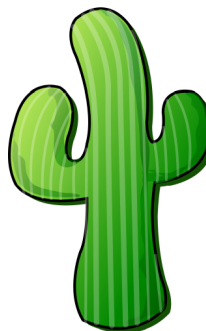


The Cacti Group

Cacti
Aggregate Plugin Version 0.75
Usage Guide



Brought to you by
Reinhard Scheck
gandalf@cacti.net



Table of Contents

Features.....	3
Installing the Plugin.....	4
Basic Installation Procedure.....	4
Providing Permissions to Users.....	6
Basic Usage.....	7
Aggregate Options Explained.....	8
Modifying the Default Traffic Graph Templates.....	13
Finding Aggregated Graphs.....	16
Color Templates.....	17
Create a new Color Template.....	17
Aggregate Cookbook.....	19
A Variation of Traffic Templates.....	19
Interface – Traffic (bits/sec) (default).....	20
Interface – Traffic (bits/sec) (AREA/STACK).....	20
Interface – Traffic (bits/sec) (peak).....	21
Interface – Traffic (bits/sec) (pos/neg).....	21
Interface – Traffic (bits/sec) (pure LINE).....	22
Examples.....	23
Using “Interface – Traffic (bits/sec) (default)”.....	23
Keep graph types, print all legend items, total similar data sources, reorder.....	23
Using “Interface – Traffic (bits/sec) (AREA/STACK)”.....	23
Convert to AREA/STACK Graph, No Totals, No Reordering.....	23
Using “Interface – Traffic (bits/sec) (peak)”.....	25
Using “Interface – Traffic (bits/sec) (pos/neg)”.....	25
Using “Interface – Traffic (bits/sec) (pure LINE)”.....	26
Convert to AREA/STACK Graph, No Totals, No Reordering.....	26
Convert to AREA/STACK Graph, Print totaling Legend Items Only, Total Similar Data Sources, No Reordering.....	27
Caveats.....	29
A Word on CDEFs.....	30
Totaling CDEFs.....	31
Make 0 CDEF.....	32
Caveat when using different Graph Templates.....	33



FEATURES

The AGGREGATE plugin was developed based on a HowTo from user Linegod (J.P. Pasnak). The main purpose is to reduce the amount of manual tasks required to create a new graph based on two or more already existing graphs: an aggregated graph.

So, basically, it does nothing magic; you can't do it all on your own. In fact, all graphs created by AGGREGATE are by all means normal Cacti graphs. You may place them into any tree, you can modify them, add new items, rename title or items.

Here's a brief list of all supported features:

- Suggest a new title based on the basic graph template, removing unwanted `|query_*|` and `|host_*|` stuff (`|query_*|` supported for data query type graphs only)
- Allow for `|host_*|` and `|query_*|` variables in AGGREGATE graph item text field (`|query_*|` supported for data query type graphs only)
- Graph item prefix defaults to `|host_hostname|` to distinguish graph items of different hosts
- optionally convert to AREA/STACK type graphs using sophisticated graph item type conversion
- optionally convert to all available LINEx graph item types (as Cacti does not yet support stacked LINEx, this is not supported by AGGREGATE as well. Cacti 0.8.8 will support all available RRDTool LINEx types)
- Support for Color Templates, e.g. all green “traffic_in” colors can be changed from light to dark green to make them visually distinguishable
- Color rotation: in case a color template holds less items than the aggregate, the color template will be applied multiple times
- programmatically add `<HR>` (line breaks) to avoid unwanted “concatenation” of graph items
- Support a “sort order”, especially required if base graph is AREA/STACK type to group similar graph items (e.g. all “traffic_in” first, all “traffic_out” next)
- Support for Totalling, either for “similar” or “all” data sources this feature auto-creates new CDEFs when necessary
- custom prefix on total GPRINTs
- Support “total only” aggregates, not printing the “elementary” graph items



INSTALLING THE PLUGIN

Basic Installation Procedure

The current manual is based on Cacti version 0.8.7g and Plugin Architecture (PIA) version 2.8. This plugin has been verified to work with this version. There has been no intention to drop support for older versions, use them at your own risk.

As a prerequisite, installing of the Plugin Architecture (PIA) is required, please see <http://cactiusers.org/> for help regarding this topic. Furthermore, it is assumed, that you already provided authorization to use “Plugin Management”.

Please download this plugin to the <path_cacti>/plugin directory and unpack the *.tgz file. All files will now reside in the <path_cacti>/plugin/aggregate directory.

CAVEAT:

Please make sure not to leave old „aggregate“, e.g. <path_cacti>/plugin/aggregate_old!

Activating this plugin depends on the Plugin Architecture (PIA) used., please go to that menu item, install and enable the new plugin.

Plugin Management (Cacti Version: 0.8.7g, Plugin Architecture Version: 2.8)

Search:

Go

Clear

Showing All 5 Rows

Actions	Name	Load Order	Description**	Type	Status	Author	Web Page	Version
	Autom8		Automate Cacti Tasks	General	Active	Reinhard Scheck	http://docs.cacti.net/plugin:autom8	0.35b4
	Aggregate		Create Aggregate Graphs	General	Not Installed	Reinhard Scheck	http://docs.cacti.net/plugin:aggregate	0.75b1
	Settings		Global Plugin Settings	System	Active	Jimmy Conner	http://cactiusers.org	0.6
	Rrdclean		RRD File Cleaner	General	Active	Reinhard Scheck	http://docs.cacti.net/plugin:rrdclean	0.40
	Nectar		Send Graphs via Email	General	Active	Reinhard Scheck	http://docs.cacti.net/plugin:nectar	0.28b6

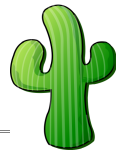
Showing All 5 Rows

NOTE: Please sort by 'Load Order' to change plugin load ordering.

NOTE: SYSTEM plugins can not be ordered.

Illustration 1: Install AGGREGATE

Select the “Install” knob, then then „Enable“. It should now look like








Showing All 5 Rows								
Actions	Name	Load Order	Description**	Type	Status	Author	Web Page	Version
	Autom8		Automate Cacti Tasks	General	Active	Reinhard Scheck	http://docs.cacti.net/plugin:autom8	0.35b4
	Aggregate		Create Aggregate Graphs	General	Active	Reinhard Scheck	http://docs.cacti.net/plugin:aggregate	0.75b1
	Settings		Global Plugin Settings	System	Active	Jimmy Conner	http://cactiusers.org	0.6
	Rrdclean		RRD File Cleaner	General	Active	Reinhard Scheck	http://docs.cacti.net/plugin:rrdclean	0.40
	Nectar		Send Graphs via EMail	General	Active	Reinhard Scheck	http://docs.cacti.net/plugin:nectar	0.28b6

Illustration 2: AGGREGATE installation finished



Providing Permissions to Users

Everybody who has Console access is now allowed to create new aggregate graphs.

But there's a second section, that comes with AGGREGATE, the Color Templates. To provide access to create, delete or modify color templates and items, please go to User Management and select the correct userid. Now, check the AGGREGATE checkboxes as shown below.

Realm Permissions

Graph Permissions

Graph Settings

Realm permissions control which sections of Cacti this user will have access to.

