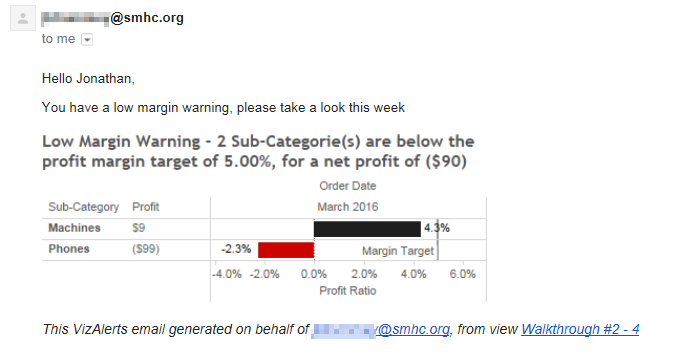
For this example we’re going to set up an Advanced Alert that sends the initial Low Margin Warning viz to the manager of each product category whenever any sub-categories in their product category meet the target. There are a few steps to setting up this kind of Advanced Alert:

1. Set up the view(s) that you want to send.
2. Set up the trigger view:
   1. Identify how many emails you’ll be sending – in other words, what defines the trigger viz Level of Detail (vizLOD)?
   2. Apply the necessary filters that will be used as the trigger for sending the alert.
   3. Identify the recipients and boilerplate content for each email.
   4. Create the necessary custom content references that will pull the views from #1.
   5. Validate that everything looks right.
3. Publish your trigger view and any necessary content views.
4. Subscribe the trigger view to the VizAlerts schedule.
5. Sit back and enjoy!

This next part goes through these steps to set up the alert:

1. The original Low Margin Warning view doesn’t need any changes, we’ll ultimately filter that view using a URL parameter.
2. For the trigger view we’re going to use a data blend. The primary source will be the same Superstore sales source used to generate the Low Margin warning, the secondary source gets us the email addresses, email body, and content references. This is an example of a situation where we want to trigger the alerts based on an operational source that is at one level of detail (individual order items) while the alerts are being set up and controlled via a separate source at a coarser level of detail (the product category) that might have a completely different owner and maintainer.  
     
   The secondary source is an Excel spreadsheet that is mostly set up with formulas, we’ll describe what’s happening in the Email Body further down.  
     
     
   You can see this spreadsheet in the [VizAlerts install folder\demo\test.xlsx file.  
   1. We know that we want just one email per category. In Tableau the VizAlertsDemo\Walkthrough #2 – 1 viz shows the Category dimension from the Superstore primary and then the Manager and Email To \* from the secondary, blending on Category:  
        
      
   2. Now we need to set up the filter(s) to only return rows that meet the criteria. In this case we only want to return Categories that have one or more Sub Category’s profit margin below the margin target. We can do that in Tableau using a variety of means, in this case we can use set up Level of Detail (LOD) expressions:
      1. Sub-Category Profit Ratio uses {INCLUDE [Sub-Category] : [Profit Ratio]} to return the Profit Ratio for each Sub-Category as a record-level value. (For the purposes of this demo we’re skipping some validation steps on this LOD expression).
      2. Sub-Category Profit Margin Warning uses IF [Sub-Category Profit Ratio] < [Margin Target] THEN 1 ELSE 0 END.\*\*\* When MAX() is wrapped as an aggregation to the vizLOD of Category we end up identifying whether any sub-categories have met the criteria – in this case, all the Categories have at least one sub-category:  
           
           
           
         Then we can put that field on the Filters Shelf, filtering for 1 in this case.  
           
         \*\*\* If you are using Tableau v9.2 or higher then you don’t need the IF statement anymore and can use just use the condition that returns a Boolean. See <http://drawingwithnumbers.artisart.org/feature-geek-how-to-have-sets-with-your-secondary-9-2-style-via-aggregated-booleans/> for more details.
   3. Now to get the recipients and boilerplate of each email. In this case the Excel file has been set up with all of those details, the Email Body is HTML formatted for a more desirable look.
   4. Finally we need to set up the custom content references, this was also done in the Excel source. In these emails the VIZ\_IMAGE() custom content reference is used to insert a PNG image inline in the body text. Here’s breaking down the parts of that VIZ\_IMAGE reference.
      1. **VIZ\_IMAGE()** – tells VizAlerts to download a PNG image, we could get a CSV or PDF if we wanted.
      2. **VizAlertsDemo/LowMarginWarning** is a custom view reference to the workbook and view name, this is what we see in our browser bar after http://[myservername]/#/views:   
           
         Note that in this case the trigger view and the custom view reference are both in the same workbook, they don’t have to be!
      3. **?Category=" & A2** uses an Excel formula that fills in the actual text for Category and generates the following output in Tableau:   
           
           
         This is a URL parameter that will be used to filter the LowMarginWarning viz when the trigger view is processed, see [Filter Views Using URL Parameters](http://kb.tableau.com/articles/knowledgebase/view-filters-url) <http://kb.tableau.com/articles/knowledgebase/view-filters-url> for more details.
      4. **"&:size=600,400** uses another Tableau URL parameter to set a custom size of the output PNG.
      5. **|vizlink** is a VizAlerts option that tells VizAlerts to make the inline PNG also have a hyperlink back to the viz so the email image is clickable (or tap-able).  
           
         For more details on these options (and more!) read the **Additional Advanced Alert Features** section below.
   5. Now to validate. Here’s the Walkthrough #2 – 3 view that has the additional fields added for the Subject and Body:  
        
        
      Now this trigger viz could work just fine as the trigger view for an Advanced Alert since VizAlerts will ignore the fields that don’t start with “ Email”. However we can clean up the view even more and get rid of those other fields, this is the Walkthrough #2 – 4 worksheet:  
      
3. Now we’re ready to publish the views to Tableau Server.
4. In practice this view would be subscribed to a monthly VizAlerts schedule. To test it out in advance you can enter a comment on the viz with just the text “test\_alert”, then wait for VizAlerts to send it (usually within a couple of minutes based on default configuration):  
     
   

When the alert is sent here’s an example email:



So that’s a simple example of how you can customize an Advanced Alert that hopefully gives you some thoughts about what you can do!

# Additional Advanced Alert Features

With Advanced Alerts come several options to allow even more customization in the content sent by the alert. These include:

* Custom Body, Header, and Footer.
* Content References
* Appended (non-inline attachments)
* Consolidated Emails
* Merged PDFs

## Custom Body, Header, and Footer

The **Email Body \*** field is a required field, though there are no required values for the field. i.e. you can leave it Null while using of other Advanced Alerts features, for example you might want to send an email that just has an attachment. The Email Body \* field passes any HTML formatting you include in it into the body of the email you compose, which allows you to customize the look of the content in your email. There are two other fields that you can customize with HTML formatting:

* **Email Header ~**
* **Email Footer ~**

The custom header and footer only really make sense to use when you are also usingthe consolidated emails functionality since otherwise you could just add the additional info to the **Email Body \*** fielddirectly, see Consolidated Emails below for more details.

Please note that using the **Email Footer ~** field in your viz will *replace* the standard VizAlerts footer that’s automatically appended to alert emails. If you wish to add your own footer, but don’t want to get rid of the standard footer at the bottom, put the content reference “**VIZALERTS\_FOOTER()**” anywhere in your footer data will cause it to be added at that location. See the VizAlerts.tdsx helper datasource for an example.

