



Extending Eclipse Test and Performance Tools Platform (TPTP)

Eugene Chan, IBM Tivoli Software

Agenda

- Overview of TPTP
 - ▣ TPTP Profiling tools
 - ▣ TPTP Monitoring
 - ▣ Extending TPTP
 - ▣ Challenges
 - ▣ What's new?





Overview - Eclipse TPTP

- Eclipse top level project
 - 2002 : Eclipse tools subproject - Hyades
 - 2004 : Eclipse top-level project - TPTP
- Mission:
 - To build a generic, extensible, standards-based platform for test and performance tracing tools.
- Goals:
 - Platform of choice for test, performance, and monitoring tools.
 - Exemplary tooling.
 - Enable value-added third-party tooling through extensibility and high-quality APIs.



Overview - TPTP project structure

- Comprised of four projects
 - Platform
 - Test
 - Trace
 - Monitoring

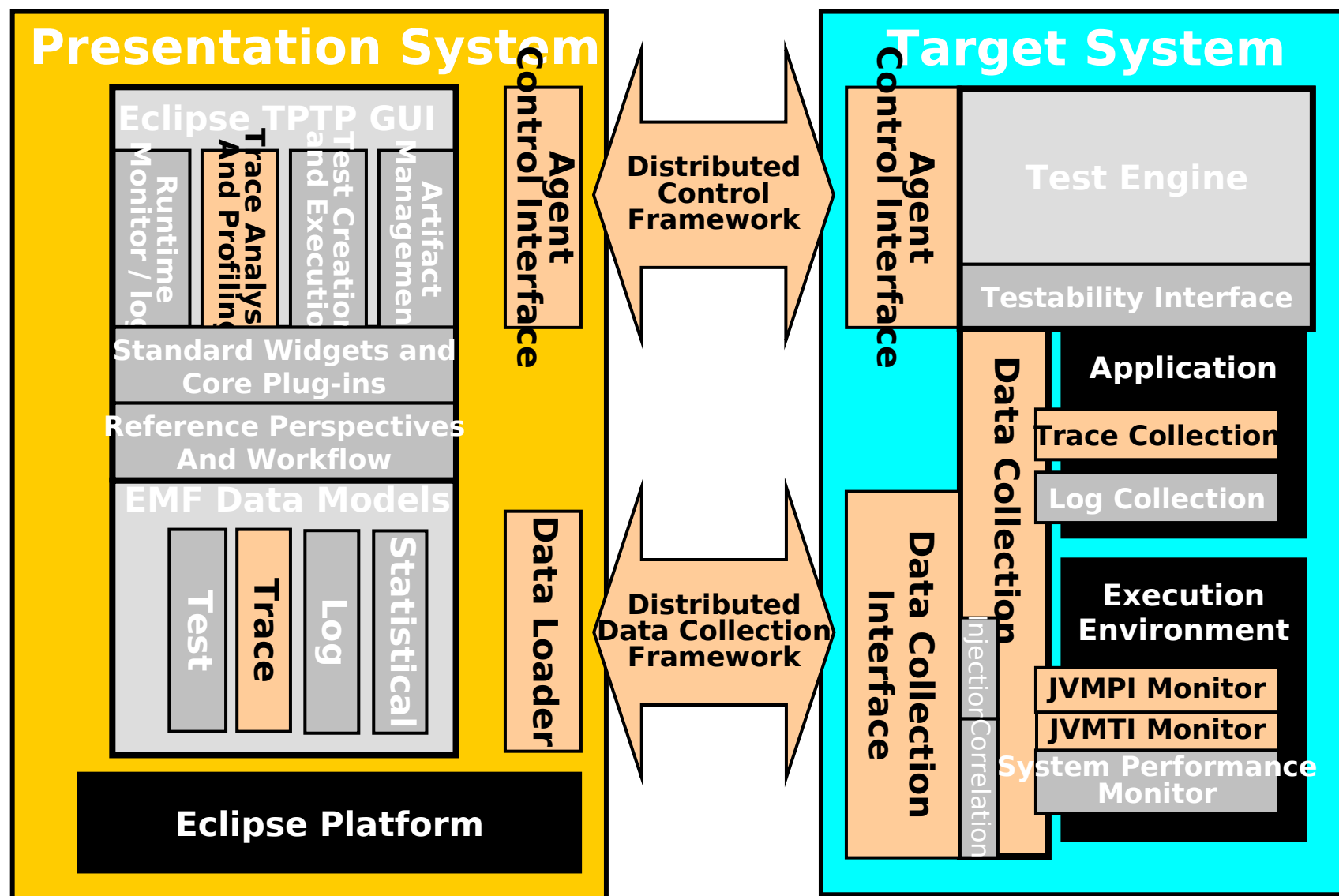
- Developed by a number of strategic contributors
 - Active contributors: Intel, IBM, OC Systems
 - Inactive contributors: Scapa Technologies, Computer Associates, Compuware, FOKUS