Realm Permissions	
<input checked="" type="checkbox"/> User Administration	<input checked="" type="checkbox"/> Update CDEF's
<input checked="" type="checkbox"/> Data Input	<input checked="" type="checkbox"/> Global Settings
<input checked="" type="checkbox"/> Update Data Sources	<input checked="" type="checkbox"/> Export Data
<input checked="" type="checkbox"/> Update Graph Trees	<input checked="" type="checkbox"/> Import Data
<input checked="" type="checkbox"/> Update Graphs	<input checked="" type="checkbox"/> Plugin Aggregate -> Create Color Template Items
<input checked="" type="checkbox"/> View Graphs	<input checked="" type="checkbox"/> Plugin Aggregate -> Create Color Templates
<input checked="" type="checkbox"/> Console Access	<input checked="" type="checkbox"/> Plugin Automate -> Maintain Automation Rules
<input checked="" type="checkbox"/> Update Round Robin Archives	<input checked="" type="checkbox"/> Plugin Management
<input checked="" type="checkbox"/> Update Graph Templates	<input checked="" type="checkbox"/> Plugin -> Nectar Reports Admin
<input checked="" type="checkbox"/> Update Data Templates	<input checked="" type="checkbox"/> Plugin -> Nectar Reports User
<input checked="" type="checkbox"/> Update Host Templates	<input checked="" type="checkbox"/> RRD Cleaner
<input checked="" type="checkbox"/> Data Queries	<input checked="" type="checkbox"/> Send Test Email

Illustration 3: Providing Permissions to Users



BASIC USAGE

Now, turn to Graph Management. It makes sense to apply a search filter and/or a graph template filter. Then, you will have to select the graphs you want to aggregate.

Graph ManagementAdd

Host: AnyTemplate: AnyGoClear

Search: trafficRows per Page: 30

Graph Title**	ID	Template Name	Size	
localhost - Traffic	136	Interface - Traffic (bits/sec)	120x600	<input checked="" type="checkbox"/>
localhost - Traffic (AREA/LINE)	140	Interface - Traffic (bits/sec)		<input type="checkbox"/>
localhost - Traffic (AREA/STACK)	141	Interface - Traffic (bits/sec)		<input type="checkbox"/>
localhost - Traffic (pure LINE)	142	Interface - Traffic (bits/sec)		<input type="checkbox"/>
wlan - Traffic	135	Interface - Traffic (bits/sec)		<input checked="" type="checkbox"/>
wlan - Traffic (AREA/LINE)	137	Interface - Traffic (bits/sec)		<input type="checkbox"/>
wlan - Traffic (AREA/STACK)	138	Interface - Traffic (bits/sec)		<input type="checkbox"/>
wlan - Traffic (pure LINE)	139	Interface - Traffic (bits/sec)		<input type="checkbox"/>

<< PreviousShowing Rows 1 to 8 of 8 [1]Next >>

Choose an action: DeleteGo

Illustration 4: Select Graphs from Graph Management

CAVEAT:
Please only select graphs based on a single graph template (see column „Template Name“). If you fail to do so, AGGREGATE will throw an error!

Now, please hit “Create Aggregate Graph”.



Aggregate Options Explained

There are a multitude of options available to tailor your aggregated graphs to your needs. Not all of them can be freely chosen and not all of them make sense, when applied to different types of graphs. But before digging deeper into use cases, let's first explain all those options.

Create Aggregate Graph [\[Click here for Help\]](#)

Are you sure you want to aggregate the following graphs?		The following data sources are in use by these graphs:	
<ul style="list-style-type: none">• Localhost - Traffic• wlan - Traffic		<ul style="list-style-type: none">• Localhost - Traffic - 192.168.1.50 - wlan0• wlan - Traffic - 192.168.1.50 - wlan0	
Title The new Title of the aggregated Graph.	<input type="text" value="Aggregate - Traffic"/>		
Prefix A Prefix for all GPRINT lines to distinguish e.g. different hosts.	<input type="text" value=" host_hostname "/>		
Graph Type Use this Option to create e.g. STACKed graphs. AREA/STACK: 1st graph keeps AREA/STACK items, others convert to STACK LINE1: all items convert to LINE1 items LINE2: all items convert to LINE2 items LINE3: all items convert to LINE3 items	<input type="button" value="Convert to AREA/STACK Graph"/>		
Totaling Please check those Items that shall be totaled in the 'Total' column, when selecting any totaling option here.	<input type="button" value="No Totals"/>		
Total Type Which type of totaling shall be performed.	<input type="button" value="Total Similar Data Sources"/>		
Prefix for GPRINT Totals A Prefix for all totaling GPRINT lines.	<input type="text" value="Total"/>		
Reorder Type Reordering of Graphs.	<input type="button" value="No Reordering"/>		

Illustration 5: Default Prompt for Aggregating Graphs (upper half)

Clicking for **help** opens this pdf file. Let me have some words on the quite complex data on that screen. On the upper left, you'll see the **list of graphs** selected previously. Please verify, that all needed graphs are included.

On the upper right, please notice the **list of related data sources**. The sequence may deviate from the graph list. Don't bother to see the same IP in this example, both graphs relate to my laptop's traffic.

The **Title** is pre-filled. The prefix always is „Aggregate“. Next comes the title taken from the first graph in raw format, but having all `|host_*|` or `|query_*|` removed as neither host nor query related variables make much sense for an aggregate. That is e.g. „`|host_description| - Traffic - |query_ifDescr|`“ turns to “ - Traffic” only. Of course, you will want to make it more meaningful to you.



Prefix allows you to distinguish graph items on the aggregate. Imagine aggregating traffic, like this example does, without it you won't be able to distinguish between all those aggregated graph items. You may discard the prefix, though. It is allowed, to use all available `|host_*|` variables here and/or any plain text you like. When using a data query, `|query_*|` type variables are allowed, too. Pay attention not to spend too many characters in order to avoid line wrap of the legend.

Graph Type is quite important to use. You may wonder why it is defaulted to create AREA/STACK graphs. Here's the reason why: Assume, you're aggregating an AREA graph. Without STACKing the second, third, ... graph item, all of them will overlap. Thus, only the last one will be seen (and perhaps parts of previous ones, if their values are higher).

- “Keep Graph Types” does not change any graph type at all
- “Convert to AREA/STACK Graph” keeps type of first graph as is. But for the next graphs, the graph type will be changed to AREA/STACK if it has been AREA or LINEx.
- “Convert to LINEx Graph”: Sometimes, it is recommended to have LINEx graphs instead. This will convert all graph types (AREA, STACK, LINEx) to the chosen LINEx (x=1,2,3).

CAVEAT:

It is recommended to change the standard Graph Template for „Traffic“. Please switch e.g. Traffic OUT to the negative y-axis. Else, Traffic IN and Traffic OUT will overlap on the positive y-axis (see Chapter “Modifying the Default Traffic Graph Templates“ below!

Totaling knows three options.

- The obvious first one is “No Totals”.
- Then, there is “Print all Legend Items”.
This one will show all legend entries (GPRINT, Comment) as usual but add some more lines for totals. The items that shall be totaled must be selected from the “Total” column (see below).
Graph Type is always LINE1, even when “Convert to AREA/STACK Graph” is used as it makes no sense to STACK totals on top of existing items.
Color is taken from Color Template of associated data source item.
- Last, there is “Print totaling Legend Items Only”.
This option will NOT print the normal legend items; it will print the totaling items only.



Graph Type is governed by the **Graph Type** chosen.
Color is taken from Color Template of associated data source item.

Total Type has two different options:

- „Total Similar Data Sources“ totals all e.g. “traffic_in” separated from “traffic_out”.
- „Total All Data Sources“ totals all data sources. Please see examples below on how to use them.

Prefix for GPRINT Totals allows you to define a prefix to be printed in front of the totaling line of the legend. The field is editable only if any “Totaling” option has been chosen. Of course, you may change that prefix.

- for “Total Similar Data Sources”, the default is “Total”,
- for “Total All Data Sources” it is “All Items”.

Reorder Type allows you to reorder graph items of the aggregate.