## Content References

Advanced Alerts supports inserting some pre-defined objects in a single alert using placeholder text, including support for views that aren’t the Advanced Alerts trigger view.

* **VIZALERTS\_FOOTER()**  
  When placed in the Email Footer ~ this adds the default VizAlerts footer to the email in addition to any other text in the Email Footer ~ field.
* **VIZ\_LINK()**This can be used in the Email Body \*, Email Header ~, or Email Footer ~ fields and will include an hyperlink to the referenced viz as an alternative to having to create http: links yourself, see below for more details.
* **VIZ\_IMAGE()**  
  This placeholder can be used in the Email Body \*, Email Header ~, Email Footer ~, or Email Attachment ~ fields and will include a PNG image of the referenced viz. When in the body, header, or footer the PNG will be included as an inline image placed within the text of the email. The optional **|vizlink** argument makes the inline image a hyperlink back to the Tableau view. When in the Email Attachment ~ field the PNG will be included as an appended (non-inline) attachment. Custom filenames are supported for appended attachments.
* **VIZ\_CSV()**  
  This placeholder can be used in only in the Email Attachment ~ field and will cause the CSV of the referenced viz to be included as an appended (non-inline) attachment. Custom filenames are supported for appended attachments.
* **VIZ\_PDF()**  
  This placeholder can be used in only in the Email Attachment ~ field and will cause a PDF of the referenced viz to be included as an appended (non-inline) attachment. Custom filenames are supported for appended attachments.

There are a number of ways we can customize the various content references, the rest of this section has more details on customizing views, using URL parameters, custom filenames, and more!

### Custom View References: Getting Almost Any View From Tableau Server

When called with no arguments, the VIZ\_LINK(), VIZ\_IMAGE(), VIZ\_CSV(), and VIZ\_PDF() content references will use the Advanced Alerts trigger view for the link, image, or data. We can go beyond that, though, by referencing most any viz we want to! For example **VIZ\_IMAGE(workbookname/viewname)** will render the given **workbookname/viewname** as a PNG.

In general the **easiest** way to construct these references is to go to Tableau Server in your web browser, find the view that you want, and copy the appropriate portion of the URL (everything after the …/views/). For example if the URL is http://myServer/views/SalesData/SalesDashboard then the argument for VIZ\_IMAGE() would be VIZIMAGE(SalesData/SalesDashboard), i.e. everything after the …/views/.

#### Permission Limitations on Custom Views

Since Advanced Alerts are generated based on the Tableau user of the trigger view any custom view references will use that user’s permissions. It’s assumed that some sort of administrator would be configuring the Advanced Alerts so they would have the necessary permissions. Also all downloaded custom views have to be in the same Site as the trigger view.

### URL Parameters: ?myParam=myValue

Content References also support URL Parameters as described in [Filter Views Using URL Parameters](http://kb.tableau.com/articles/knowledgebase/view-filters-url) <http://kb.tableau.com/articles/knowledgebase/view-filters-url>. For example you could use VIZ\_IMAGE(VizAlertsDemo/Product?Region=East) to send only the East data in an email. Keep in mind that you do need to URL encode any values, for example VIZ\_PDF(VizAlertsDemo/Product?Customer%20Segment=Corporate). If you want to filter the trigger view without explicitly referencing it you can do that by adding just the URL parameter, for example VIZ\_CSV(?Customer%20Segment=Corporate) would filter the trigger view.

Note that VizAlerts uses the “|” (pipe) character as a delimiter for additional arguments after the URL parameter for filenames, etc. so if your custom views are requiring URL parameters using “|” then you will need to URL encode those “|” characters.

Also, VizAlerts automatically adds the **:refresh=y** option to make sure that your outgoing emails contain the latest data.

#### Controlling PNG Size

A given VIZ\_IMAGE() content reference will be rendered as one of two sizes:

1. If the view is a dashboard with a fixed size or or minimum size that is larger than the VizAlerts default, that size will be used.
2. The VizAlerts default size that was configured by your administrator when VizAlerts was installed.

Thankfully there’s a 3rd option: Tableau Server has an undocumented URL parameter for controlling the size of PNG images. You can add **:size=width,height** as an additional URL parameter to you viz reference. For example VIZ\_IMAGE(mySalesData/mySalesDashboard?Region=East:size=400,600).

There are a couple of limitations on controlling the PNG size when the referenced viz is a dashboard (based on Tableau’s default behavior):

* If a dashboard has a range of sizes then you can't set the image size smaller than the minimum range or bigger than the maximum range.
* If a dashboard has a fixed size you can’t change it.

### Custom Filenames for Appended Attachments: |filename

For appended attachments the default filename is YYYYMMDDHHMMSSUUUUU\_[workbookname]-[viewname].[filetype] where:

* YYYYMMDDHHMMSSUUUUU is the Year, Month, Day, Hour, Minute, Second, Microsecond
* [workbookname] [viewname] are the referenced viz
* [filetype] is png, csv, or pdf.

We can override the default filename by adding **|filename=[filename]** after the custom view and URL parameters where the leading character is “|” (pipe delimiter) and [filename] is the desired filename (including any spaces). VizAlerts will automatically append the filetype based on the content reference. For example VIZ\_CSV(AnnualReport/Overview|filename=Annual Report) would append a file named “Annual Report.csv”. See the sections on appended attachments and merged PDFs for more details.

#### Timestamps and Other Dynamic Elements for Custom Filenames

If you want a timestamp or some other per-view element like a unique ID on a custom filename then you can build your own using calculated fields in Tableau as you build the trigger view or using functions in your data source. For example if I’ve got a data source with Alice, Bob, and Carlos as records and I want the monthly productivity dashboard PDF to have a name like “Alice Productivity Feb 2016.pdf” then I could build the content reference to be: “VIZ\_PDF(Productivity/Dashboard?Employee=” + [Employee] + “|filename=” + [Employee] + “ Productivity “ + LEFT(DATENAME(‘month’,TODAY()),3) + “ “ + DATENAME(‘year’,TODAY())”

### Hyperlink Text References: VIZ\_LINK()

When using the VIZ\_LINK() content reference the default text of the link will be the reference to the trigger view (if just VIZ\_LINK() is used) or the custom view reference, including any URL parameters. For example if the reference is VIZ\_LINK(VizAlertsDemo/Shipping?Region=East) then the generated HTML would be something like “<a href=http://myTableauServer/views/ VizAlertsDemo/Shipping?Region=East>VizAlertsDemo/Shipping?Region=East</a>”. By using the **|filename** option we can use our own text, for example VIZ\_LINK(VizAlertsDemo/Shipping?Region=East|filename=Shipping Dashboard) would generate the following HTML:

<a href=”http://myTableauServer/views/ VizAlertsDemo/Shipping?Region=East”>Shipping Dashboard</a>

If you leave the |filename option blank then VizAlerts will use the workbook/view reference so VIZ\_LINK(VizAlertsDemo/Shipping?Region=East) would create the following HTML:

<a href=”http://myTableauServer/views/ VizAlertsDemo/Shipping?Region=East”>VizAlertsDemo/Shipping?Region=East</a>

If you have no custom view reference at all and just use VIZ\_LINK() then the HTML will be use the workbook/view of the trigger view.

