

Learning Series - How to record JDBC Transactions of Enterprise Dashboard and Virtualize the same using DevTest Broker ?

DevTest Broker provides an Java Agent using which you record JDBC transactions done from Enterprise Dashboard and convert it into a virtual service.

TESTED WITH: 10.4

Pre-Requisites:

DevTest Broker should be up and running. Check pfbroker.log and make sure there are no problems in starting this service. Refer to this blog if you face any troubles [What to check if Broker Startup Fails with weird error in pfbroker log file](#)

Setting Up Java Instrumentation Agent for ED & IAM

Add the line below to LISA_HOME/bin/EnterpriseDashboard.vmoptions or LISA_HOME\bin\EnterpriseDashboardService.vmoptions

```
-javaagent:C:\Progra~1\CA\DevTest\agent\InsightAgent.jar=url=tcp://localhost:2009,name=DRADIS
```

Here; url refers to the URL of DevTest Broker. Name can be any descriptive identifier. You would find its usage later in the article

It is very important to follow the same syntax as shown above. Using folder names with space or using double quotes may lead to startup failure of ED

Start ED from command prompt from LISA_HOME\bin, run the command **EnterpriseDashboard.exe** or start it from Windows Services

For IAM, add the following entry to file C:\Program Files\CA\DevTest\IdentityAccessManager\bin\standalone.bat

```
set "JAVA_OPTS=-Dprogram.name=%PROGNAME% -javaagent:"C:\Program Files\CA\DevTest\agent\InsightAgent.jar"=url=tcp://localhost:2009,name=IAMCL %JAVA_OPTS%"
```

In case of IAM, you can start IAM from windows services

NOTE: Trying to use javagent with Registry fails to start registry.

Checking If ED is Connected to Broker

As soon as ED starts, it connects to the broker. There are 5 different ways this can be confirmed.

Method 1 - Checking pfbroker log file

```
[DevTest AGENT:BC][INFO][7468][183][ActiveMQ Session Task][Jan 23, 2020 7:56:55 PM] Agent online: [ID: 69429598][NAME: IAMCL][IP: 169.254.92.143][MC: org.jboss.modules.Main][PID: 9132]
[DevTest AGENT:BC][INFO][7468][183][ActiveMQ Session Task][Jan 23, 2020 7:57:24 PM] Agent online: [ID: 2024515099][NAME: DRADIS][IP: 169.254.92.143][MC: com.install4j.runtime.launcher.WinLauncher][PID: 18332]
```

In this case, I have set javaagent property to both ED and IAM.

Whenever the process that was set with javaagent stops, you would see an entry similar to the one shown below in pfbroker.log file

```
Agent offline: [ID: 69429598][NAME: IAMCL][IP: 169.254.92.143][MC: org.jboss.modules.Main][PID: 9132]
```

Method 2 - From ATK (Agent Tool Kit Tool)

This is a simple UI tool to show agents (yes, any process that has the javaagent configured is called as agent)

Launch atk.exe from LISA_HOME\bin folder and you should be able to see the ONLINE agents

