USING APPLICARE › TRANSACTIONS

Business Transactions User Guide

Applicare 9.2 · Updated September 13, 2024 · **Pending technical review**

Overview

Key Business Transactions enables you to capture performance statistics and set SLAs for Web or Java method calls. It also allows you to track individual transactions with detailed performance metrics and call graphs.

Web key business transactions are defined by specifying one or more URI patterns. Any request matching these defined patterns will be intercepted and statistic will be gathered under the given business transaction name.

Java key business transactions operate similarly, but instead of URIs, you define class and method name patterns.

The Profiling Agent must be enabled for business transactions to function on a server. For web based business transactions to work you need to deploy both HTTP Profiling and Business Transactions aspects. You can enable or deploy these on the Configuration -> Configure Dynamic Instrumentation page.

Configuring Business Transactions

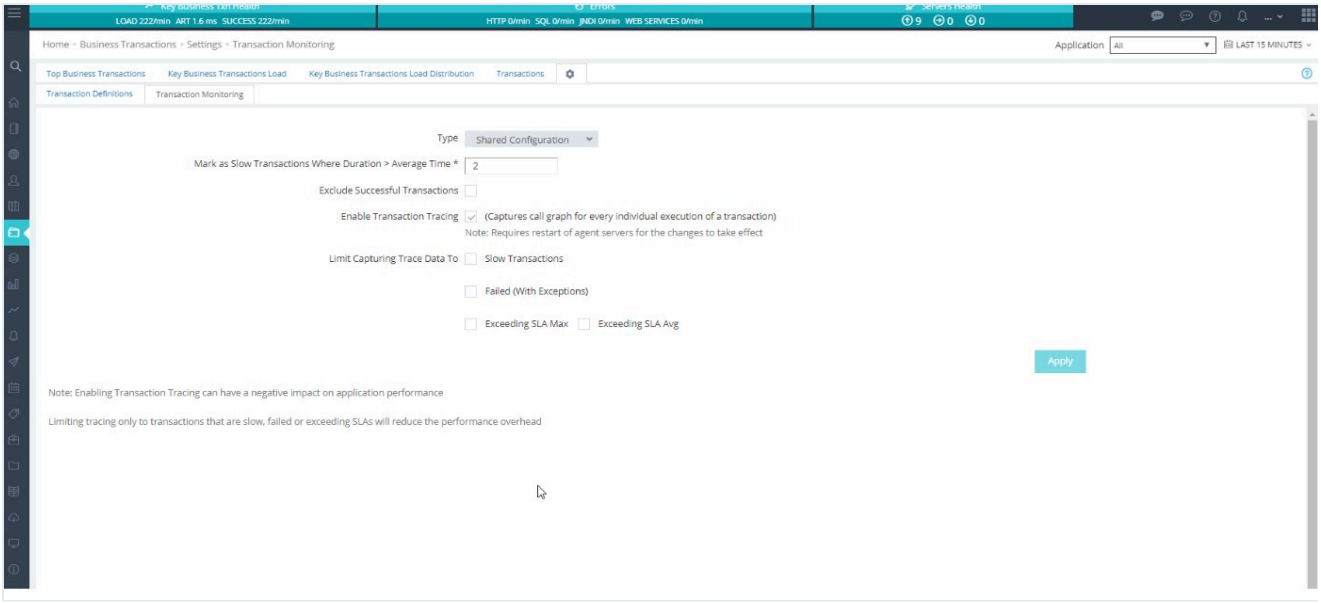
Prerequisites

Before Applicare 7.1.4 you must enable Standard Mode Instrumentation and activate the instrumentation agent on the servers where you want to track business transactions. See Configuring Dynamic Instrumentation on how to enable Standard Mode instrumentation on servers.

For Applicare 7.1.4 onwards At a minimum you must enable HTTP, SQL, JDBC Connection and Business Transactions profiling by deploying those aspects in Configuring Dynamic Instrumentation In Applicare 7.1.4 onwards, 'Enable Per-Transaction Monitoring' in Business Transactions configuration has no impact and BTM wouldn't start until 'Business Transactions' aspect is deployed along with other prerequisites in 'Configure Dynamic Instrumentation'.

Configuration

You can access the business transaction configuration page by navigating to Business Tansactions -> Configure (wheel button on right) Transactions Monitoring. On Transaction Monitoring tab you can configure the amount of details to be captured on configured transactions.



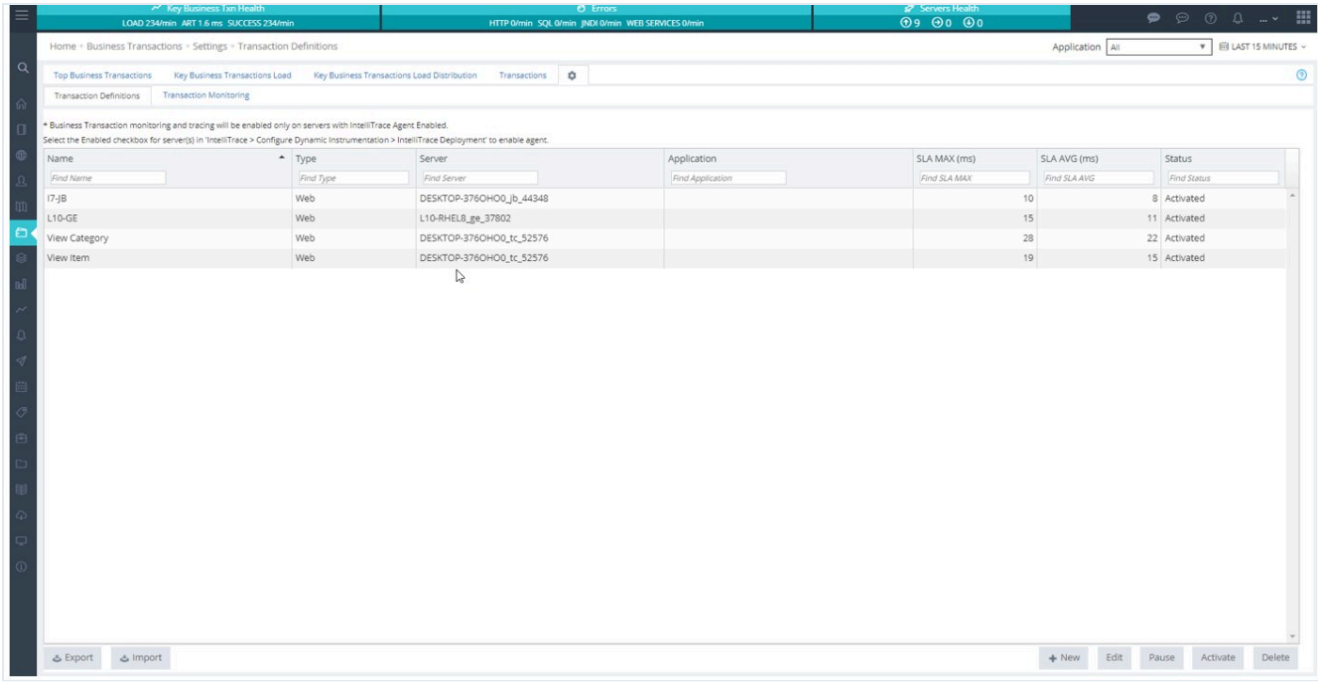
Exclude Successful Transactions allows you to exclude successful/good transactions from Busniss Transaction Monitoring. This helps concentrate on bad transactions which needs a fix or improvement.