- For more information visit <http://www.eclipse.org/tptp/>

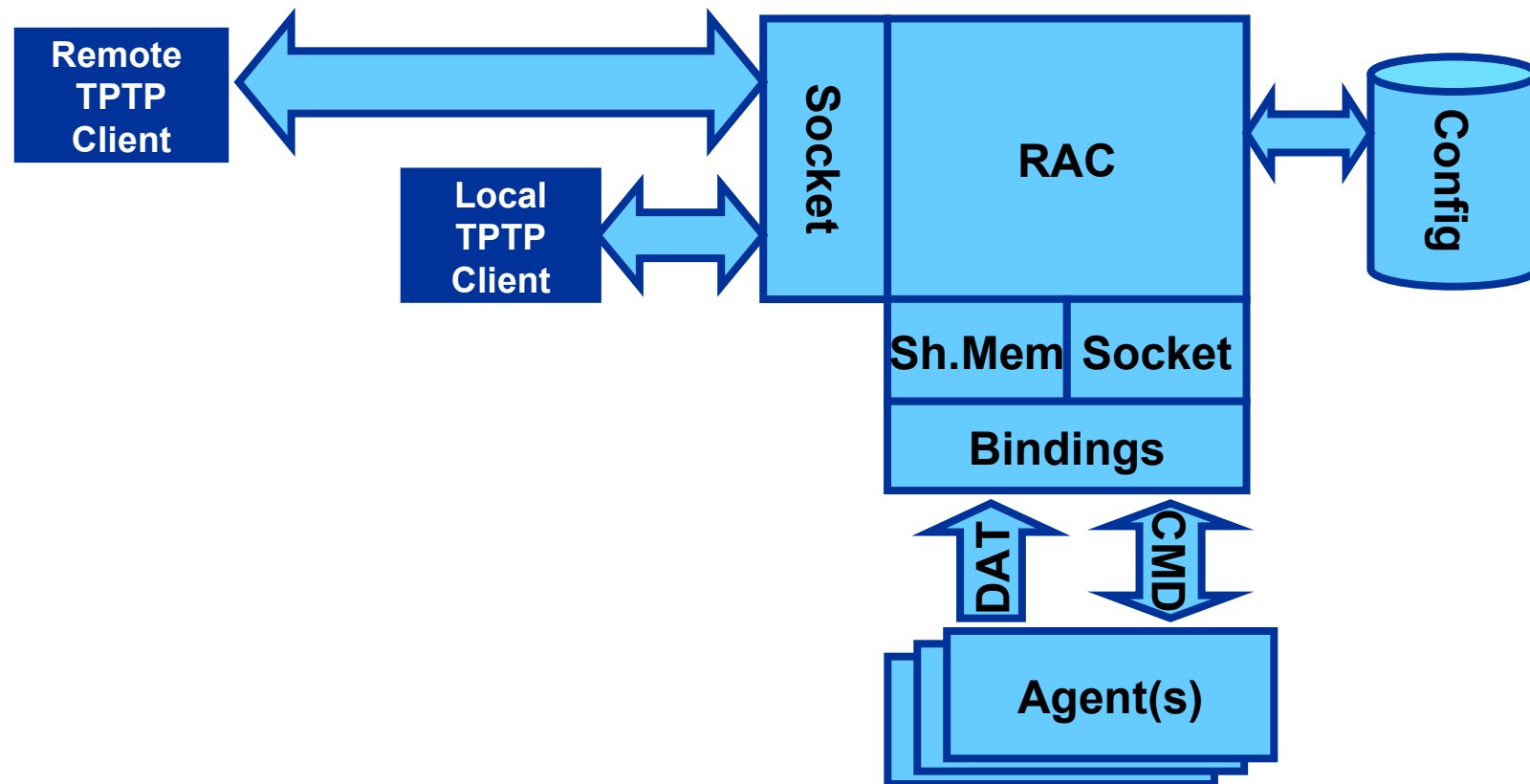
Overview - TPTP framework

- **TPTP offers a common, extendable framework** as well as reference implementations for the following functions:
 - Testing
 - Profiling
 - Static code analysis
 - Static and dynamic Bytecode Instrumentation (BCI)
 - Application monitoring and log analysis

TPTP Architecture



Agent Controller Architecture



Agenda

- ✓ Overview of TPTP
- **TPTP Profiling tools**
- ▣ TPTP Monitoring
- ▣ Extending TPTP
- ▣ Challenges
- ▣ What's new?