The screenshot displays the CA Application Trace Kit interface. At the top, the title bar reads "CA Application Trace Kit (v.10.4.0.355 - i.5.3.1.2) [Dev:natsa03@NATSA03-7520.ca.com][16319157][Broker: tcp://localhost:2009?daemon=true]". Below the title bar is a "Broker URL" field containing "tcp://localhost:2009". A navigation bar includes icons for Overview, Classes, Threads, VS Play, Scripting, Terminal, Explorer, Log, and Extensions. The left sidebar contains a "Pathfinder" section with links for Live Paths, Saved Paths, Database, and Tickets. Below it is the "Manage Agents" section, which lists several agents: DRADIS, DRADISSERVICE (12968), IAMCL (4340), IAMCLASSES, and REGISTRY. Red lines are drawn around the DRADISSERVICE and IAMCL entries. At the bottom of the sidebar is a "Manage Consoles" dropdown. The main content area is titled "Overview" and shows general information for the selected agent: DRADIS (ID: 2024515099) (PID: -1). The information includes Agent Version (10.4.0.355), Agent Directory (C:\Program Files\CA\DevTest\lib\core), OS (Windows 10 10.0 (amd64)), Machine (NATSA03-7520.ca.com (169.254.92.143)), VM Name (7468@NATSA03-7520 (Java HotSpot(TM) 64-Bit Server VM)), VM Version (1.8.0_221 (25.221-b11)), Working Directory (c:\Program Files\CA\DevTest), Cmd Line (C:\Program Files\CA\DevTest\bin\BrokerService.exe -XX:HeapDumpPath="C:\Progra), and Flags (UNCONFIGURED). Below this information are two empty graphs for "CPU Usage" and "Memory Usage" with x-axis labels at 8:13:30 and 8:14:00. At the bottom of the main area are sections for "Details" and "Settings".

Method 3 - From Portal (Yes)

Just login to Portal and go to Settings -> Agents as shown below. It shows active as well as inactive agents.

The screenshot shows the DevTest Portal interface. On the left is a navigation menu with options like Home, Create, Manage, Monitor, Application Insight, Reporting, Settings, Access Control, Roles, Resource Groups, and Resources. The main area is titled 'Agents' and shows a list of agents under a 'DEFAULT' group. The agents listed are DRADIS, DRADISSERVICE, IAMCL, IAMCLASSES, and REGISTRY, all with version 10.4.0.355. On the right, there is a 'JVM Performance' section with a 'CPU Usage (%)' graph showing a value of 30%.

Method 4 - Using PF Invoke API

<http://localhost:1505/lisa-pathfinder-invoke/api/v1/agents>

NOTE: This is what the portal uses internally

Method 5 - Using Workstation

Created a VS by recording. Choose java as protocol.

Virtual Service Image Recorder

Please provide us with some basic information about what is to be recorded and select the appropriate protocol(s) involved. Some transport protocols do not allow for a data protocol.

Basics Notes

Write image to: Files\CA\DevTest\Projects\SE_Issues_Jan_2020\DE439223\3m.vsi

Create Merge into

Import traffic:

Transport protocol: Java

De-identify (transport layer)