If you want the generated to be just the http: hyperlink then use the **|rawlink** option, for example VIZ\_LINK(VizAlertsDemo/Shipping?Region=East|rawlink) would generate just:

http://myTableauServer/views/ VizAlertsDemo/Shipping?Region=East

### Hyperlinked Inline Images: |vizlink

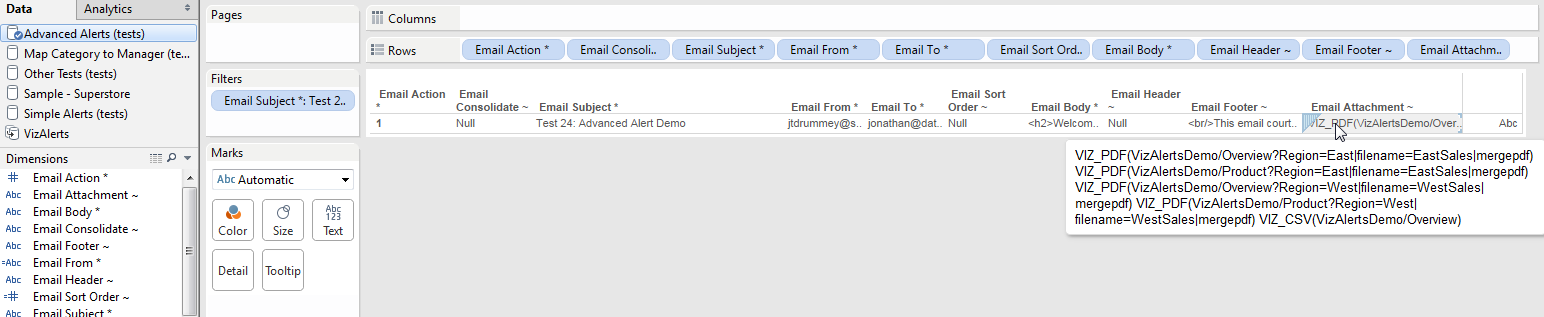
When VIZ\_IMAGE() is used to embed images inline we might want to make the image clickable so that users will automatically go to the Tableau Server view. Adding **|vizlink** after the custom view tells VizAlerts to add a hyperlink to the image. For example VIZ\_IMAGE(VizAlertsDemo/Product?Region=East|vizlink) would make the inline image also a hyperlink back to the image on Tableau Server.

### Merge Multiple PDFs : |mergepdf

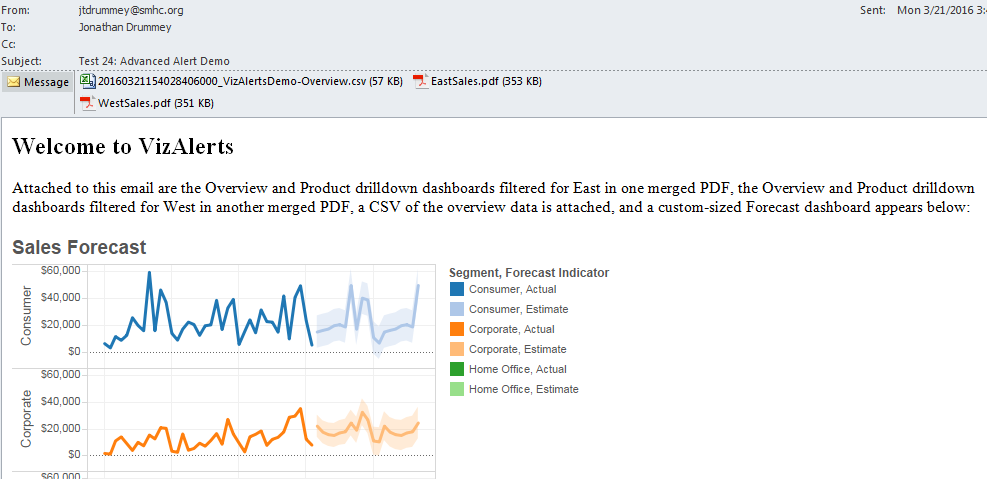
If there are multiple PDFs in a single VizAlerts email it’s possible to merge sets of them together into one or more PDFs. This option requires that a custom **|filename** be used for each output PDF that you want. The option is configured by adding **|mergepdf** to the content reference argument. For example an email might have two content references VIZ\_PDF(VizAlertsDemo/Overview?Region=East|filename=EastSales|mergepdf) and VIZ\_PDF(VizAlertsDemo/Product?Region=East|filename=EastSales|mergepdf) those would be merged into a single EastSales.pdf attachment. See below for more details on merging PDFs.

## Appended (non-inline) Attachments

We can add one or more content references to the **Email Attachment ~** field to append non-inline file attachments of downloaded views. For example, the VizAlertsDemo/AdvancedAlertsDemo worksheet has a set of content references that include two merged PDFs and a CSV:



These show up in our email application as file attachments to the message:



CSVs and PDFs can only be included in VizAlerts emails as appended attachments, PNG attachments can be included inline in the body, footer, and/or header, or be appended.

If we want to append multiple attachments to a given email there are two ways this can be configured:

* Put multiple content references into a single Email Attachment ~ field as in the above example.
* Use consolidated emails. All appended attachments in all rows for a given consolidated email will be attached. See the Consolidated Emails section below for more info.

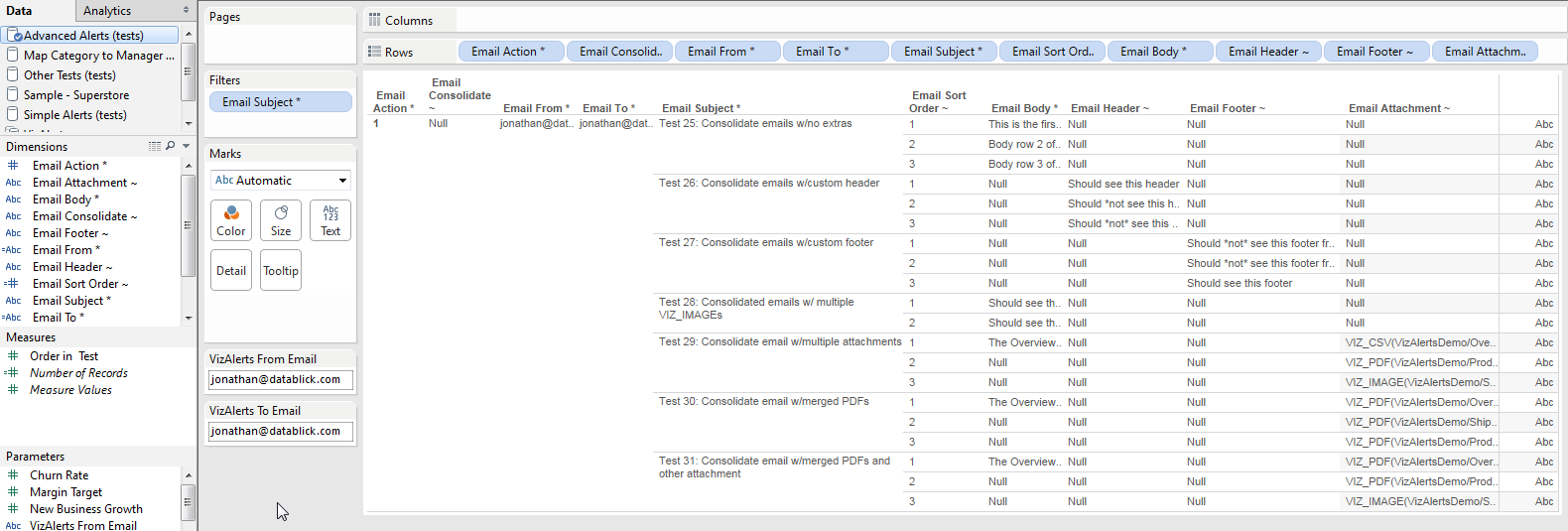
Note that VizAlerts will prevent collisions in filenames, if you try to make totally different attachments in the same email have the have the same filename then VizAlerts will revert to the default filenames for the 2nd and successive attachments. If you want to merge multiple PDFs into the same filename then you can use the **|mergepdf** option.

