Cacti

„Aggregate Graph“ Plugin

Usage Guide

Brought to you by

Reinhard Scheck alias gandalf@forums.cacti.net

The Cacti Group

Table of Contents

Installing the Plugin....2
Providing Authorization to Users....4
Basic Usage....5
Changing the Default Traffic Graph Templates....8
Example: Color Templates Usage....10
Example: Create an Aggregate Graph using „Total All Data Sources“....12
Example: Create an Aggregate Graph using „Total Similar Data Sources“....14
Example: Create an Aggregate Graph off of „95th Percentile“....16
A Word on CDEFs....18
Caveat when using different Graph Templates....20

Installing the Plugin

As a prerequisite, the Plugin Architecture is required. This is not described here, please see http://cactiusers.org/ for help regarding this topic. Then, please download this plugin to the <path_cacti>/plugin directory. Next, 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!

Edit <path_cacti>/include/global.php (od config.php if running cacti 086 code) and add aggregate in the plugin section as usual. Assuming, you already have provided authorization to Plugin Management, please go to that menu item and select the Uninstalled tab. Hit „Install“.

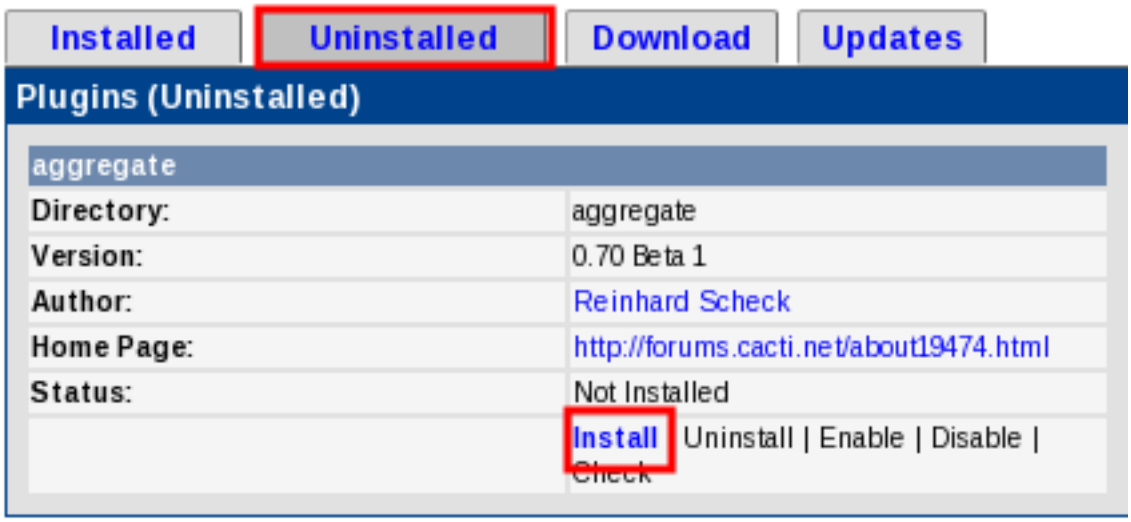

Illustration 1: Install Aggregate, Step 1

Select the „Installed Tab, hit „Enable“. It should now look like

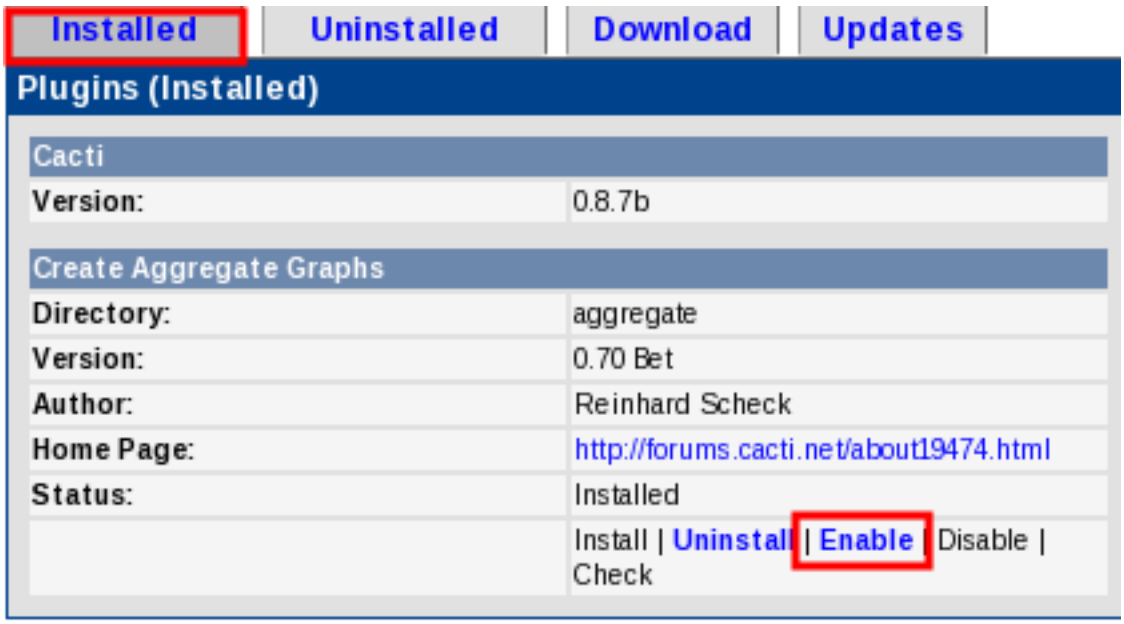

Illustration 2: Install Aggregate, Step 2

The final result is

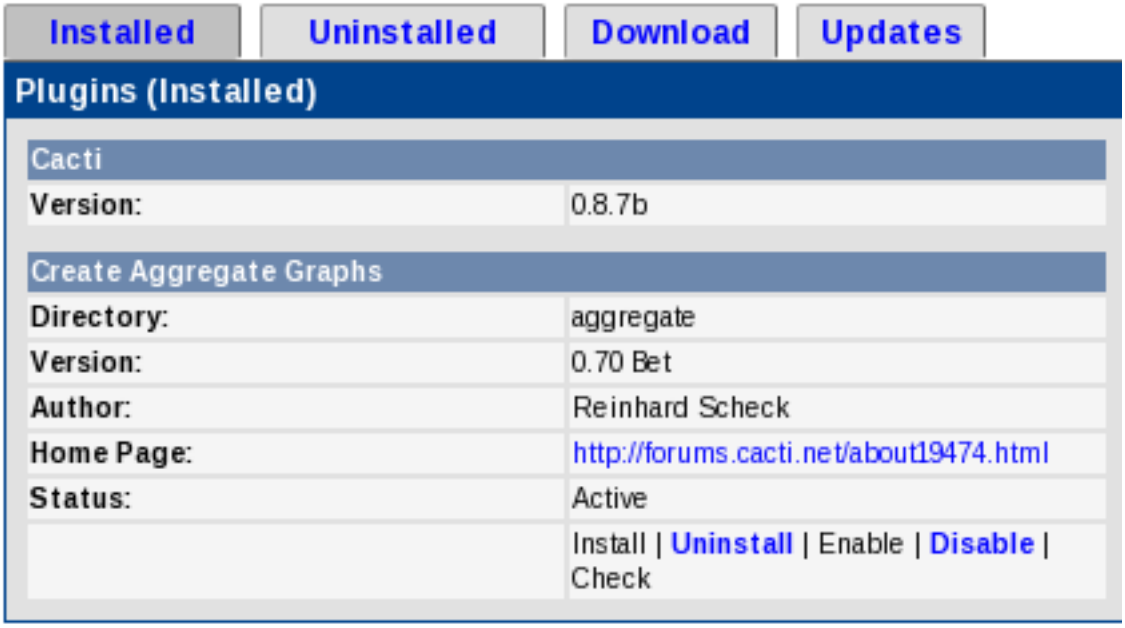

Illustration 3: Installation, Step 3 (final)

Providing Authorization to Users

Everybody who has Console access is 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, please go to User Management and select the correct userid. Now, check the Aggregate checkbox.