Treat all transactions as stateless

Allow duplicate specific transactions

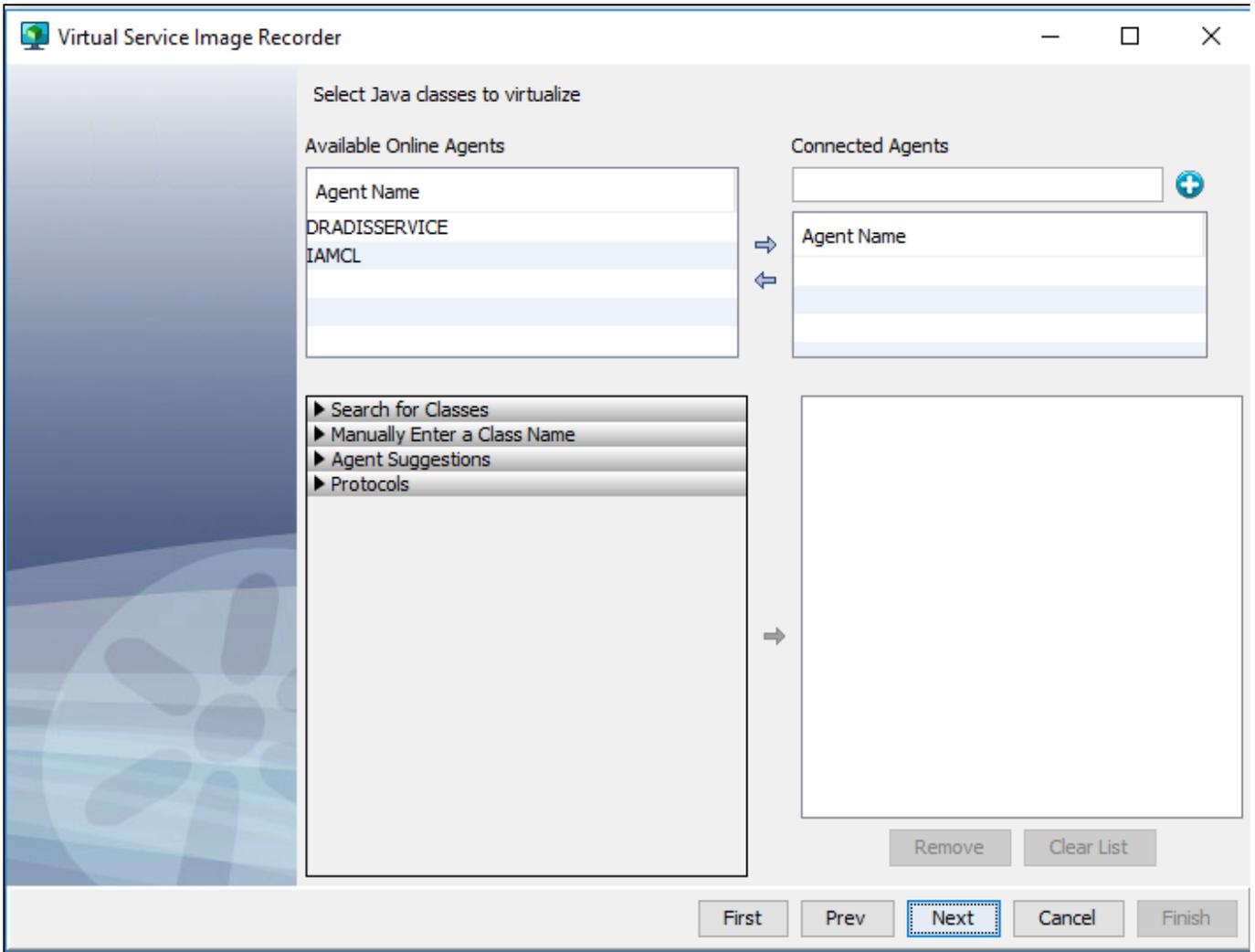
Default navigation: WIDE Last: LOOSE

Export to:

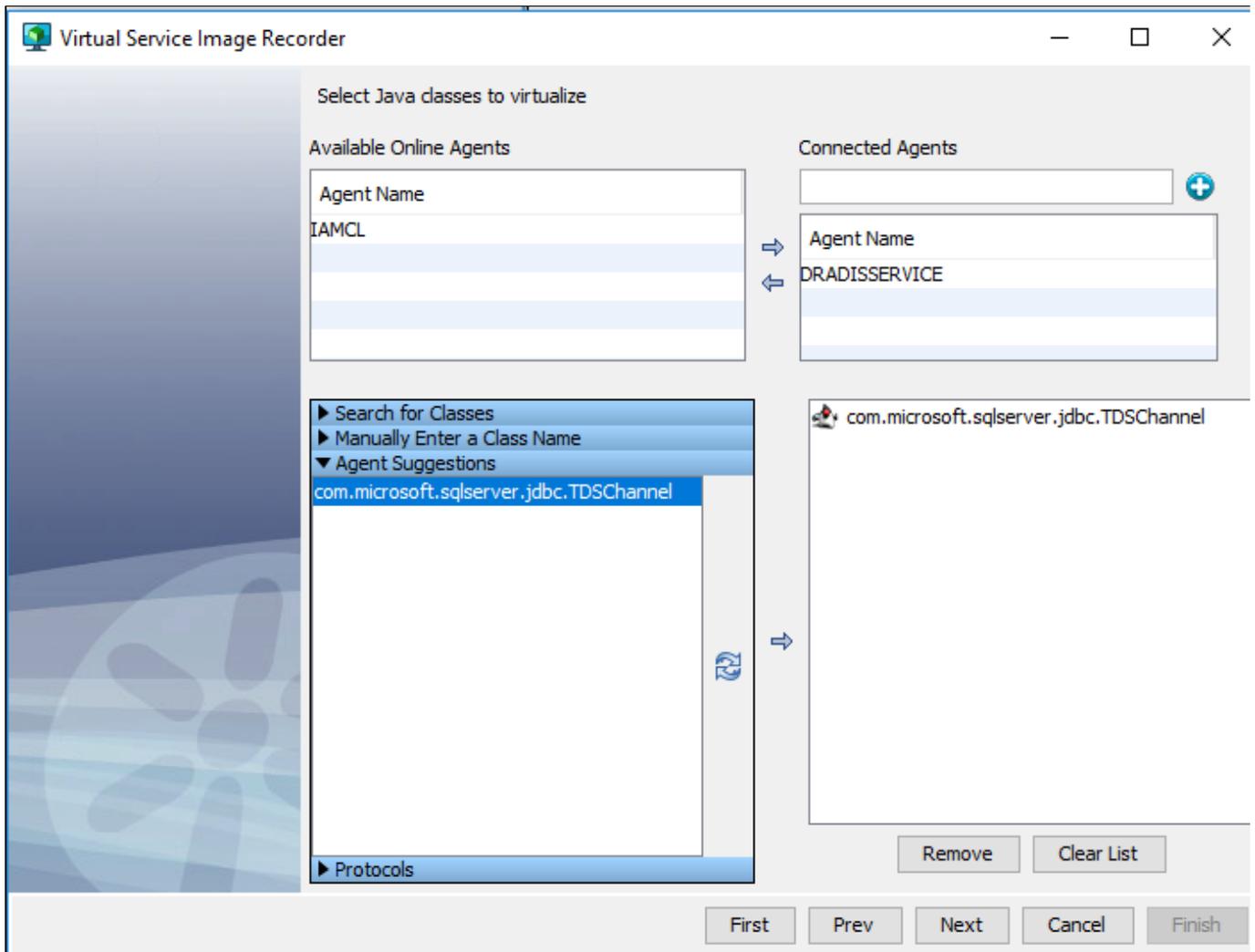
Model file: Files\CA\DevTest\Projects\SE_Issues_Jan_2020\DE439223\3m.vsm

VS Model style: More flexible More efficient

You would see online agents appear in next screen



Choose agent, move it to right and choose class from 'Agent Suggestions'



In the next screen you would see JDBC traffic being captured and you can move further and finish the VS

Virtual Service Image Recorder

Recording has begun. Exercise the target service while we capture the service image. Click the Next button when you have recorded what you want (recording will be stopped automatically).