- “No Reordering”
This means, that the selected graph items of each graph, graph by graph, will be used for the aggregate. E.g. for a traffic graph, this will lead to a sequence of “Traffic In – Traffic Out – Traffic In – Traffic Out ...”.
Totaled items will not be reordered; they will be placed at the end of the legend.
- “Data Source, Graph”
This will first order by data source, then by graph. E.g. for a traffic graph, this will lead to a sequence of “Traffic In – Traffic In ... – Traffic Out – Traffic Out ...”.
For STACKed graphs, totaled items will not be reordered; they will be placed at the end of the legend.
For non-STACKed graphs, totaled items will be reordered as well, e.g. totaled “Traffic In” will be placed after all “Traffic In” items and before “Traffic Out” items.



Graph Templates Items is build based on the first graph selected, showing up in the lower part of the aggregate screen.. Please pay attention to this section to make the most of your new aggregate.

Graph Template Items							
Graph Item	Data Source	Graph Item Type	CF Type	Item Color	Color Template	Skip	Total
Item # 1	Inbound	AREA	AVERAGE	00FF00	None	<input type="checkbox"/>	<input type="checkbox"/>
Item # 2	Current:	GPRINT	LAST			<input type="checkbox"/>	<input type="checkbox"/>
Item # 3	Average:	GPRINT	AVERAGE			<input type="checkbox"/>	<input type="checkbox"/>
Item # 4	Maximum:	GPRINT	MAX			<input type="checkbox"/>	<input type="checkbox"/>
Item # 5	Outbound	AREA	AVERAGE	00A0FF	None	<input type="checkbox"/>	<input type="checkbox"/>
Item # 6	Current:	GPRINT	LAST			<input type="checkbox"/>	<input type="checkbox"/>
Item # 7	Average:	GPRINT	AVERAGE			<input type="checkbox"/>	<input type="checkbox"/>
Item # 8	Maximum:	GPRINT	MAX			<input type="checkbox"/>	<input type="checkbox"/>

Please confirm

Cancel
Continue

Illustration 6: Aggregate Graphs, showing List of Graph Items (lower half)

There are three columns to pay attention to.

Color Template governs the coloring of the aggregated graph items. Why's this? Using a single graph template will usually result in graph items like e.g. „Traffic In“ having same color on all graphs. On an aggregate, you would thus not be able to distinguish between them. That's where color templates come in. They simply define a sequence of colors, each of them assigned to an aggregated graph item in turn. And if the aggregate has more items than the Color Template, we will wrap the Color Template automatically.

Creating a color template like a „rainbow“ of colors allows you now to assign a set of colors in a single run! Do not forget to assign different color templates to different graph items!

As totaling will add a new line that was not yet present, the color for that item will be taken from the associated data source item or the related color template, if chosen.

Skip allows you to skip the checked item in the aggregate. There's a new, automatic <HR> mechanism to cope with <hard_returns> that will be dropped now. A skipped <HR> will percolate up the list to the previous item to keep line feeds in place. The mechanism will even introduce new <HR>'s at end of a graph template.

Else, concatenating two graphs without <HR> in between will create ugly legends. This is required as well to make automatic legend adjustment work (but be aware, that this



requires always a fixed sized font for legends).

Total governs the magic of totaling graph items. I did not make up my mind to create an algorithm for it. So you are required to check exactly those lines you want to see on the total legend.

CAVEAT:

It is recommended to check required items! It is NOT sufficient to simply check the totaling options in the upper half.

CAVEAT:

When using **Total All Data Sources**, it makes no sense in most cases to check more than a “single set” of data sources. E.g. on a traffic graph, it's ok to select “*input traffic items*” only. In case you check “*output traffic items*” as well, you will get two totaling lines showing the same data: “total all data sources”.

For the totaling items, **Prefix for GPRINT Totals** is always used instead of **Prefix** to prepend the legend line.

Totaling Similar Data Sources creates e.g. a total of each different data source referred to in the graph. E.g. In this example, there are two: „Traffic In“ and „Traffic Out“. So you want to check at least the AREA/LINEx graph items and all additional GPRINT items.

But when using e.g. 95th percentile graphs or bandwidth COMMENTS, you will want to skip them on the total.

Please see examples below for more.



Modifying the Default Traffic Graph Templates

The default Cacti Graph Templates are mimicked after some very famous templates like those used by MRTG. “traffic_in” and “traffic_out” are both plotted to the positive y-axis, the latter as a LINE1.

For use with AGGREGATE, this is not the best choice. And people often want to plot outbound traffic to the negative y-axis for a better understanding.

You may either copy the graph templates you're going to change to preserve standard templates or change the standard to apply changes to all existing graphs with a simple swish of your magic wand called knowledge.

As a first step, a CDEF is required to “Turn Bytes into Bits, make negative”:

CDEF's [edit: Turn Bytes into bits, make negative]

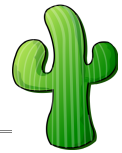
Name
A useful name for this CDEF.

```
cdef=CURRENT_DATA_SOURCE, 8, *, -1, *
```

CDEF Items			Add
Item	Item Value		
Item #1	Special Data Source: CURRENT_DATA_SOURCE	⬇ ⬆	✖
Item #2	Custom String: 8	⬇ ⬆	✖
Item #3	Operator: *	⬇ ⬆	✖
Item #4	Custom String: -1	⬇ ⬆	✖
Item #5	Operator: *	⬇ ⬆	✖

Illustration 7: Turn Bytes into Bits, make negative

Please apply this CDEF to the outbound traffic item and make it an AREA as well:

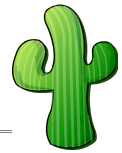


Graph Template Items [edit graph: Interface - Traffic (bits/sec)]

Data Source [Field Not Templated] The data source to use for this graph item.	Interface - Traffic - (traffic_out)
Color The color to use for the legend.	00A0FF
Opacity/Alpha Channel The opacity/alpha channel of the color. Not available for rrdtool-1.0.x.	60%
Graph Item Type How data for this item is represented visually on the graph.	AREA
Consolidation Function How data for this item is represented statistically on the graph.	AVERAGE
CDEF Function A CDEF (math) function to apply to this item on the graph.	Turn Bytes into bits, make negative
Value The value of an HRULE or VRULE graph item.	
GPRINT Type If this graph item is a GPRINT, you can optionally choose another format here. You can define additional types under "GPRINT Presets".	Normal
Text Format Text that will be displayed on the legend for this graph item.	Outbound
Insert Hard Return Forces the legend to the next line after this item.	<input type="checkbox"/> Insert Hard Return
Sequence	6

Illustration 8: Apply CDEF to Outbound Traffic Item

Now, one more tweak is required for the template itself. The default auto-scaling option does not allow for negative numbers to be plotted on the graph. That's why we now switch to *-alt-autoscale (ignore given limits)*. See



Graph Template	
Title (--title)	<input type="text" value=" host_description - Traffic"/>
<input type="checkbox"/> Use Per-Graph Value (Ignore this Value)	
Image Format (--imgformat)	<input type="text" value="PNG"/>
<input type="checkbox"/> Use Per-Graph Value (Ignore this Value)	
Height (--height)	<input type="text" value="120"/>
<input type="checkbox"/> Use Per-Graph Value (Ignore this Value)	
Width (--width)	<input type="text" value="600"/>
<input type="checkbox"/> Use Per-Graph Value (Ignore this Value)	
Slope Mode (--slope-mode)	<input checked="" type="checkbox"/> Slope Mode (--slope-mode)
<input type="checkbox"/> Use Per-Graph Value (Ignore this Value)	
Auto Scale	<input checked="" type="checkbox"/> Auto Scale
<input type="checkbox"/> Use Per-Graph Value (Ignore this Value)	
Auto Scale Options	<input checked="" type="radio"/> Use --alt-autoscale (ignoring given limits)
<input type="checkbox"/> Use Per-Graph Value (Ignore this Value)	
	<input type="radio"/> Use --alt-autoscale-max (accepting a lower limit)
	<input type="radio"/> Use --alt-autoscale-min (accepting an upper limit, requires rrdtool 1.2.x)
	<input type="radio"/> Use --alt-autoscale (accepting both limits, rrdtool default)

