

Enhanced Full Transaction Trace Detail Tracer

By Keane Yu, keane.yu@ca.com

Contents

What is it	2
Why.....	2
Features	2
Compatibility.....	3
Limitation	3
Ground rules	3
Installation	3
Configuration Example.....	4
FAQ.....	4
List of Changes	5

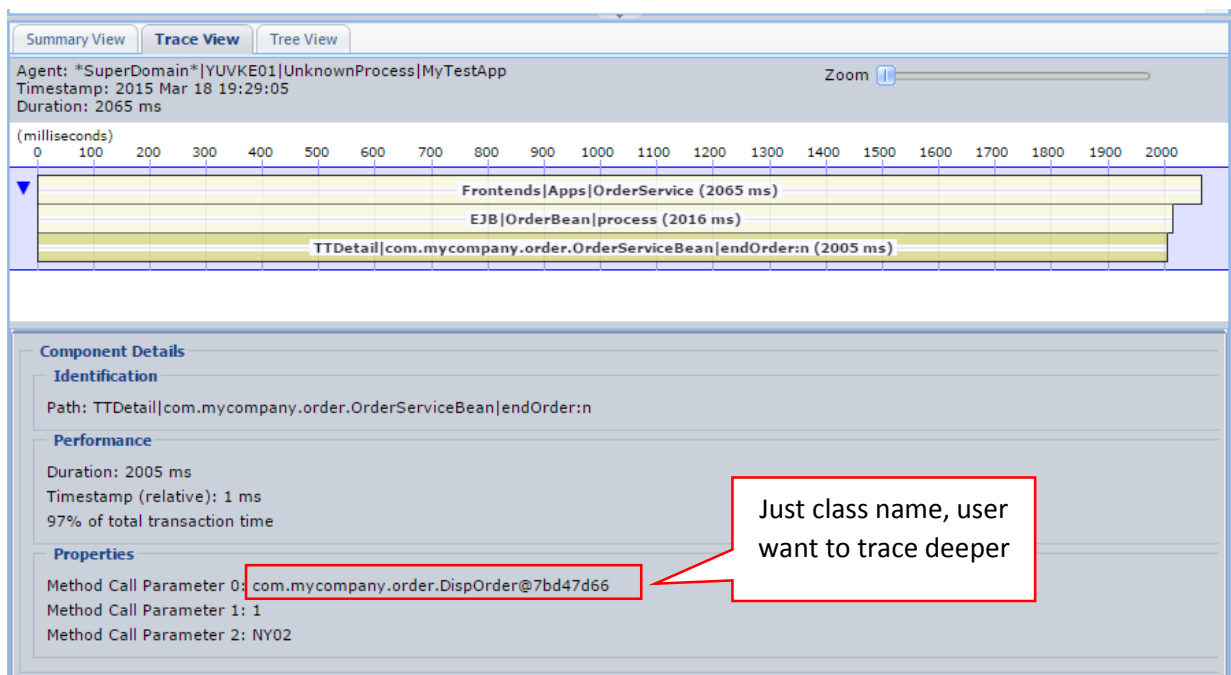
What is it

A tracer to extract deeper value by calling specified methods on method parameters or return value, and then attached those values into transaction trace component.

This tracer enhanced the great fieldpack "Full Transaction Trace Detail Tracer (new_mode)", which originally developed by Omar Ocampo.

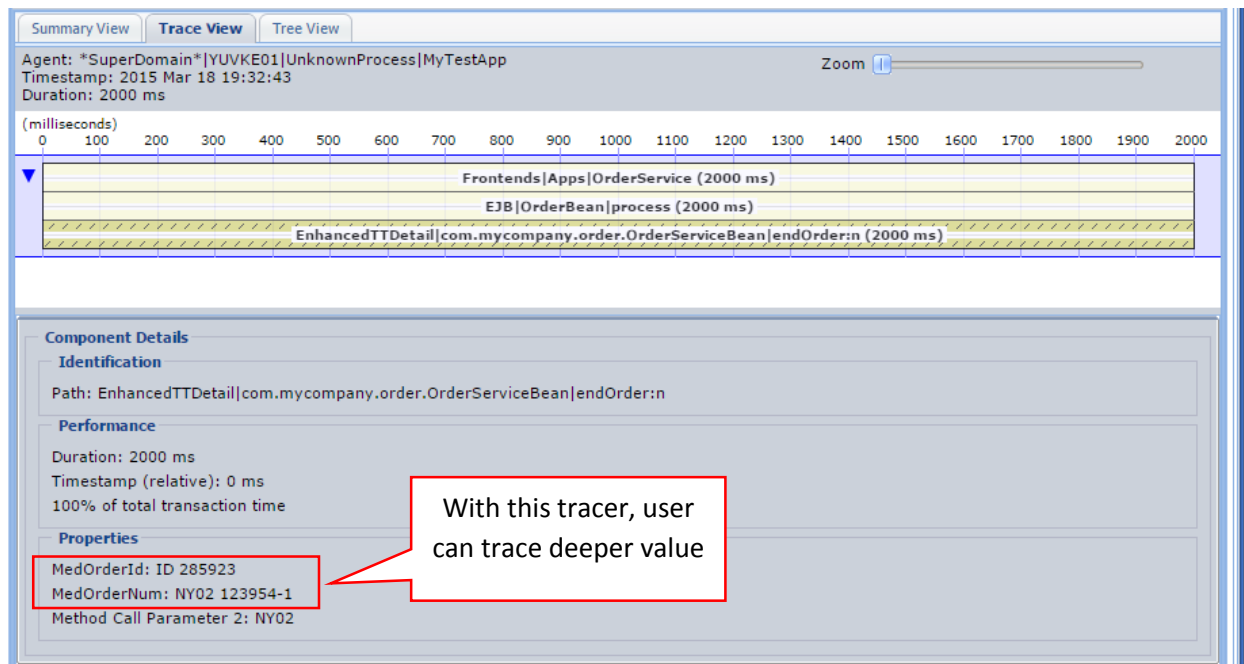
Why

APM user want to get deeper and useful value from method parameter or return value for diagnosis purpose. However, sometimes parameter or return value is complex object instead of simple object (String, int, Boolean, etc.), and original FullTTDetail tracer will only show class name of complex object.