Enabling Per-Transaction monitoring gives you the benefit of monitoring your business transactions performance over time as a whole and on a per server basis. Performance overhead of gathering this level of data is minimal and can be enabled even on production systems.

Enabling Transaction Tracing will capture execution graph of each transaction including Java method calls, SQLs and their parameters. Performance overhead of gathering this level of data might be noticable specially in production systems with very high number of business transaction executions per minute. You can reduce the overhead by limiting call graph tracing to transaction executions that are slow, failed or exceeding SLA limits.

Business Transaction Definition

To view the Business Transaction Definition page, navigate to: **Applicare Home -> Business Transactions -> Configure Business Transactions -> Transaction Definitions**.

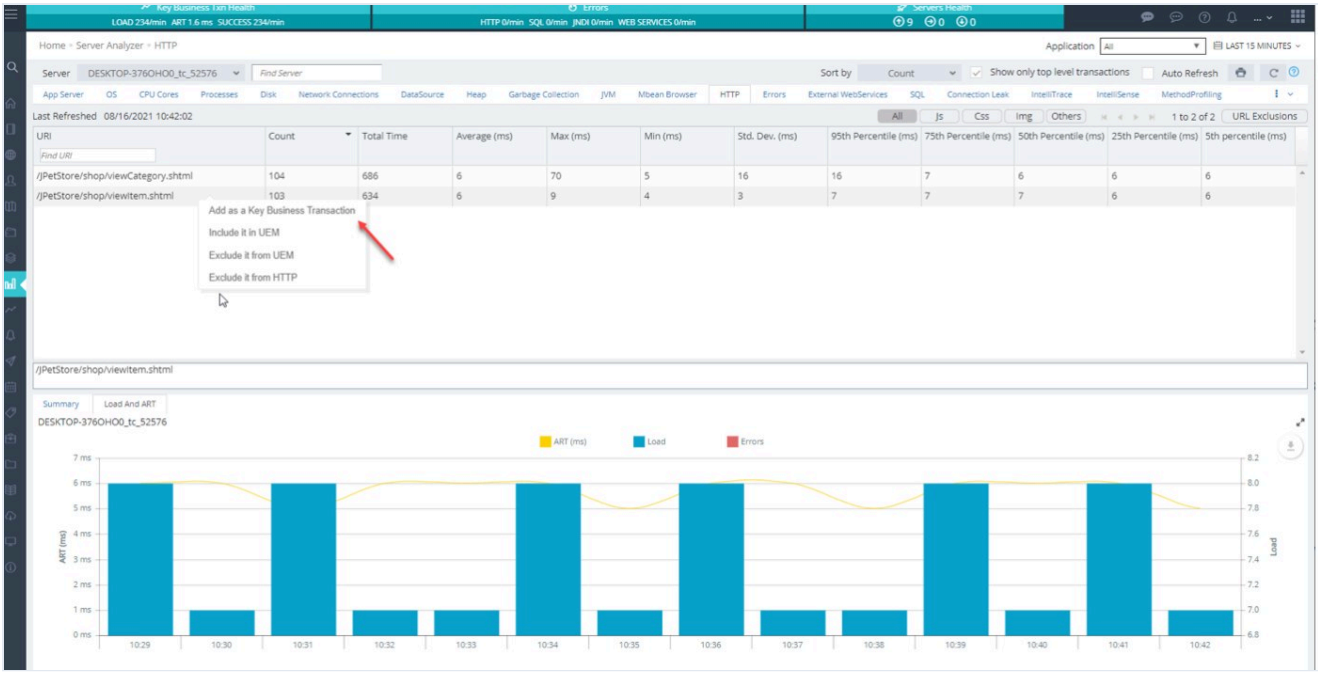


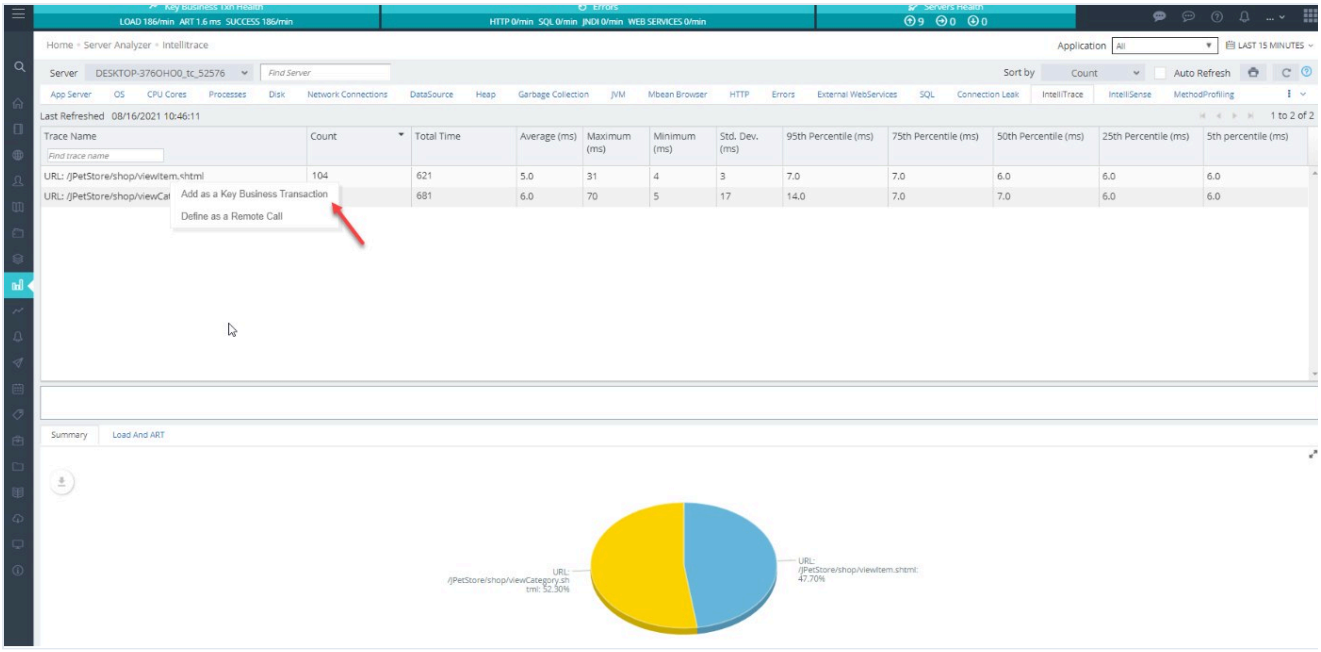
This page shows all defined Key Business Transactions and lets you create, edit, or delete them. Changes made here take effect immediately across the domain without requiring a server restart.