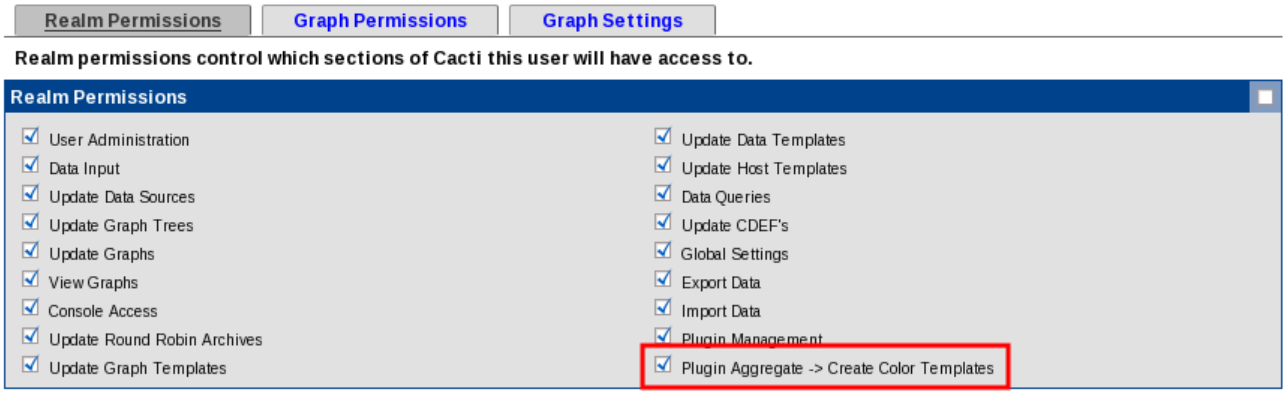

Illustration 4: Realm Permission for Aggregate Color Templates

Basic Usage

Now, turn to Graph Management. First, you will have to select the graphs you want to aggregate.

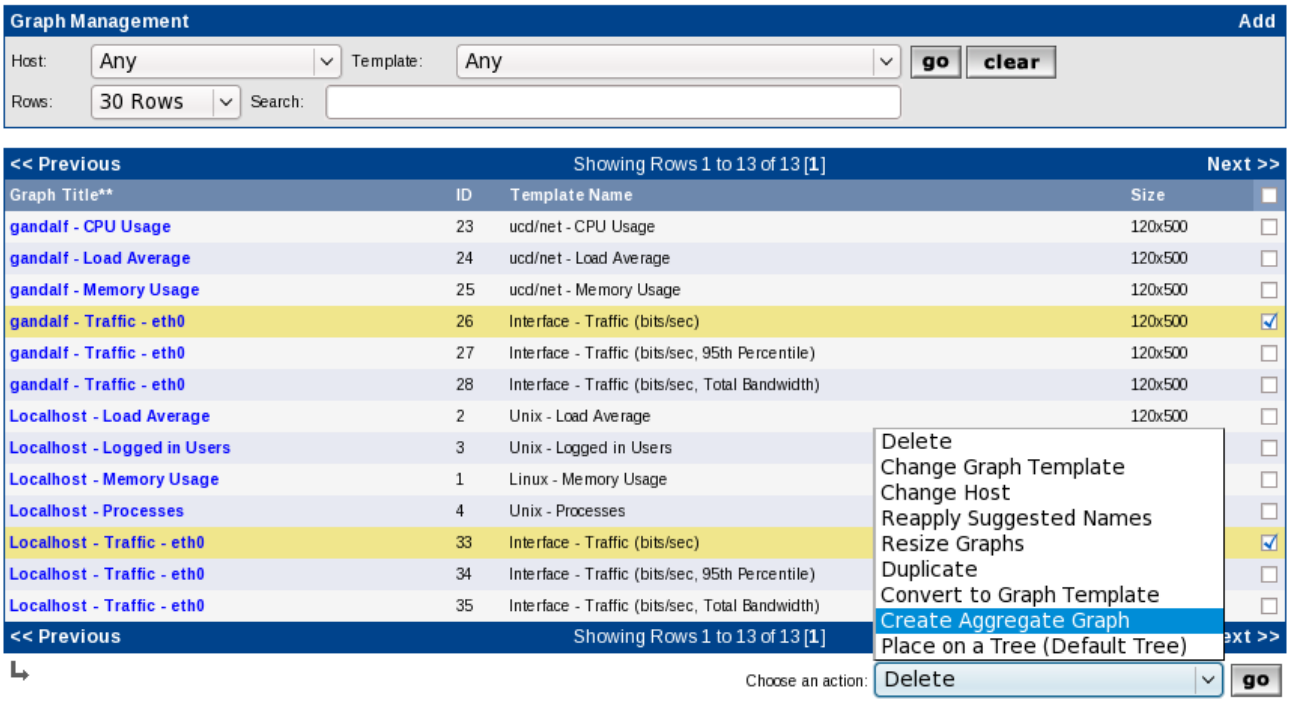

Illustration 5: Select Graphs from Graph Management

CAVEAT: Please only select graphs based on a single graph template (see column „Template Name“). Else you will get funny looking aggregates!