## Consolidated Emails

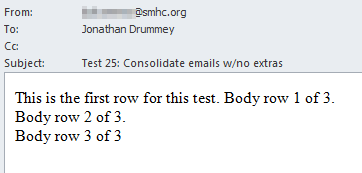
There are a few use cases for consolidated multiple alerts into a single email:

* VizAlerts is so awesome that your users sign up for a ton of alerts! But then they get bombarded with emails. By consolidating emails their inbox won’t be quite so full.
* There’s a manager who wants to get individualized dashboards for each of her direct reports but doesn’t want an email for each person, just one email with all of the reports.
* You want to generate an image or PDF for each region, product category, customer segment, etc. but can’t get the Pages Shelf to do what you want. VizAlerts can come to your rescue!

Consolidated emails are turned on by having **Email Consolidate ~** in the trigger view. When this field is present in the trigger view, rather than sending **one email per row** of data, VizAlerts will consolidate the Body field and appended attachments across **multiple rows**, as long as the Subject and **all** recipient fields (To, From, CC, BCC) are the same. So in this view below the three rows for Test 25: Consolidate emails w/no extras will be consolidated into one email:



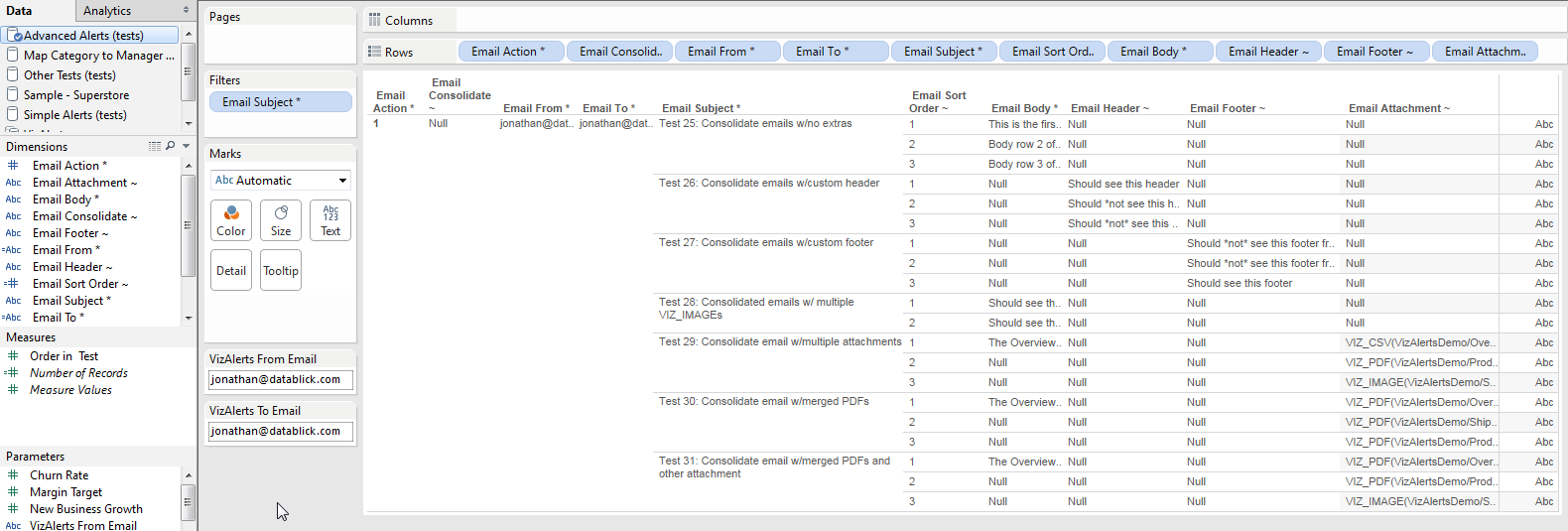
Here’s the email:



Note that email consolidation works at the trigger view level, and not at the individual row level—you cannot use it for some rows in your trigger view, but not others. If Email Consolidate ~ is present VizAlerts will always attempt to consolidate emails across the Subject, To, From, CC, and BCC fields.

### Sorting Consolidated Emails

When we’re using consolidated emails we often want the content to be generated in a specific order within each email. The **Email Sort Order ~** field lets us do that. When it is present the values in the field are used to alphanumerically sort each row in the trigger view for each email. Here’s an example setup from the VizAlertsDemo/Advanced Alerts worksheet:

   
  
In the above example the numbering restarts for each consolidated email, however it doesn’t have to. There are a number of ways you could get the sort order without too much effort, here are a couple of additional examples:

* If you were trying to generate an alphabetical list of product names then you could use the [Product Name] field values in the Email Sort Order ~ field.
* The INDEX() or RANK() functions in Tableau could generate a number that you could use.

### Headers and Footers in Consolidated Emails

The optional **Email Header ~** and **Email Footer ~** fields find their use in consolidated emails. The contents of the Email Header ~ field in the **first** row from the trigger view for a given consolidated email are pre-pended to the body of the email, while the contents of the Email Footer ~ field from the **last** row for that consolidated are appended to the body of the email.

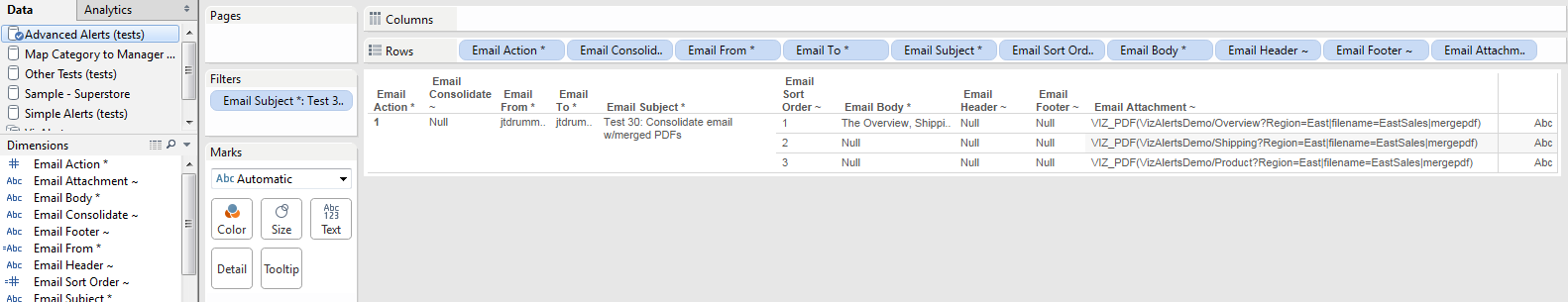
For example you might use the Email Header for some intro text and an image of an overview dashboard, and then the Footer would contain a link to the dashboard, and all the other rows of the consolidated email would have body text, or maybe no body text and just attachment content references.

