

# Visualizing IBM Integration Bus statistics data using Grafana

10

Simon Stone

Published on August 18, 2016

In IBM Integration Bus, message flow statistics and accounting data can be collected to record performance and operating details of one or more message flows. Message flow statistics and accounting data captures dynamic information about the runtime behavior of a message flow. For example, it indicates how many messages are processed and how large those messages are, as well as processor usage and elapsed processing times.

## Introduction

The IBM Integration Bus runtime has built-in support for publishing the message flow statistics and accounting data in XML or JSON format on MQ or MQTT topics, writing SMF records on z/OS, or recording to user trace.

One of the less publicized features in IBM Integration Bus 10.0.0.5 is the introduction of a new C plugin API (see the header file `BipCsi.h` in the product installation) that allows developers to write C plugins that can access the statistics data directly. This provides an alternative route to capturing the XML or JSON publications from MQ or MQTT, and transforming those publications into the correct format before sending them on to another application for storage or visualization.

One such application that is seeing a lot of interest at the moment is Grafana. Grafana is usually found as part of a stack of open source products. That stack includes – StatsD, a statistics gathering daemon – Graphite, a time-series database and graphing tool – and Grafana, a tool for visualizing time-series data in the form of dashboards.

In order to demonstrate how you can use the new C plugin API, we have developed and published a sample plugin that writes IBM Integration Bus statistics to the StatsD, Graphite, and Grafana stack. By installing and configuring this sample plugin, you can publish statistics data from multiple integration nodes across multiple systems and build a single dashboard showing the aggregated data.

## Installing the sample plugin

The source code for the sample plugin has been published to our GitHub repository and can be downloaded and forked from:

<https://github.com/ot4i/iib-stats-writer-statsd>

The source code is licensed under the MIT license, so feel free to download, fork, modify, and submit pull requests!

The GitHub link includes links to prebuilt versions of the sample plugin for Linux/x86-64, Windows, and Mac OS X. The source code should compile on other platforms, but we have not tested it on any other platforms at the time of writing. If you are feeling brave, then the build instructions for the sample plugin are included in the README.md file on the GitHub page.

Once you have downloaded or built the `statsdsw.lil` file for your target system, then copy it into the product installation directory (under `{install root}/server/lil`) and restart all affected integration servers. You must restart the integration servers so that the plugin is loaded and available for use.

## Setting up StatsD, Graphite, and Grafana

The easiest way to set up this stack of products for test purposes is to use a Docker image that contains a pre-configured environment. We have been using the following Docker image for our testing of the sample plugin:

[https://hub.docker.com/r/kamon/grafana\\_graphite/](https://hub.docker.com/r/kamon/grafana_graphite/)

The Docker Hub instructions include information on how to access Grafana after running `docker run` and setting up the Graphite data source. You must follow all of these steps before proceeding.

As part of the `docker run` command, you must expose port 8125 from the Docker container so that the sample plugin can write statistics data to StatsD. This is done via the `-p 8125:8125/udp` argument of the sample `docker run` command provided on Docker Hub. If desired, you can modify the `docker run` command to expose port 8125 as a different port – but remember to use that port in the below

instructions for configuring the sample plugin!

#### [Configuring IBM Integration Bus to publish statistics data to StatsD](#)

Now that the sample plugin has been installed, and the StatsD, Graphite, and Grafana stack is up and running, it is time to configure IBM Integration Bus to publish statistics into StatsD. StatsD will then forward the statistics data into Graphite and Grafana for storage and visualization.

Configure the sample plugin with the hostname and port of StatsD by running the following command:

```
mqsichangeproperties NODE -e SERVER -o StatsdStatsWriter -n hostname,port -v localhost,8125
```

Replace localhost with the hostname of the server running the Docker container (if not the same server running IBM Integration Bus), and port 8125 with the correct port (if modified from the default).

The StatsdStatsWriter is only available after the sample plugin has been installed and the integration server has been restarted. If you do not follow the previous steps correctly, then you will see errors from this command.

Next, you must enable message flow statistics so that the sample plugin is called to publish the statistics into StatsD. Deploy the message flows you wish to monitor, and run the following command:

```
mqsichangeflowstats IB10NODE -s -e default -j -c active -t basic -n advanced -o statsd
```

Statistics will now be published into StatsD every 20 seconds. The command will enable message flow statistics for every message flow deployed to the integration server. See the help for the mqsichangeflowstats command for further information on restricting this to specific message flows.

In order to get “interesting” statistics published into StatsD, you must drive some workload through the message flow. There are various tools for automatically driving workload through a message flow – for example, SoapUI for SOAP/REST traffic, or our very own PerfHarness tool which is available from:

<https://github.com/ot4i/perf-harness>

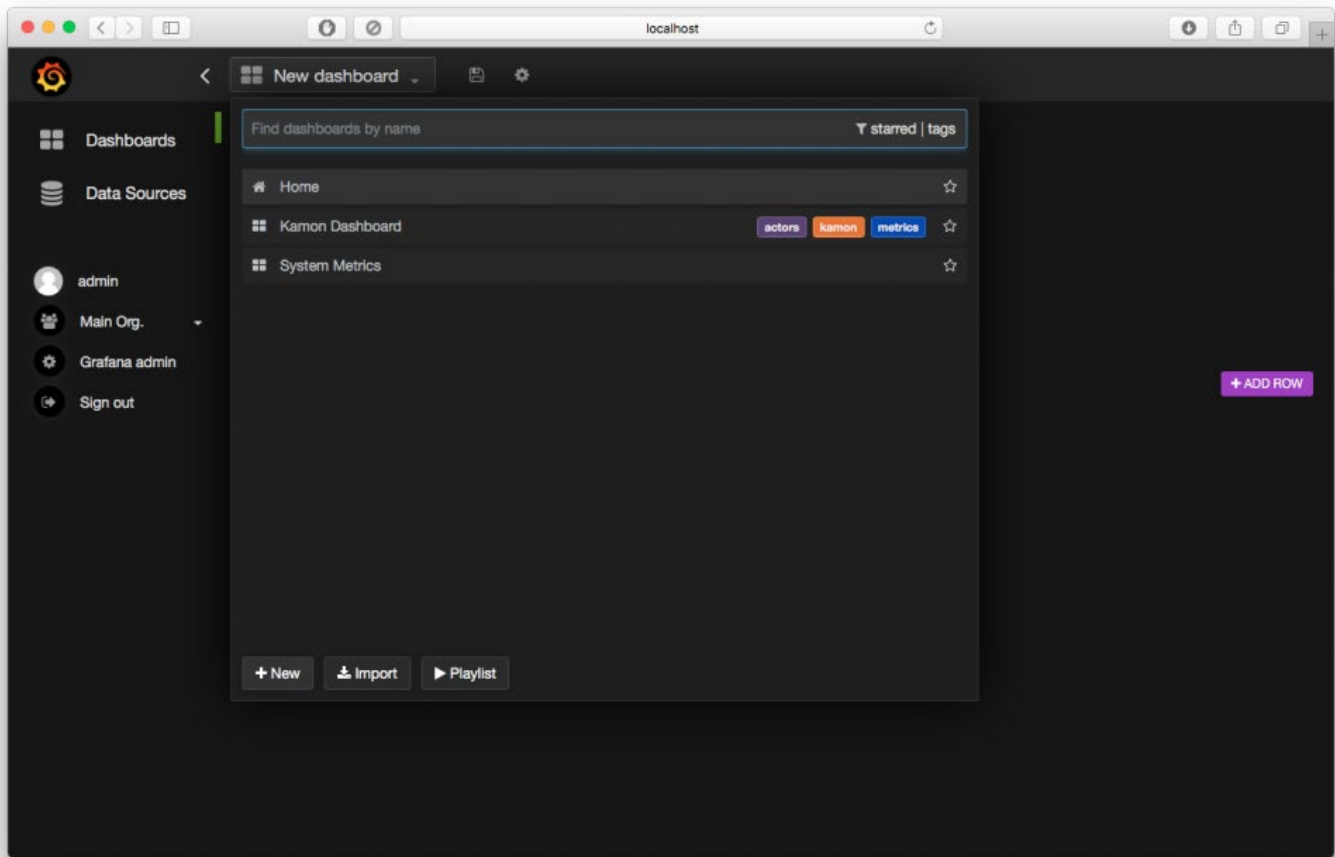
Ensure that you have workload going through your message flows before proceeding.