You can export definitions to an XML file or import them into another Applicare environment using the Export/Import buttons.

Defining Business Transactions

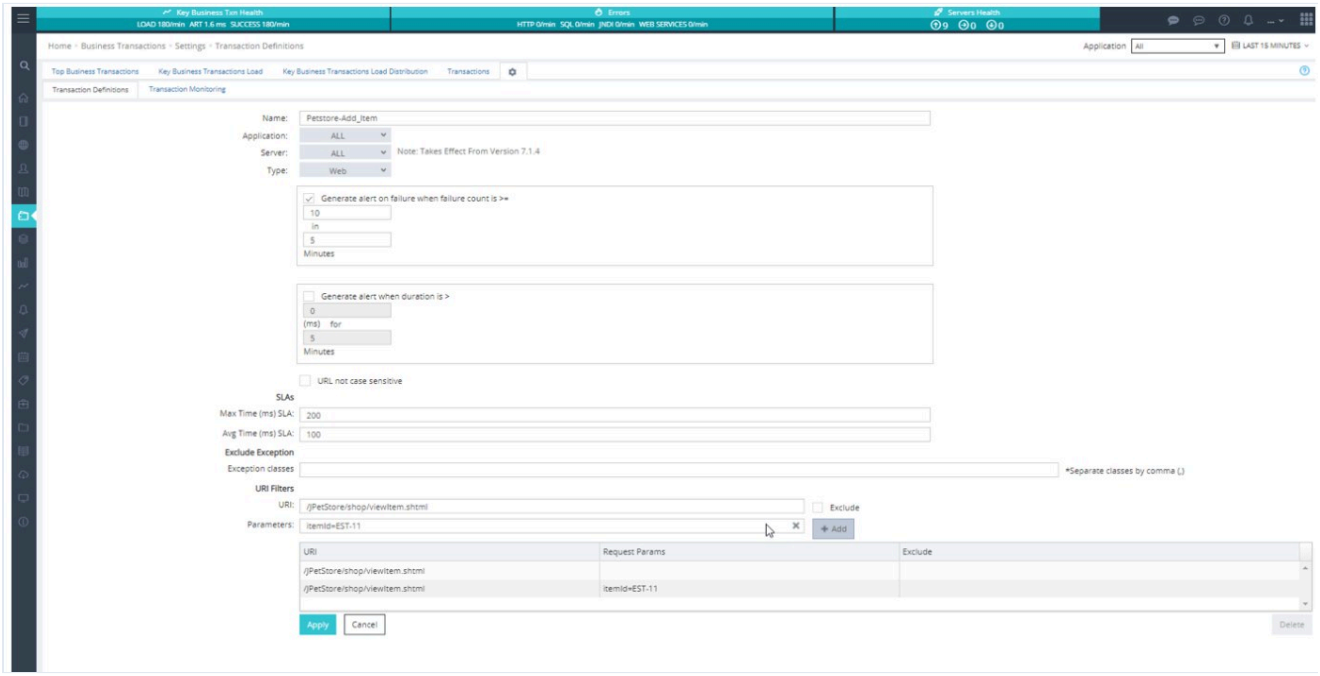
Business transactions can be defined either manually from this page or within the IntelliTrace Analyzer. If IntelliTrace Standard Mode or Advanced Mode with IntelliTrace Profiling is enabled, you can navigate to the HTTP or IntelliTrace tabs in IntelliTrace Analyzer. Right-click on any autodetected URL or Java method to define a Business Transaction for that URL or method.





Add a New Web Business Transaction

To manually add a new Web Business Transaction, click the **New** button on the Transaction Definition page.



1. Provide a **Name** for the Business Transaction

2. Select **Web** from the **Type** drop down box

3. Provide average and maximum SLA values. Leave them blank or set to zero if you don't want to enforce SLA values for this transaction definition

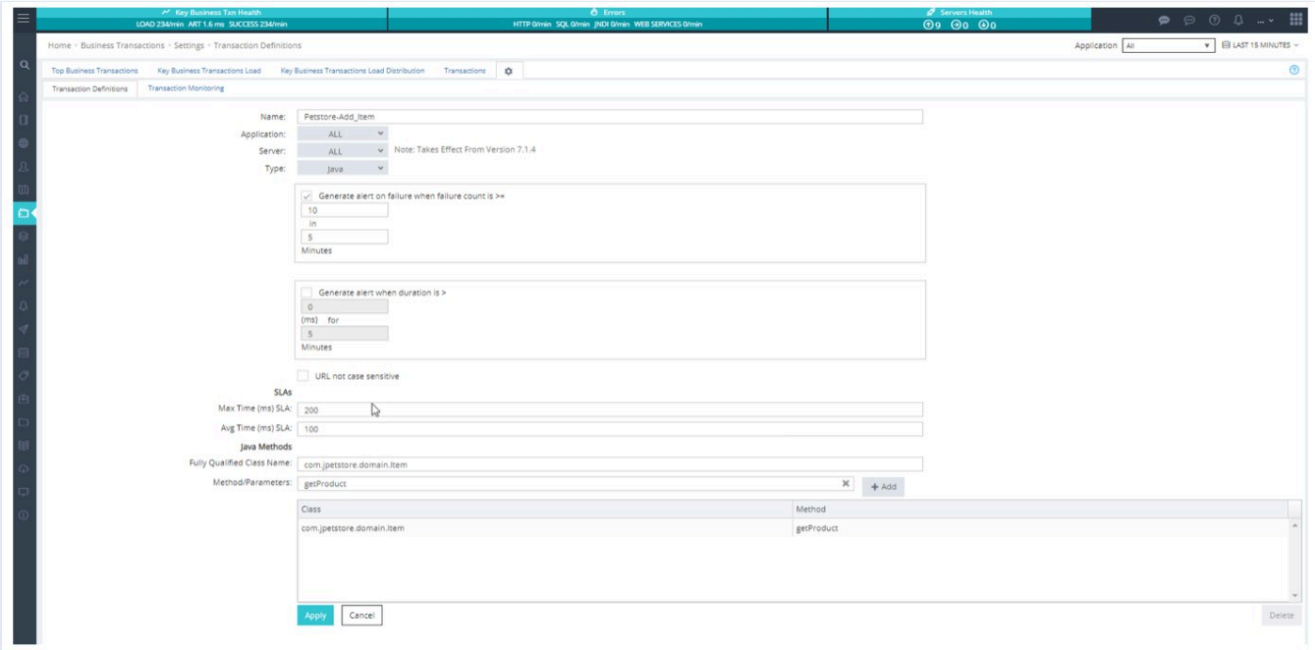
4. Provide a URI pattern (Note: You can use * character as a wild card here)

5. Optionally provide a HTTP parameter pattern if you want to filter further by HTTP parameters. This follows the normal HTTP convention of separating key and value with = sign and separating multiple parameters with & sign. (Note: You can use * character as a wild card for values)

6. Click **Add** button to add the URI pattern to definition. You must add at least one URI pattern. Multiple URI patterns can be added to a definition and HTTP requests matching at least one URI pattern will be included in the Business Transaction. 7. Once you have added all URI Patterns, click **Save** button to save the Business Transaction Definition.

