Transaction Snapshots

On this page:

Related pages:

- Transaction Snapshot Collection
- Call Graphs
- Access Database Visibility from Application Monitoring Views

AppDynamics monitors every execution of a business transaction in the instrumented environment, and the metrics reflect all such executions. However, for troubleshooting purposes, AppDynamics takes snapshots of specific instances of a transaction. A transaction snapshot gives you a cross-tier view of the processing flow for a single invocation of a transaction.

Call drill downs, where available, detail key information including slowest methods, errors, and remote service calls for the transaction execution on a tier. A drill down may include a partial or complete call graph. Call graphs reflect the code-level view of the processing of the business transaction on a particular tier.

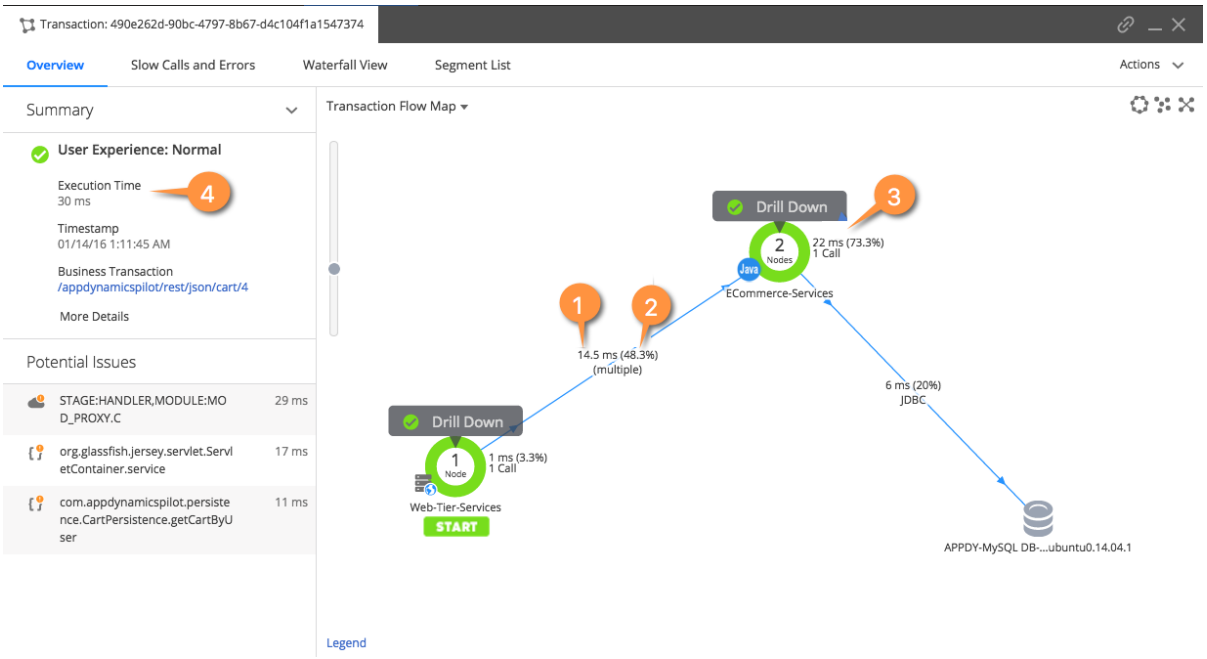
This topic covers how to view transaction snapshot data to monitor and troubleshoot business transaction performance. To learn more about when AppDynamics takes transaction snapshots or how to configure transaction snapshot settings, see Transaction Snapshot Collection.

View Transaction Snapshots

You can access business transaction snapshots from the following locations in the AppDynamics Controller:

- *Troubleshooting* > *Slow Response Times* or *Troubleshooting* > *Errors* left navigation tree for a business application
- Transaction Snapshots tab on the Business Transaction Dashboard

Double-click a business transaction snapshot to display the snapshot viewer. The following screen capture and its accompanying table identify the metrics available in the transaction flow map:



Callout	Metric Name	Explanation
1	Tier Response Time (ms)	The total response time for the call as measured at the calling tier. This includes the processing time on the called tier as well as on any tiers and backends it calls in turn.
2	Percentage of Time Spent (%)	The time spent downstream processing at all downstream tiers and backends as measured by the calling tier and represented as a percentage of the entire execution lifespan of a business transaction. This metric does not include the processing time of asynchronous activities, if any.
3	Asynchronous Activity Processing Time (ms)	The processing time of all asynchronous activities at this tier. This metric does not contribute to the overall tier response time because the activity is asynchronous by nature. This metric is calculated by adding the execution times of all asynchronous activities at a tier and the time spent in communication between other tiers and backends as follows: *Asynchronous Activity Processing Time = Asynchronous-activity-1-processing-time + Asynchronous-activity-2-processing-time + so on.* In the Metric Browser, you can view the Average Async Processing Time metric, which shows the average of the async activity processing time over the selected time range.
4	Execution Time (ms)	Total time spent processing by the business transaction in all affected tiers and communication with other tiers and backends. This metric does not include the processing time of the asynchronous activities. How ever, in the case of Wait-for-Completion, the originating business transaction will take longer to process the request due to blocking and waiting for all the activities to complete before proceeding. The formula for this metric is calculated by summing up the processing times of a Business Transaction at a particular Tier/communication between Tiers/Backends as follows: *Execution Time = Time-spent-processing-in-Tier-1 + Time-spent-processing-in-Tier-2 + Time-spent-communicating-with-Tier-2 + so on.*

The Potential Issues panel highlights slow methods and slow remote service help you investigate the root causes for performance issues. Click an item in the Potential Issues list to view the call in the call graph.