TDSChannel.getReader

```

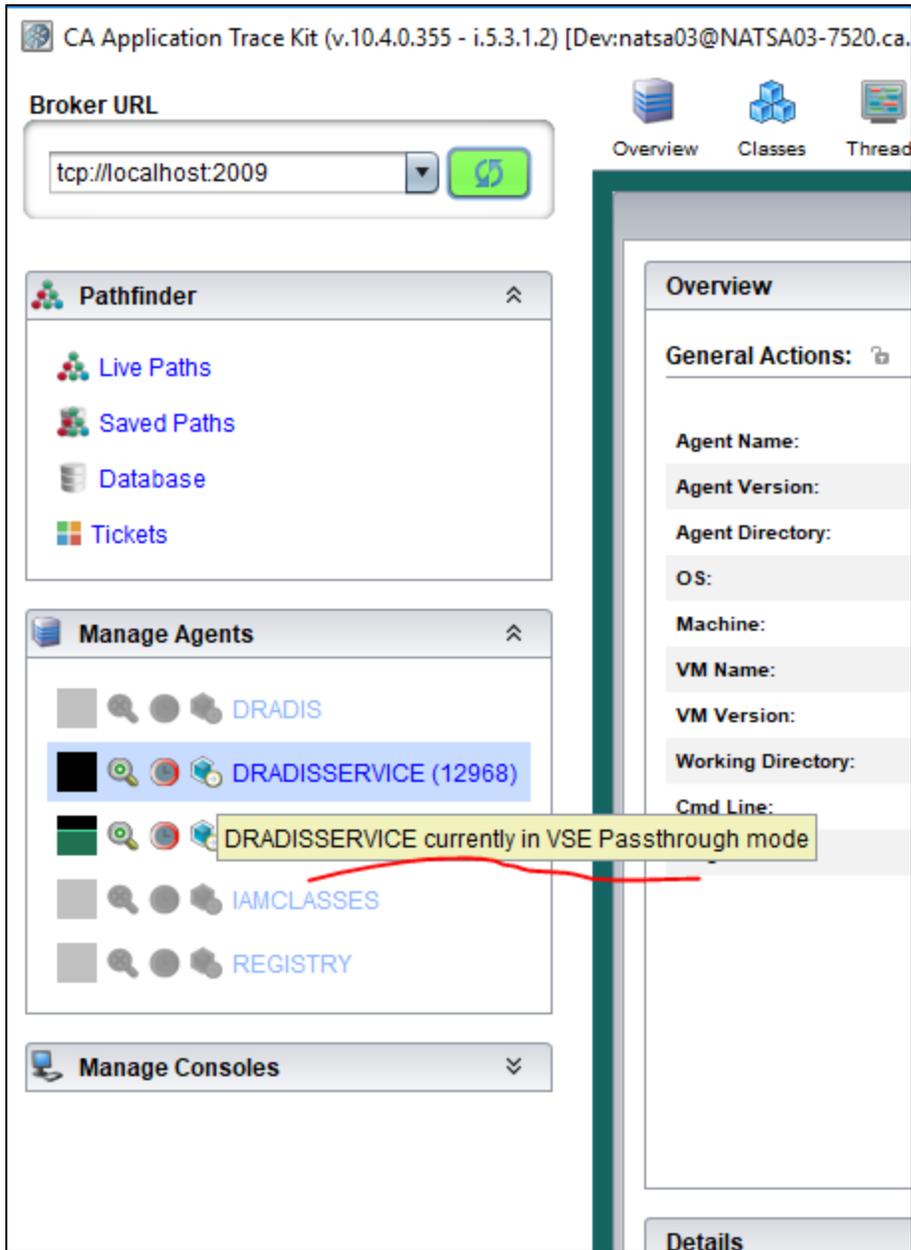
arg0_COM_MICROSOFT_SQLSERVER_JDBC_SQLSERVERSTATEMENT_-STMTEXECMD_TRACEID: __NULL
arg0_COM_MICROSOFT_SQLSERVER_JDBC_SQLSERVERSTATEMENT_-STMTEXECMD_TDSWRITER: __NU
arg0_COM_MICROSOFT_SQLSERVER_JDBC_SQLSERVERSTATEMENT_-STMTEXECMD_TDSREADER: __NU
arg0_COM_MICROSOFT_SQLSERVER_JDBC_SQLSERVERSTATEMENT_-STMTEXECMD_INTERRUPTREASON:
arg0_COM_MICROSOFT_SQLSERVER_JDBC_SQLSERVERSTATEMENT_-STMTEXECMD_LOGCONTEXT: SQL
arg0_COM_MICROSOFT_SQLSERVER_JDBC_SQLSERVERSTATEMENT_-STMTEXECMD_TIMEOUTTIMER_TI
arg0_COM_MICROSOFT_SQLSERVER_JDBC_SQLSERVERSTATEMENT_-STMTEXECMD_TIMEOUTTIMER_TI
arg0_COM_MICROSOFT_SQLSERVER_JDBC_SQLSERVERSTATEMENT_-STMTEXECMD_TIMEOUTTIMER_CC
arg0_COM_MICROSOFT_SQLSERVER_JDBC_SQLSERVERSTATEMENT_-STMTEXECMD_TIMEOUTTIMER_C/
arg0_COM_MICROSOFT_SQLSERVER_JDBC_SQLSERVERSTATEMENT_-STMTEXECMD_INTERRUPTLOCK:
arg0_COM_MICROSOFT_SQLSERVER_JDBC_SQLSERVERSTATEMENT_-STMTEXECMD_INTERRUPTSENABL
arg0_COM_MICROSOFT_SQLSERVER_JDBC_SQLSERVERSTATEMENT_-STMTEXECMD_WASINTERRUPTED
arg0_COM_MICROSOFT_SQLSERVER_JDBC_SQLSERVERSTATEMENT_-STMTEXECMD_REQUESTCOMPLET
arg0_COM_MICROSOFT_SQLSERVER_JDBC_SQLSERVERSTATEMENT_-STMTEXECMD_ATTENTIONPENDIN
arg0_COM_MICROSOFT_SQLSERVER_JDBC_SQLSERVERSTATEMENT_-STMTEXECMD_PROCESSEDRESPO
arg0_COM_MICROSOFT_SQLSERVER_JDBC_SQLSERVERSTATEMENT_-STMTEXECMD_READINGRESPONSI
arg0_COM_MICROSOFT_SQLSERVER_JDBC_SQLSERVERSTATEMENT_-STMTEXECMD_INTERRUPTCHECKE