Add New Java Business Transaction

To manually add a new Java Business Transaction, click the **New** button on the Transaction Definition page



1. Provide a **Name** for the Business Transaction

2. Select **Java** from the **Type** drop down box

3. Provide average and maximum SLA values. Leave them blank or set to zero if you don't want to enforce SLA values for this transaction definition

4. Provide fully qualified class name (Note: You can use * character as a wild card for the class name)

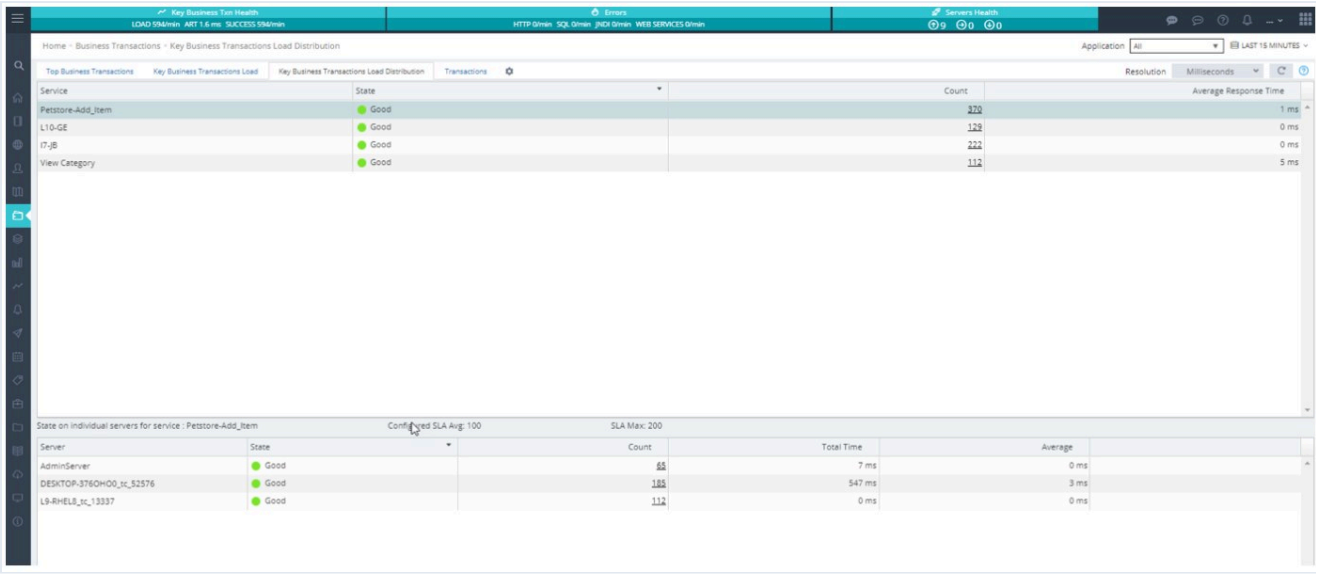
5. Provide a method name (Note: You can use * character as a wild card here)

6. Click **Add** button to add the class/method pattern to definition. You must add at least one class/method pattern. Multiple patterns can be added to a definition and method calls matching at least one pattern will be included in the Business Transaction.

7. Once you have added all patterns, click **Save** button to save the Business Transaction Definition.

Key Business Transactions Load Distribution View

This view displays the state of Key Business Transactions across all servers.



If a Transaction meets SLA values defined for it across all servers, it is shown as in **Good** state. If its not meeting SLA values across all servers, the state will be **Failed**. If its not meeting SLA values on some servers then the state will be **Partial Failure**.

Clicking on a Business Transaction will display the state of the transaction on individual servers in the lower table.

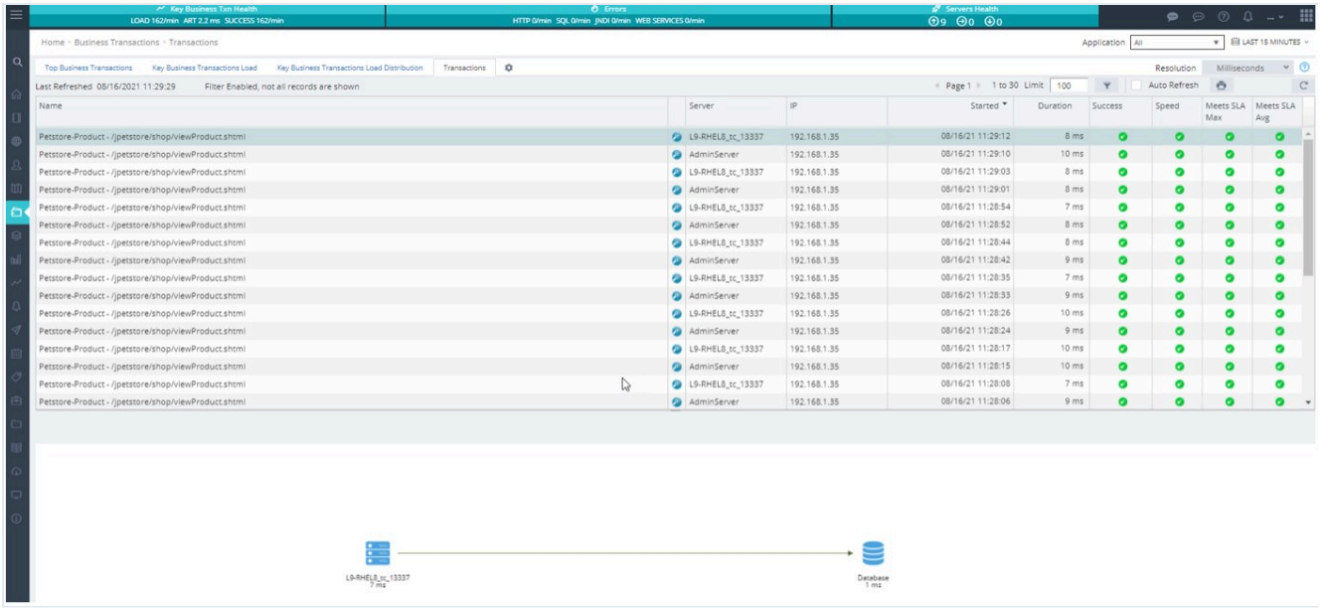
Double clicking on a Transaction in the top table will take you to the Transactions tab and will set the search filter to the selected transaction.

Double clicking on a Transaction in the bottom table will take you to the Transactions tab and will set the search filter to the selected transaction and server.

The states displayed on Summary View are calculated periodically for performance reasons and are not updated real time.