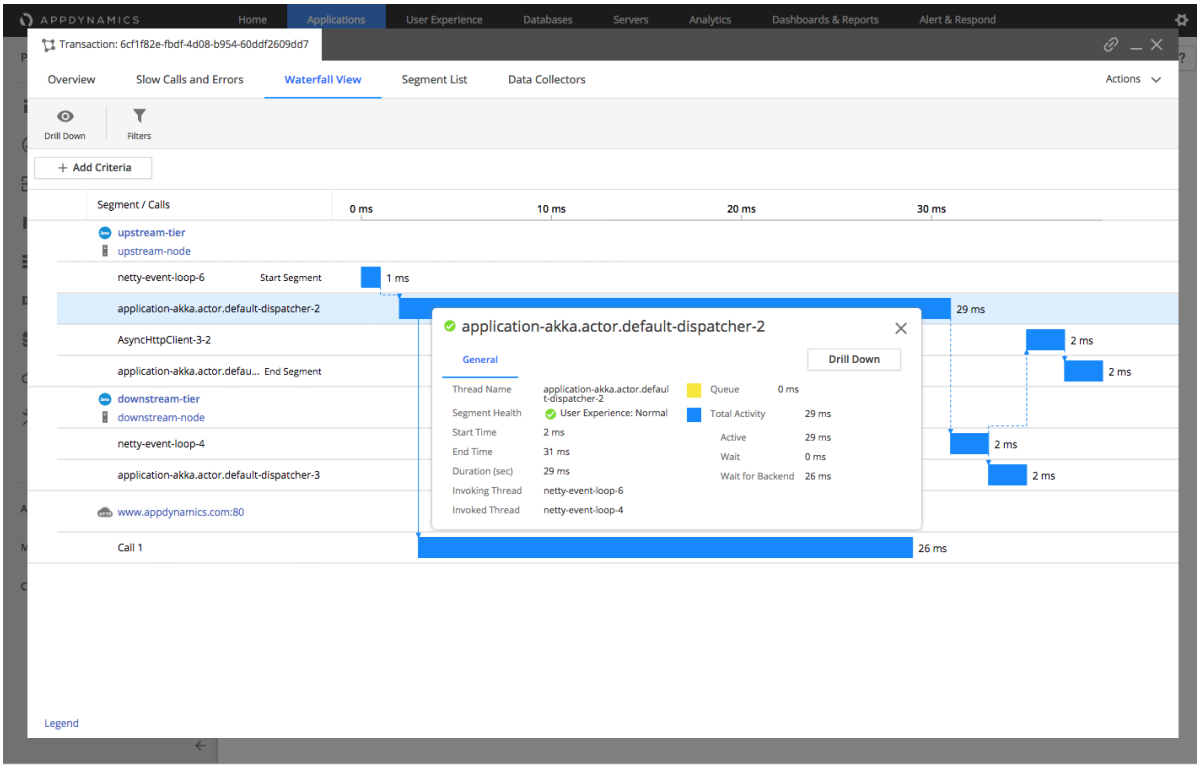
In the flow map for a business transaction snapshot, a tier with a *Drill* Down link indicates AppDynamics has taken a call graph for that tier.

To drill into a call graph for a correlated application in a transaction with cross application flow, you need view permissions to the destination business application.

If you use the Java Agent and monitor Oracle databases with Database Monitoring and you have configured snapshot correlation between the Java Agent and the Database Agent, the flow map may also include database drill downs.

The flow map or Overview is one of several views of the business transaction in the snapshot viewer. Other views include:

- *Slow Calls and Errors*, which presents information on the slowest database and remote service calls, slowest methods, and errors. You can gain further insight into these slow calls and errors either by viewing their details on by drilling down into their call graphs.
- *Waterfall View*, which shows the execution of an individual business transaction broken into execution segments. Each segment (shown as a blue bar) represents the time spent executing code on a particular thread within an instrumented application runtime, or the time spent waiting for responses from uninstrumented backends. Handoffs between segments are shown as solid lines for synchronous requests, and dotted for asynchronous requests. The waterfall view allows you to quickly determine which calls consumed the transaction time for a given snapshot. You can click on a segment to view the resource wait time for its business transaction.