Illustration 9: Use --alt-autoscale

You may want to apply those changes to all traffic graph templates using bits. That makes

- [Interface - Traffic \(bits/sec\)](#)
- [Interface - Traffic \(bits/sec, 95th Percentile\)](#)
- [Interface - Traffic \(bits/sec, Total Bandwidth\)](#)



FINDING AGGREGATED GRAPHS

Due to the aggregated nature of the graphs, they are in general not associated to a single host. When searching an aggregate, you will therefor not find in when searching for a specific host. In fact, you'll find them when filtering for Host "None"; in other words: filter for all non-host related graphs:

Graph Management

Host: **None**

Template: **Any**

Search: **traffic**

Rows per Page: **30**

<< Previous

Showing Rows 1 to 1 of 1 [1]

Graph Title**	ID	Template Name	Size
Aggregate - Traffic	143	None	120

<< Previous

Showing Rows 1 to 1 of 1 [1]

Illustration 10: Find Aggregated Graphs by Host Filter

The same applies to the "template" filter. You may argue, that all aggregated graphs stem from the same basic graph template, expecting the aggregated graph to be related to that very graph template as well.

But hold on: The new aggregate will have many more graph items, and, if you want, even totaling items that have not been present at all. That's why it is not possible to associate it with any existing graph template.

In fact, if you want to search by template, you will again have to set the template filter to "None":

Graph Management

Host: **Any**

Template: **None**

Search: **traffic**

Rows per Page: **30**

<< Previous

Showing Rows 1 to 1 of 1 [1]

Graph Title**	ID	Template Name	Size
Aggregate - Traffic	143	None	120

<< Previous

Showing Rows 1 to 1 of 1 [1]

Illustration 11: Find Aggregated Graphs by Graph Template Filter



COLOR TEMPLATES

When adding some graph items of same type, e.g. “traffic_in” and “traffic_out” to a single aggregated graph, it makes sense not to use a single color for all graph items of same type.

E.g. instead of using green (RGB x'00FF00) for “traffic_in”, you may want to associate colors from light green to dark green. That's what a Color Template is used for.

Create a new Color Template

Now let's see how to create them. Make sure, that your userid was provided the realm to access “Color Templates”. Then, move to that menu option:

Color Templates		Add
Search:	<input type="text"/>	Rows per Page: 30 Go Clear
<< Previous Showing Rows 1 to 4 of 4 [1] Next >>		
Template Title**	<input type="checkbox"/>	
Green: dark-light, 16	<input type="checkbox"/>	
Red: light yellow-dark red, 8	<input type="checkbox"/>	
Red: light-dark, 16	<input type="checkbox"/>	
Yellow: light-dark, 4	<input type="checkbox"/>	
<< Previous Showing Rows 1 to 4 of 4 [1] Next >>		
Choose an action:		Delete Go

Illustration 12: Color Templates

Add a new Color Template by clicking “Add”:

Color Template [new]	
Name A useful name for this Template.	<input type="text" value="Color Template Test"/>
Cancel Create	

Illustration 13: Add a new Color Template

Hit “Create” and “Add” new “Color Template Items”, one by one:



Save Successful.

Color Template [edit: Color Template Test]

Name
A useful name for this Template.

Color Template Test

Color Template Items [edit: Color Template Test] [Add](#)

Color Item	Seq	Item Color
No Items		

Return

Save

Illustration 14: Add Items to Color Template



AGGREGATE COOKBOOK

A Variation of Traffic Templates

The cookbook is based on variations of the default “Interface Traffic” Graph Template. I've chosen this one because almost everyone should be able to reproduce the examples. To show the different use cases, I've copied this graph template multiple times; varying different items of the template. Here's an overview:

Graph TemplatesAdd

Search: GoClear

<< Previous	Showing Rows 1 to 9 of 9 [1]	Next >>
Template Title**		<input type="checkbox"/>
Interface - Traffic (bits/sec) (default)		<input type="checkbox"/>
Interface - Traffic (bits/sec) (AREA/STACK)		<input type="checkbox"/>
Interface - Traffic (bits/sec) (peak)		<input type="checkbox"/>
Interface - Traffic (bits/sec) (pos/neg)		<input type="checkbox"/>
Interface - Traffic (bits/sec) (pure LINE)		<input type="checkbox"/>
Interface - Traffic (bits/sec, 95th Percentile)		<input type="checkbox"/>
Interface - Traffic (bits/sec, Total Bandwidth)		<input type="checkbox"/>
Interface - Traffic (bytes/sec)		<input type="checkbox"/>
Interface - Traffic (bytes/sec, Total Bandwidth)		<input type="checkbox"/>

<< Previous

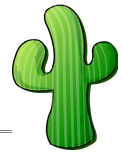
Showing Rows 1 to 9 of 9 [1]

Next >>

↳

Choose an action: Delete ↕ Go

Illustration 15: Predefined Traffic Graph Templates



Interface – Traffic (bits/sec) (default)

This is the default Traffic Graph Template. It shows “traffic_in” as an AREA and “traffic_out” as a LINE1.

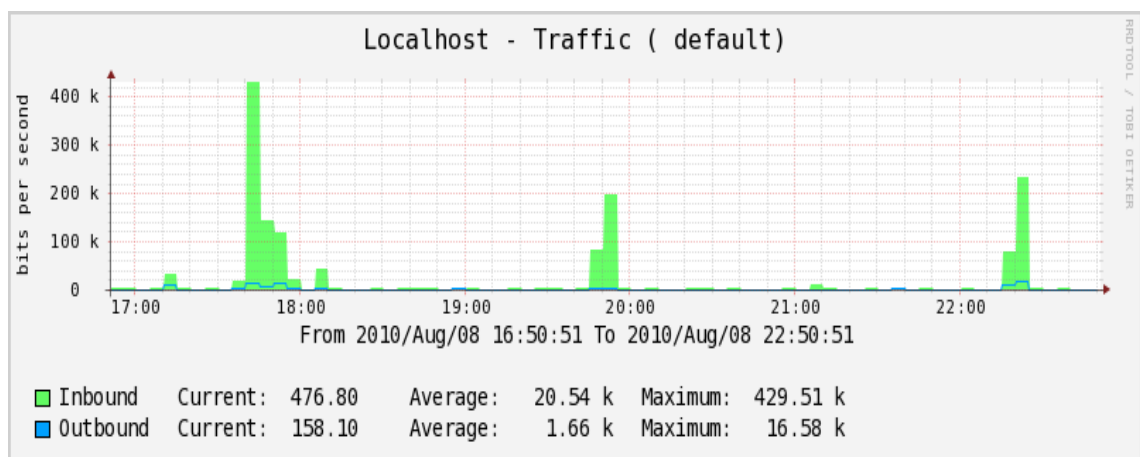


Illustration 16: Interface – Traffic (bits/sec) (default)

Interface – Traffic (bits/sec) (AREA/STACK)

It shows “traffic_in” as an AREA and “traffic_out” as a n AREA/STACK. This does not make much sense as a traffic template; but keep in mind that this is used as an example.