Now, please hit Create Aggregate Graph to see

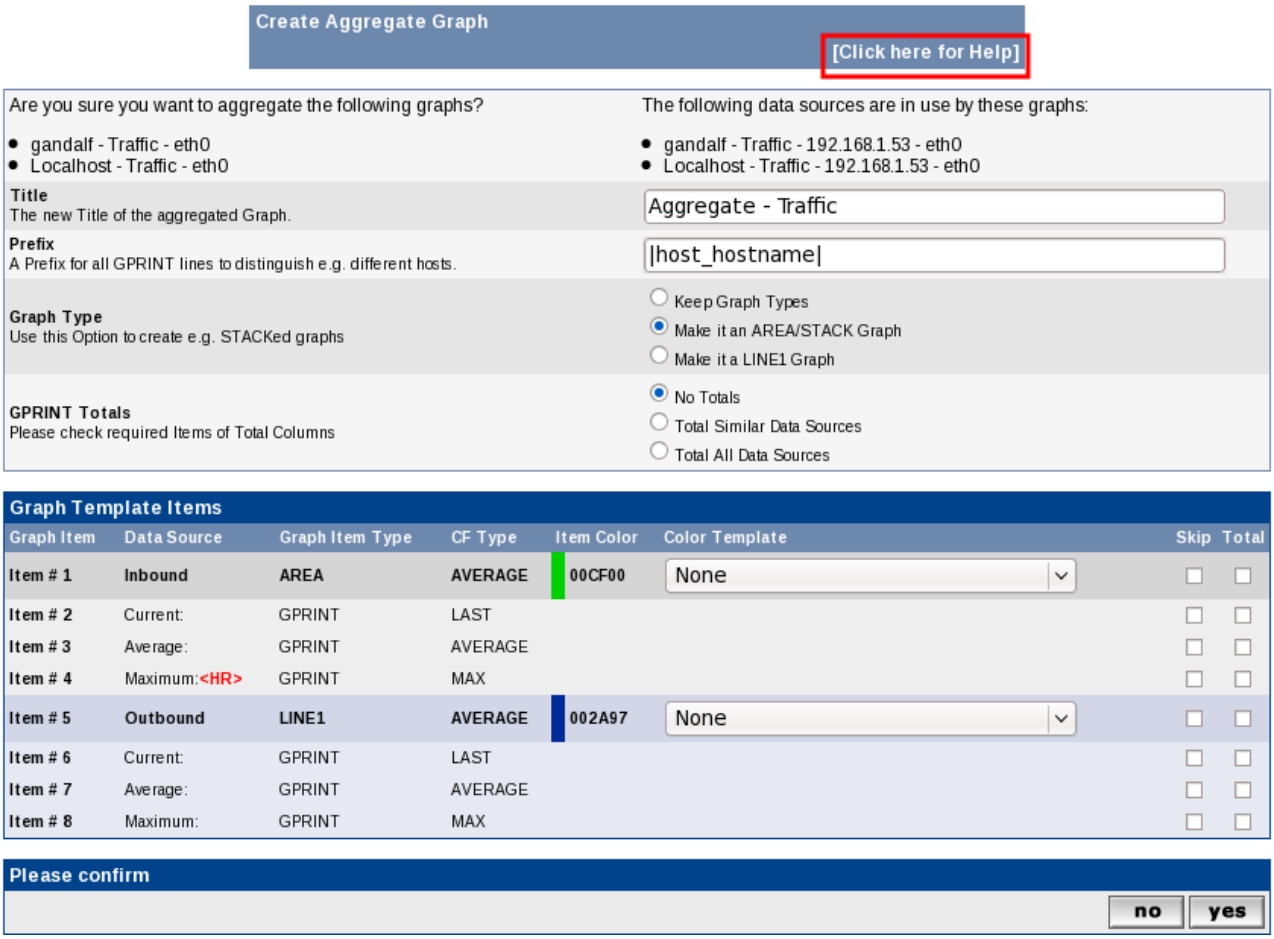

Illustration 6: Default Prompt for Aggregating Graphs

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, that is e.g. „|host_description| - Traffic - |query_ifDescr|“. As neither host nor query related variables make much sense for an aggregate, all |host_*| and |query_*| stuff is removed silently.

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. 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). Sometimes, it is

recommended to have LINEx graphs instead. That's the third radio button. The first one keeps graph types as is.

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 an the positive y-axis (see Chapter “Changing the Default Traffic Graph Templates“ below!

GPRINT Totals is an option, that automatically generates totaling lines not yet present on any of the selected graphs. It comes is useful, you you want to total traffic of e.g. interfaces of different hosts. There are two different options: „Total Similar Data Sources“ and „Total All Data Sources“. Please see examples below on how to use them.

Graph Templates Items is build based on the first graph selected. Please pay attention to this section to make the most of your new aggregate. 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 now 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. 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!

Skip allows you to skip the checked item in the aggregate. There's a new, automatic <HR> mechanism to cope with <hard_returns> that now will be dropped. 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 radio button!

On the first run, you may want to check each line. That does not make sense, in general. 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 for totaling all data sources, the data source type is not taken into account. Again checking the same graph items as done with „similar data sources“ will result in two (or more) identical totaling legend entries.

And 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.

Changing 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 “Make Stack negative, turn Bytes into Bits”:

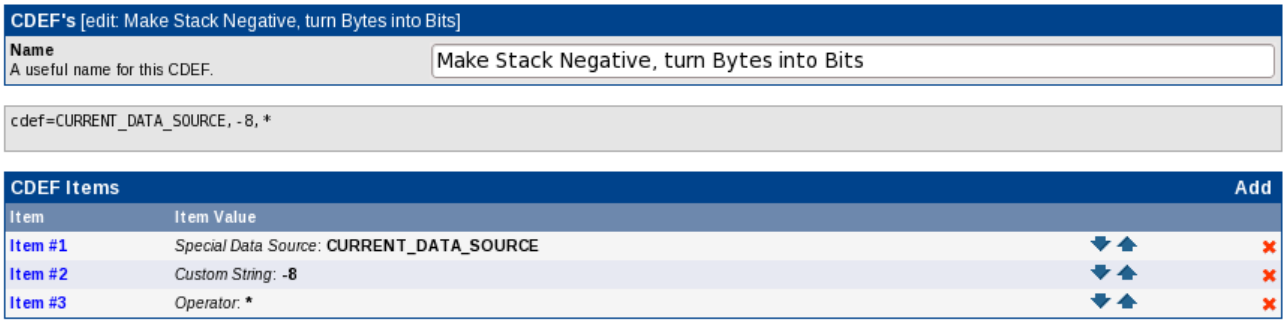

Illustration 7: Make Stack negative, turn Bytes into Bits

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

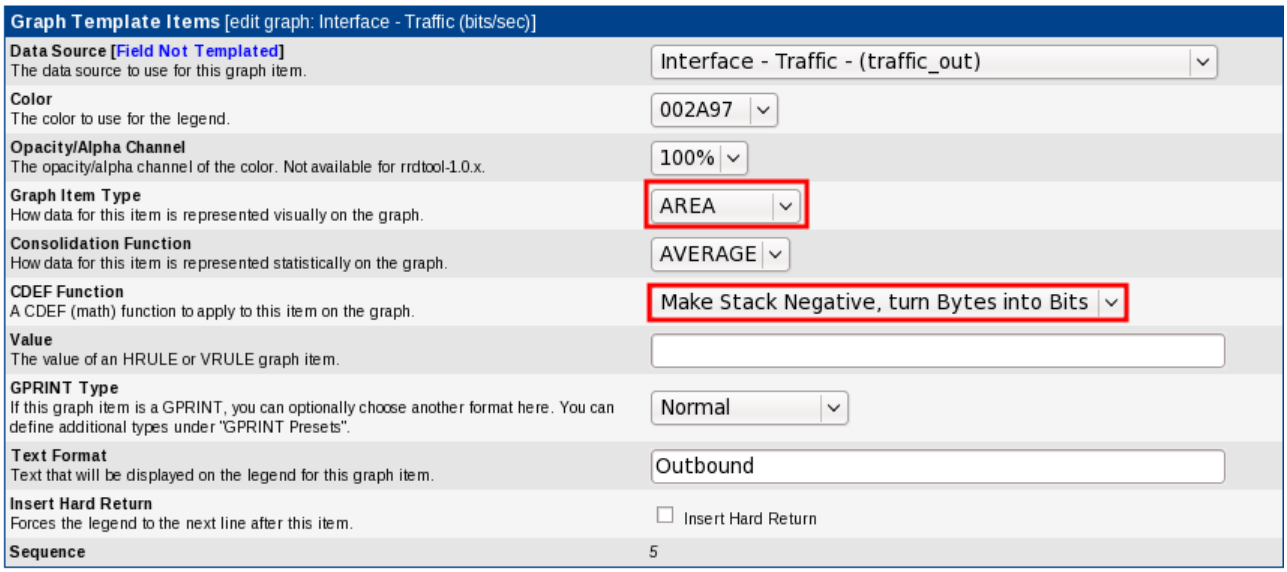

Illustration 8: Apply CDEF to Outbound Traffic Item

Now, one more tweaks 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

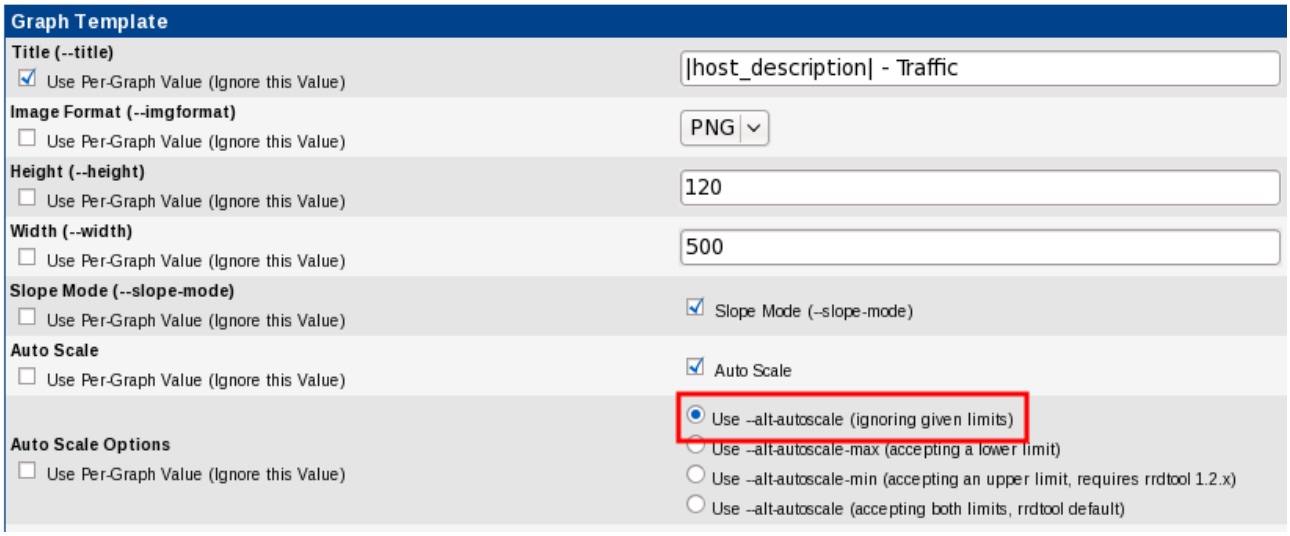

Illustration 9: Use --alt-autoscale

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)

Example: Color Templates Usage

In this example, both AREA/LINE items were related to different color templates by means of the dropdown.

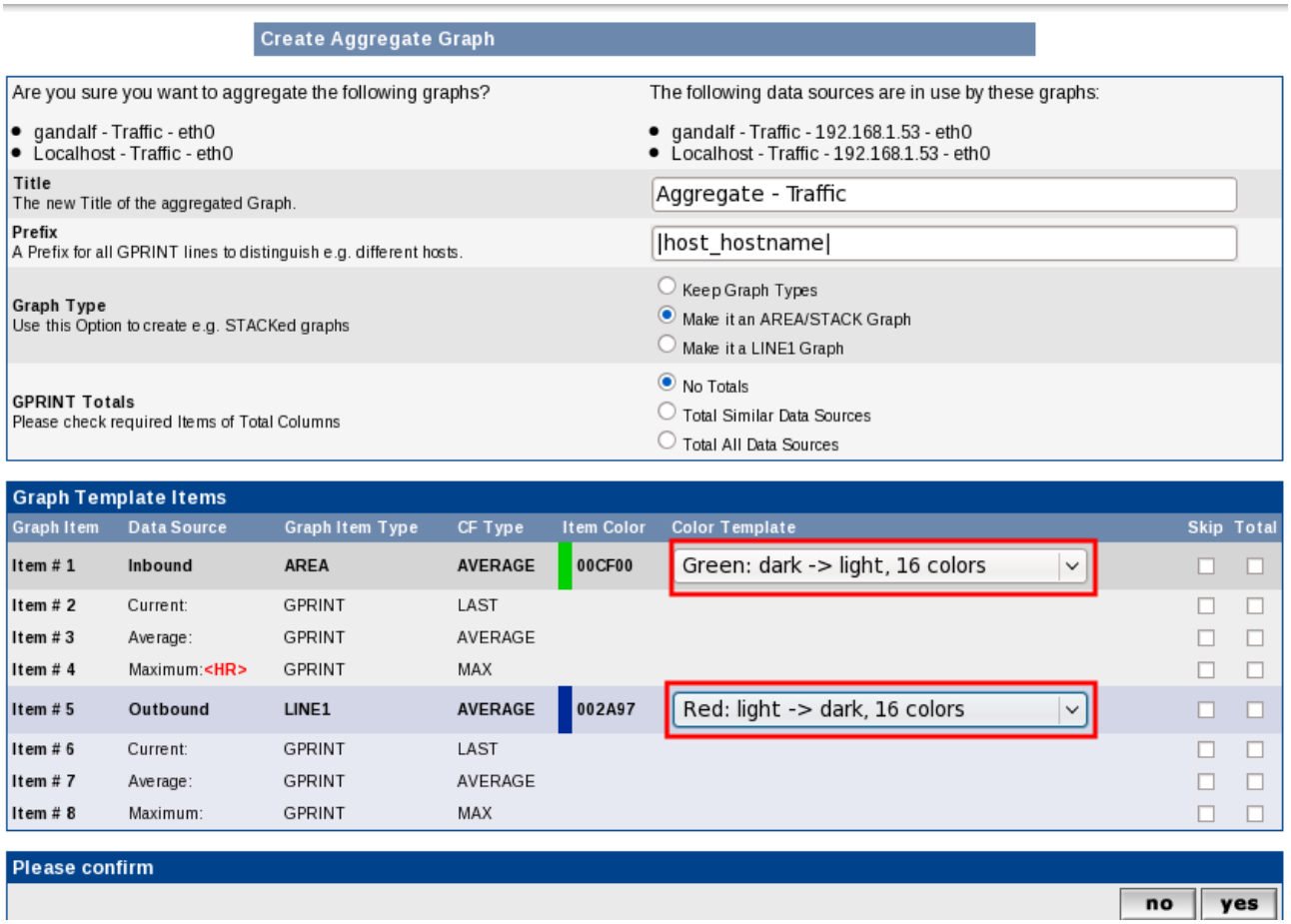

Illustration 10: Use of Color Templates

The results shows different colors assigned to all colored graph items. You may also notice, that the *Graph Item Type* was changed to AREA/STACK as required.

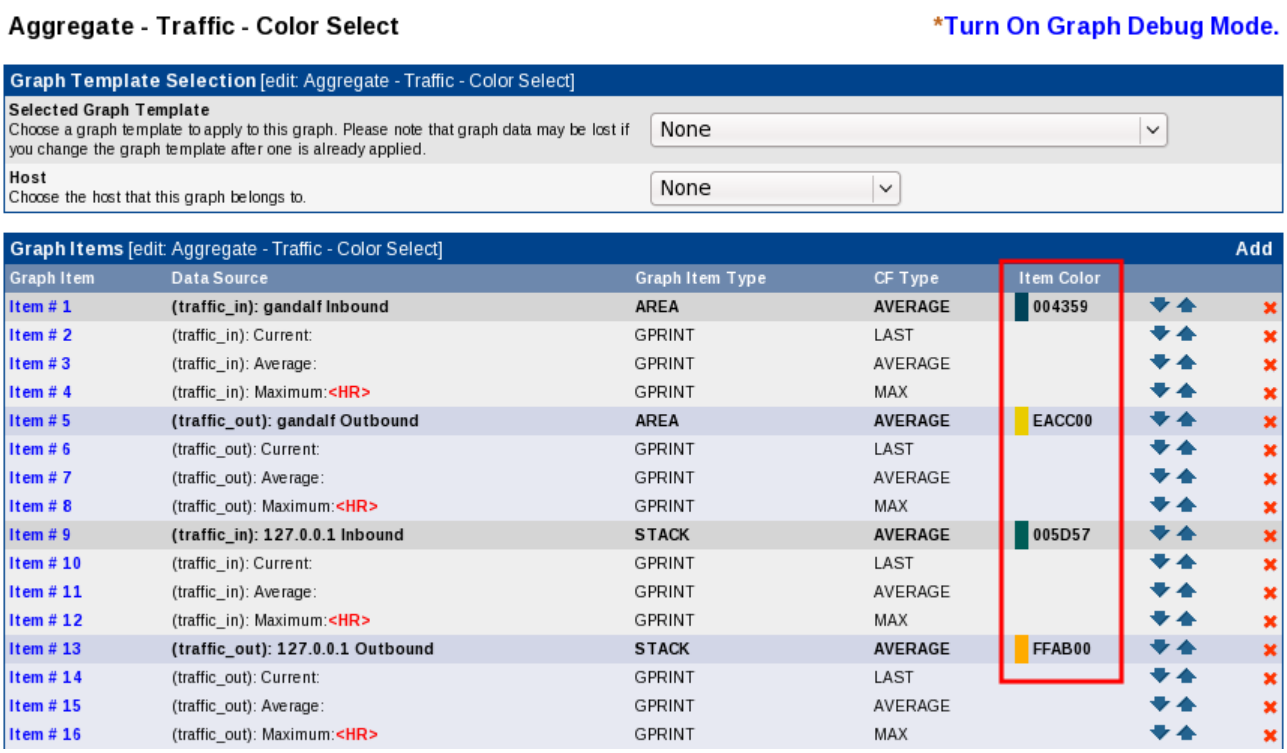

Illustration 11: Result of using Color Templates

Please notice, that even the items “(traffic out): Maximum” now have <HR> even if the original graphs don't.

Example: Create an Aggregate Graph using „Total All Data Sources“

We now select the totaling option “Total All Data Sources”. As already said, it is necessary to check required items for this options from *Graph Template Items* table.

As we want to see the total as a line within the graph, item#1 is checked. Totals will always be represented as a LINE1.

To created legend entries for total numbers, *current*, *average* and *maximum*, items#2-4 are checked as well.

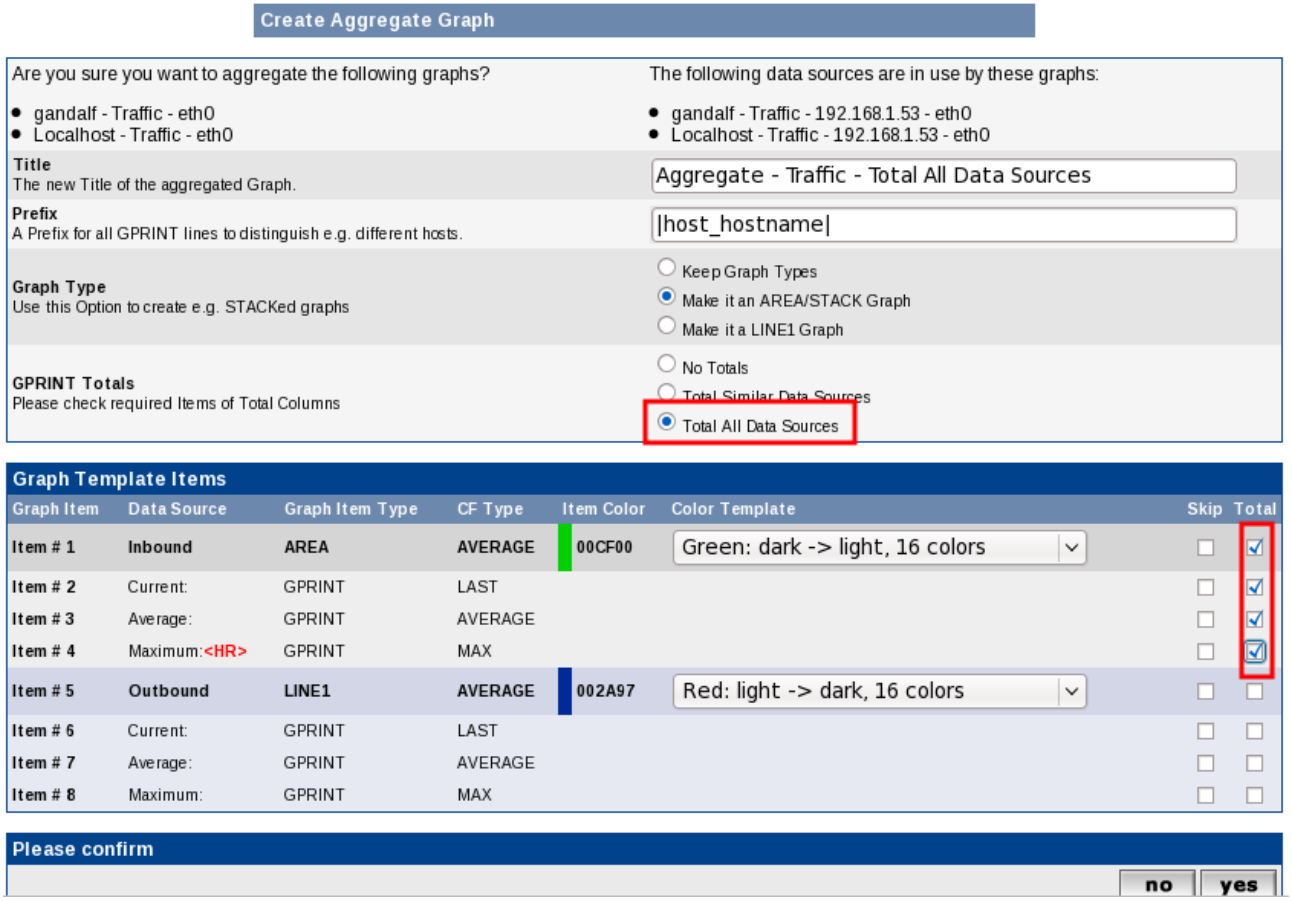

Illustration 12: Create an Aggregate with Total All Data Sources

Please find the result below. The automatically created total graph items are marked. Please notice, that the graphed item (item#17) is created as a LINE1.

Aggregate - Traffic - Total All Data Sources *Turn On Graph Debug Mode.

Graph Template Selection [edit: Aggregate - Traffic - Total All Data Sources]

Selected Graph Template
Choose a graph template to apply to this graph. Please note that graph data may be lost if you change the graph template after one is already applied. None

Host
Choose the host that this graph belongs to. None

Graph Items [edit: Aggregate - Traffic - Total All Data Sources] Add

Graph Item	Data Source	Graph Item Type	CF Type	Item Color
Item # 1	**(traffic_in): gandalf 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): gandalf Outbound**	**AREA**	**AVERAGE**	**EACC00**
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): 127.0.0.1 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): 127.0.0.1 Outbound**	**STACK**	**AVERAGE**	**FFAB00**
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): ALL ITEMS**	**LINE1**	**AVERAGE**	**157419**
Item # 18	(traffic_in): Current:	GPRINT	LAST	
Item # 19	(traffic_in): Average:	GPRINT	AVERAGE	
Item # 20	(traffic_in): Maximum:<HR>	GPRINT	MAX	

Illustration 13: Created Aggregate with Total All Data Sources

Example: Create an Aggregate Graph using „Total Similar Data Sources“

Next, we select the totaling option “Total Similar Data Sources”. We want to see totals on inbound and outbound traffic as well as on the legend. Thus, all items are checked.

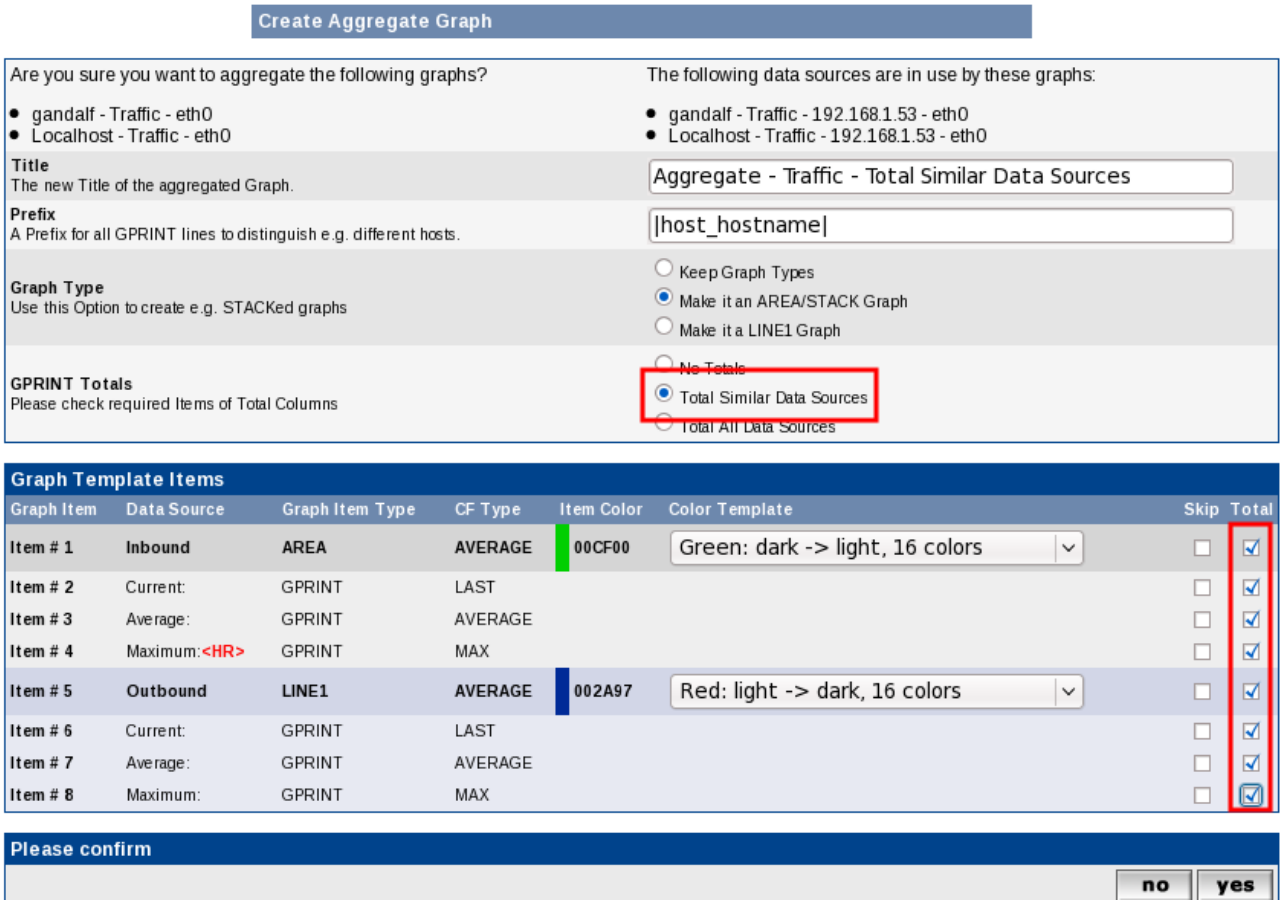

Illustration 14: Create an Aggregate with "Total Similar Data Sources"

The result is shown below. The automatically created total graph items are marked. Please notice, that the graphed item (item#17 and 21) is created as a LINE1.

Graph Template Selection [edit: Aggregate - Traffic - Total Similar Data Sources]

Selected Graph Template
Choose a graph template to apply to this graph. Please note that graph data may be lost if you change the graph template after one is already applied. None

Host
Choose the host that this graph belongs to. None

Graph Items [edit: Aggregate - Traffic - Total Similar Data Sources] Add

Graph Item	Data Source	Graph Item Type	CF Type	Item Color
Item # 1	**(traffic_in): gandalf 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): gandalf Outbound**	**AREA**	**AVERAGE**	**EACC00**
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): 127.0.0.1 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): 127.0.0.1 Outbound**	**STACK**	**AVERAGE**	**FFAB00**
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 Inbound**	**LINE1**	**AVERAGE**	**157419**
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 Outbound**	**LINE1**	**AVERAGE**	**EA8F00**
Item # 22	(traffic_out): Current:	GPRINT	LAST	
Item # 23	(traffic_out): Average:	GPRINT	AVERAGE	
Item # 24	(traffic_out): Maximum:<HR>	GPRINT	MAX	

Illustration 15: Created Aggregate with "Total Similar Data Sources"

Example: Create an Aggregate Graph off of „95th Percentile“

This is a quite advanced example, touching different features of Aggregate. I selected two graphs, both based on the 95th percentile traffic graph template.

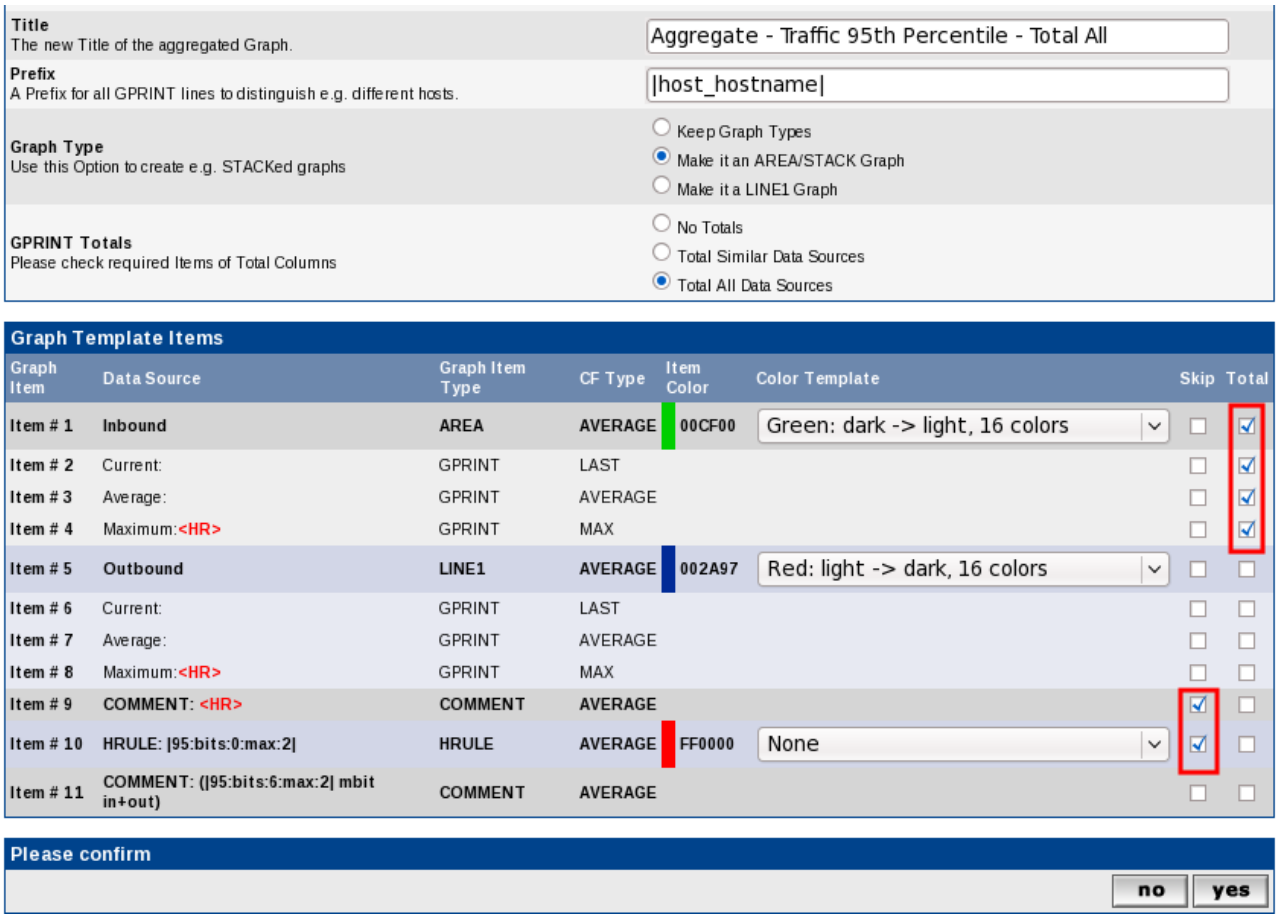

Illustration 16: Create an Aggregate Graph off of „95th Percentile“

The totaling option is selected as “Total All Data Sources”. So we have to check all required graph items; this makes items#1-4.

But we want to skip the empty COMMENT and the red HRULE.

The result is as follows:

Aggregate - Traffic 95th Percentile - Total All *Turn On Graph Debug Mode.

Graph Template Selection [edit: Aggregate - Traffic 95th Percentile - Total All]

Selected Graph Template Choose a graph template to apply to this graph. Please note that graph data may be lost if you change the graph template after one is already applied.	None
Host Choose the host that this graph belongs to.	None

Graph Items [edit: Aggregate - Traffic 95th Percentile - Total All] Add

Graph Item	Data Source	Graph Item Type	CF Type	Item Color
Item # 1	**(traffic_in): gandalf 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): gandalf Outbound**	**AREA**	**AVERAGE**	**EACC00**
Item # 6	(traffic_out): Current:	GPRINT	LAST	
Item # 7	(traffic_out): Average:	GPRINT	AVERAGE	
Item # 8	(traffic_out): Maximum:<HR>	GPRINT	MAX	
Item # 9	**COMMENT: gandalf (|95:bits:6:max:2| mbit in+out)<HR>**	**COMMENT**	**AVERAGE**	
Item # 10	**(traffic_in): 127.0.0.1 Inbound**	**STACK**	**AVERAGE**	**005D57**
Item # 11	(traffic_in): Current:	GPRINT	LAST	
Item # 12	(traffic_in): Average:	GPRINT	AVERAGE	
Item # 13	(traffic_in): Maximum:<HR>	GPRINT	MAX	
Item # 14	**(traffic_out): 127.0.0.1 Outbound**	**STACK**	**AVERAGE**	**FFAB00**
Item # 15	(traffic_out): Current:	GPRINT	LAST	
Item # 16	(traffic_out): Average:	GPRINT	AVERAGE	
Item # 17	(traffic_out): Maximum:<HR>	GPRINT	MAX	
Item # 18	**COMMENT: 127.0.0.1 (|95:bits:6:max:2| mbit in+out)<HR>**	**COMMENT**	**AVERAGE**	
Item # 19	**(traffic_in): ALL ITEMS**	**LINE1**	**AVERAGE**	**157419**
Item # 20	(traffic_in): Current:	GPRINT	LAST	
Item # 21	(traffic_in): Average:	GPRINT	AVERAGE	
Item # 22	(traffic_in): Maximum:<HR>	GPRINT	MAX	

Illustration 17: Created Aggregate Graph „95th Percentile“

Please again notice, that the COMMENT now has a <HR>, even if the graph doesn't.

A Word on CDEFs

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

Well, 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.

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.

Please find attached the list of CDEFs on my system after having created all the examples from above.