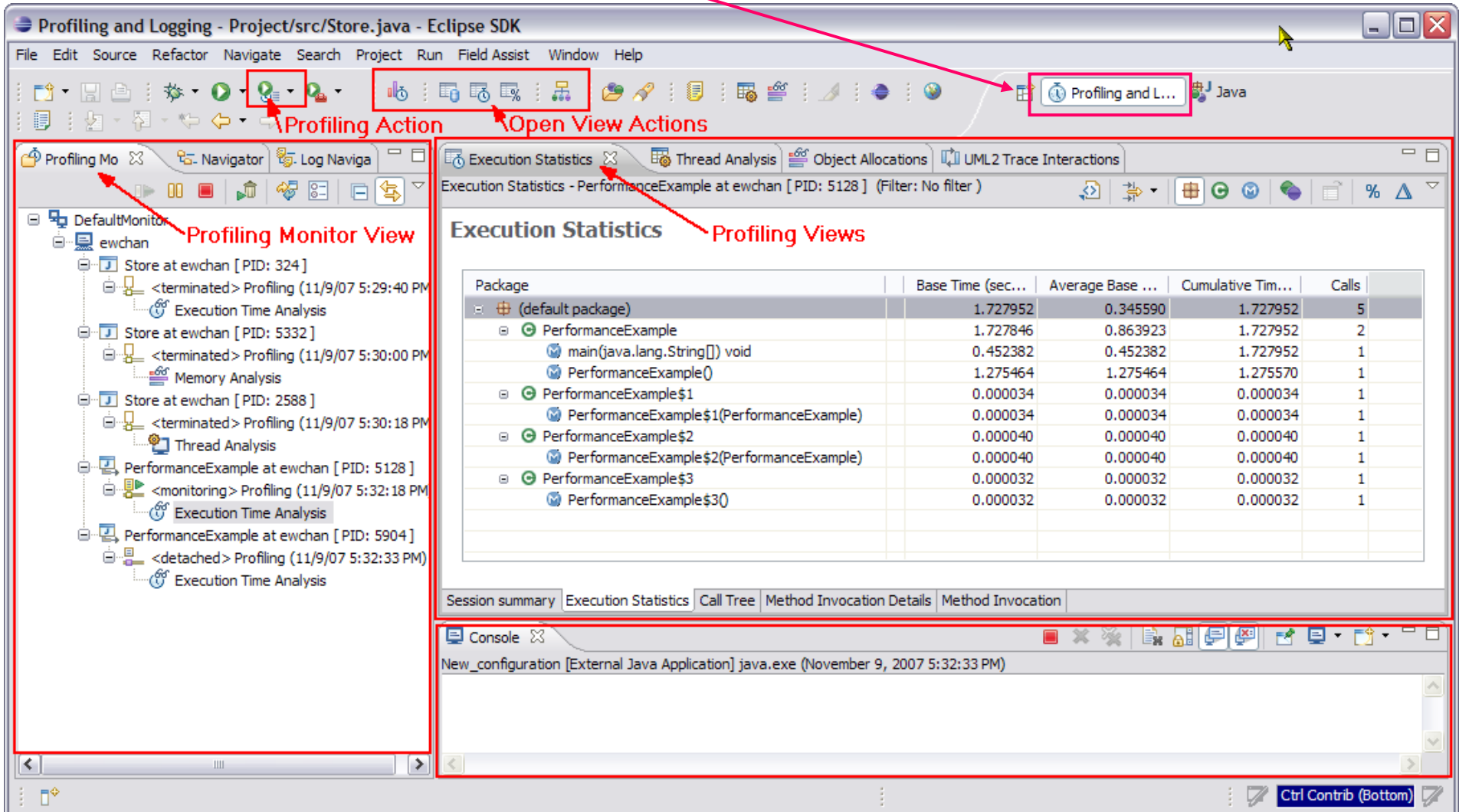


Overview - TPTP Profile tools

- Common framework for profiling simple Java applications or complex Web applications running on multiple platforms, on different hosts.
 - Common perspectives and views for interacting with target systems and resources.
 - User actions to interact with the profiled application.
 - Start, stop, attach, detach from the profiled application.
 - Views framework to analyze the profiled application.
 - Views can be extended and customized using extension points.
 - Standard EMF data model, query framework and assets repository.
 - Common data collection and execution framework on local and remote targets.
- Reference implementation of the Profiling framework:
 - Java Profiling tools based on JVMPI and JVMTI.
 - Actions to interact with the profile application: run garbage collection, collect object references, filter and sort data
 - Views to analyze collected data: execution, memory and thread analysis.

Profiling Tools

➤ Profiling Perspective



Profiling Action

Open View Actions

Profiling Monitor View

Profiling Views

Package	Base Time (sec...)	Average Base ...	Cumulative Tim...	Calls
(default package)	1.727952	0.345590	1.727952	5
PerformanceExample	1.727846	0.863923	1.727952	2
main(java.lang.String[]) void	0.452382	0.452382	1.727952	1
PerformanceExample()	1.275464	1.275464	1.275570	1
PerformanceExample\$1	0.000034	0.000034	0.000034	1
PerformanceExample\$1(PerformanceExample)	0.000034	0.000034	0.000034	1
PerformanceExample\$2	0.000040	0.000040	0.000040	1
PerformanceExample\$2(PerformanceExample)	0.000040	0.000040	0.000040	1
PerformanceExample\$3	0.000032	0.000032	0.000032	1
PerformanceExample\$3()	0.000032	0.000032	0.000032	1