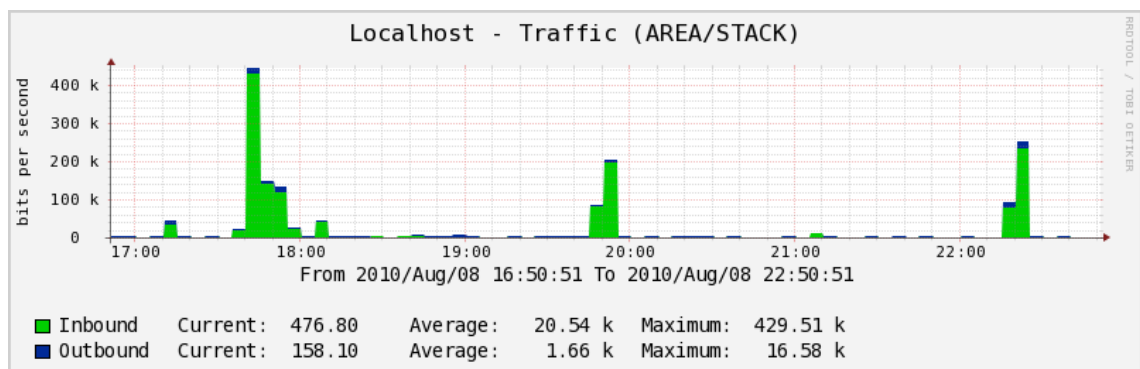


Illustration 17: Interface – Traffic (bits/sec) (AREA/STACK)



Interface – Traffic (bits/sec) (peak)

This is much like the template below, but an additional LINE1 item is added for “traffic_in” and “traffic_out”, showing the MAX consolidation function.

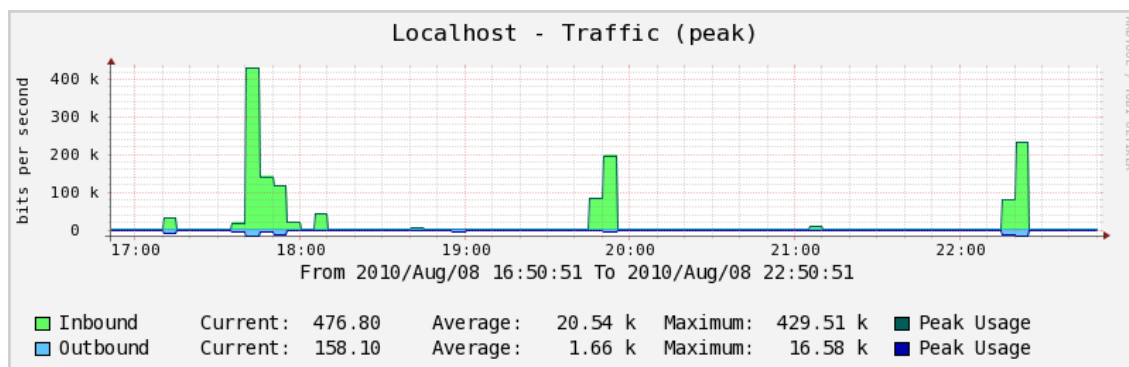


Illustration 18: Interface – Traffic (bits/sec) (peak)

Interface – Traffic (bits/sec) (pos/neg)

It shows “traffic_in” as an AREA on the positive y-axis and “traffic_out” as an AREA on the negative y-axis.

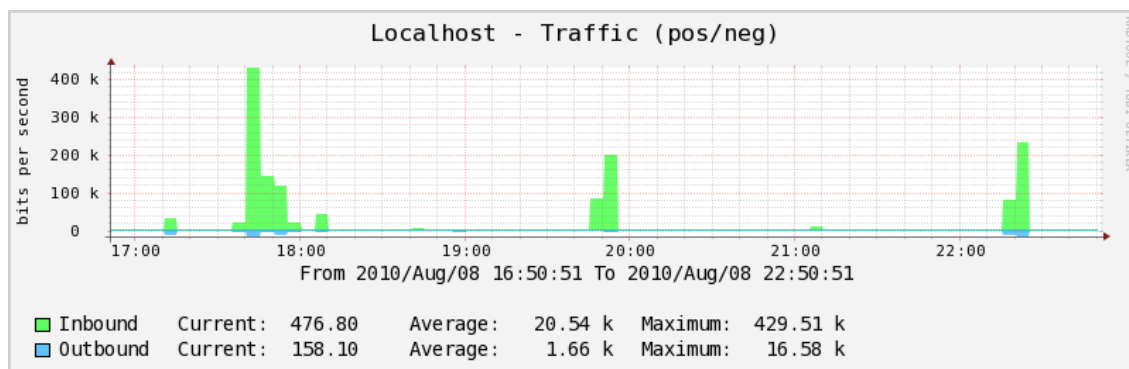
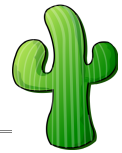


Illustration 19: Interface – Traffic (bits/sec) (pos/neg)



Interface – Traffic (bits/sec) (pure LINE)

This one looks very much like the default traffic graph template. But in this case, both “traffic_in” and “traffic_out” are drawn as LINE1 on the positive y-axis.

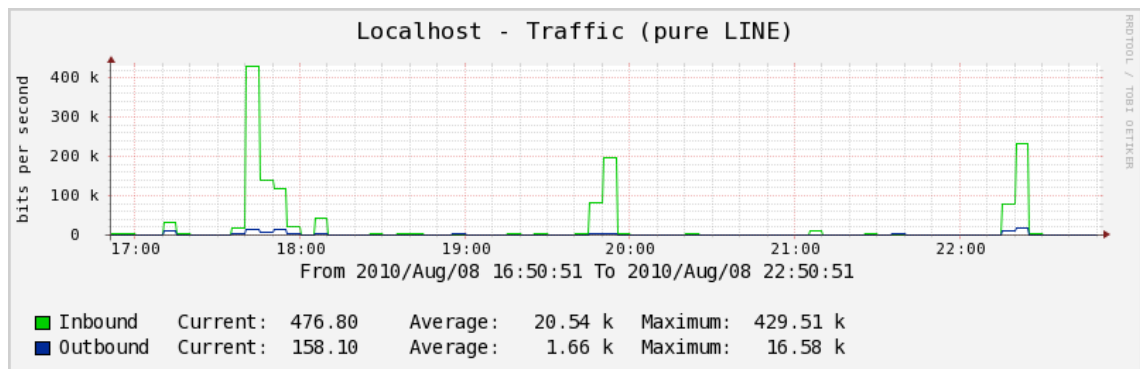


Illustration 20: Interface – Traffic (bits/sec) (pure LINE)



Examples

Using “Interface – Traffic (bits/sec) (default)”

Keep graph types, print all legend items, total similar data sources, reorder

Base graph is Interface – Traffic (bits/sec) (default) on page 20. The aggregate shows the same host twice, once named “gandalf-wlan” and once “localhost (127.0.0.1)”.