- Segment List, which shows the various legs of the transaction in descending order of duration and give access to their snapshots and allows you to drill down into their details.

Call Drill Downs

A call drill down contains details for that business transaction execution on a particular tier. It takes you to the code-level information for the transaction.

The contents of a transaction snapshot containing asynchronous segments look slightly different if you access the snapshot via the Business Transaction view or via the App/Tier/Node views:

- Initially the Business Transaction view only displays the originating segments for the transaction. You have the option drill down into the asynchronous segments as desired.
- The App/Tier/Node views surface all the segments that are relative to that specific entity. Therefore you access one of these views, you can see all segments, originating and asynchronous.

Node Drill Down

The node drill down organizes diagnostic data among the following tabs:

- The Overview tab includes a problem summary, execution times, CPU time stamp, tier, node, process ID, thread name, etc.
- The Call Graphs tab lists call graphs showing the execution flow for the transaction on a given tier. For details, see Call Graphs .
- The Slow Calls and Errors tab lists all the slow method calls and calls that resulted in an error. You can use the *Hot Spots* slide r to sort calls by execution time with the most expensive calls in the snapshot at the top.
- The Error Details tab exposes exception stack traces and HTTP error codes.
- The DB & Remote Service Calls tab shows all SQL query exit calls to databases and exit calls to other remote services such as web services, message queues or caching servers. See Database queries and batching for more information about how AppDynamics handles SQL exit calls.
- The Server tab displays graphs for hardware—CPU Memory, Disk IO, Network IO— Memory—Heap, Garbage Collection, Memory Pools—JMX, and more. If you have Server Visibility, you will have access to full performance details for the server hardware and operating system
- For customers using Network Visibility, the Network tab shows charts related to the impact of the network on the transaction and other pertinent data. For information on network KPIs and troubleshooting see KPI Metrics in Network Dashboard and Application Flow Map and KPI Metrics in Right-Click Dashboards.
- The Data Collectors tab shows pertinent application data for the transaction snapshot. For configuration options, see Data Collectors.

  HTTP Data: HTTP payloads contain basic data such as the URL and session ID, and additional data for Servlet entry points, Struts, JSF, Web Services, etc. You can use HTTP data collectors to specify which query parameter or cookie values should be captured in the transaction snapshot.

  Cookies: The snapshot can use cookie values to help identify the user who initiated the slow or error transaction.

  User Data: User data from any method executed during a transaction, including parameter values and return values, to add context to the transaction. You can use method invocation data collectors to specify the method and parameter index.
  In cases where an exit call is made just before a business transaction starts, exit call information can show up in this field, particularly if the transaction is marked as slow or having errors. Please note that sensitive information on the exit call may be shown in this situation.

- The More tab shows hows metrics for the node that deviate the most from the established baselines as Node Problems. It also shows all the Service Endpoints invoked during the snapshot and the Servlet URI and Process ID of the transaction.

Database queries and batching

AppDynamics normalizes SQL queries and by default does not display raw/bind values. You can configure SQL capture settings to monitor raw SQL data in the queries. Individual calls that take less than 1 second are not reported.

When returning data to a JDBC client, database management systems often return the results as a batched response. Each batch contains a subset of the total result set, with typically 10 records in each batch. The JDBC client retrieves a batch and iterates through the results. If the query is not satisfied, the JDBC client gets the next batch, and so on.

In the SQL query window, a number followed by an X in the Query column means that the query ran the number of times indicated within a batch. The value in the Count column indicates the number of times that the batch job executed.

Database Drilldown

Database drill-downs tell you the following about the transaction:

- Queries lists the queries consuming the most time in the database as top SQL statements and Stored Procedures. Comparing the query weights to other metrics such as SQL wait times may point you to SQL that requires tuning.
- Clients displays the hostname or IP addresses of the Top N clients using the database. A database client is any host that accesses the database instance.
- Sessions displays the Session ID of the Top N sessions using the database sorted by time spent.
- Schemas shows the names of the Top N busiest schemas on the database server.

Using the Transaction Snapshot List

You can view transaction snapshots generated in the UI time range from the *Transaction Snapshots* tab of the application, tier, node, or business transaction dashboards. From there you can:

- Compare Snapshots shows the performance of calls in two snapshots as a side-by-side comparison.
- Identify the most expensive calls / SQL statements in a group of Snapshots shows the calls that take the most time across the snapshots you have selected. You can select up to 30 snapshots.
- Find snapshots using the filter options.

The Controller purges transaction snapshots after two weeks by default, but you can configure the snapshot retention period. You can archive a snapshot to preserve it beyond the normal snapshot lifespan. For example, if you want to retain a snapshot associated with a particular problem for future analysis. To archive a snapshot, select it from the list and choose *Actions* > *Archive*.

 To archive a snapshot, you need the *Application level - Can create applications* permission.

The file cabinet icon in the far right column indicates that the snapshot is an archive snapshot ().

To display only archived snapshots in the snapshot list, filter the snapshot list and check *Only Archived*.