arg0_COM_MICROSOFT_SQLSERVER_JDBC_SQLSERVERSTATEMENT_-STMTEXECMD_STMT_PROCEDURE
arg0_COM_MICROSOFT_SQLSERVER_JDBC_SQLSERVERSTATEMENT_-STMTEXECMD_STMT_TDSREADER
arg0_COM_MICROSOFT_SQLSERVER_JDBC_SQLSERVERSTATEMENT_-STMTEXECMD_STMT_INOUTPARAI
arg0_COM_MICROSOFT_SQLSERVER_JDBC_SQLSERVERSTATEMENT_-STMTEXECMD_STMT_LASTSTMTEX
arg0_COM_MICROSOFT_SQLSERVER_JDBC_SQLSERVERSTATEMENT_-STMTEXECMD_STMT_RESULTSET:
arg0_COM_MICROSOFT_SQLSERVER_JDBC_SQLSERVERSTATEMENT_-STMTEXECMD_STMT_CURSORNAN
arg0_COM_MICROSOFT_SQLSERVER_JDBC_SQLSERVERSTATEMENT_-STMTEXECMD_STMT_AUTOGENER
    
```

Showing the 100 most recently added transactions.

Total sessions: 3

Total transactions: 45

Now that you have just done the recording and the VS not deployed, you would be able to see that the agent is in passthrough mode in ATK



As soon as you deploy the VS that was recorded, you would see that the status of agent changes to PLAYBACK mode

IN case of an active customer defect (DE439223), the status doesnt change to PLAY BACK and remains in PASSTHROUGH mode after deploying the VS

CA Application Trace Kit (v.10.4.0.355 - i.5.3.1.2) [Dev:natsa03@NATSA]

Broker URL: tcp://localhost:2009

Overview Classe

Pathfinder

- Live Paths
- Saved Paths
- Database
- Tickets

Manage Agents

- DRADIS
- DRADISSERVICE (12968)**
- DRADISSERVICE currently in VSE Playback mode
- IAMCLASSES
- REGISTRY

Manage Consoles

Overview

General Act

Agent Name

Agent Versi

Agent Direct

OS:

Machine:

VM Name:

VM Version:

Working Dir

Cmd Line: