

Information Points

On this page:

- [Information Point Data](#)
- [Business Metrics](#)
- [Creating Information Points](#)
- [Viewing Information Point Data](#)
- [Watch the Video](#)

Related Pages:

- [Configure Information Points for PHP](#)

Works with:



Information points let you define custom metrics for your application based on code points you specify in your configuration.

Information points are very similar to data collectors. However, data collectors capture application data only in the context of a business transaction. An information point collects data from a given point in code across all business transactions. For information on data collectors, see [Collecting Application Data](#).

Information Point Data

Every information point comes with default metrics called code metrics. The code metrics are:

- Total call count
- Calls per minute count
- Errors per minute
- Average response time

Business Metrics

You can supplement the code metrics that you get by default by defining custom metrics (business metrics) for the information point. Business metrics reflect the value of runtime data, such as the method parameter, return value, or a value captured by getter chain on the object on which the identified method was invoked. The business metric value represents either the sum or average of the values of the code point you identify as the information point.

Information points can give you significant insight into how the performance of an application corresponds to business performance. For example, depending on the nature of your application, you could use it to resolve business questions such as:

- What is the average value of the credit card total?
- How many credit cards did my application process in a certain time period, regardless of the business transaction?
- What was the average time spent processing a credit card transaction?

A example of a practical use of an information point are ignored exceptions. Exceptions, especially one that occurs frequently, can contribute to CPU spikes in a JVM. If you configure the exception to be ignored in AppDynamics, for example, if it is generated in the underlying application framework and does not have a direct bearing on your application performance, it may not be readily evident to you when the exception is affecting your application. An information point that counts the exception occurrence can help you identify the additional overhead.

Creating Information Points

Before attempting to configure an information point, note that to apply the information point to .NET and JVM 1.5 applications you will need to restart the application server.

Applies to



The easiest way to create an information point is from the call drill down view in the transaction snapshot for a business transaction that uses the method you want to instrument as an information point. When you find an invocation of the method in the call graph, right click on it and select the option for configuring instrumentation on this class/method.

The screenshot shows the AppDynamics Call Graph interface. The top navigation bar includes 'Overview', 'Call Graph', 'Slow Calls & Errors', 'DB & Remote Service Calls', 'Server', 'Data Collectors', and 'More'. The main area displays a call graph with a table of nodes. The selected node is 'com.appdynamicspilot.rest.items.getItems', which has a context menu open over it. The menu options are: 'View Details', 'Copy to clipboard', 'Collapse', 'Expand All', 'Set Root', 'Reset Root', 'Configure Instrumentation for this Class/Method' (highlighted), and 'Export Call Graph starting here'. The table columns are 'Name', 'Time (ms)', 'Percent %', and 'Exit Calls / Threads...'. The selected node shows 0 ms (self) and 0%.

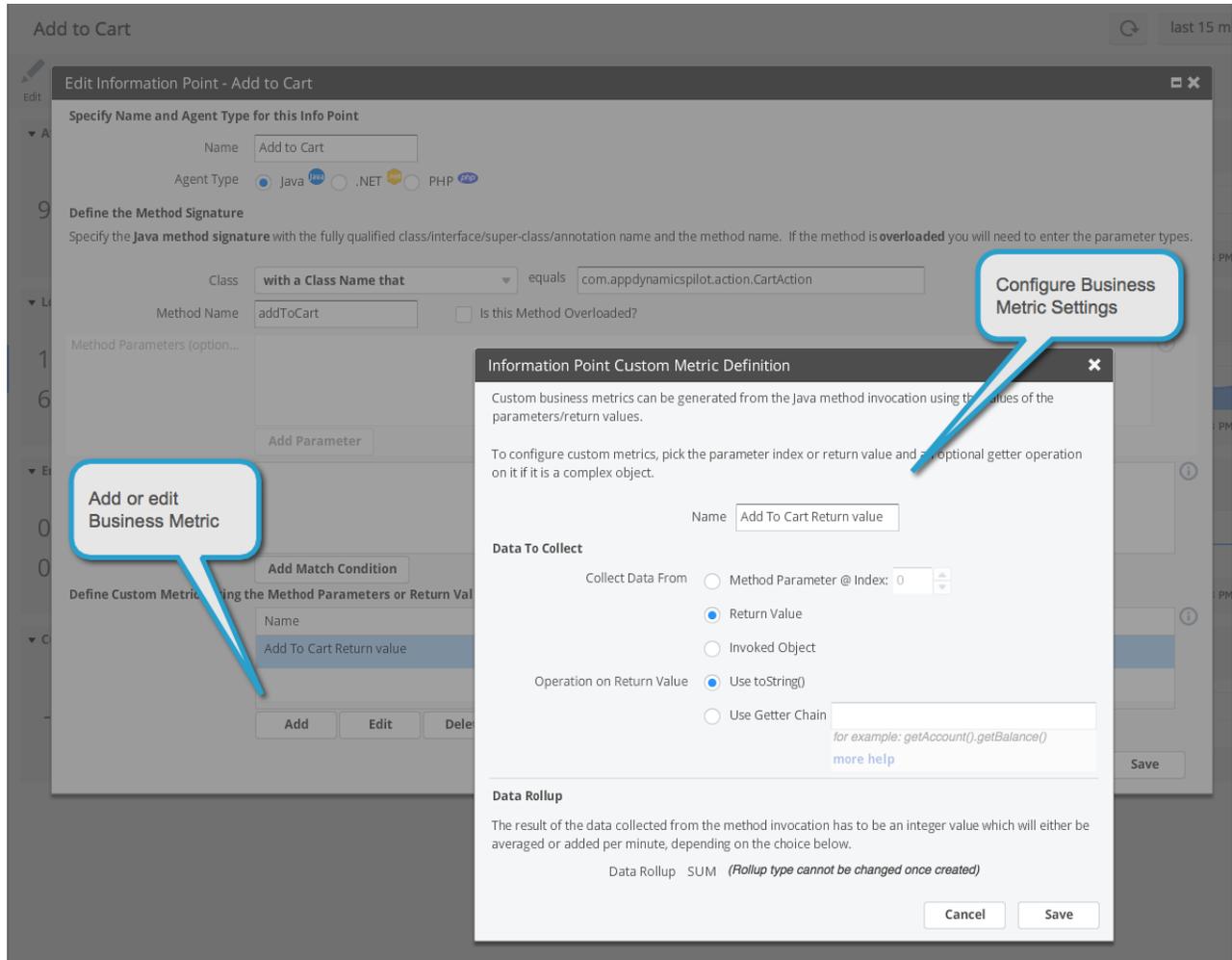
Name	Time (ms)	Percent %	Exit Calls / Threads...
REST Service - org.glassfish.jersey.server.model.internal.AbstractJavaResourceMethodDispatcher:dispatch:11	0 ms (self)	0%	
REST Service - org.glassfish.jersey.server.model.internal.AbstractJavaResourceMethodDispatcher:invoke	0 ms (self)	0%	
Proxy For REST Service - org.glassfish.jersey.server.model.internal.AbstractJavaResourceMethodDispatcher:run	0 ms (self)	0%	
com.appdynamicspilot.rest.items.getItems	0 ms (self)	0%	
Spring Bean - org.springframework.transaction.interceptor.TransactionInterceptor:execute	0 ms (self)	0%	
Proxy For Spring Bean - ItemServiceTarget:invoke	0 ms (self)	0%	
Spring Bean - ItemServiceTarget:getAllItems	0 ms (self)	0%	
Proxy For Spring Bean - ItemPersistence:invoke	0 ms (self)	0%	
Spring Bean - ItemPersistence:getAllItems	19,401 ms (Total)	100%	JDBC

Alternatively, create information points from **More > Information Points** page. You can also modify and remove existing information points .

Specify the class and method for which you want to collect information, along with other identifying information. For details on using the UI to configure code matching for an information point, see the equivalent information on configuring data collectors on [Collecting Application Data](#). While data collectors differ from information points in the scope of the data collected, the configuration for each is very similar.

For information on configuring information points for PHP applications, see [Configure Information Points for PHP](#).

To collect business metrics from the method, add a custom metric for the information point. The custom metric configuration settings let you specify the data to collect as the business metric.

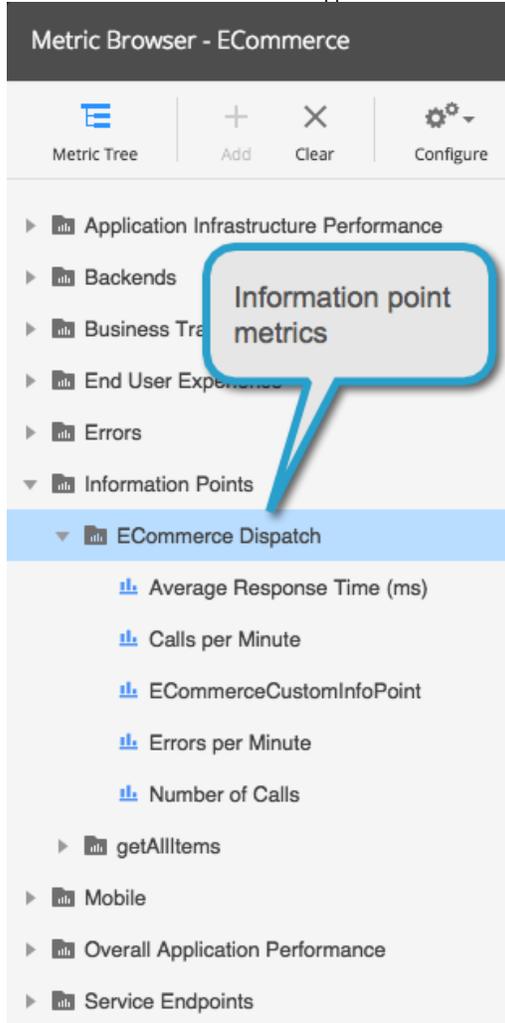


Viewing Information Point Data

You can view information point values in various places in the Controller UI, including the following:

- **Information Points page:** The primary page for viewing and administering information points is the **Information Points** page. From there, click on an information point to view a histogram of the metrics for the information point for the selected time period.

- **Metric Browser:** The metrics also appear in the Metric Browser under the Information Points node:



- **REST API:** Business metrics can be accessed from the [AppDynamics REST API](#).
- **Business Transaction Snapshots:** Information point data does not appear in transaction snapshots by default, but you can configure it to appear by setting the `enable-info-point-data-in-snapshots` node property to `true`. When the `enable-info-point-data-in-snapshots` node property is set, information point calls appear in the User Data section of the snapshot. For more information, see [App Agent Node Properties](#).

Watch the Video

For full-screen viewing, click [What is an Information Point?](#)