This examples shows two features:

- Totaling “Similar Data Sources”, printing all legend items
- Reordering by data sources

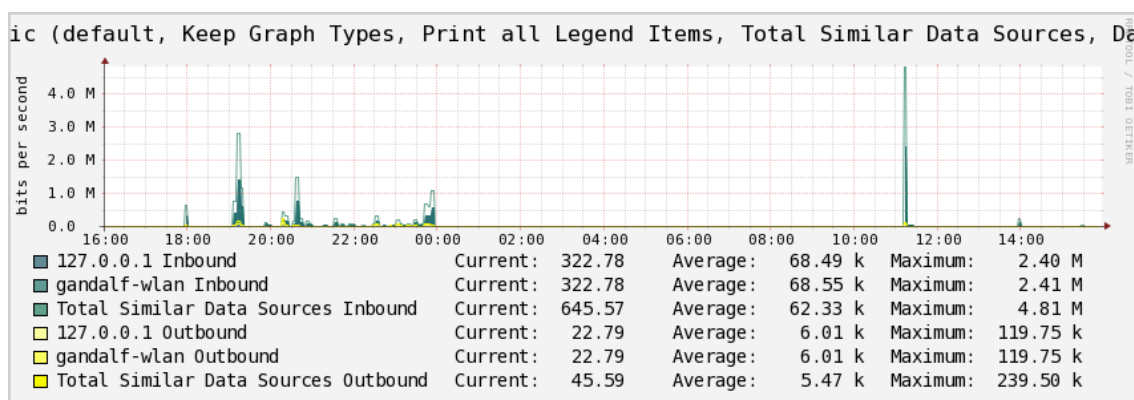


Illustration 21: Total similar data sources and reorder

First, all items are added to the new aggregate graph. In this example, the graph types are kept (AREA and LINE1, respectively)

Then, the total is build; in this case “Total Similar Data Sources”, both for inbound and outbound traffic. The total is drawn as a line (hard to notice on this graph).

At last, the reordering takes place. So all inbound items are grouped and all outbound as well.

Using “Interface – Traffic (bits/sec) (AREA/STACK)”

Convert to AREA/STACK Graph, No Totals, No Reordering

Base graph is Interface – Traffic (bits/sec) (AREA/STACK) on page 20. The aggregate



shows the same host twice, once named “gandalf-wlan” and once “localhost (127.0.0.1)”.

When using a base graph of type Interface – Traffic (bits/sec) (AREA/STACK) without conversion (“Keep Graph Types”), the result is

Graph Items [edit: Aggregate - Traffic (Area/Stack, Keep Graph Types, No Totals, Total Similar Data Sources, No Reordering)]						Add
Graph Item	Data Source	Graph Item Type	CF Type	Item Color		
Item # 1	(traffic_in): 127.0.0.1 Inbound	AREA	AVERAGE	004359	↕ ↗	✗
Item # 2	(traffic_in): Current:	GPRINT	LAST		↕ ↗	✗
Item # 3	(traffic_in): Average:	GPRINT	AVERAGE		↕ ↗	✗
Item # 4	(traffic_in): Maximum:<HR>	GPRINT	MAX		↕ ↗	✗
Item # 5	(traffic_out): 127.0.0.1 Outbound	STACK	AVERAGE	FAFD9E	↕ ↗	✗
Item # 6	(traffic_out): Current:	GPRINT	LAST		↕ ↗	✗
Item # 7	(traffic_out): Average:	GPRINT	AVERAGE		↕ ↗	✗
Item # 8	(traffic_out): Maximum:<HR>	GPRINT	MAX		↕ ↗	✗
Item # 9	(traffic_in): gandalf-wlan Inbound	AREA	AVERAGE	005D57	↕ ↗	✗
Item # 10	(traffic_in): Current:	GPRINT	LAST		↕ ↗	✗
Item # 11	(traffic_in): Average:	GPRINT	AVERAGE		↕ ↗	✗
Item # 12	(traffic_in): Maximum:<HR>	GPRINT	MAX		↕ ↗	✗
Item # 13	(traffic_out): gandalf-wlan Outbound	STACK	AVERAGE	F9FD5F	↕ ↗	✗
Item # 14	(traffic_out): Current:	GPRINT	LAST		↕ ↗	✗
Item # 15	(traffic_out): Average:	GPRINT	AVERAGE		↕ ↗	✗
Item # 16	(traffic_out): Maximum:<HR>	GPRINT	MAX		↕ ↗	✗

Illustration 22: Keep Graph Types, No Totals, No Reordering

In fact, that graph consists of AREA and STACK only. But as explained in Caveats on page 29, both AREAs do overlap. It is very likely that this won't make you happy.

You will get a different result, when using “Convert to AREA/STACK graph”, even if this sounds weird. This conversion option will sweep the resulting graph, keeping a single AREA only and converting the rest to STACK.

The result may look better than the first one



Graph Items [edit: Aggregate - Traffic (Area/Stack, Convert to AREA/STACK Graph, No Totals, Total Similar Data Sources, No Reordering)]						Add
Graph Item	Data Source	Graph Item Type	CF Type	Item Color		
Item # 1	(traffic_in): 127.0.0.1 Inbound	AREA	AVERAGE	004359	↕ ↗	✖
Item # 2	(traffic_in): Current:	GPRINT	LAST		↕ ↗	✖
Item # 3	(traffic_in): Average:	GPRINT	AVERAGE		↕ ↗	✖
Item # 4	(traffic_in): Maximum:<HR>	GPRINT	MAX		↕ ↗	✖
Item # 5	(traffic_out): 127.0.0.1 Outbound	STACK	AVERAGE	F4FD9E	↕ ↗	✖
Item # 6	(traffic_out): Current:	GPRINT	LAST		↕ ↗	✖
Item # 7	(traffic_out): Average:	GPRINT	AVERAGE		↕ ↗	✖
Item # 8	(traffic_out): Maximum:<HR>	GPRINT	MAX		↕ ↗	✖
Item # 9	(traffic_in): gandalf-wlan Inbound	STACK	AVERAGE	005D57	↕ ↗	✖
Item # 10	(traffic_in): Current:	GPRINT	LAST		↕ ↗	✖
Item # 11	(traffic_in): Average:	GPRINT	AVERAGE		↕ ↗	✖
Item # 12	(traffic_in): Maximum:<HR>	GPRINT	MAX		↕ ↗	✖
Item # 13	(traffic_out): gandalf-wlan Outbound	STACK	AVERAGE	F9FD5F	↕ ↗	✖
Item # 14	(traffic_out): Current:	GPRINT	LAST		↕ ↗	✖
Item # 15	(traffic_out): Average:	GPRINT	AVERAGE		↕ ↗	✖
Item # 16	(traffic_out): Maximum:<HR>	GPRINT	MAX		↕ ↗	✖

Illustration 23: Convert to AREA/STACK Graph, No Totals, No Reordering

Using “Interface – Traffic (bits/sec) (peak)”

Base graph is Interface – Traffic (bits/sec) (peak) on page 21. The aggregate shows the same host twice, once named “gandalf-wlan” and once “localhost (127.0.0.1)”.

This is a quite difficult beast. At the first glance, it makes sense to “Keep Graph Types”. But then, the AREAs overlap.

Then, you may want to “Convert to AREA/STACK”. This will handle the AREAs well, but will convert the “Peak” LINE1 to STACK as well. That makes no sense at all. If you want to choose this conversion, it is best to skip the “Peak” items. If you want to keep them, some kind of post-processing is required to change the “Peak STACKs” to “Peak LINE1” again. That's my favorite for this type of graphs.

So, at last, there's “Convert to LINE1”. Nothing bad will happen here, but you will receive a lot of LINES. Looks ugly.

Using “Interface – Traffic (bits/sec) (pos/neg)”

Base graph is Interface – Traffic (bits/sec) (pos/neg) on page 21. The aggregate shows the same host twice, once named “gandalf-wlan” and once “localhost (127.0.0.1)”.

This graph looks quite the same as Interface – Traffic (bits/sec) (peak) on page 21, except for the missing peaks. Referring to the discussion above, this eases the “Convert to AREA/STACK” solution.

We will use quite the same parameter set as in Using “Interface – Traffic (bits/sec)



(default)” on page 23. You will expect a quite similar result. And yes, the only difference is that the output related data is drawn on the negative y-axis. Even the totaled output LINEx is automagically printed on the lower half.

Graph Items [edit: Aggregate - Traffic (Pos/Neg, Convert to AREA/STACK Graph, Print all Legend Items, Total Similar Data Sources, Data Source, Graph)]						Add
Graph Item	Data Source	Graph Item Type	CF Type	Item Color		
Item # 1	(traffic_in): 127.0.0.1 Inbound	AREA	AVERAGE	004359	↕ ↗	✖
Item # 2	(traffic_in): Current:	GPRINT	LAST		↕ ↗	✖
Item # 3	(traffic_in): Average:	GPRINT	AVERAGE		↕ ↗	✖
Item # 4	(traffic_in): Maximum: <HR>	GPRINT	MAX		↕ ↗	✖
Item # 5	(traffic_in): gandalf-wlan Inbound	STACK	AVERAGE	005D57	↕ ↗	✖
Item # 6	(traffic_in): Current:	GPRINT	LAST		↕ ↗	✖
Item # 7	(traffic_in): Average:	GPRINT	AVERAGE		↕ ↗	✖
Item # 8	(traffic_in): Maximum: <HR>	GPRINT	MAX		↕ ↗	✖
Item # 9	(traffic_out): 127.0.0.1 Outbound	AREA	AVERAGE	FAFD9E	↕ ↗	✖
Item # 10	(traffic_out): Current:	GPRINT	LAST		↕ ↗	✖
Item # 11	(traffic_out): Average:	GPRINT	AVERAGE		↕ ↗	✖
Item # 12	(traffic_out): Maximum: <HR>	GPRINT	MAX		↕ ↗	✖
Item # 13	(traffic_out): gandalf-wlan Outbound	STACK	AVERAGE	F9FD5F	↕ ↗	✖
Item # 14	(traffic_out): Current:	GPRINT	LAST		↕ ↗	✖
Item # 15	(traffic_out): Average:	GPRINT	AVERAGE		↕ ↗	✖
Item # 16	(traffic_out): Maximum: <HR>	GPRINT	MAX		↕ ↗	✖
Item # 17	(traffic_in): Total Similar Data Sources Inbound	LINE1	AVERAGE	00694A	↕ ↗	✖
Item # 18	(traffic_in): Current:	GPRINT	LAST		↕ ↗	✖
Item # 19	(traffic_in): Average:	GPRINT	AVERAGE		↕ ↗	✖
Item # 20	(traffic_in): Maximum: <HR>	GPRINT	MAX		↕ ↗	✖
Item # 21	(traffic_out): Total Similar Data Sources Outbound	LINE1	AVERAGE	F5F800	↕ ↗	✖
Item # 22	(traffic_out): Current:	GPRINT	LAST		↕ ↗	✖
Item # 23	(traffic_out): Average:	GPRINT	AVERAGE		↕ ↗	✖
Item # 24	(traffic_out): Maximum: <HR>	GPRINT	MAX		↕ ↗	✖

Illustration 24: Convert to AREA/STACK Graph, Print all Legend Items, Total Similar Data Sources, Reorder

Using “Interface – Traffic (bits/sec) (pure LINE)”

Base graph is Interface – Traffic (bits/sec) (pure LINE) on page 22. The aggregate shows the same host twice, once named “gandalf-wlan” and once “localhost (127.0.0.1)”.

Convert to AREA/STACK Graph, No Totals, No Reordering

This serves as an example how to convert a plain LINEx graph to AREA/STACK. This results in



Graph Items [edit: Aggregate - Traffic (Line, Convert to AREA/STACK Graph, No Totals, Total Similar Data Sources, No Reordering)]						Add
Graph Item	Data Source	Graph Item Type	CF Type	Item Color		
Item # 1	(traffic_in): 127.0.0.1 Inbound	AREA	AVERAGE	004359	↕ ↗	✖
Item # 2	(traffic_in): Current:	GPRINT	LAST		↕ ↗	✖
Item # 3	(traffic_in): Average:	GPRINT	AVERAGE		↕ ↗	✖
Item # 4	(traffic_in): Maximum:<HR>	GPRINT	MAX		↕ ↗	✖
Item # 5	(traffic_out): 127.0.0.1 Outbound	STACK	AVERAGE	FAFD9E	↕ ↗	✖
Item # 6	(traffic_out): Current:	GPRINT	LAST		↕ ↗	✖
Item # 7	(traffic_out): Average:	GPRINT	AVERAGE		↕ ↗	✖
Item # 8	(traffic_out): Maximum:<HR>	GPRINT	MAX		↕ ↗	✖
Item # 9	(traffic_in): gandalf-wlan Inbound	STACK	AVERAGE	005D57	↕ ↗	✖
Item # 10	(traffic_in): Current:	GPRINT	LAST		↕ ↗	✖
Item # 11	(traffic_in): Average:	GPRINT	AVERAGE		↕ ↗	✖
Item # 12	(traffic_in): Maximum:<HR>	GPRINT	MAX		↕ ↗	✖
Item # 13	(traffic_out): gandalf-wlan Outbound	STACK	AVERAGE	F9FD5F	↕ ↗	✖
Item # 14	(traffic_out): Current:	GPRINT	LAST		↕ ↗	✖
Item # 15	(traffic_out): Average:	GPRINT	AVERAGE		↕ ↗	✖
Item # 16	(traffic_out): Maximum:<HR>	GPRINT	MAX		↕ ↗	✖

Illustration 25: Convert to AREA/STACK Graph, No Totals, No Reordering

Convert to AREA/STACK Graph, Print totaling Legend Items Only, Total Similar Data Sources, No Reordering

This example shows a “totaling only” aggregate. Especially when aggregating many items, the legend may rise and the graph area may look confusing.

The option to use is “Print Totaling Legend Items Only”. This will not draw the basic items used for the aggregate.

Graph Items [edit: Aggregate - Traffic (Line, Convert to AREA/STACK Graph, Print totaling Legend Items Only, Total Similar Data Sources, No Reordering)]						Add
Graph Item	Data Source	Graph Item Type	CF Type	Item Color		
Item # 1	(traffic_in):	AREA	AVERAGE	004359	↕ ↗	✖
Item # 2	(traffic_out):	STACK	AVERAGE	FAFD9E	↕ ↗	✖
Item # 3	(traffic_in):	STACK	AVERAGE	005D57	↕ ↗	✖
Item # 4	(traffic_out):	STACK	AVERAGE	F9FD5F	↕ ↗	✖
Item # 5	(traffic_in): Total Similar Data Sources Inbound	STACK	AVERAGE	00694A	↕ ↗	✖
Item # 6	(traffic_in): Current:	GPRINT	LAST		↕ ↗	✖
Item # 7	(traffic_in): Average:	GPRINT	AVERAGE		↕ ↗	✖
Item # 8	(traffic_in): Maximum:<HR>	GPRINT	MAX		↕ ↗	✖
Item # 9	(traffic_out): Total Similar Data Sources Outbound	STACK	AVERAGE	F5F800	↕ ↗	✖
Item # 10	(traffic_out): Current:	GPRINT	LAST		↕ ↗	✖
Item # 11	(traffic_out): Average:	GPRINT	AVERAGE		↕ ↗	✖
Item # 12	(traffic_out): Maximum:<HR>	GPRINT	MAX		↕ ↗	✖

Illustration 26: Convert to AREA/STACK Graph, Print totaling Legend Items Only, Total Similar Data Sources, No Reordering

This graph seems to include too many items. You are expecting to find totaling items only, do you?

But this won't work. “Total Similar Data Sources Nodups” requires, that the items to be



totalled are present! AGGREGATE therefor does not skip them. It uses the “MAKE 0” CDEF to suppress the data that would else be printed on the graph area.

As a conversion too AREA/STACK is requested, the totals are printed as STACKs. Those STACKs are based on the first AREA, that has been CDEFed to 0. All in all you get what you want.

Caveats

Here's a small compilation of selectable parameters that won't make sense due to various reasons mentioned below.

No.	Graph Type	Conversion	Totaling	Total Type	Reordering	Remarks
1	AREA + LINE	Keep Graph Types	Any	Any	Any	AREAs overlap; e.g. only last AREA may be visible
2	Any	Any	Print totaling Legend Items Only	Any	Any	Reordering has no visible effect when totaling entries are the only one printed.
3	AREA + STACK	Keep Graph Types	Any	Any	by data source	Two or more AREAs will appear; they again will overlap. The STACKed items will be stacked onto the last AREA, not onto the “related” AREA. This is how RRDTool works.
4	AREA + STACK	Keep Graph Types	Print all Legend Items	Any	by data source	At least one totaling LINE1 will drawn in between normal entries. All STACKed entries shwoing up after this totaling LINE1 will be stacked onto that very LINE1, not onto the “related” AREA.
5	AREA, STACK + LINE _x	Convert to AREA/STACK	Print totaling Legend Items Only	Any	No	This type of graph is e.g. used when graphing AVERAGE and MAX in one template. The conversion will change the LINE _x to STACK which may not be what you want. Consider using conversion type “Keep Graph Types”
6	AREA (pos) + AREA (neg)	Keep Graph Types	Any but “Print totaling Legend Items Only”	Any	Any	AREA is drawn on positive and negative y-axis. Without conversion of graph type, the AREAs will overlap; see item 3. When “Print totaling Legend Items Only” is chosen, this may still make sense.



A Word on CDEFs

Well, until now everything seems to be quite straightforward. CDEFs were already mentioned in Chapter “Modifying the Default Traffic Graph Templates“. What's wrong with them?

When aggregating graph items, this is more or less beading them one after another, mostly just as they are in the selected graphs. Then, there is some color magic, skipping code, prefixing text and <HR> stuff.

But totaling is worse, much worse. The graph items themselves do not require any change. But the totaling line requires a CDEF that holds something like “TOTAL_ALL_DATA_SOURCES_NODUPS” or “SIMILAR_DATA_SOURCES_NODUPS” where currently “CURRENT_DATA_SOURCE” is listed.

So this plugin generates new CDEFs. It fetches the CDEF from the original graph item and resolves it to plain text. Then, depending on the totaling action selected, “CURRENT_DATA_SOURCE” is replaced by the totaling magic required.

Now we have a new CDEF. To avoid storing duplicate CDEFs, all existing CDEFs now are scanned and compared to the new CDEF. On match, the existing CDEF is used. If no match is found, the new CDEF is stored.

The title of the new CDEF is taken from the original CDEF, but a string is prepended. Either “_AGGREGATE ALL “ or “_AGGREGATE SIMILAR” is used to distinguish. The underscore is used for sake of sorting them to the bottom of the CDEF list.

This is a list of all CDEFs on my system after quite a lot of aggregation:

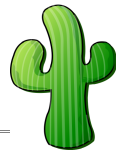


<< Previous		Showing Rows 1 to 16 of 16 [1]	Next >>
CDEF Title**	<input type="checkbox"/>		
Make Per 5 Minutes	<input type="checkbox"/>		
Make Stack Negative	<input type="checkbox"/>		
Multiply by -1024	<input type="checkbox"/>		
Multiply by 1024	<input type="checkbox"/>		
Total All Data Sources	<input type="checkbox"/>		
Total All Data Sources, Multiply by 1024	<input type="checkbox"/>		
Total similar data sources	<input type="checkbox"/>		
Turn Bytes into Bits	<input type="checkbox"/>		
Turn Bytes into bits, make negative	<input type="checkbox"/>		
_AGGREGATE ALL Turn Bytes into Bits	<input type="checkbox"/>		
_AGGREGATE ALL Turn Bytes into bits, make negative	<input type="checkbox"/>		
_AGGREGATE SIMILAR Multiply by -1024	<input type="checkbox"/>		
_AGGREGATE SIMILAR Multiply by 1024	<input type="checkbox"/>		
_AGGREGATE SIMILAR Turn Bytes into Bits	<input type="checkbox"/>		
_AGGREGATE SIMILAR Turn Bytes into bits, make negative	<input type="checkbox"/>		
_MAKE 0	<input type="checkbox"/>		
<< Previous		Showing Rows 1 to 16 of 16 [1]	Next >>

Illustration 27: Complete list of CDEFs

Totaling CDEFs

CDEF Title	CDEF
_AGGREGATE ALL Turn Bytes into Bits	ALL_DATA_SOURCES_NODUPS,8,
_AGGREGATE ALL Turn Bytes into bits, make negative	ALL_DATA_SOURCES_NODUPS,8,*,-1,*
_AGGREGATE SIMILAR Multiply by -1024	SIMILAR_DATA_SOURCES_NODUPS,-1024,*
_AGGREGATE SIMILAR Multiply by 1024	SIMILAR_DATA_SOURCES_NODUPS,1024,*
_AGGREGATE SIMILAR Turn Bytes into Bits	SIMILAR_DATA_SOURCES_NODUPS,8,
_AGGREGATE SIMILAR Turn Bytes into bits, make negative	SIMILAR_DATA_SOURCES_NODUPS,8,*,-1,*



Make 0 CDEF

Another special CDEF is the “Make 0” CDEF. When you want totals only, we need a special treatment of “original data source” that you don't want to see.

For the totaling magic, you will need the “DEF” statements created by cacti when using the “original data source”. But you don't want to have them printed. This sounds like an inconsistency. But it can be resolved by applying a CDEF that renders all data to 0.

This can be achieved by

CDEF Title	CDEF
_MAKE 0	CURRENT_DATA_SOURCE,0,*



Caveat when using different Graph Templates

In previous version of AGGREGATE, there was no verification of graph templates used. This may lead to buggy graphs, because always the first graph is used as a model for all other graphs. Thus, if the first graph as eight items (default traffic graph template) and the second one has eleven (95th percentile traffic template), funny things will happen.

To prevent this, AGGREGATE now checks the templates used for the graphs. See example below.

<< Previous		Showing Rows 1 to 6 of 6 [1]		Next >>
Graph Title**	ID	Template Name	Size	<input type="checkbox"/>
Localhost - Traffic (default)	152	Interface - Traffic (bits/sec) (default)	120x700	<input checked="" type="checkbox"/>
Localhost - Traffic (AREA/STACK)	141	Interface - Traffic (bits/sec) (AREA/STACK)	120x700	<input type="checkbox"/>
Localhost - Traffic (peak)	144	Interface - Traffic (bits/sec) (peak)	120x700	<input type="checkbox"/>
Localhost - Traffic (pos/neg)	136	Interface - Traffic (bits/sec) (pos/neg)	120x700	<input type="checkbox"/>
Localhost - Traffic (pure LINE)	142	Interface - Traffic (bits/sec) (pure LINE)	120x700	<input type="checkbox"/>
Localhost - Traffic - wlan0	1281	Interface - Traffic (bits/sec, 95th Percentile)	120x500	<input checked="" type="checkbox"/>
<< Previous		Showing Rows 1 to 6 of 6 [1]		Next >>

Illustration 28: Erraneously selected graphs using different templates

In this example, I purposely made a mistake. The result screen will be as follows

Create Aggregate Graph [\[Click here for Help\]](#)

The Graphs chosen for AGGREGATE refer to different Graph Templates. This will break AGGREGATE

- Interface - Traffic (bits/sec) (default)
- Interface - Traffic (bits/sec, 95th Percentile)

Please click CANCEL

and choose different Graphs

Cancel Continue

Illustration 29: Error response when choosing wrong graphs