Transactions View

Transaction view displays transaction execution stats if Per-Transaction Monitoring is enabled. Double clicking on any transaction displayed here will take you to the detailed view where you can see the in parameters, exceptions if any and trace details if transaction tracing is enabled.

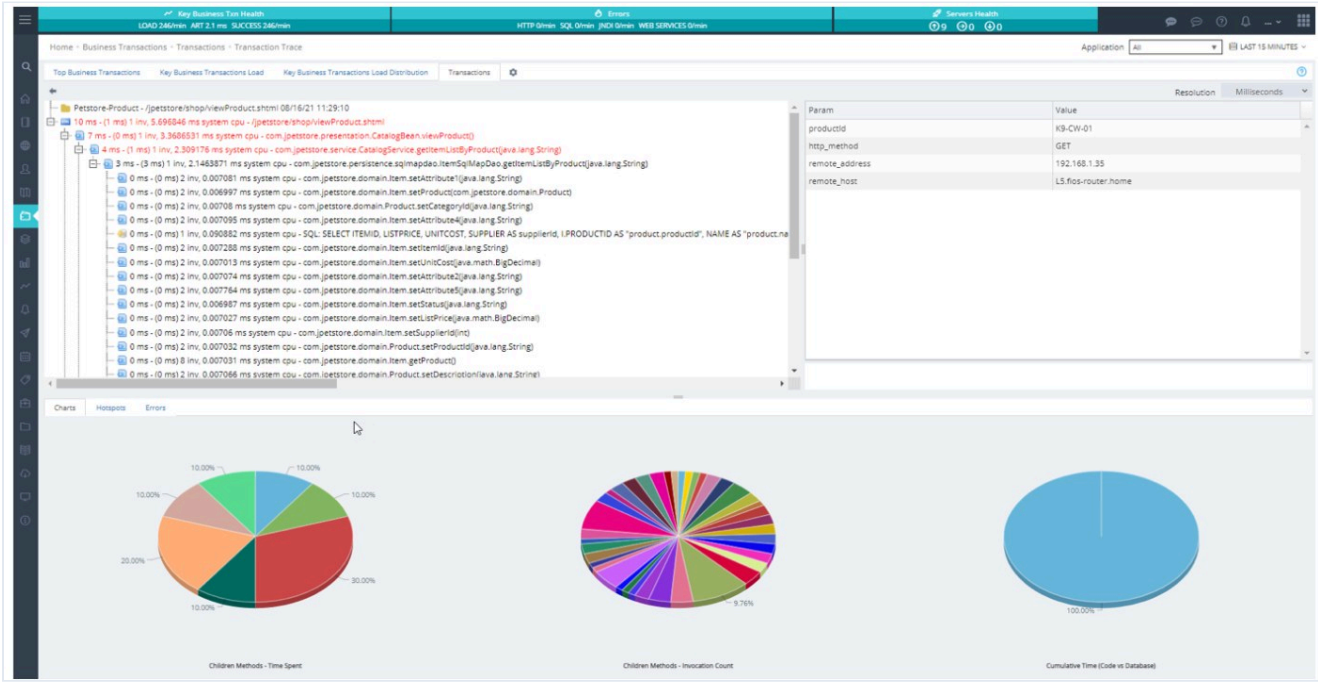


Success - If transaction was completed normally without an exception thrown from the method (applies only to Java transactions).

Speed - If transaction duration was less than the average * (User defined value in Business Transaction Configuration).

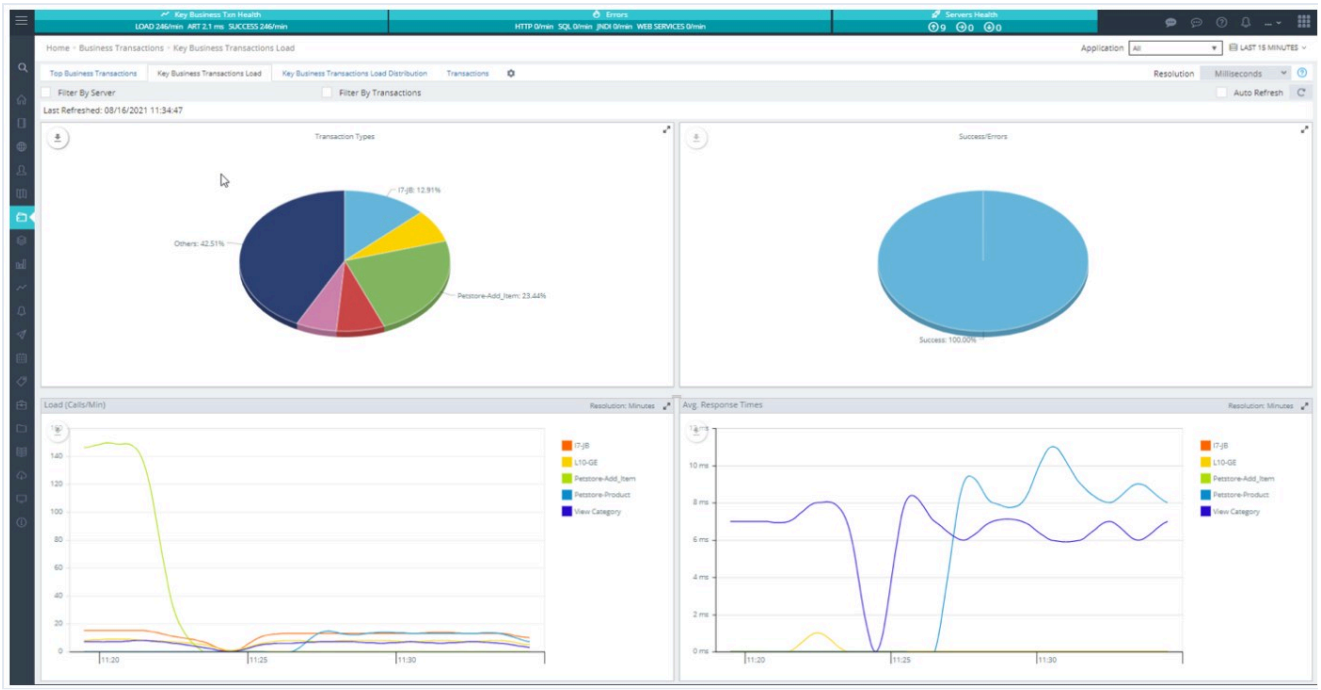
Meets SLA Max - Indicates transaction duration met the configured SLA Max value.

Meets SLA Avg - Indicates transaction duration met the configured SLA Average value.



Key Business Transactions Load View

Using this tab you can view the performance stats of key business transactions across the environment as well as on a per server basis. You must enable Business Transaction Profiling in order to gather data to be displayed on the dashboard.