Features

- All features included in original Full Transaction Trace Detail Tracer
- Control which parameters to show
- Control whether to show return value or not
- If parameter is specified to show, user can specify the display name and method chain for this parameter. For each parameter, user can specify multiple method chains to get multiple values.
- If return value is specified to show, user can specify the display name and method chain for this parameter. User can specify multiple method chains to get multiple values.



Compatibility

APM Java Agent 9.1.x

APM Java Agent 9.5.x

APM Java Agent 9.6.x

APM Java Agent 9.7.x

Limitation

Version 2015-03-18 doesn't support deep method call with parameter

Ground rules

This is a field pack from CA APM User Community, and thus currently not an officially supported product from CA Technologies. Consult your field rep/SE for assistance, rather than contact CA Technical Support.

It is recommended that you use this tool at your own RISK in production. This module has not undergone CA Technology official QA cycles.

Installation

1. Place file "EnhancedFullITTDDetail.jar" into folder "<agent_home>/core/ext"
2. Place file "EnhancedFullITTDDetail.pbd" into folder "<agent_home>/core/config/hotdeploy"
3. Restart agent JVM.

Configuration Example

Refer file "EnhancedFullTTDetail.pbd".

FAQ

- **How to control which parameters to show?**

If you want to analysis a parameter, explicitly specify its index (n-1), and analysis mode as your tracer parameter in PBD file.

Supported analysis modes:

"Normal" --- Tracer will call method "toString" of this parameter object. Usually use this mode when target is simple object (e.g. String and int).

"Deep" --- Tracer will use specified method chain to get final target object. Usually use this mode when parameter object is complex object and expected value is deeply hidden behind the parameter object.

"Disable" --- Tracer will not analysis this parameter object. Any parameter without explicit analysis mode will use "Disable" mode.

In following example, the tracer will use "Deep" mode to analysis the fifth parameter

SetTracerParameter: MyEnhancedFullDetailTTDifferentMethods 4 Deep

- **How to control whether to show return value or not?**

If you want to analysis return value, explicitly specify its analysis mode as your tracer parameter in PBD file.

Supported analysis modes are similar with above parameter analysis.

In following example, the tracer will use "Normal" mode to analysis the return value

SetTracerParameter: MyEnhancedFullDetailTTDifferentMethods ReturnTrace Normal

- **If parameter is specified to show, how to specify the display name and method chain for this parameter?**

For example, targeted method is

```
public EndOrderResult endOrder(  
    DispOrder dispOrder,  
    Long prevOrderId,  
    String ShopCode,  
    Boolean remarkFlag,  
    Boolean PersistenceEnabled,  
    Boolean legacyPersistenceEnabled)
```

{

You want to get Order ID from the first parameter “dispOrder” object, and the method chain is “dispOrder.getInterimOrder().getFinalOrder().getId()”, the setting should be:

SetTracerParameter: MyEnhancedFullDetailTTDDifferentMethods 0_0_Name MedOrderId

SetTracerParameter: MyEnhancedFullDetailTTDDifferentMethods 0_0_Exp getInterimOrder.getFinalOrder.getId

If you also want to get Order Number from the same parameter, the setting should be:

SetTracerParameter: MyEnhancedFullDetailTTDDifferentMethods 0 Deep

SetTracerParameter: MyEnhancedFullDetailTTDDifferentMethods 0_Count 2

SetTracerParameter: MyEnhancedFullDetailTTDDifferentMethods 0_0_Name MedOrderId

SetTracerParameter: MyEnhancedFullDetailTTDDifferentMethods 0_0_Exp getInterimOrder.getFinalOrder.getId

SetTracerParameter: MyEnhancedFullDetailTTDDifferentMethods 0_1_Name MedOrderNum

SetTracerParameter: MyEnhancedFullDetailTTDDifferentMethods 0_1_Exp getInterimOrder.getFinalOrder.getOrderNum

- **If return value is specified to show, how to specify the display name and method chain for this parameter?**

Similar to method parameter, setting should be:

SetTracerParameter: MyEnhancedFullDetailTTDDifferentMethods ReturnTrace Normal

SetTracerParameter: MyEnhancedFullDetailTTDDifferentMethods ReturnTraceName MyReturnValueName

SetTracerParameter: MyEnhancedFullDetailTTDDifferentMethods ReturnTraceExp getSomething.getAgain

- **How to set ParamCount in PBD?**

This parameter is mandatory, its value should be the number of parameter in your targeted method. In above method “endOrder”, there’s 6 parameters, then setting should be:

SetTracerParameter: MyEnhancedFullDetailTTDDifferentMethods ParamCount 6

List of Changes

2015-03-18:

- First try