For this example, I have two integration nodes, NODE1 and NODE2. Each integration node has two integration servers, SERVER1 and SERVER2. All four integration servers have been configured with a REST API called MonitoringDemo and all four integration servers have been configured to publish statistics to StatsD.

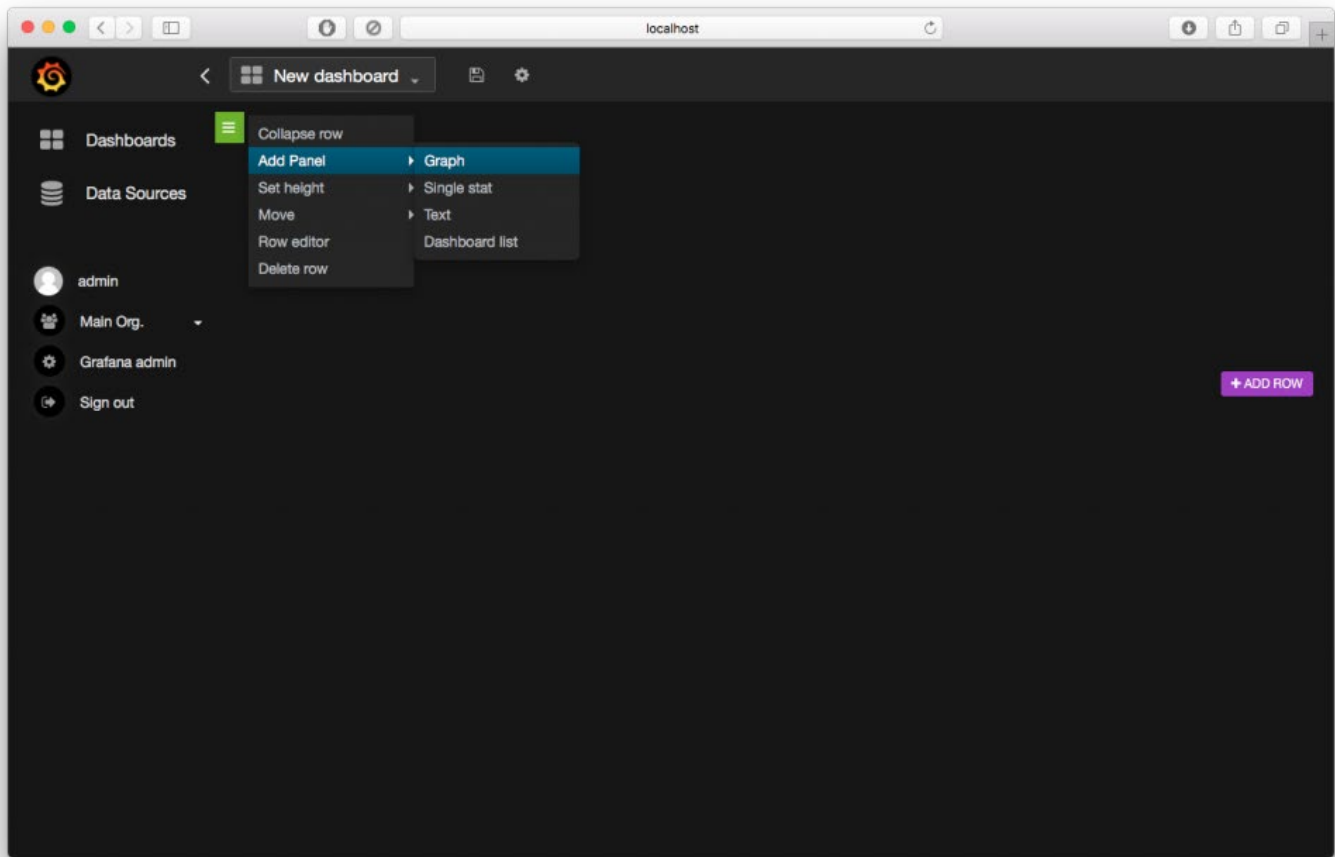
#### [Building a Grafana dashboard from IBM Integration Bus statistics data](#)

Once IBM Integration Bus statistics data is being published into StatsD, you can visualize it by building a Grafana dashboard.

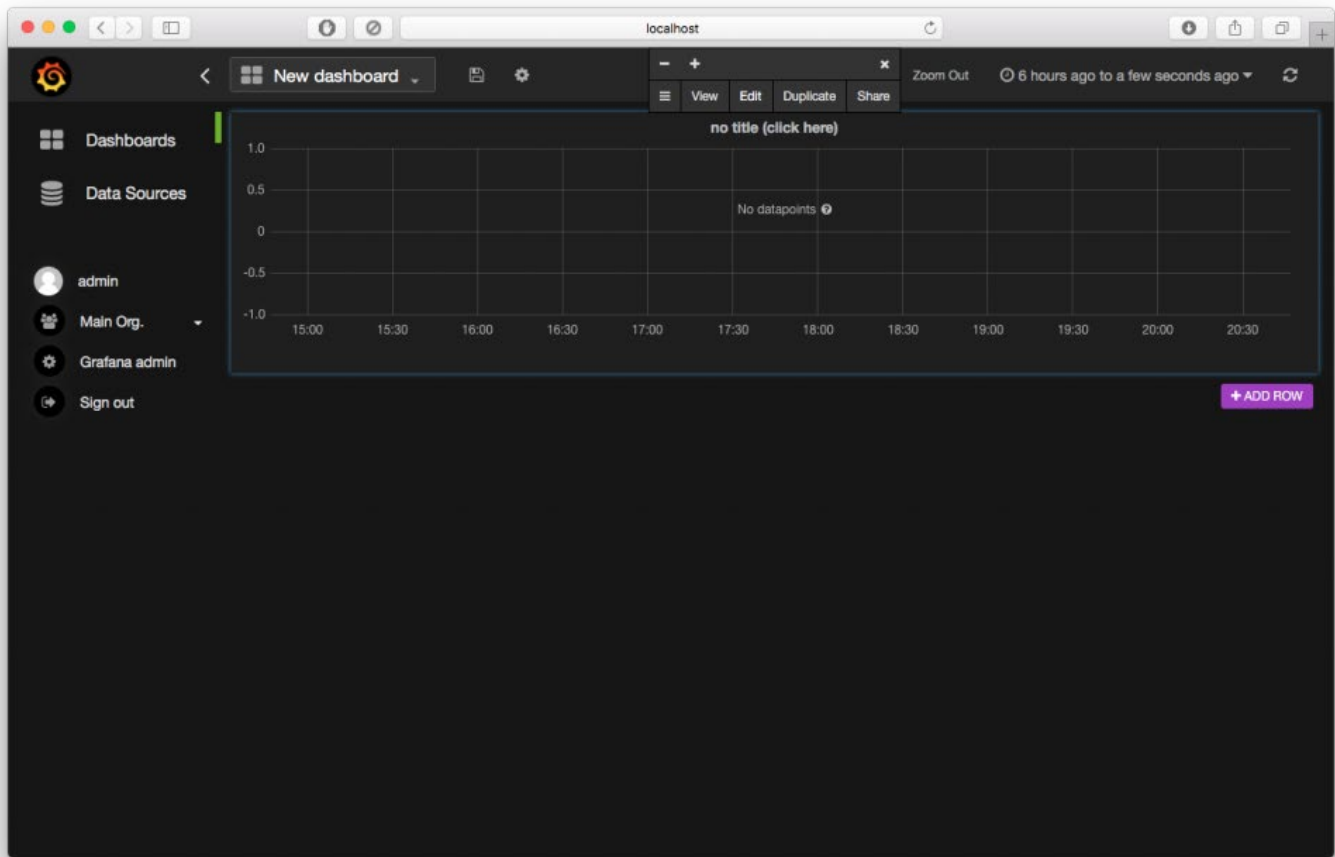
In Grafana, create a new dashboard by clicking the “Home” button at the top of the page, and then clicking the “+ New” button:



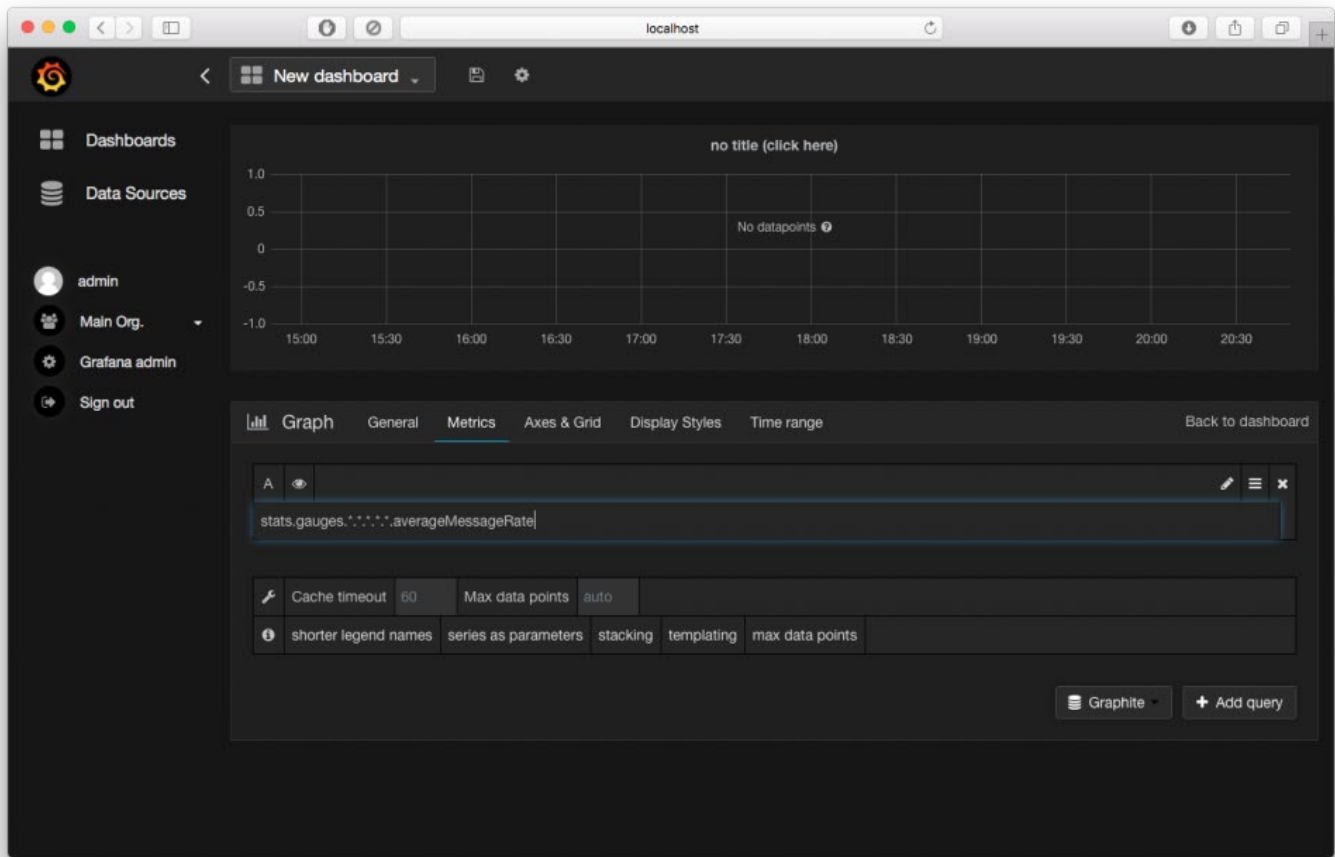
Click the green line that appears on the new dashboard, and create a new panel by clicking on “Add Panel”, and then “Graph”:



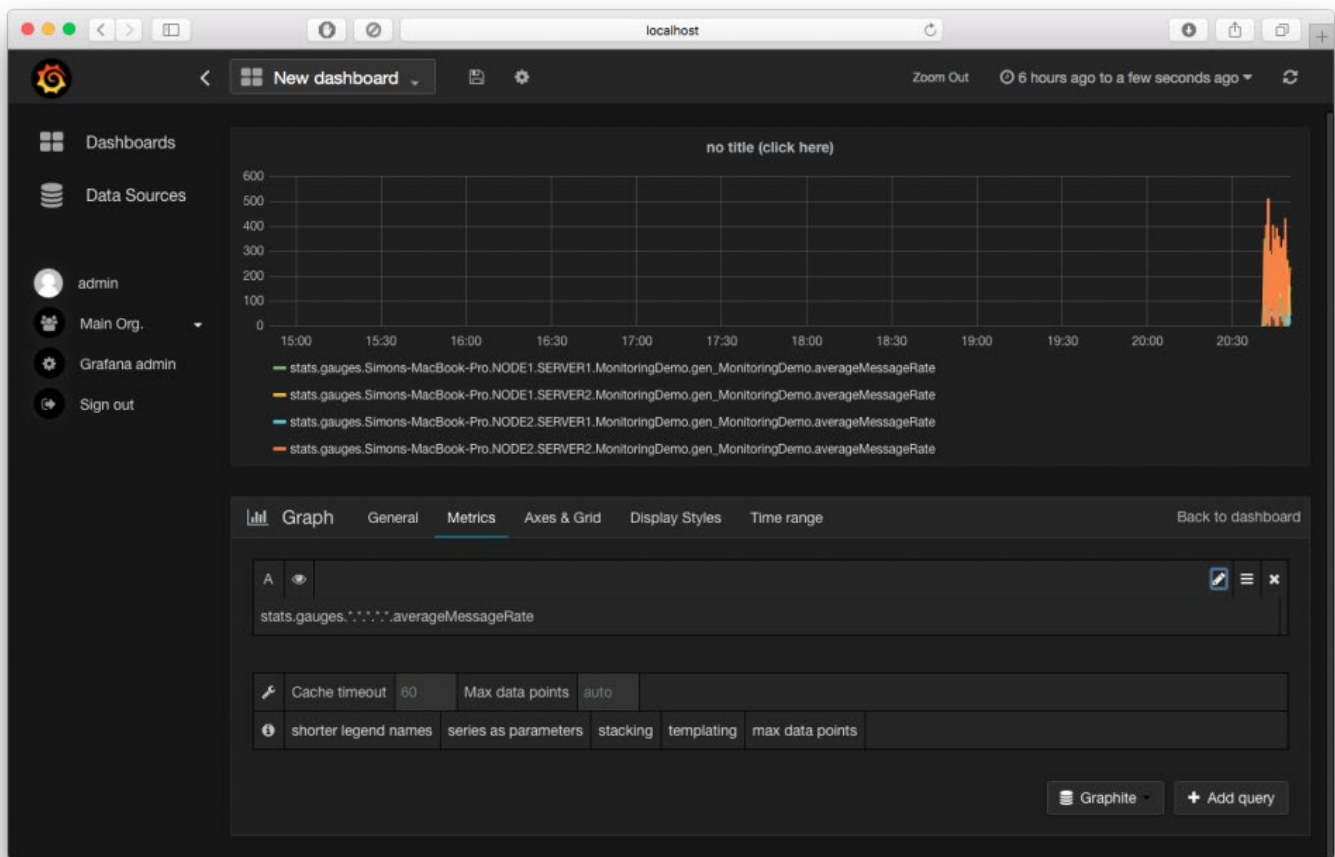
Click on the title of the new panel (“no title (click here)”) and then click on “Edit”:



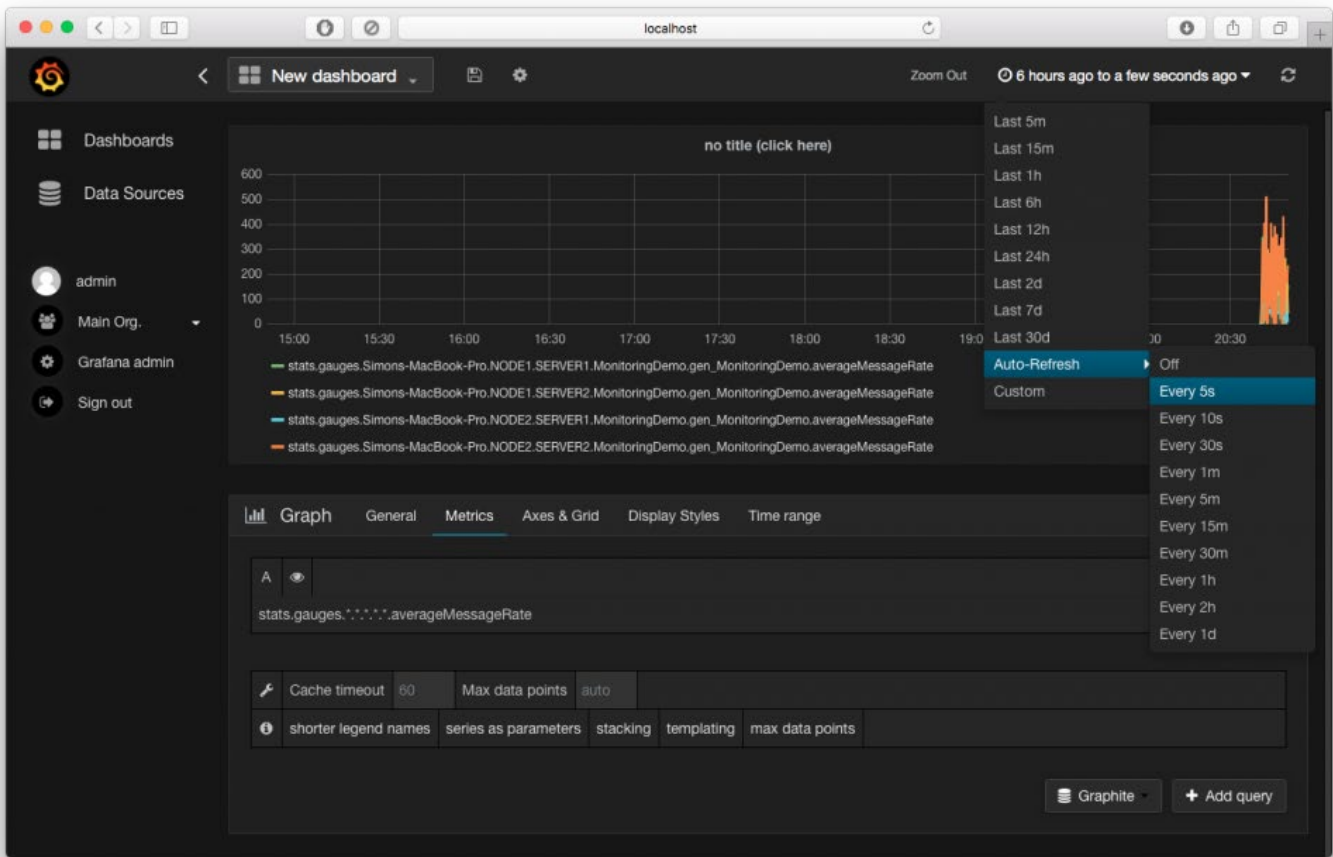
In the line that contains the text “select metric”, click the pencil icon and enter the text “stats.gauges.\*.\*.\*.\*.averageMessageRate” (without quotes):



Once done, click the pencil icon again. You should see statistics start to appear!

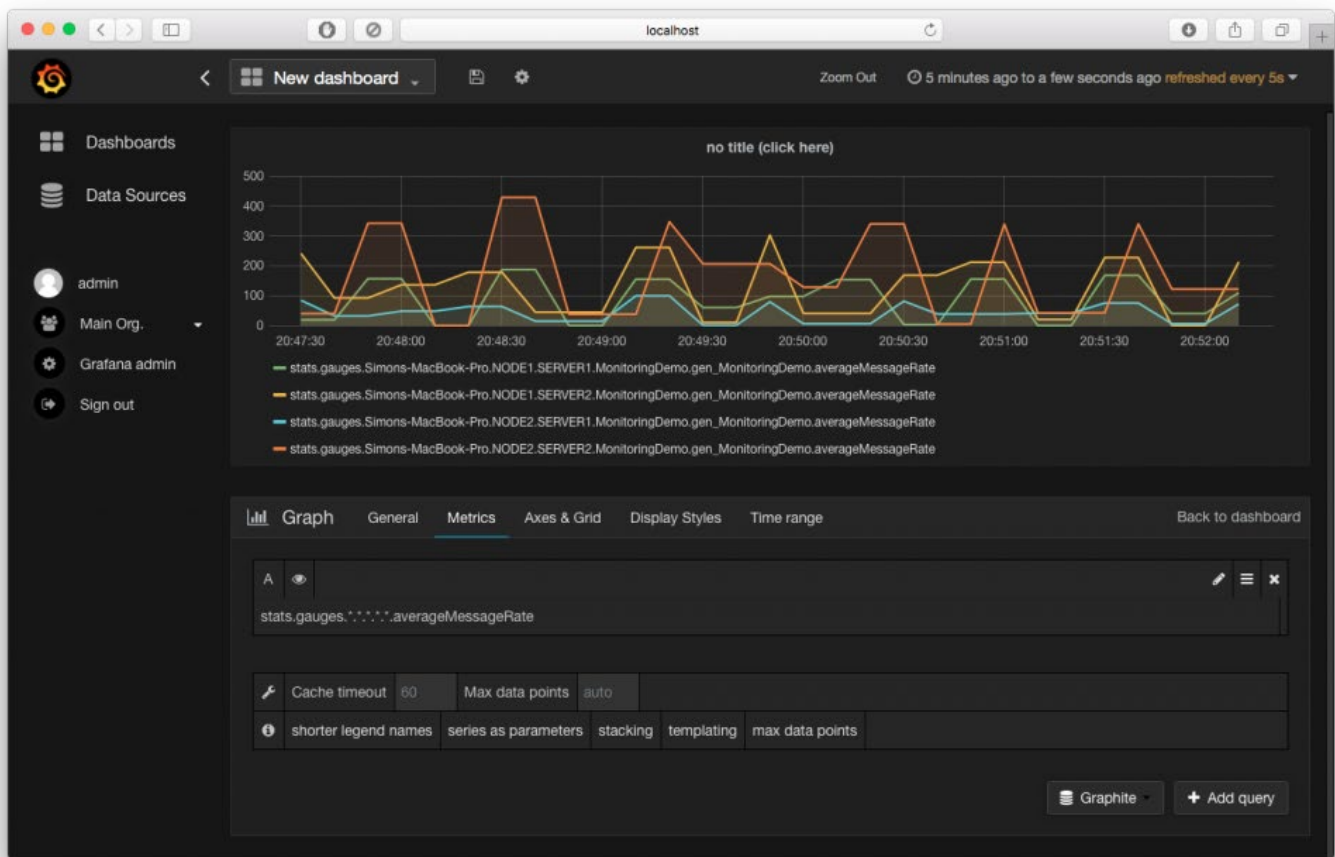


It's a bit bunched up, so change the timing window to "Last 5m" by clicking the time range at the top of the window. Additionally, enable "Auto-Refresh", so the graph automatically updates with the latest data:

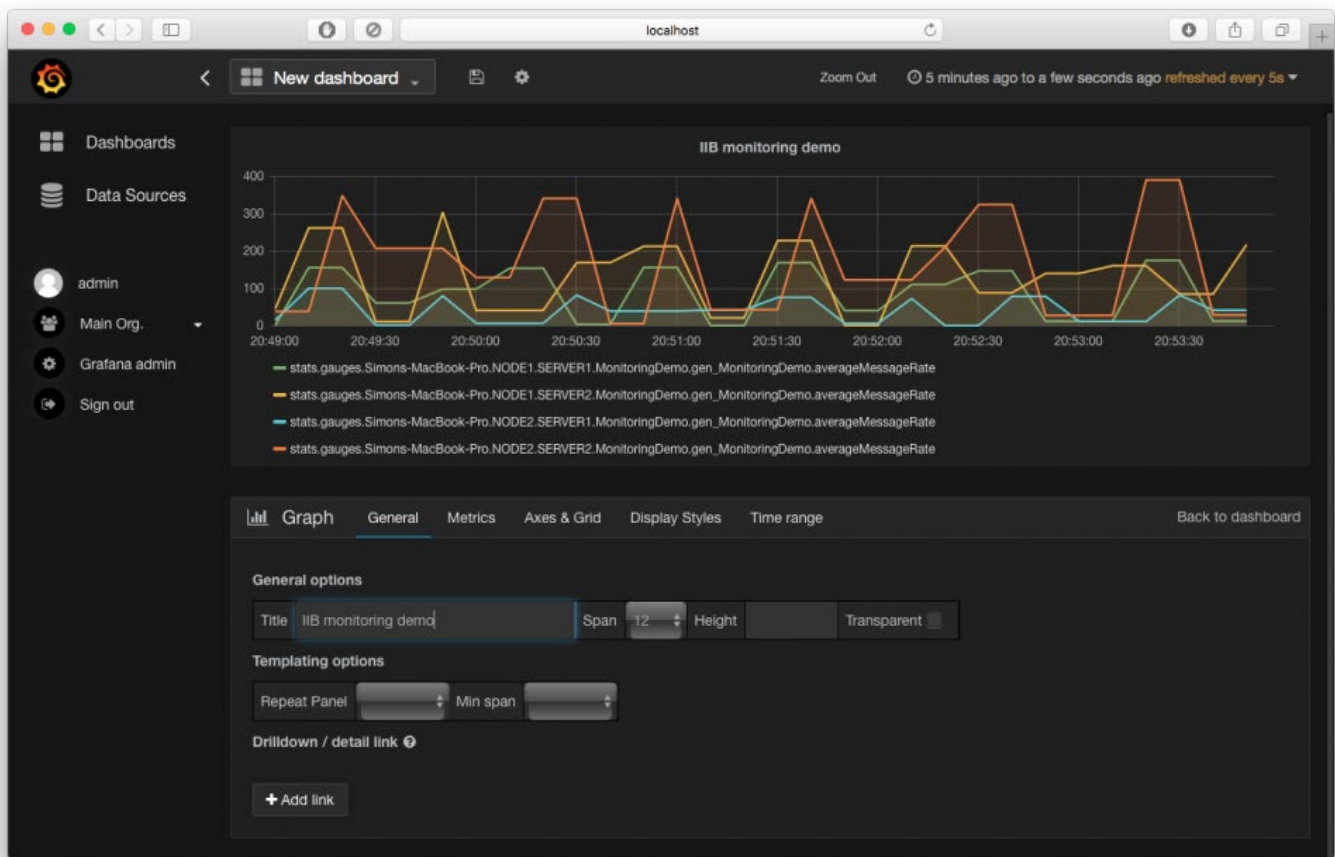


You should start to see the average message rate of all message flows across all integration nodes and integration servers start to scroll past:

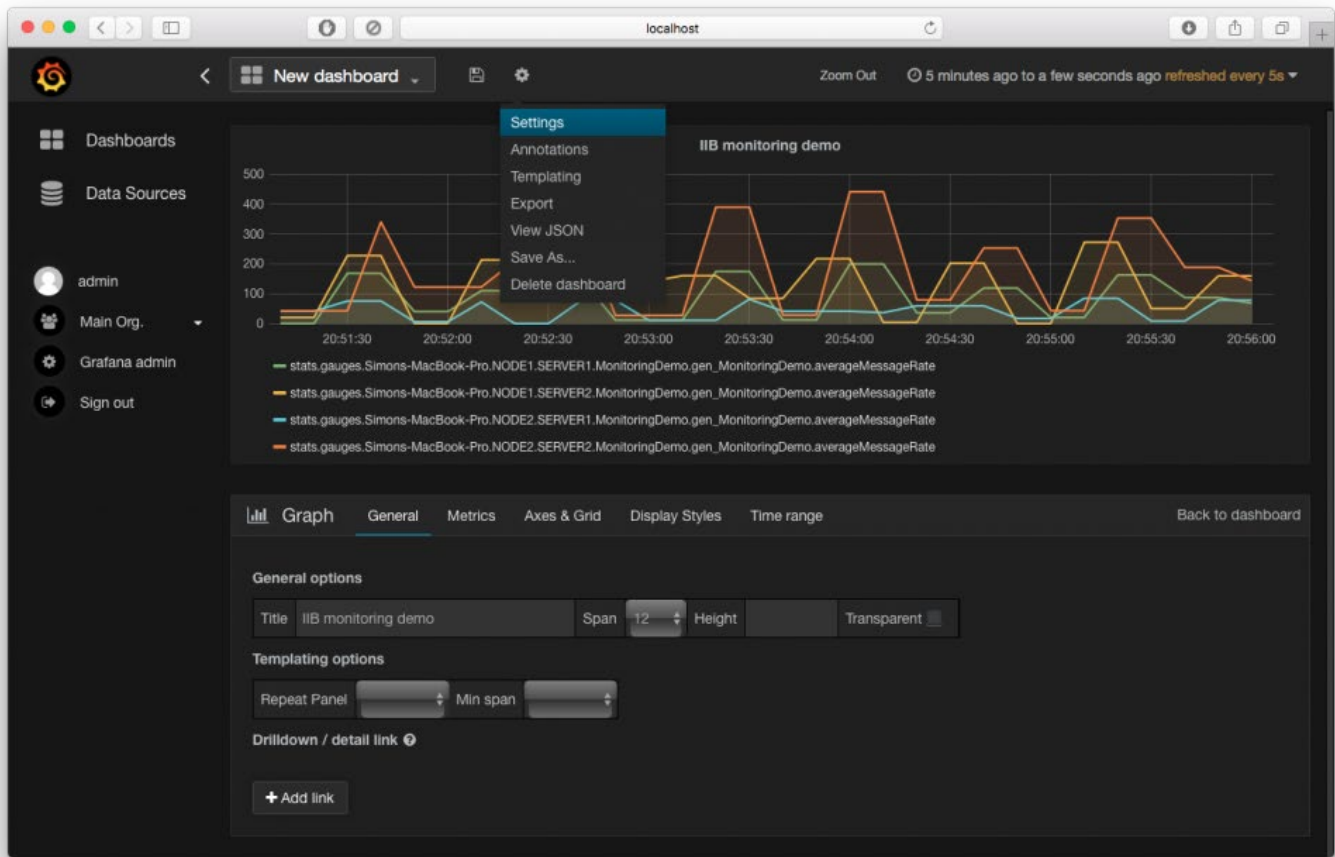




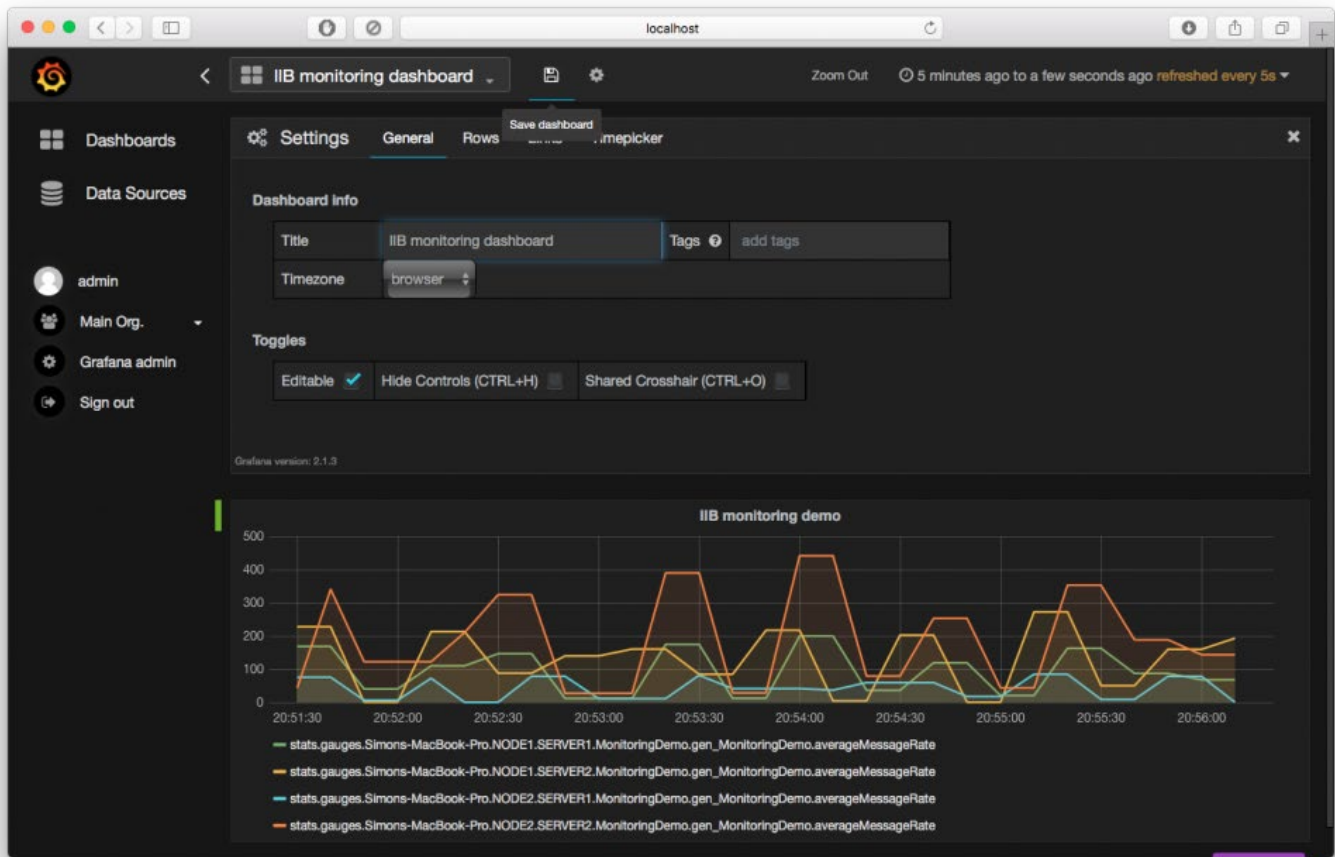
You can give the graph a title by clicking on the “General” tab at the bottom:



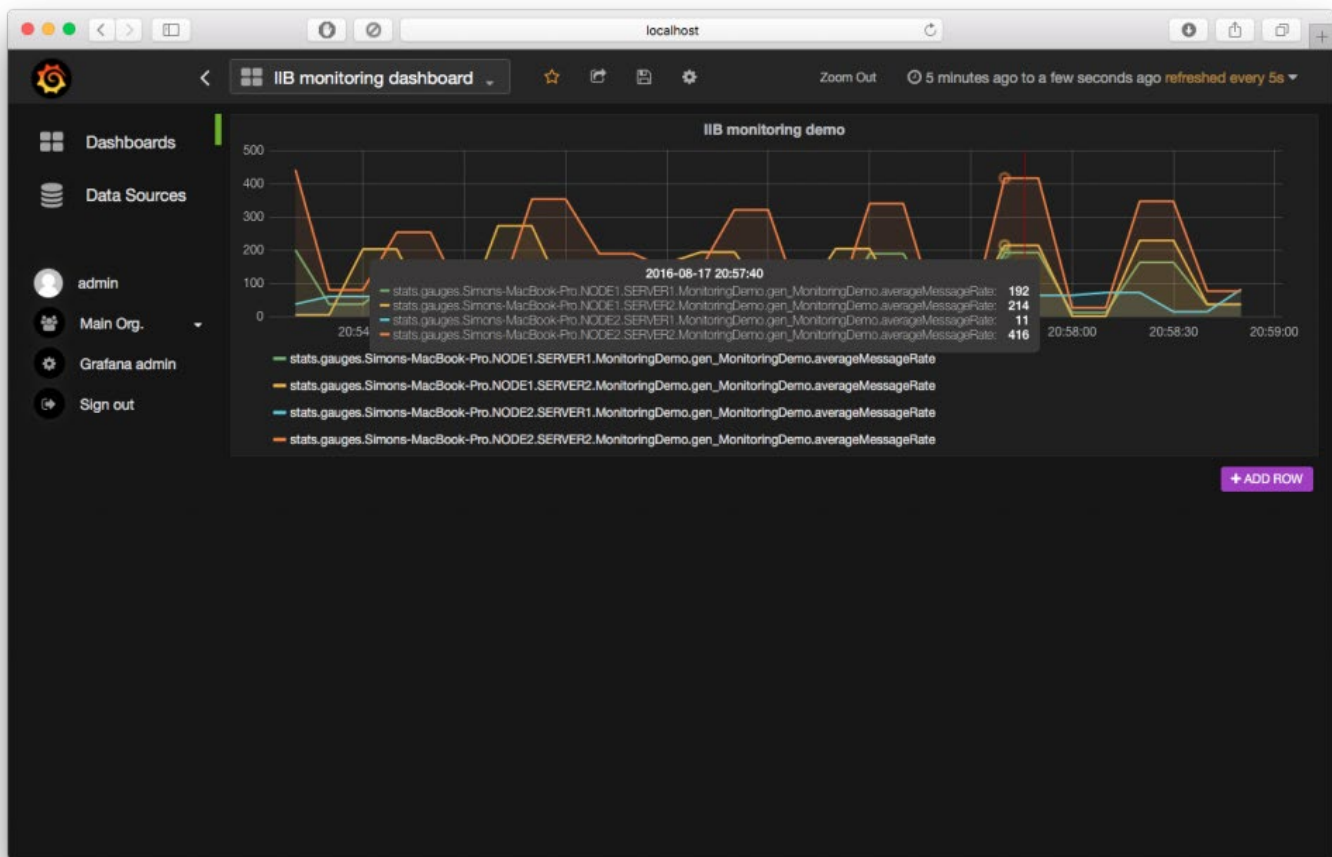
You can also give the dashboard a name by clicking the cog icon at the top of the screen and then click "Settings":



Finally, save the dashboard by clicking the floppy disk icon at the top of the screen:



At this point you can monitor the average message rates of message flows across multiple integration nodes and integration servers. You can hover over the graph for specific values for the average message rate at any particular point in the graph:



You can narrow scope of the data displayed by modifying the metric expression used. For example, if I only want to see the metrics for SERVER2 on NODE1, I would use the metric expression “stats.gauges.\*.NODE1.SERVER2.\*.\*.averageMessageRate” (without quotes).

Other metrics are also provided by the sample plugin, including minimum, maximum, and average CPU and elapsed times. The full set of metrics available at time of writing are:

- hostname.nodename.servername.uniqueflowname.minimumCPUtime
- hostname.nodename.servername.uniqueflowname.maximumCPUtime
- hostname.nodename.servername.uniqueflowname.minimumElapsedTime
- hostname.nodename.servername.uniqueflowname.maximumElapsedTime
- hostname.nodename.servername.uniqueflowname.averageMessageRate
- hostname.nodename.servername.uniqueflowname.averageCPUtimePerMessage
- hostname.nodename.servername.uniqueflowname.averageElapsedTimePerMessage

The C plugin API provides access to the full set of statistics data available from the existing XML and JSON formats. For further documentation on the metrics that are available from the C plugin API, examine the header file BipCsi.h in the product installation.

### Conclusion

In this blog entry, you have seen how the new C plugin API introduced in IBM Integration Bus 10.0.0.5 can be used to access message flow statistics data. You have seen how a C plugin built against the new C plugin API can receive message flow statistics and publish the message flow statistics data into a popular open-source statistics visualization product such as Grafana.

## 10 comments on "Visualizing IBM Integration Bus statistics data using Grafana"

vamshi January 05, 2018

ON Linux x86-64 bit:

IIB version 10.0.0.6

Copied statsdsw.lil to /opt/IBM/mqsi/10.0/iib-10.0.0.6/server/lil

Made sure broker id has read/execute permissions

Restarted the broker

Ran the command: ( localhost is not right but wanted to test the command/config)

```
mqsichangeproperties -e -o StatsdStatsWriter -n hostname,port -v localhost,8125
```

Error:

BIP2087E: Integration node XXXXX was unable to process the internal configuration message.

The entire internal configuration message failed to be processed successfully.

Use the messages following this message to determine the reasons for the failure. If the problem cannot be resolved after reviewing these messages, contact your IBM Support center. Enabling service trace may help determine the cause of the failure.

BIP4041E: Integration server 'XXXXXXX' received an administration request that encountered an exception.

While attempting to process an administration request, an exception was encountered. No updates have been made to the configuration of the integration server.

Review related error messages to determine why the administration request failed.

BIP2212E: Invalid configuration message containing action 'Change' which is not valid for target object 'StatsdStatsWriter'.

The integration node received a configuration message containing the action 'Change' which is not valid for the target object 'StatsdStatsWriter'. This can be caused by a mismatch in levels between the Integration Toolkit and the integration node, or as a result of a user or third party written user-defined node where the implementation library installed at the integration node does not match the node definition held at the Integration Toolkit.

Ensure that the levels of code installed at the Integration Toolkit and integration node are all consistent. If they are, identify the supplier of the target object and report the problem to them. If this is IBM, contact your IBM support center.

BIP8036E: Negative response received.

This command sends an internal configuration message to the integration node. The response received indicated that the internal configuration message was unsuccessful.

Check the system log for further information.

[Reply \(Edit\)](#)

Oscar December 12, 2017

the statics don't send to graphite and for consequence don't apper in grafana, any idea?

[Reply \(Edit\)](#)

Ian\_Larner December 13, 2017

@Oscar Hi,

Please check your set up of StatsD, Graphite, and Grafana, such as exposing port 8125 from the Docker container so that the sample plugin can write statistics data to StatsD, and using that port for configuring the sample plugin. (See the sections in the blog post above, in particular: "Setting up StatsD, Graphite, and Grafana" and "Configuring IBM Integration Bus to publish statistics data to StatsD".)

I've passed your comments on to the IIB Support to see if they can help.

Regards

Ian

[Reply \(Edit\)](#)

Oscar December 13, 2017

Hi Ian\_Larner thanks for you answer, but I find that problem is into machine the comunication localhost with Statsd docker in port 8125 on linux all is on VirtualBox, but if ran Integration Bus 10 on Window 7 and then configure the plugin for send statics to docker in virutal box docker is succfeull, but I will continue checking cause don't work on linux send statics on localhost with docker.

Thank you

[Reply \(Edit\)](#)

Asim April 19, 2017

Hi Simon,

On IIB 10.0.0.7; while running below command on windows broker, I am getting error as shown below.

Command: mqsireportproperties TESTNODE\_admin -e default -o StatsdStatsWriter -r

Error:BIP2212E: Invalid configuration message containing action 'Report' which is not valid for target object 'StatsdStatsWriter'.

\*\* I have already copied statsdsw.lil and statsdsw.pdb files to bin directory

Please advise.Any comments related to 10.0.0.7?

[Reply \(Edit\)](#)

vamshi January 09, 2018

I am facing the same issue after following the instructions in this article

@ Ian\_Larner, can you help us with this ?

Thanks

[Reply \(Edit\)](#)

dp08 March 07, 2017

if we enable this plugin what will be the overall performance impact of all messages flows( per Node ) , Can we use this in production ?

[Reply \(Edit\)](#)

Garth November 21, 2016

What is the advantage of using the C API to access IIB statistics, compared the the existing model?

Thanks,

Garth

[Reply \(Edit\)](#)

Igor Lysenko August 22, 2016

Hi Simon,

Is there any way to see an accounting origin?

Thanks,

Igor.

[Reply \(Edit\)](#)

Simon Stone August 23, 2016

Hi Igor,

The accounting origin is available from the C plugin interface, but the sample C plugin code does not currently use this field.

You can access the accounting origin via the "accountingOrigin" field in the CsiStatsRecordMessageFlow object. That object is available as the "messageFlow" field of the CsiStatsRecord object that gets passed into the "write" callback for the C plugin.

The header file BipCsi.h contains all of the definitions of the structures and fields that are passed from IIB into the C plugin across the new C plugin interface.

Regards,

Simon

[Reply \(Edit\)](#)