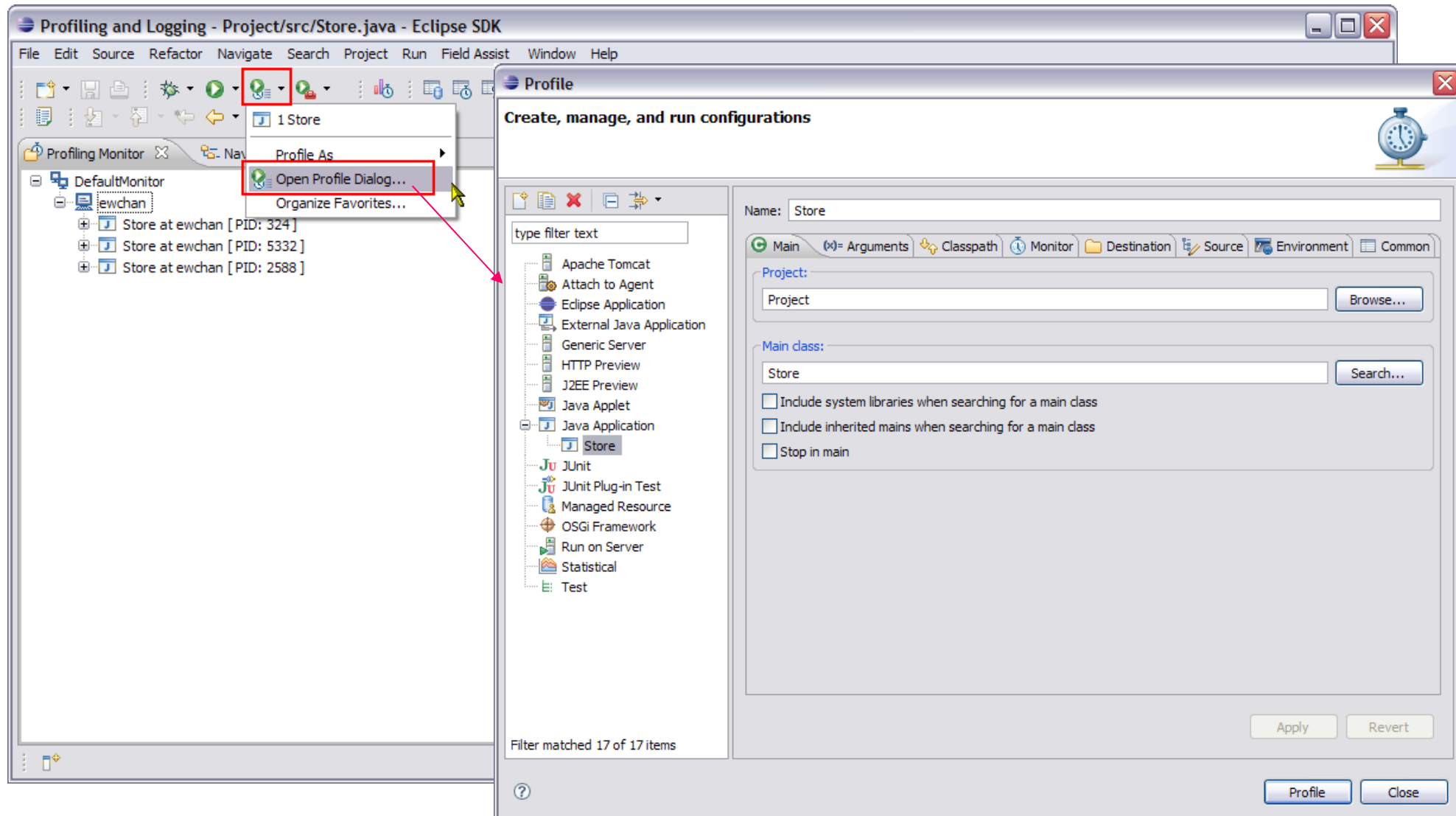
Session summary | Execution Statistics | Call Tree | Method Invocation Details | Method Invocation

Console

New_configuration [External Java Application] java.exe (November 9, 2007 5:32:33 PM)