## Merged PDFs

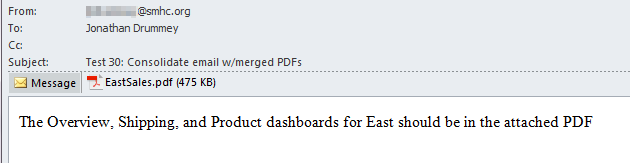
From time immemorial (ok, sometime in the 21st century) Tableau users have wanted to be able to do something close to traditional batch reporting where a whole bunch of reports are merged together into a single PDF suitable for the CxO to read on her next cross-country plane flight. VizAlerts delivers!

The configuration of merged PDFs is described above in Merge Multiple PDFs – you simply add the **|filename=[filename]** and **|mergepdf** arguments to the **VIZ\_PDF()** content reference. Then when the email is generated VizAlerts will scan through all the VIZ\_PDF() content references (whether they come from multiple content references in a given row or from multiple rows in a consolidated email, or both) for the ones with the same fieldname and merge them together in order. If you have multiple filenames with the |mergepdf option you’ll get multiple merged PDFs.

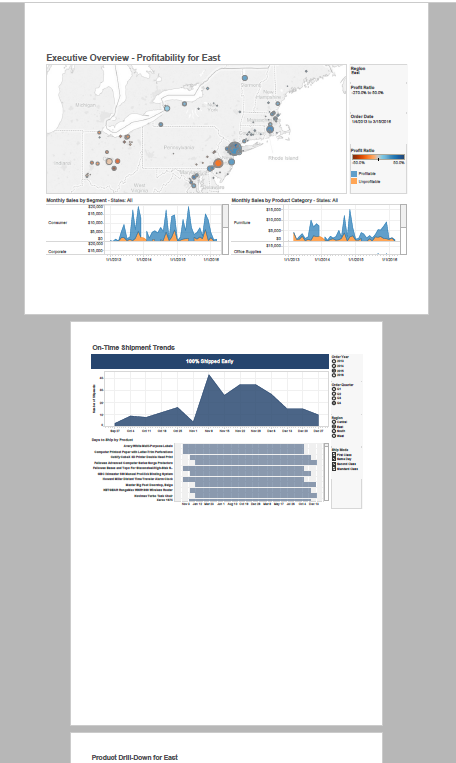
Here’s an example where three dashboards are merged into a single PDF using a consolidated email using test 30 from the VizAlertsDemo\Advanced Alerts worksheet:



The email:



One nice feature here is that VizAlerts respects the Landscape/Portrait orientation of each referenced view in the merged PDF, here are screenshots of the first two pages of the above example:



# SMS Text Message Alerts

SMS Alerts are a variation on Advanced Alerts that are SMS text messages delivered to mobile devices. There are two ways to deliver text message alerts from VizAlerts:

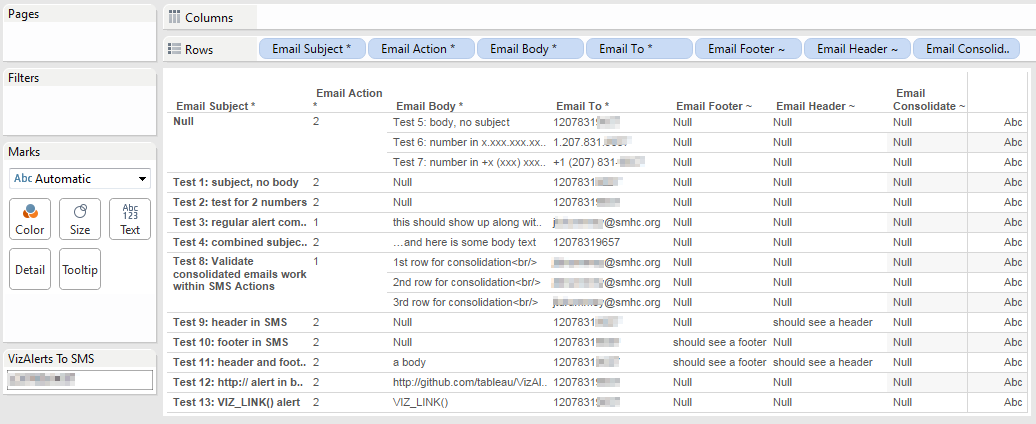
* If your cellphone provider has a free email-to-text gateway then you can use the given email address for the mobile phone as the Email To \* address for an email advanced alert, for example for AT&T in the USA the address is [xxxxxxxxx@txt.att.net](mailto:xxxxxxxxx@txt.att.net) where xxxxxxxxx is the area code and full phone number of the mobile phone. As a bonus some cellular providers will even automatically convert emails with inline images for you into MMS messages.
* Use the built-in integration to an SMS gateway provider, by configuring an Advanced Alert trigger view with Email Action \* = 2. These messages are SMS only, inline images are not supported.

At this time the only supported SMS gateway is with [Twilio](https://www.twilio.com/), to use the Twilio integration your Tableau Server admin will need to have set up the VizAlerts configuration file with the necessary information. See the VizAlerts Install Guide for details.

The rest of this section is about configuring and using SMS Alerts with the gateway since configuring Email Advanced Alerts is covered above.

## SMS Alert Configuration

SMS Alerts are set up in the exact same way as Email Alerts and you can even make a trigger view that has rows for both Email Alerts and SMS Alerts. Here’s a screenshot of the SMS test view located at …demo\SMSDemo.twb that has a couple of outgoing emails along with various SMS alerts:



The trigger view for SMS Alerts uses the exact same field names as Email Alerts, however the field values are different, here’s a description:

|  |  |  |
| --- | --- | --- |
| **Field Name (with leading space)** | **Required?** | **Structure & Description** |
| Email Action \* | Yes | Value = 2, tells VizAlerts that this is an SMS Advanced Alert |
| Email To \* | Yes | Phone number(s) separated by commas or semicolons. Phone numbers need the country dialing code, area code (if applicable), and destination number. |
| Email From ~ |  | Optional originating phone number (you will need this supplied by the SMS gateway provider) |
| Email CC ~ |  | Additional phone number(s) |
| Email BCC ~ |  | Additional phone number(s) |
| Email Subject \* | Yes | Text for subject, can be left blank/Null |
| Email Header ~ |  | Optional additional header text & inline images |
| Email Body \* | Yes | Text for body, can be left blank/Null |
| Email Footer ~ |  | Optional additional footer text |
| Email Attachment ~ |  | Not used for SMS alerts |
| Email Consolidate ~ |  | No value, just needs to be present in viz to trigger consolidation |
| Email Sort Order ~ |  | Any alphanumeric values, used for sorting consolidated emails |