Configuring Distributed Transaction Tracing

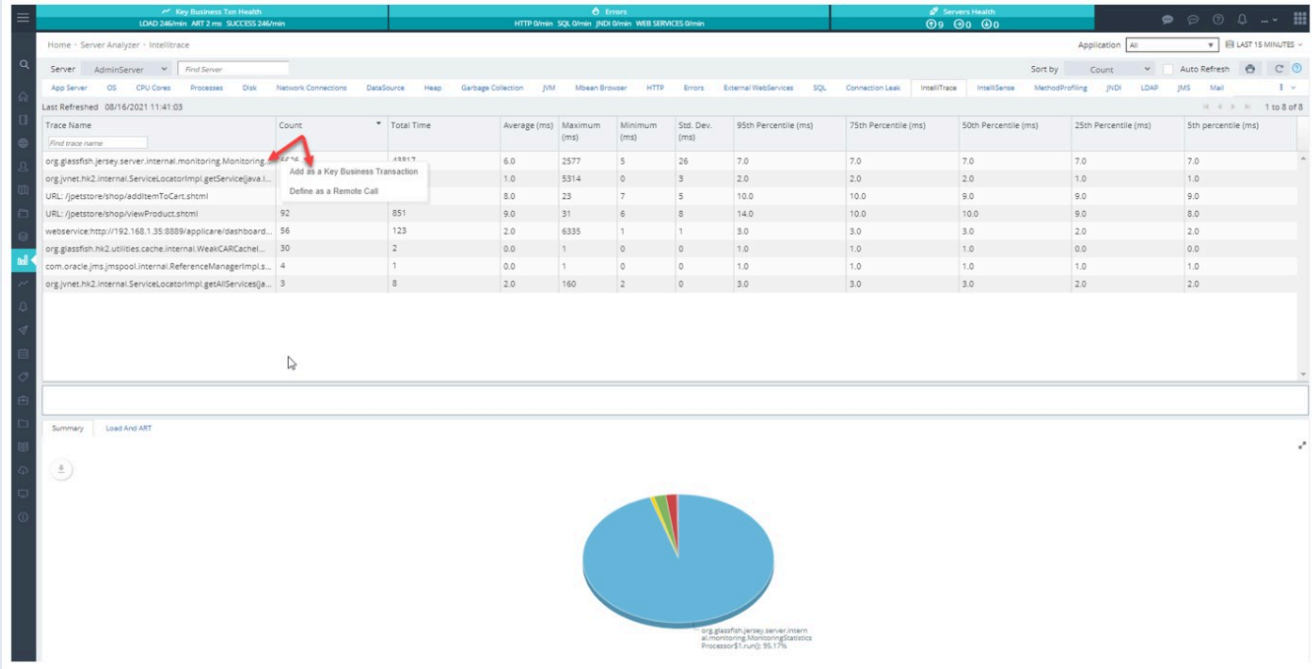
Applicare BTM can track transactions that span multiple servers (JVMs) and display the consolidated call graph consisting of executions on multiple servers in a single UI.

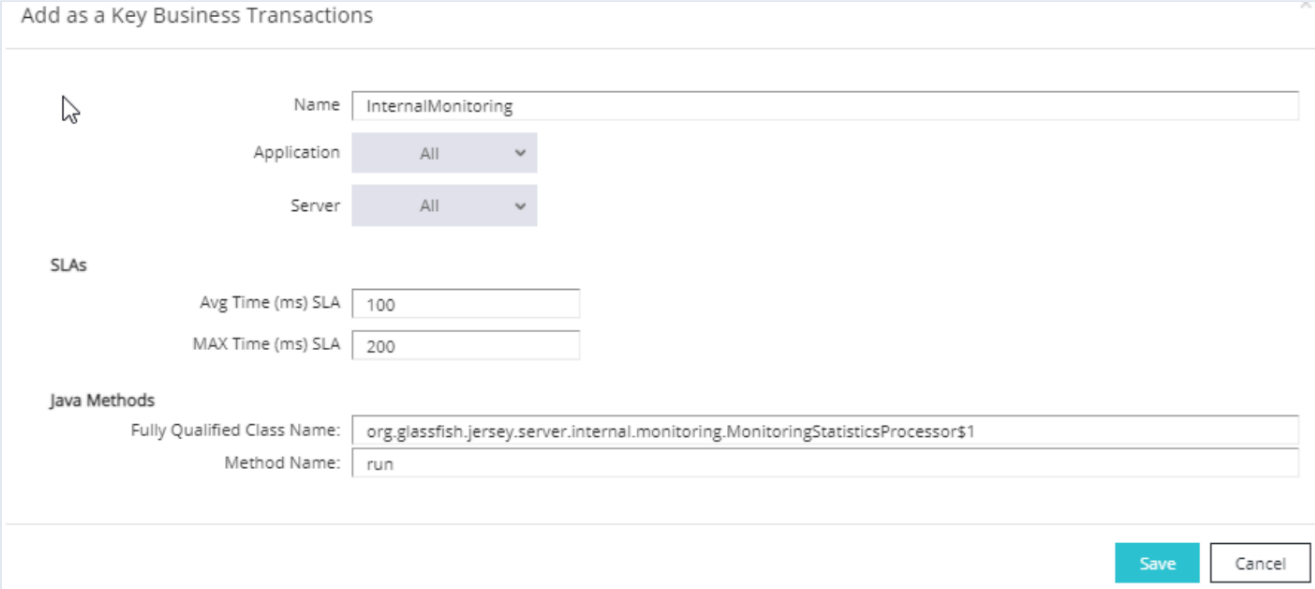
Note: All servers that participates in distributed transactions must have their system clocks synchronized (E.g. Using NTP) for remote transaction tracing to work properly.

There are two ways to enable transaction tracking between two servers.

First Method

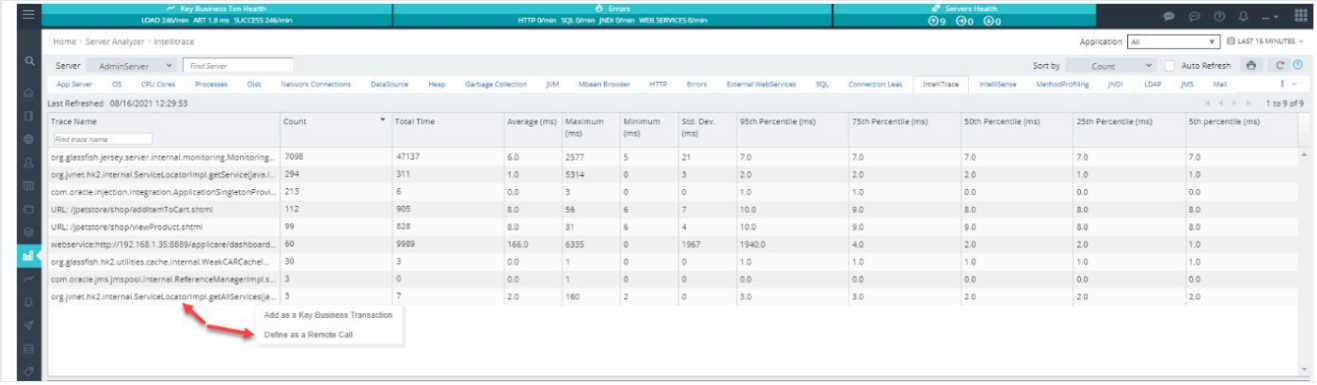
First you must define the remote service as a Java business transaction. You can right click your service method which accepts the remote parameters in IntelliTrace window and select "Define as a Business Transaction" option.