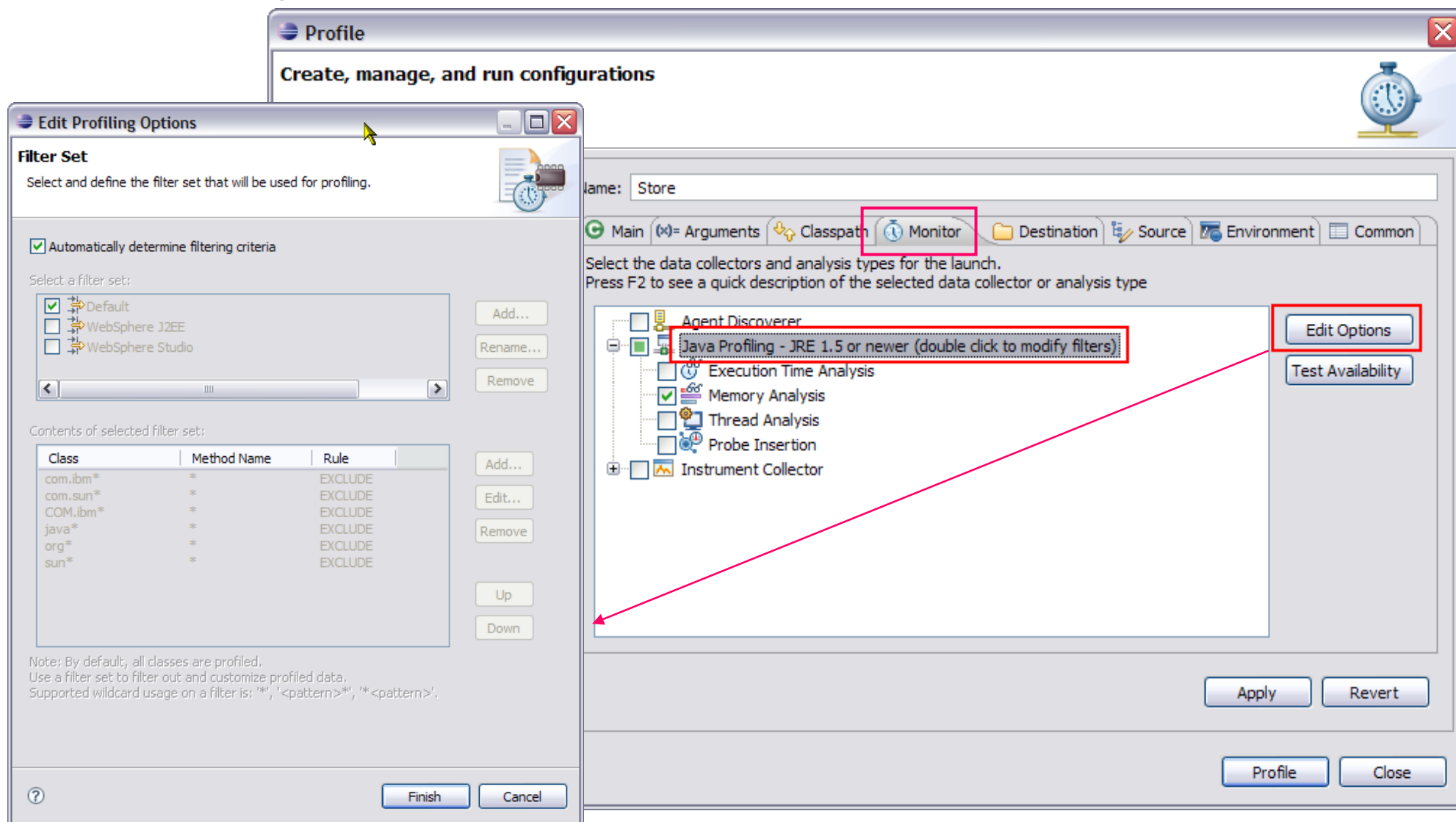
Profiling Tools

➤ Start a profiling session...



Profiling Tools

➤ Profiling Filter



Profile
Create, manage, and run configurations

Name: Store

Main (x) Arguments Classpath **Monitor** Destination Source Environment Common

Select the data collectors and analysis types for the launch.
Press F2 to see a quick description of the selected data collector or analysis type

- Agent Discoverer
- Java Profiling - JRE 1.5 or newer (double click to modify filters)**
- Execution Time Analysis
- Memory Analysis
- Thread Analysis
- Probe Insertion
- Instrument Collector

Edit Options
Test Availability

Apply Revert

Profile Close

Edit Profiling Options

Filter Set
Select and define the filter set that will be used for profiling.

☒ Automatically determine filtering criteria

Select a filter set:

- ☒ Default
- ☐ WebSphere J2EE
- ☐ WebSphere Studio

Add...
Rename...
Remove

Contents of selected filter set:

Class	Method Name	Rule
com.ibm*	*	EXCLUDE
com.sun*	*	EXCLUDE
COM.ibm*	*	EXCLUDE
java*	*	EXCLUDE
org*	*	EXCLUDE
sun*	*	EXCLUDE

Add...
Edit...
Remove

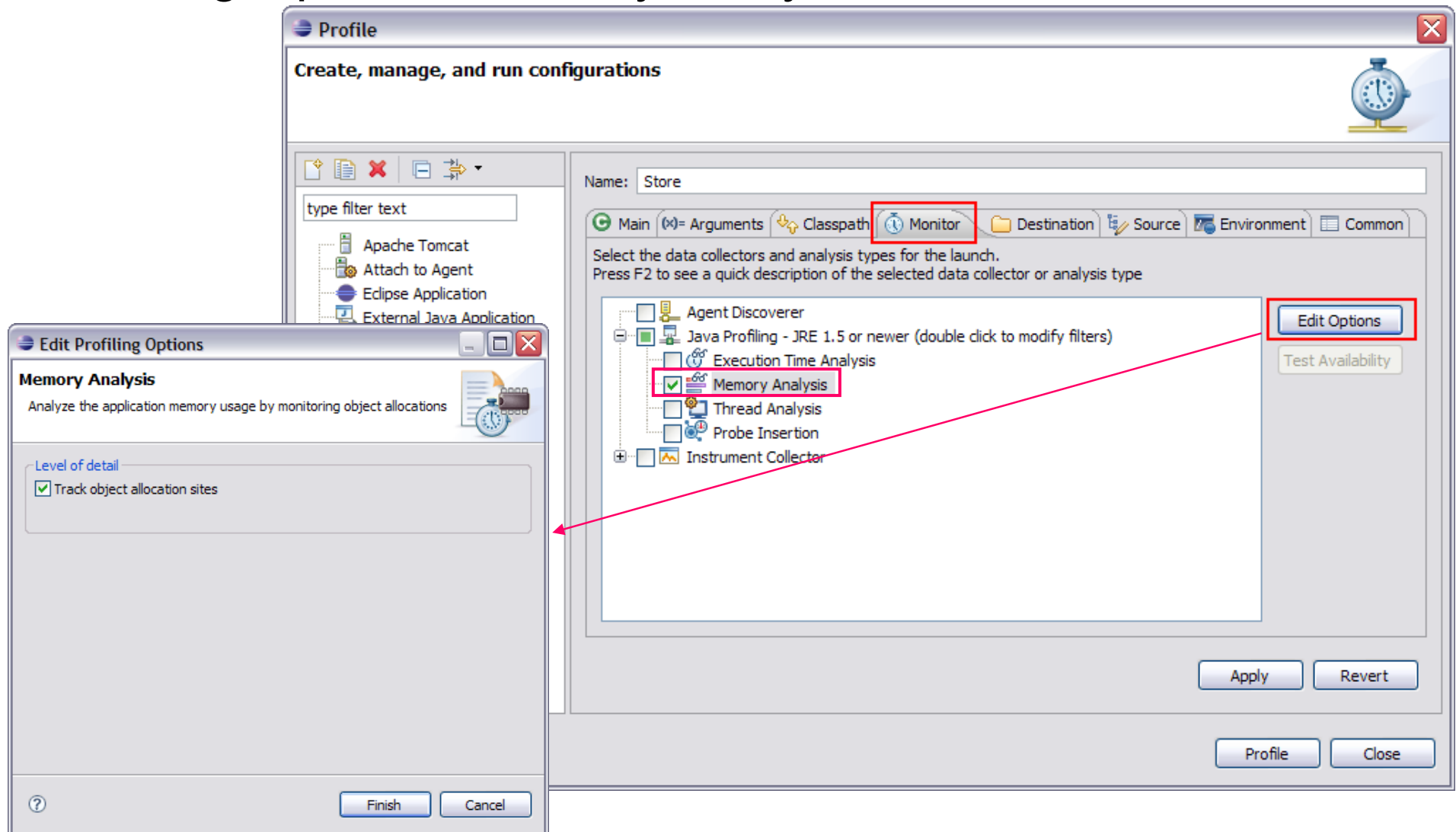
Up
Down

Note: By default, all classes are profiled.
Use a filter set to filter out and customize profiled data.
Supported wildcard usage on a filter is: '*', '<pattern>*', '*<pattern>'.
Supported wildcard usage on a filter is: '*', '<pattern>*', '*<pattern>'.

