

AppDynamics Integration Guide

AppFirst (<http://www.appfirst.com>) delivers the only application-aware operational intelligence platform that provides organizations with unparalleled visibility into their IT ecosystem. Customers are using AppFirst's patented data collection to create a continuous application footprint, enabling IT to support business stakeholders through re-imagined performance based capacity management, application-specific SLAs, and proactive issue identification. Only AppFirst's patented "Miss Nothing" data provides an inside-out view of your application infrastructure by providing the details of every process, network connection, file operation, access to registry, and thread created. Through our extensible, HBase platform, we normalize and correlate this data alongside virtually any other data source you are leveraging (Nagios scripts, log files, StatsD, etc). This guide will show you how to integrate with AppDynamics application metrics.

In AppDynamics:

Account Information

From AppDynamics the following information should be gathered from your AppDynamics Account:

- Controller Name (Url)
- Account Name
- User Name
- Account Access Key/Password

An example of this account information is shown below:

```
-----
Your AppDynamics Controller information is:
Controller Hostname: appfirst.saas.appdynamics.com
Account Name: appfirst
Account Access Key is: 
This Controller hostname must be specified in the Agent configuration for the Agents to report data.

To access your AppDynamics web-based user interface from your browser, use the URL below. Use the login credentials detailed below to access the UI and login.

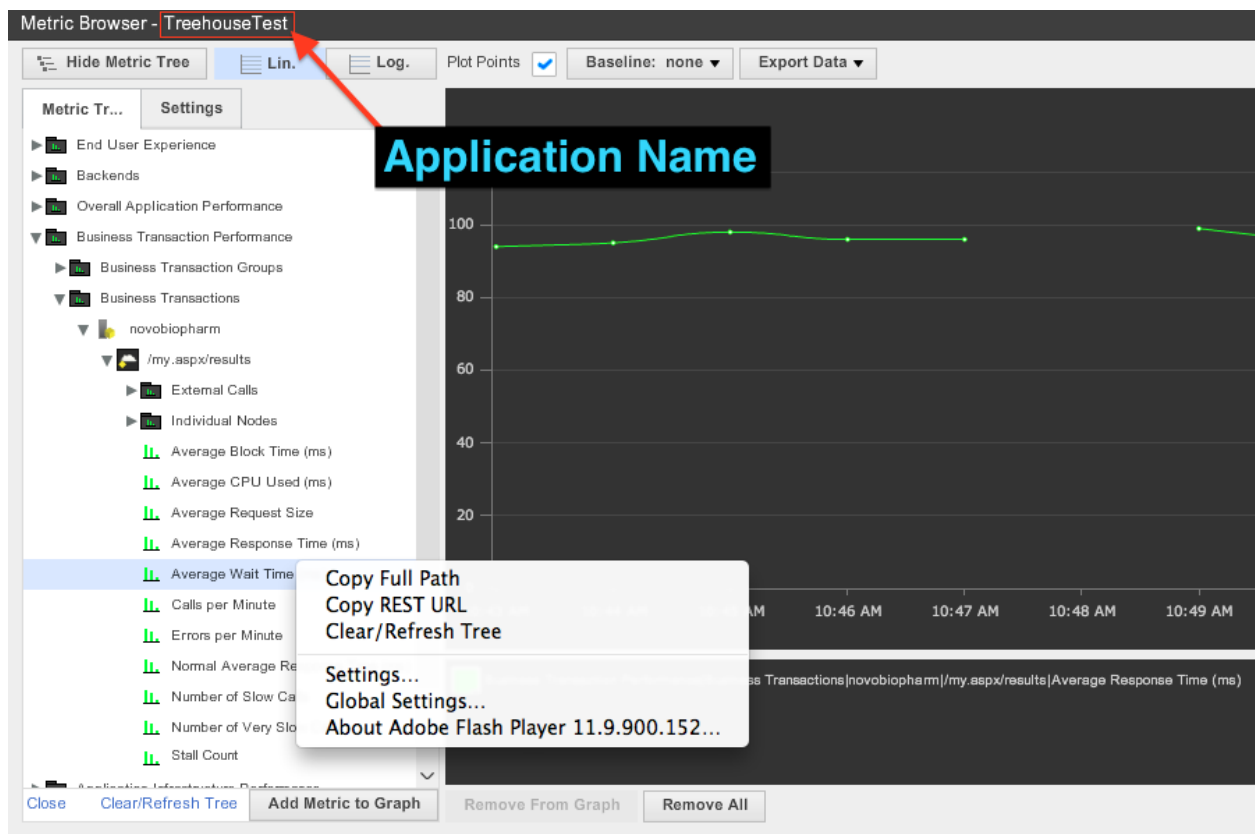
AppFirst URL: http://appfirst.saas.appdynamics.com/controller

Account Name: appfirst
User: appfirst
Password: 
```

AppDynamics Metrics

Next, get the metric path names for each metric that will be collected in the AppFirst product. This can be accomplished using the “metric browser” which lists a tree of all metrics available for an application instance.

Using the metric browser, use either “Copy Full Path” or “Copy REST URL.” For this example, we’ll copy the Full Path. Ensure you also take note of the *Application* - for example, “TreehouseTest” in our example screenshot.



Also note, with the Metric option, you can specify an asterisk to grab all metrics available in the metric tree as listed in the AppDynamics browser.

Alternatively, a `check_appdynamics.sh` script is provided.

In AppFirst:

Once the AppDynamics metrics have been chosen, it’s time to finish the AppFirst configuration. This includes installing the AppDynamics package, configuring metrics to be collected, and creating dashboards or correlated views.

Package Installation

Start by installing the AppDynamics plugin to the system where the AppFirst collector is running. This can be done either through [AppFirst's GitHub](#), or from the package downloaded from the AppDynamics Exchange. This package has been cloned to the repo directory `/usr/share/appfirst/plugins/libexec`, which is the default storage location for AppFirst:

```
/usr/share/appfirst/plugins/libexec$ git clone https://github.com/appfirst/poll-aroid/  
Cloning into 'poll-aroid'...  
remote: Counting objects: 19, done.  
remote: Compressing objects: 100% (15/15), done.  
remote: Total 19 (delta 1), reused 15 (delta 1)  
Unpacking objects: 100% (19/19), done.
```

The following supporting python modules should also be installed:

- *requests*
- *afstatsd*

Python for Linux

In order to install Python modules, from the Linux command line, utilize your favorite python package manager. For example, PIP:

```
sudo pip install requests  
sudo pip install afstatsd
```

Python for Windows

For Windows, ActiveState Python supplies PyPM. Run the PyPM terminal, and use the command:

```
pypm install requests
```

For AfStatsD on windows, you'll need to download our windows extended collector at: https://github.com/appfirst/statsd_clients/tree/stable/python/AFStatsd .

Once downloaded, from the command line run:

```
python setup.py install
```

Configuration of AppDynamics Metrics

It's time to build the script execution and add the AppDynamics metrics to the AppFirst configuration.

For AppDynamics the options are:

Usage: AfPoller.py - AppFirst Poller for AppDynamics

Options:

- h, --help show this help message and exit
- P PLUGIN, --plugin=PLUGIN
[REQUIRED] Name of plug in appdynamics or cloudwatch
- u USERNAME, --username=USERNAME
AppDynamics username (usually name@domain format)
- p PASSWORD, --password=PASSWORD
AppDynamics password
- a APPNAME, --application=APPNAME
Name of your AppDynamics Application
- H HOSTNAME, --hostname=HOSTNAME
Host name (including port) of your AppFirst Controller
- m METRICPATH, --metricpath=METRICPATH
AppDynamics path (with \ | separators) to the metric to poll. You may use wildcards '*'

As an example for Business Transactions, we arrive at the following command:

```
/usr/bin/python /usr/share/appfirst/plugins/libexec/poll-aroid/AfPoller/AfPoller.py --plugin=appdynamics -u  
username@account -p access_key -H controller_hostname -m "Business Transaction Performance|Business  
Transactions|novobiopharm|/my.aspx/results|Average Response Time (ms)" -a TreehouseTest --test --dry-run
```

The --test and --dry-run options execute the command, verbosely, without sending data to our backend. This is a good way to verify the information you are collecting. Once verified, you can remove these options and save the command in your Administration | Collector configuration.

Admin | Collector Config

Within the AppFirst product, navigate to Admin | Collectors and click on the 'Polled Data Config' tab. Below is an example of the command that should be added to the config file and what it looks like in the polled data config editor:

```
command[ADPoll.TreehouseTest]=/usr/bin/python  
/usr/share/appfirst/plugins/libexec/poll-aroid/AfPoller/AfPoller.py --plugin=appdynamics -u  
appfirst@appfirst -p access_key -H appfirst.saas.appdynamics.com -m "Business Transaction
```

Performance|Business Transactions|novobiopharm|/my.aspx/results|Average Response Time (ms)" -a TreehouseTest

Collector Settings

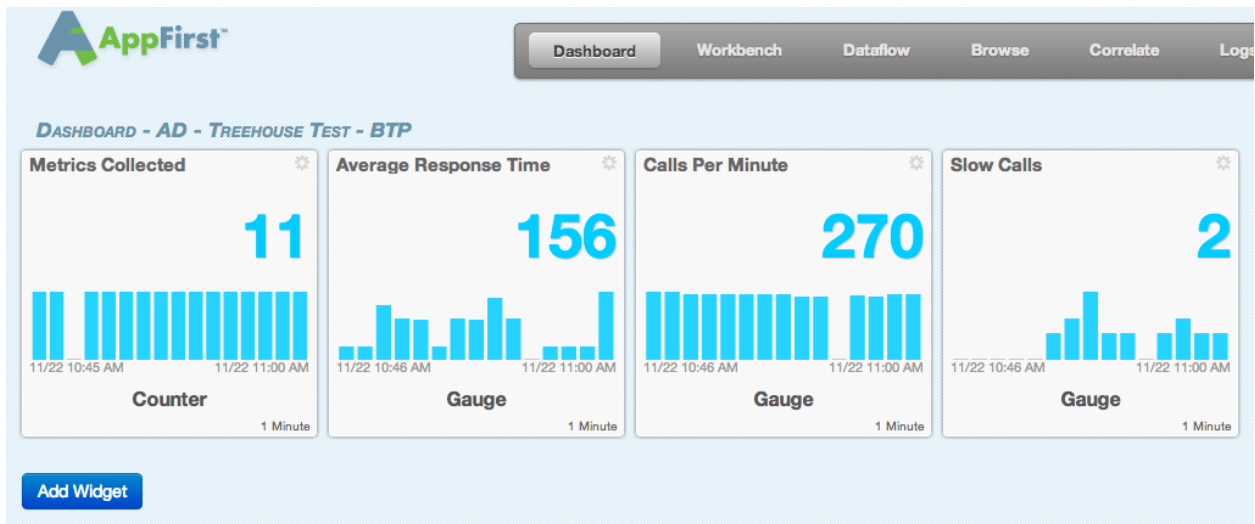
Collector Config **Polled Data Config** **Collector URL Config**

```
# Scrape AppDynamics
command[ADPoll.TreehouseTest]=/usr/bin/python /usr/share/appfirst/plugins/libexec/poll-aroid/AfPol
command[Poll_aroid]=/usr/share/appfirst/plugins/libexec/poll-aroid/check_appdynamics.sh
```

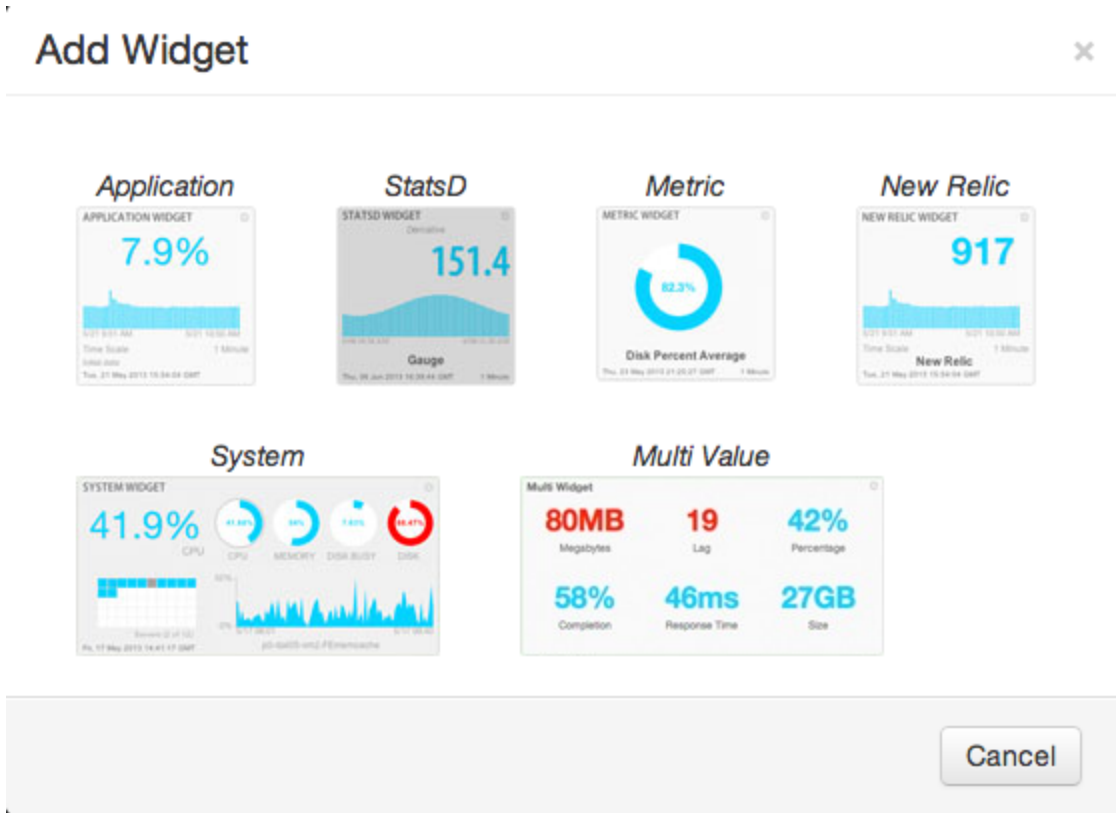
AppFirst Dashboard

Once you have the AppDynamics metrics configured, it is then possible to view and correlate these values within the AppFirst product. Below are examples of dashboard widgets showing AppDynamics values and also how these values can be correlated in other AppFirst views.

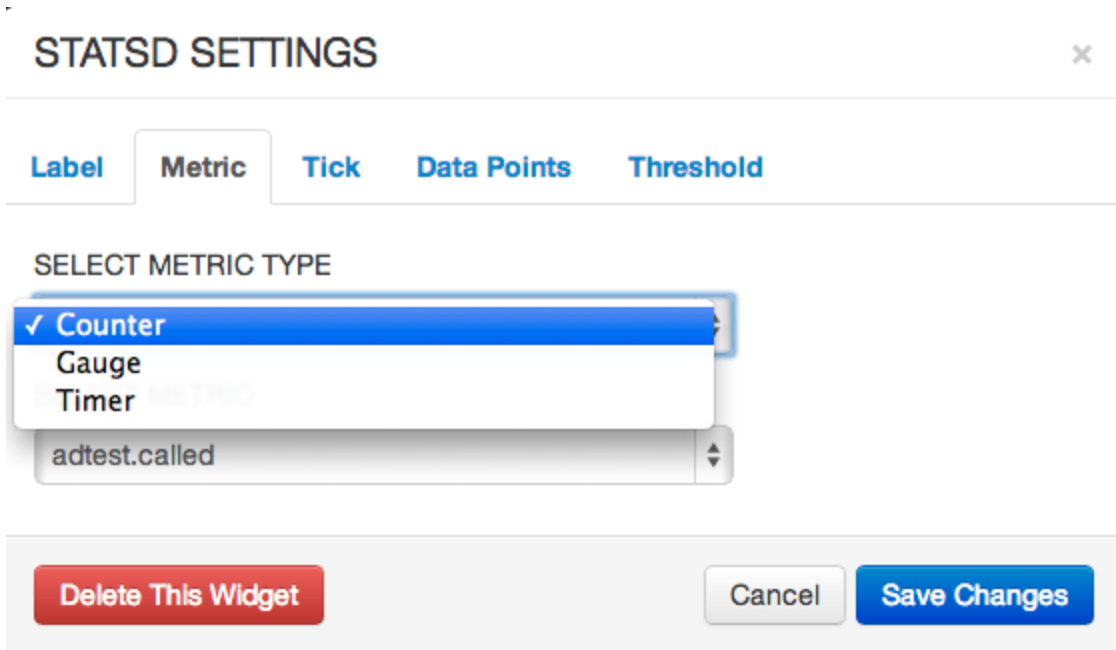
Once deployed, metrics will be available under StatsD widgets in the Dashboard and Correlate.



Adding a dashboard widget is easy. First, use the 'Add Widget' button at the bottom of the page.



Select the StatsD from the gallery. Then, go into settings using the gear at the top right of the new widget. Select the metric type 'Gauge'.



Then select from the metric list which item you'd like to track:

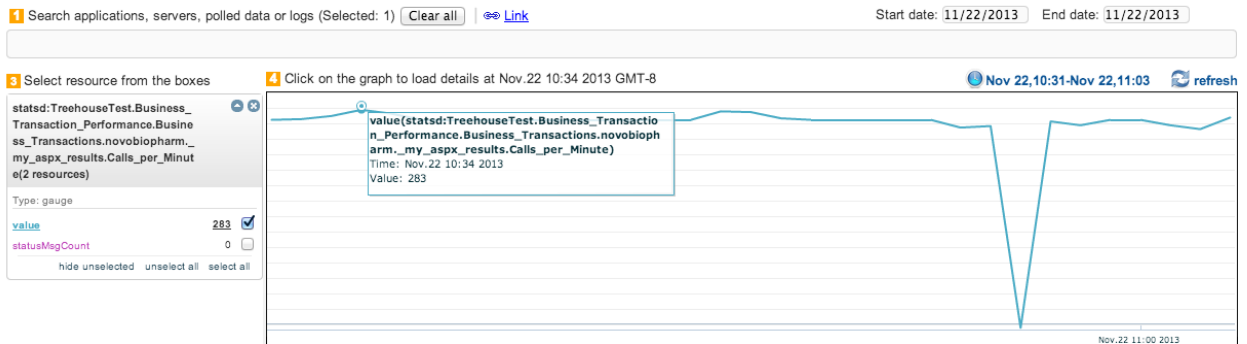
```

✓ LampDev.shell_interrupt
Netstat.*
Netstat.Artemis2.Active.gauge
Netstat.Connections.Active.gauge
TreehouseTest.*
TreehouseTest.Business_Transaction_Performance.*
TreehouseTest.Business_Transaction_Performance.Business_Transactions.*
TreehouseTest.Business_Transaction_Performance.Business_Transactions.novobiopharm.*
TreehouseTest.Business_Transaction_Performance.Business_Transactions.novobiopharm._my_aspx_results.*
TreehouseTest.Business_Transaction_Performance.Business_Transactions.novobiopharm._my_aspx_results.Average_Block_Time_ms
TreehouseTest.Business_Transaction_Performance.Business_Transactions.novobiopharm._my_aspx_results.Average_CPU_Used_ms
TreehouseTest.Business_Transaction_Performance.Business_Transactions.novobiopharm._my_aspx_results.Average_Request_Size
TreehouseTest.Business_Transaction_Performance.Business_Transactions.novobiopharm._my_aspx_results.Average_Response_Time_ms
TreehouseTest.Business_Transaction_Performance.Business_Transactions.novobiopharm._my_aspx_results.Average_Wait_Time_ms
TreehouseTest.Business_Transaction_Performance.Business_Transactions.novobiopharm._my_aspx_results.Calls_per_Minute
TreehouseTest.Business_Transaction_Performance.Business_Transactions.novobiopharm._my_aspx_results.Errors_per_Minute
TreehouseTest.Business_Transaction_Performance.Business_Transactions.novobiopharm._my_aspx_results.Normal_Average_Response_Time_ms
TreehouseTest.Business_Transaction_Performance.Business_Transactions.novobiopharm._my_aspx_results.Number_of_Slow_Calls
TreehouseTest.Business_Transaction_Performance.Business_Transactions.novobiopharm._my_aspx_results.Number_of_Very_Slow_Calls
TreehouseTest.Business_Transaction_Performance.Business_Transactions.novobiopharm._my_aspx_results.Stall_Count

```

Save Changes and your new widget is streaming the data collection configured earlier. Clicking on the number displayed in the widget or using our Correlate features, you can view the data historically for that StatsD item as well:

CORRELATE



For questions or comments visit the Appfirst Community at support.appfirst.com.