## Supported Mobile Number Formats

For convenience the mobile numbers can be in most any format, the key elements are:

* Country code
* Area code (if applicable)
* Dialing code

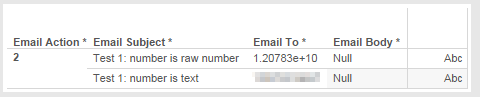
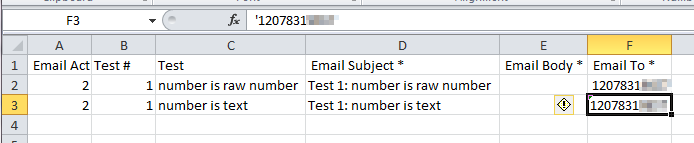
The valid characters in mobile numbers are the ten digits 0-9, hyphen -, space , period ., and parentheses (), and plus sign +. For example the following number formats are supported:

* 12071234567
* +12071234567
* +1-207-123-4567
* +1 (207) 123-4567
* 1.207.123.4567

If you put multiple phone numbers in a single Email To \* field they can be separated by commas (,) or semicolons (;) such as 12071234567, 12069876543.

## Important Info about SMS Advanced Alerts

There are a number of complexities to configuring SMS Advanced Alerts:

* SMS messages that are over 160 characters will be sent as multiple SMS messages, so rates could be higher.
* If you use both the Email Subject \* and Email Body \* to generate the SMS then VizAlerts will prepend “Subj: ” to the subject and “Body: ” to the body, using up 12 characters of the SMS.
* HTML will not render as HTML in a text message, so you’ll need to leave that out of the structure (especially to save space). However http:// and emailto:// links are generally hyperlinked on smartphones.
* For Excel and text sources using the default connector Tableau will render 10 digit mobile numbers in scientific notation, so 12071234567 becomes 1.207e10:  
    
    
    
  The workaround is to force the number to be text. In Excel you can do this by prepending ‘ (single quote) to a number field, see how the formula bar is different from the display in this screenshot:  
    
    
    
  Alternatively in Tableau you can cast the number as a string using the STR() function.
* SMS Providers will only deliver one text message per second.

## Content References for SMS Advanced Alerts

The supported custom content references for SMS Advanced Alerts are:

* **VIZALERTS\_FOOTER()**Though this can be used, it generates HTML code that doesn’t look very good.
* **VIZ\_LINK()**This content reference behaves slightly differently than email alerts. No matter the options, only the URL (whether a custom one or to the trigger view) is sent in the SMS Alert. So the VIZ\_LINK() content reference in an SMS Advanced Alert always acts as if the **|rawlink** option is set. This is done because HTML doesn’t render in SMS messages and to save space in the messages.

# Advanced Alert Use Cases

### Tableau Server Monitoring

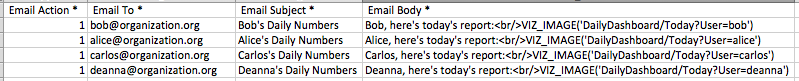
Here’s a link to a thread of a number of ways to use VizAlerts for monitoring Tableau Server: <https://community.tableau.com/message/383607#383607>

### Extract Failure to Refresh Notifications

There are multiple ways to set up automated notifications for failure to refresh extracts (the above link has one), here’s another: <https://community.tableau.com/message/465996#465996>

### Bulk Mailing

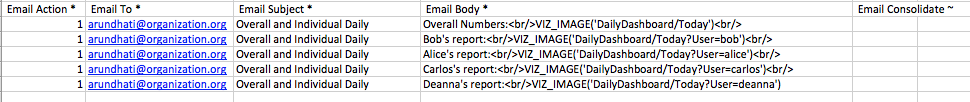
Using an Advanced Alert with custom URL parameters it is possible to have very fine-grained controls for sending bulk emails from Tableau Server of any view(s) you want. For example a user with the correct permissions could set up a view that had a list of email addresses and body that included custom URL parameters, for example to email out a daily dashboard filtered for certain data:



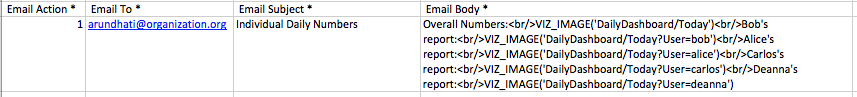
Since the data source could be generated with some simple formulas in Excel, Tableau, or the data source, this offers a lot of flexibility for configuring delivery.

### Consolidated Emails

As noted above there’s an “ Email Consolidate ~” option for Advanced Alerts that will consolidate across the rows in the triggering viz. For example this data would send out one email to Arundhati and automatically include all the images of her direct reports:



You can also embed multiple VIZ\_IMAGE() calls in the header, footer, and/or body. So the same email could be generated using a single (larger) email body:



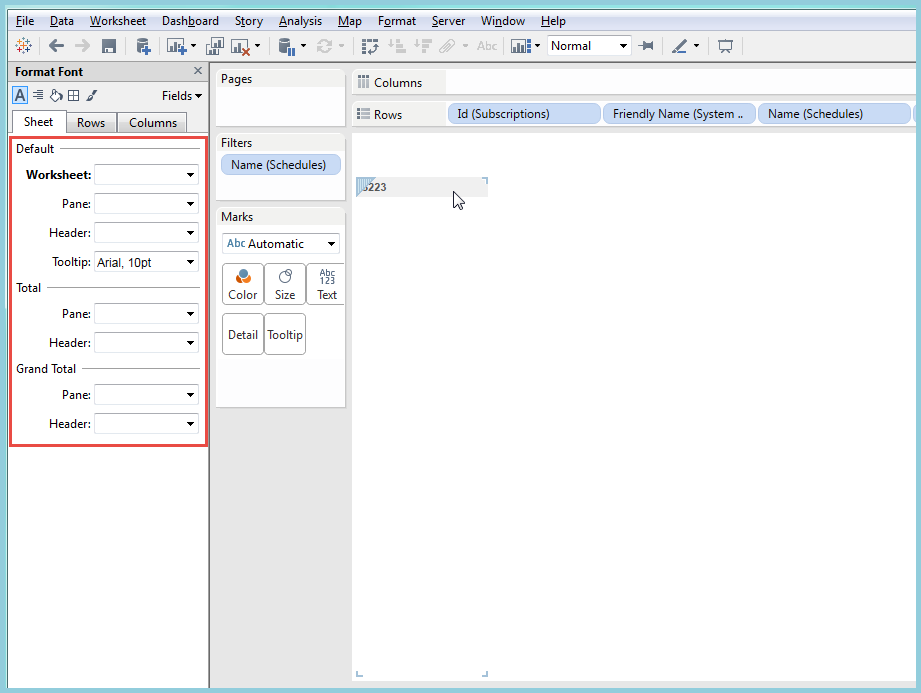
### Dashboard with Hidden Embedded Alert

This is an alternative method to setting up Advanced Alerts for a Tableau dashboard that doesn’t require using custom VIZ\_IMAGE() references. There are two important things to know when you set up a Dashboard with a Hidden Embedded Alert:

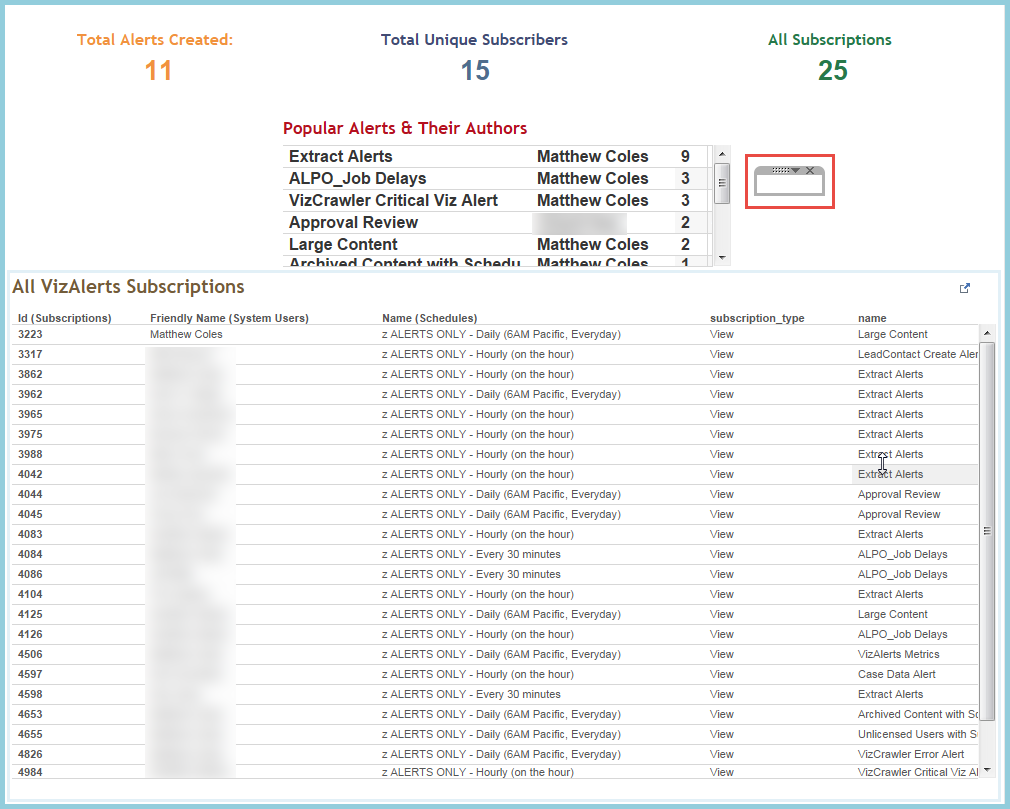
1. Only one View within the Dashboard can be used to trigger the Alert
2. The triggering View is the one whose name comes first in alphabetical order

Given that, the way to ensure you get what you need out of the Dashboard w/Hidden Embedded Alert is to **embed a specific View that serves as the trigger** for the alert, and name it “-(Dashboard Name) Trigger”, e.g. “-Sales Quota Dashboard Trigger”. The hyphen should ensure it comes first and is therefore used as the trigger.

However, what if you don’t actually want to *display* the trigger view? For Advanced Alerts in particular, you often don’t. To make it invisible, use the Format menu to change text colors to the background of the dashboard and remove borders:



Then, place it in your Dashboard as a “floating” view, and drag the container around it into as small a rectangle as possible. Place it somewhere unobtrusive:

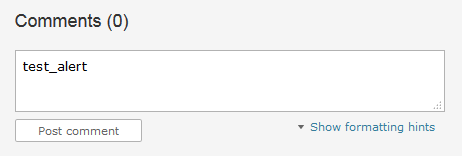


Now, just subscribe to the Dashboard just like any other Alert based on a standard View, and you’re good to go!

### 

# Testing an Alert

Occasionally you may wish to test an alert prior to setting it up on a Schedule, as you may not be certain it will execute properly. For Advanced Alerts in particular, testing them first is highly recommended! VizAlerts allows an alert to be tested on a one-off basis without being scheduled through the use of a specific **comment**:



If the owner of the View adds a Comment with the content “**test\_alert**” (case insensitive), then the Alert will be tested one time only. If another comment with the same content is added, the Alert will be tested again. Removing the comments, however, will not trigger an additional test. Allow at least three minutes for your alert to show up.

Note that this doesn’t work for Customized Views, only standard Views, and the commenter must also be the owner.

# FAQ / Common Issues / Troubleshooting

## I got a failure email instead of an alert!

When VizAlerts cannot process your alert, by default you and the Administrator are sent a notification email with the details. Issues can range from simple timeouts, permissions issues to datasources, non-allowed destination email domains, bad custom view references, and internal Tableau Server errors. The two most common errors are:

* **Timeouts**  
  If timeouts are the case, improve the speed of your view! Read <http://www.tableausoftware.com/learn/whitepapers/designing-efficient-workbooks> and follow the instructions there. If that’s not possible, work with your Tableau Server Admin to find a way to increase the timeouts your alert runs under.
* **Unable to Download Custom View**  
  If the custom view reference is causing failures then go to the desired view in Tableau Server, get the URL, and check it against the view reference in your Advanced Alerts trigger view. For example you might have set up a view reference like VIZ\_IMAGE(myDashboard/Quarterly Report) and the actual viewname is http://myServer/views/myDashboard/QuarterlyReport with no space in “QuarterlyReport”.

## I’m getting Simple Alerts when I should be getting Advanced Alerts.

The most likely reasons are that either the **“ Email Action \*”** field isn’t present in your trigger view or the Email Action \* field doesn’t have the required leading space in the field name

## What if I don’t get an alert email when I expect one?

There are a couple of reasons this might occur:

1. First, ensure that your filter criteria is set up correctly, and that you DID have data in your alert for when it was scheduled to run.
2. Check to see if your alert was filtered and sent to Junk Mail rather than your Inbox. If it was, make sure to “always allow” mail from the address you received it from in your Junk Mail settings.
   1. If it’s neither of those things, contact your Tableau Server administrator, as they should have received a failure email if your alert truly did not work correctly.

## How can I avoid getting continual alerts for the same data?

One strategy for this is to use a Relative Date filter against your data. You should set that filter to a span of time that matches the Schedule you subscribed to your Alert on. For example, if you have subscribed to an Alert on a daily schedule, you should set your relative date to show only data for the last 24 hours (or 1440 minutes).

## I’m getting my Alerts late.

Discuss with your Admin, and if they aren’t sure what to do, ask them to read the “FAQ” section of the install\_guide.docx file.

## There is no default footer on my emails.

If you’ve added the Email Footer ~ field to your trigger view then that overrides the VizAlerts default footer. You can add a calculated field using the string “VIZALERTS\_FOOTER()” to put the default footer back in, or build your own calculation.

## My CSV is Bad!

If you’ve ever tried to download a CSV of a Tableau dashboard you’ve likely run into a situation where the data you got was not what you expected to get…the same can occur with VIZ\_CSV content references. There are two problems that can occur:

* Wrong worksheet’s data
* Wrong data structure (too many fields, measures are in multiple rows instead of multiple columns)

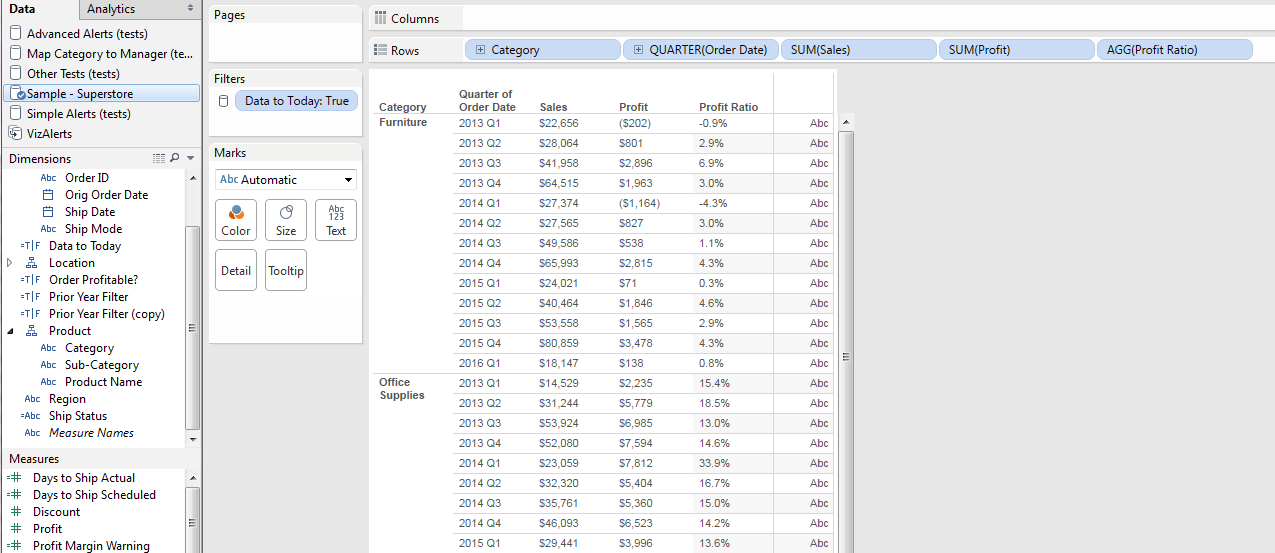
Here are workarounds for both:

### Wrong worksheet’s data

When we download a CSV from a Tableau dashboard (for example VIZ\_CSV(myViz/myDashboard) Tableau pulls the data for the worksheet that comes first when sorted alphanumerically. So to control which worksheet gets downloaded first you can prepend the desired worksheet’s name with a number or space character. For more information see ~~~link to AndyK’s post.

### Wrong data structure

When downloading the CSV Tableau pulls all the pills used in the viz and if we’ve used Measure Names/Measure Values generates a set of rows for each of the measure names. To have more control over this the solution for this is to build a separate worksheet with the fields that we want as discrete pills on rows:

 Don’t worry at all about formatting this viz, it’ll never be seen by the user. Once you’ve built this, you have a few options for publishing this worksheet for download:

* Publish as a completely separate viz. This requires making sure that you have permissions properly configured.
* Publish as an additional view in the workbook, give it an appropriate name, and then you’d reference it as something like VIZ\_CSV(myViz/DataForDownload). When using tabs this guarantees that you’ll have one extra tab, which is sometimes undesirable.
* Publish as an invisible (and hidden) sheet in a dashboard. In this case you’d name the sheet using the tip above so that it is first in the alphanumeric sort order, Hide the worksheet, and make it invisible in the dashboard. There are two ways to make it invisible:
  + Make the worksheet for download tiny and float some other element over the worksheet.
  + Float the worksheet for download and then place it outside the borders of dashboard.

Then when VIZ\_CSV(myViz/myDashboard) is called the worksheet for download will be the one referenced, and the data will be in the desired format.

## My Emails aren’t Consolidating.

The **“ Consolidate Email ~”** field has to be present in your trigger view and must have the leading space. In addition each set of rows that you want to consolidate must have the same Subject, From, To, CC, and BCC fields. Make sure there aren’t any leading or trailing spaces that are causing those fields to be subtly (and perhaps invisibly) different.

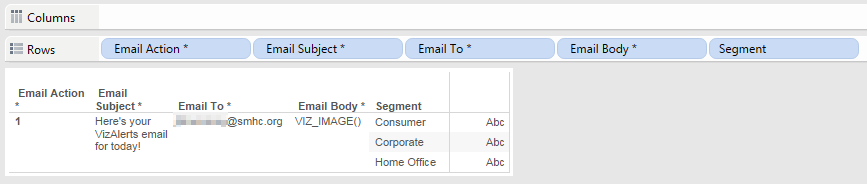
## My PDFs aren’t Merging.

If the PDFs aren’t merging then you need to check each VIZ\_PDF() content reference to make sure each includes both the following arguments:

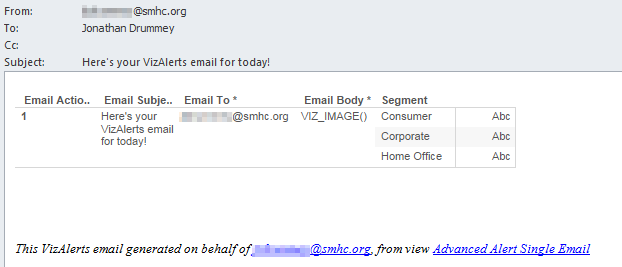
* **|filename=[filename]** where all the [filename] references are the same
* **|mergepdf**

## I’m getting too few emails, it looks like they are Over-Consolidating.

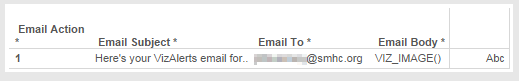
If you set up a view like this to generate 3 emails, one for each customer Segment:



You’re only going to get one email of the trigger view:



The reason why is that VizAlerts ignores all fields in the trigger viz except for the Email… fields, so effectively there’s only one row to be emailed:



The solution is to make the Email Subject \* or Email Body \* fields include the Segment or another field that will make them unique and generate the multiple emails.

## Inline and/or Appended Attachments aren’t showing up in my email.

There are a couple of tests you can run here:

1. Try running the VizAlertsDemo tests and see if they work. If they do, then your content references are incorrect in some way and need to be checked.
2. If the attachment tests for the VizAlertsDemo aren’t showing up in your email, then try checking your email via a different application or device, and try running the tests again to a different email address on a different server. VizAlerts has been tested on a variety of applications and devices (Exchange Server, Apple Mail, iPhones, web browser, Outlook, etc.) but may be missing yours. If you are still having issues with certain application/device combinations then please submit a bug at <https://github.com/tableau/VizAlerts>.

# Getting VizAlerts Help

First of all, check with your local Tableau Server Admin and any local documentation that might exist. After that, the center for all things VizAlerts is the VizAlerts Group on the Tableau Community <https://community.tableau.com/groups/tableau-server-email-alert-testing-feedbac>.

# Contributing to VizAlerts

VizAlerts is an open source project distributed under the MIT License. If you’d like to contribute ideas or code to VizAlerts, please visit the VizAlerts GitHub site at <https://github.com/tableau/VizAlerts>.

1. VizAlerts does have the ability to consolidate multiple rows of data into one email. [↑](#footnote-ref-2)