



# Business Transactions

## On this page:

- [View Business Transactions](#)
- [Transaction Entry Points and Exit Points](#)
- [Sample Business Transaction](#)
- [Refine Business Transactions](#)
- [Business Transaction Limits](#)
- [Correlate Business Transaction Logs](#)

## Related pages:

- [Organize Business Transactions](#)
- [Transaction Detection Rules](#)
- [Service Endpoints](#)

In the AppDynamics application model, a business transaction represents the end-to-end, cross-tier processing path used to fulfill a request for a service provided by the application. This topic introduces and describes business transactions.

## View Business Transactions

To view business transactions for a business application, click **Business Transactions** in the application navigation tree. The business transaction list shows key metrics for business transactions for the selected time range.

Only business transactions that have performance data in the selected time range appear in the list by default. You can show inactive business transactions for the time range by modifying the filter view options.

Other ways to modify the default view include showing transactions that belong to business transaction groups or transactions that exceed a configurable average response time. You can also choose which performance metrics appear for business transactions in the list from the **View Options** menu.

To see the actions you can perform on business transactions, open the **More Actions** menu. Actions include viewing health rule violations, configuring thresholds, renaming business transactions, grouping business transactions, [starting a diagnostic session](#) for the transaction, and classifying a business transaction as a [background task](#).

## Transaction Entry Points and Exit Points

When you install an app agent, the agent detects incoming calls and registers transactions based on the default transaction detection rules. Automatic detection rules describe entry points for transactions based on supported frameworks.

Usually, more than one tier participates in the processing of a transaction. A request to the originating tier may invoke services on:

- Another instrumented tier, called a downstream tier.
- Remote services that are not instrumented.

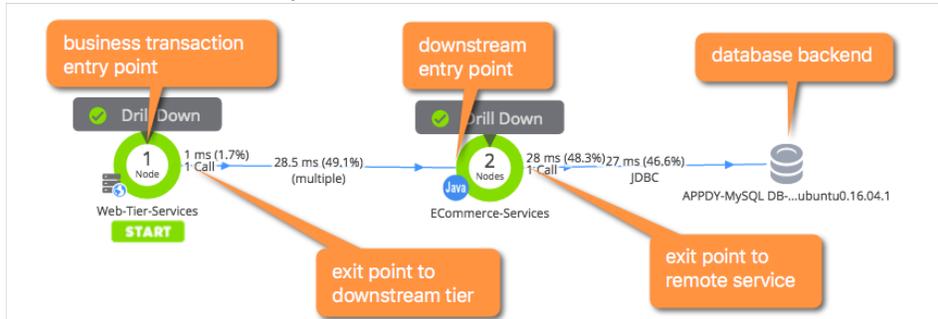
Outbound requests from an instrumented application tier are called exit points. Downstream tiers may in turn have exit points that invoke other services or backend requests.

App agents tag exit point calls with metadata describing the existing transaction. When an agent on a downstream tier detects an entry point that includes transaction metadata from another AppDynamics app agent, it treats the entry point as a continuation of the transaction initiated on the upstream tier. This linking of upstream exit points to downstream entry points is called correlation. Correlation maintains the client request context as it is processed by various tiers in your business application.

## Sample Business Transaction

Consider, for example, the fictional ACME Online application. The application exposes a checkout service at <http://acmeonline.example.com/checkout>. A user request to the service triggers the following distributed processing flow and actions:

1. The business transaction entry point at the originating tier is /checkout URI, which is mapped to a Servlet called CheckoutServlet.
2. The request results in the originating tier invoking the createOrder method on a downstream tier, the ECommerce-Services server.
3. The inventory tier application calls a backend database, which is an exit point for the business transaction. The request context is maintained across tiers, including calls to backend tiers.



4. Any user request on the entry point is similarly categorized as this business transaction, the Checkout business transaction.

To enable detection of all the components in a distributed business transaction, downstream agents must be at the same AppDynamics release or newer than upstream agents. This is true whether the tiers are all built on the same platform (for example all Java) or multiple platforms (a Node.js tier calling a Java tier, calling a .NET tier and so on).

## Refine Business Transactions

While the default rules can go a long way towards getting you a useful list of business transactions to track, an important part of implementing AppDynamics is verifying and refining the business transactions used to monitor your application. The business transaction you are monitoring should reflect those operations that are important to your application and business. It is important to consider the [limits on business transactions](#), and apply your refinements accordingly.

Refining your business transaction list requires a solid understanding of the important business processes in your environment. Identify the 5 to 20 most important operations in the application. These are the key operations that must work well for the application to be successful.

Important services can be indicated by the number of calls or calls per minute received by the business transactions generated for the services. You can refine the list of transactions you want to monitor by locking down critical transactions and enabling automatic cleanup of stale transactions. For the Java and .NET environments, you can use interactive [Live Preview](#) tools to help identify important transactions.

You can add business transactions manually from a virtual business transaction called "All Other Traffic", which is populated with transactions once the business transaction registration limits are reached, as described below.

To customize the business transaction list, you can use either of these approaches:

- You can modify existing business transactions by grouping, renaming, or removing the business transactions. Most of these operations are available from the business transaction list. Use this approach to apply relatively minor, small scale changes to the current business transaction list. For more information, see [Organize Business Transactions](#).
- You can affect how business transactions are created by modifying the automatic discovery rules. You can modify rule to similarly achieve business transaction grouping and naming, and to exclude transactions. Discovery rules also enable you to define new entry points for business transactions. Discovery rule modification is a powerful mechanism for changing transaction discovery on a larger scale. For more information, see [Transaction Detection Rules](#).

## Business Transaction Limits

When reviewing and refining your business transaction limits, it's important to consider the business transaction limits for the Controller and app server agents. Business transaction limits prevent boundless growth of the business transaction list.

The default limits are:

- Business Application Limits: Each application is limited to 200 registered business transactions.
- App Server Agent Limits: Each agent is limited to 50 registered business transactions.

There is no limit at the tier level.

Also note that the app agent limit applies to each app agent, not to the machine. If you have multiple app agents on a single machine, the machine the business transactions originating from the machine could be up to the number of agents times 50.

## Correlate Business Transaction Logs

For those times when tracing application code doesn't provide enough clues to track down the cause of a problem, AppDynamics provides visibility into the transaction logs that can be correlated to specific business transaction requests. Log correlation visibility requires a license for both Transaction Analytics and Log Analytics. See [Business Transaction and Log Correlation](#).