Illustration 18: CDEFs

This is “_AGGREGATE ALL Turn Bytes into Bits”

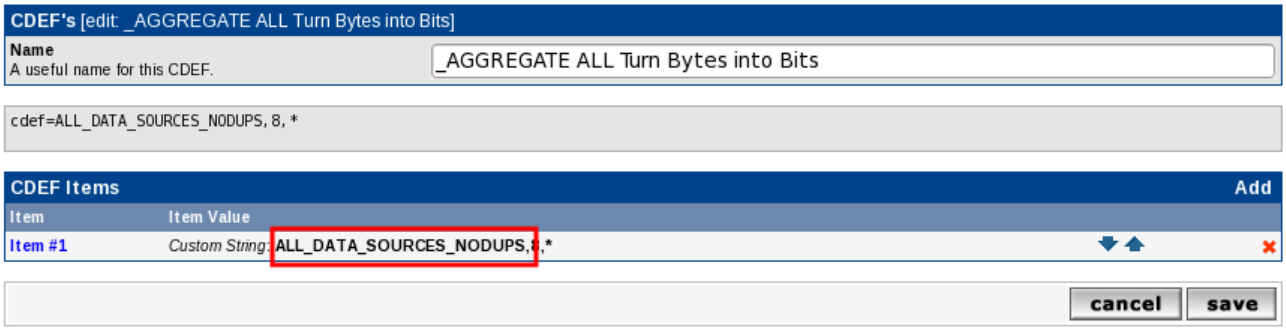

Illustration 19: AGGREGATE ALL Turn Bytes into Bits

Now "_AGGREGATE SIMILAR Make Stack negative, turn Bytes into Bits"

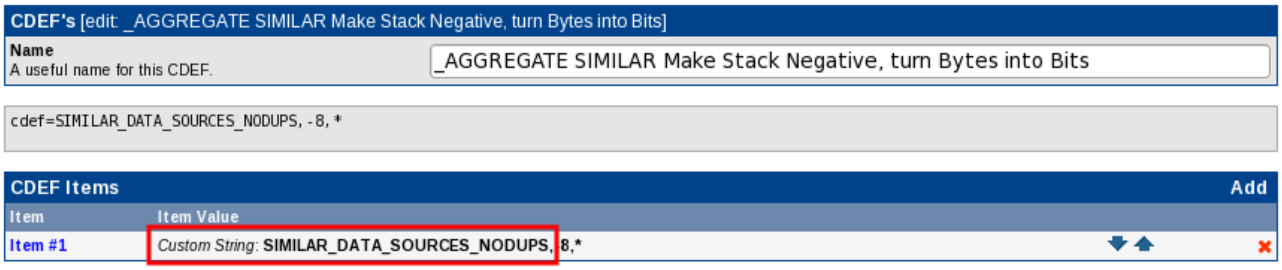

Illustration 20: AGGREGATE SIMILAR Make Stack negative, turn Bytes into Bits

Last "_AGGREGATE SIMILAR Turn Bytes into Bits"

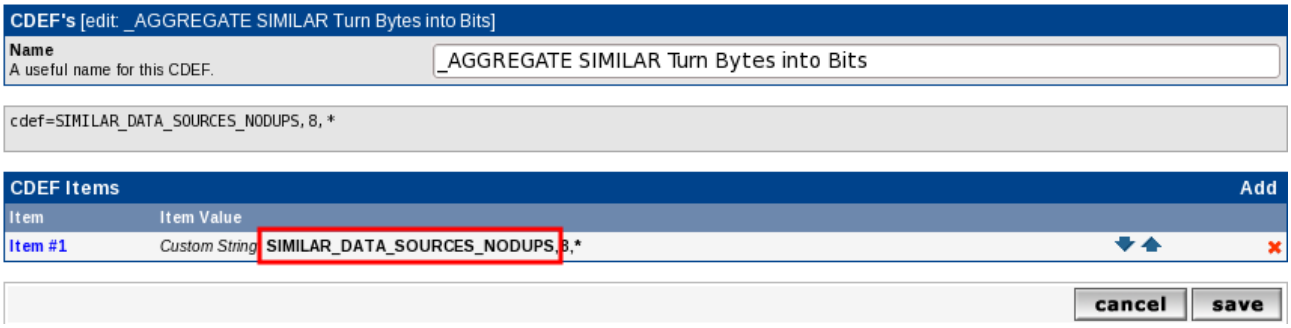

Illustration 21: AGGREGATE SIMILAR Turn Bytes into Bits

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.

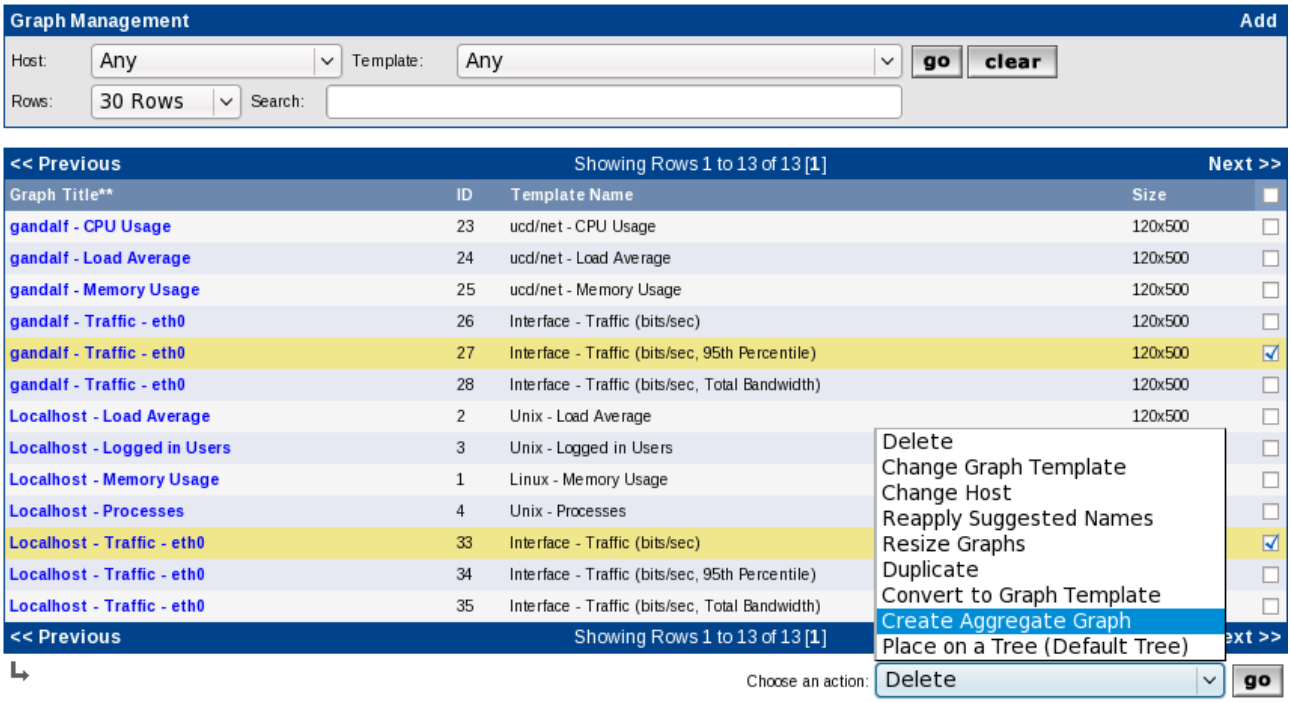

Illustration 22: Wrong Graph Selection

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