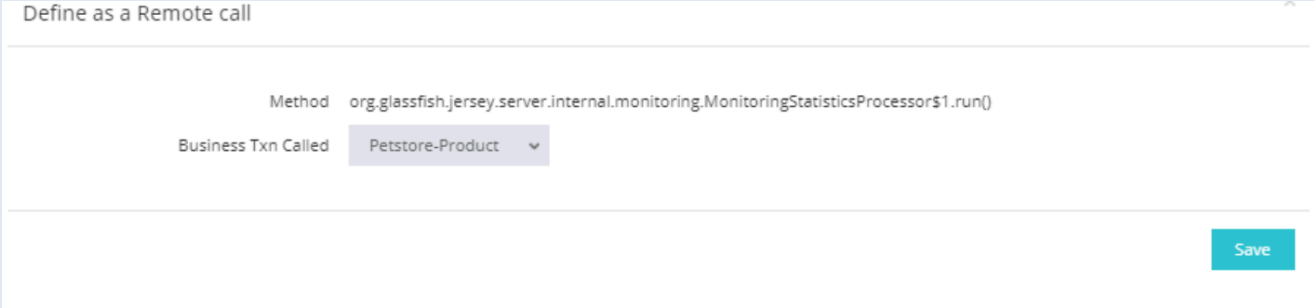




In the above example we are defining the org.glassfish.jersey.server.internal.monitoring.MonitoringStatisticsProcessor$1.run() method which is the service method that parameter as a Business Transaction named "**Petstore-Product**".

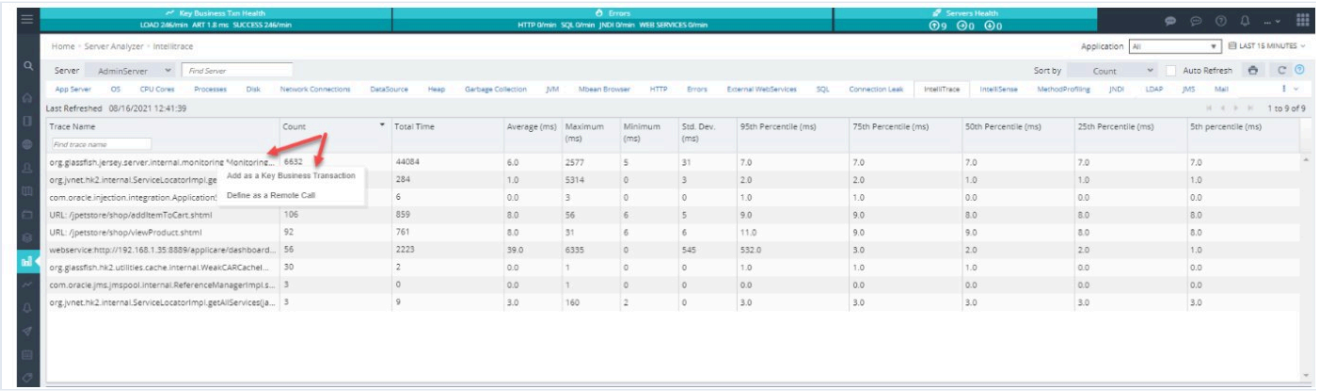
Next we define the method that calls the remote service as a remote method by right clicking the method name in IntelliTrace or in the BTM trace window and selecting "**Define as a Remote Call**" option. Be sure to choose the method that accepts the parameters which are sent to the remote service.

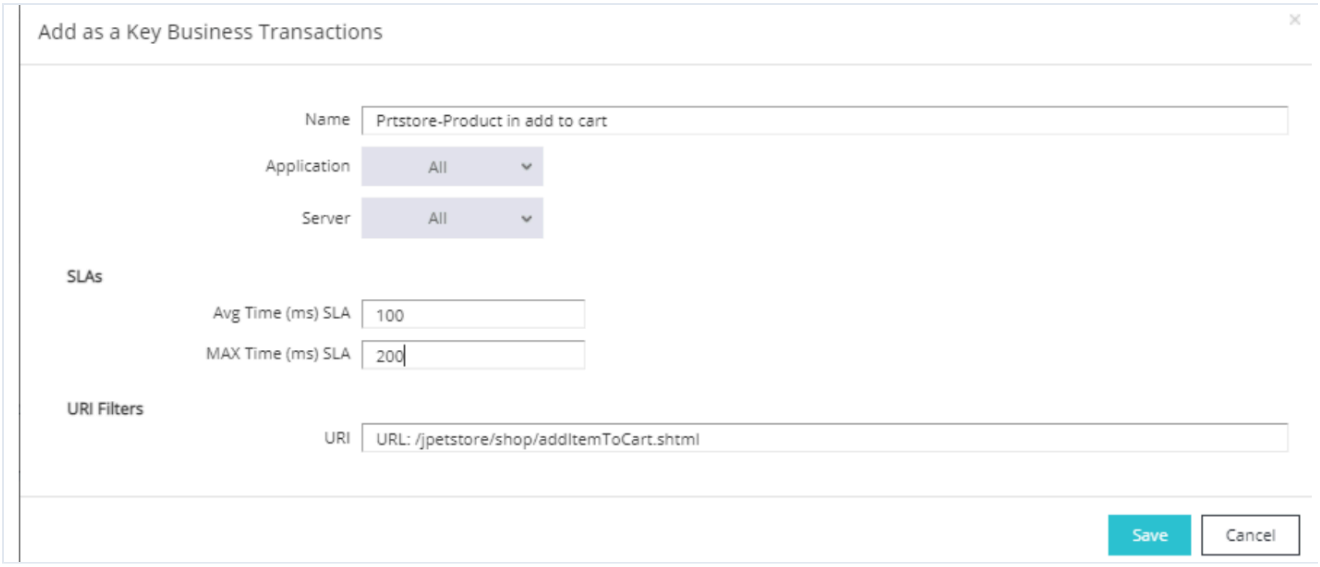




In the example above we define the method jpetstore/shop/addItemToCart.shtml as a remote method call and we select the previously defined **Pet store-Product in add to cart** service as the Business Transaction called by this method.

