

## Configure Instrumentation

### On this page:

- [Access Instrumentation Settings](#)
- [Other Configuration Topics](#)

In AppDynamics, instrumentation refers to how app agents interact with your application software to gather performance data and report it back to the AppDynamics Controller. The first step to instrumenting your application is [installing the app agents](#) on the servers where the code runs.

App agents ship with default instrumentation settings that cover the most common types of application frameworks and programming patterns. If your requirements are more complex, you can customize the instrumentation behavior to suit your environment.

See [Configure Instrumentation Overview](#) for information on how AppDynamics applies instrumentation settings across your business application.

## Access Instrumentation Settings

Navigate to **Configuration > Instrumentation** to customize the following:

- [Transaction detection rules](#) that define entry points for [business transactions](#).
- [Backend detection rules](#) that define exit points for your [business transactions](#).
- [Error detection](#) settings that define which transactions qualify as [error transactions](#).
- [Service endpoints](#) to view metrics specific to services and transaction segments outside the context of a single business transaction.
- [Diagnostic data collectors](#) for an app agent to extract additional context about a specific transaction at transaction snapshot time. You can also configure data collectors for [Application Analytics](#).
- [Call Graph Settings](#) to tune the amount of data the app agent collects for call graphs.
- [JMX metrics](#) that the Java agent collects for specific products and frameworks. See "Configure JMX Metric Rules" on [Configure JMX Metrics from MBeans](#).
- Memory monitoring settings for agents that collect instance tracking data for specific classes or objects. See the following topics:
  - [Object Instance Tracking for Java](#)
  - [Object Instance Tracking for .NET](#)
  - [Object Instance Tracking for Node.js](#)
- [Asynchronous transaction demarcators](#) to set end points for asynchronous transactions.

## Other Configuration Topics

In addition to the instrumentation configuration options covered in this topic, there are many other ways to customize AppDynamics for your environment. For more information on the items on the **Configuration** page, refer to the following topics:

- "Configure Thresholds" on [Transaction Thresholds](#).
- "View Baselines" on [Dynamic Baselines](#).
- [Information Points](#)
- [Development Level Monitoring](#)

If you have an End-User monitoring license, see the following topics for more information on **User Experience App Integration**:

- [Correlate Business Transactions for Browser RUM](#)
- [Automatic Injection of the JavaScript Agent](#)
- [Assisted Injection](#)