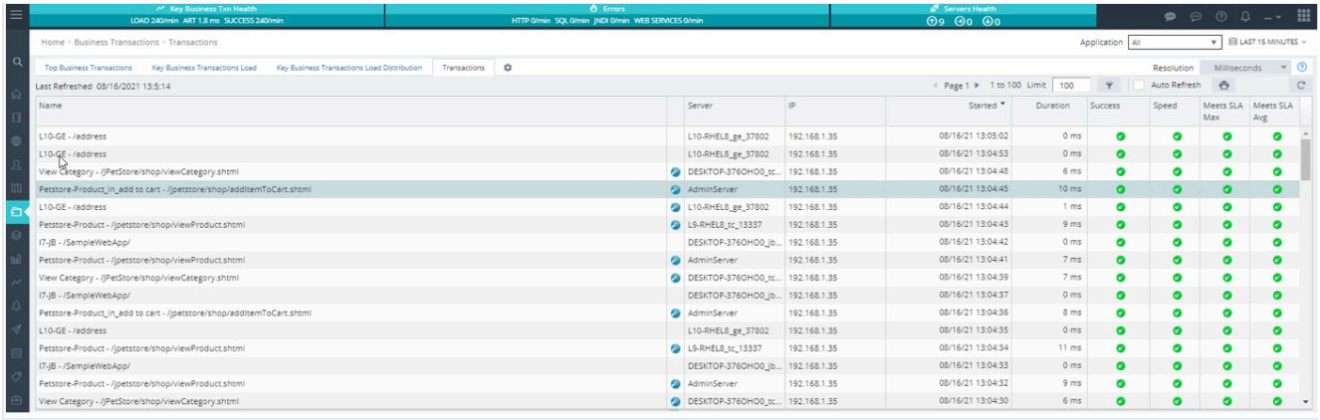
Next we will define the HTTP call which makes the remote method call as a business transaction too

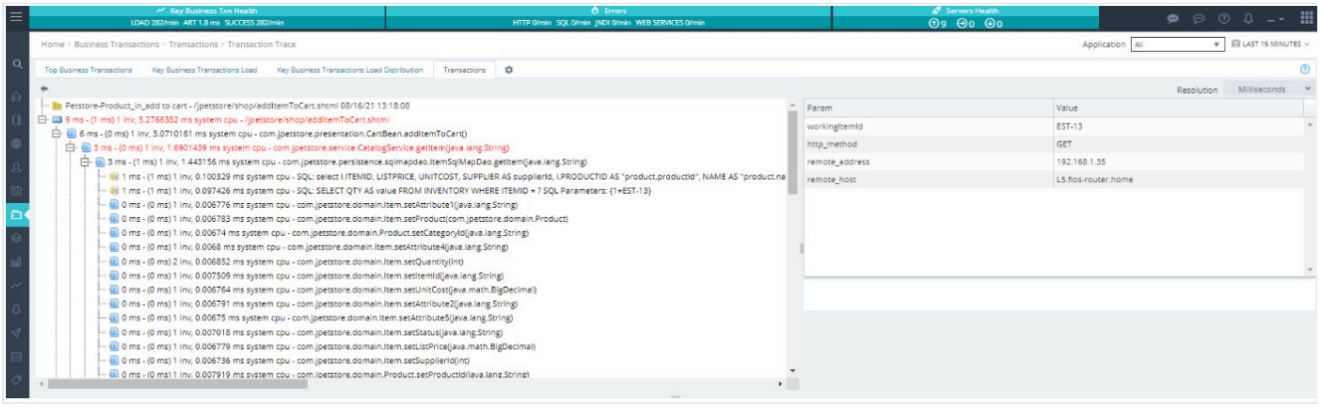




Now that both transactions and remote method calls are defined, you can monitor the transactions as they are invoked on the server and track their flow across different servers.

The following two images illustrate a **Validate CC** transaction invoked on the AdminServer. By double-clicking, you can view the call graph for the AdminServer and the call graph for the remote service on the Tomcat server, which was invoked as part of the transaction on the AdminServer.



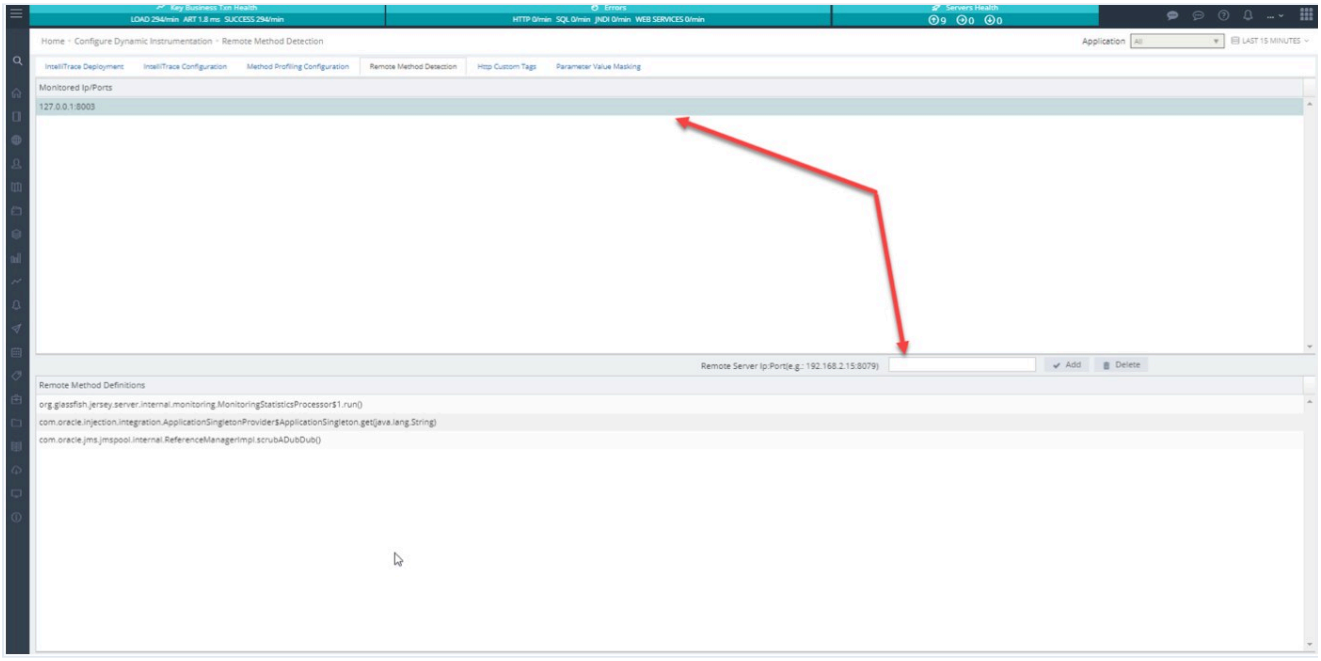


Second Method

Second approach is to use Network Connection Profiling to track transactions and as in the first method, you will have to define the calling and the called business transactions. You will not need to define the method that calls the remote transaction as a Remote Method.

For this approach to work, Network Connection Profiling must be enabled on all participating servers. You can do this by going to **Profiling -> Configure Dynamic Instrumentation** page and deploying Network Connection Profiling in Advanced mode.

Once you enable Network Connection Profiling, go to the Remote Method Definition tab on the same page and add the server IP and port where the remote transactions are running on.



Monitoring with Applicare