Finish Cancel

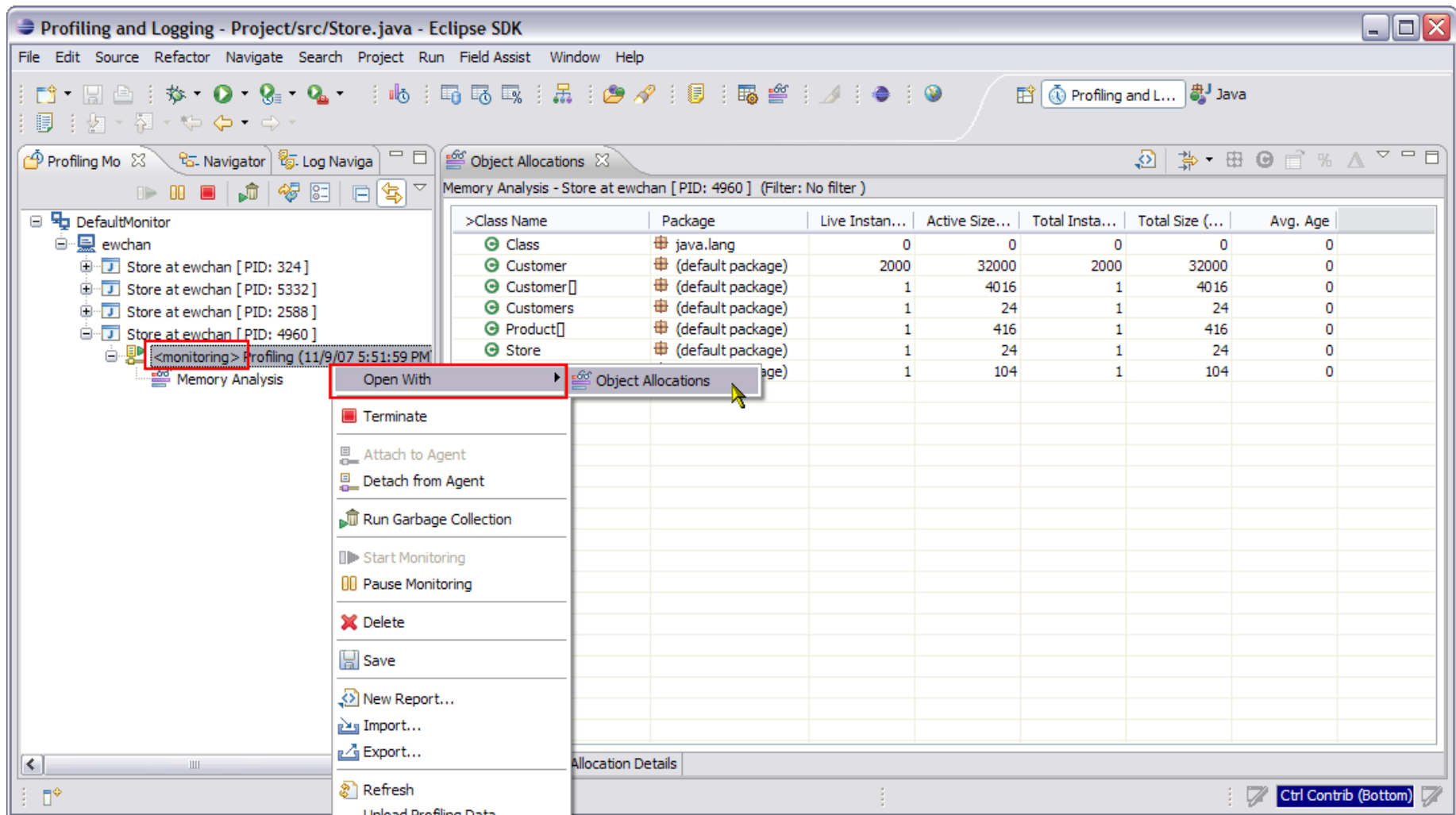
Profiling Tools

➤ Profiling Option – Memory Analysis



Profiling Tools

➤ Open Profiling View



Profiling and Logging - Project/src/Store.java - Eclipse SDK

File Edit Source Refactor Navigate Search Project Run Field Assist Window Help

Profiling Mo Navigator Log Naviga

Object Allocations

Memory Analysis - Store at ewchan [PID: 4960] (Filter: No filter)

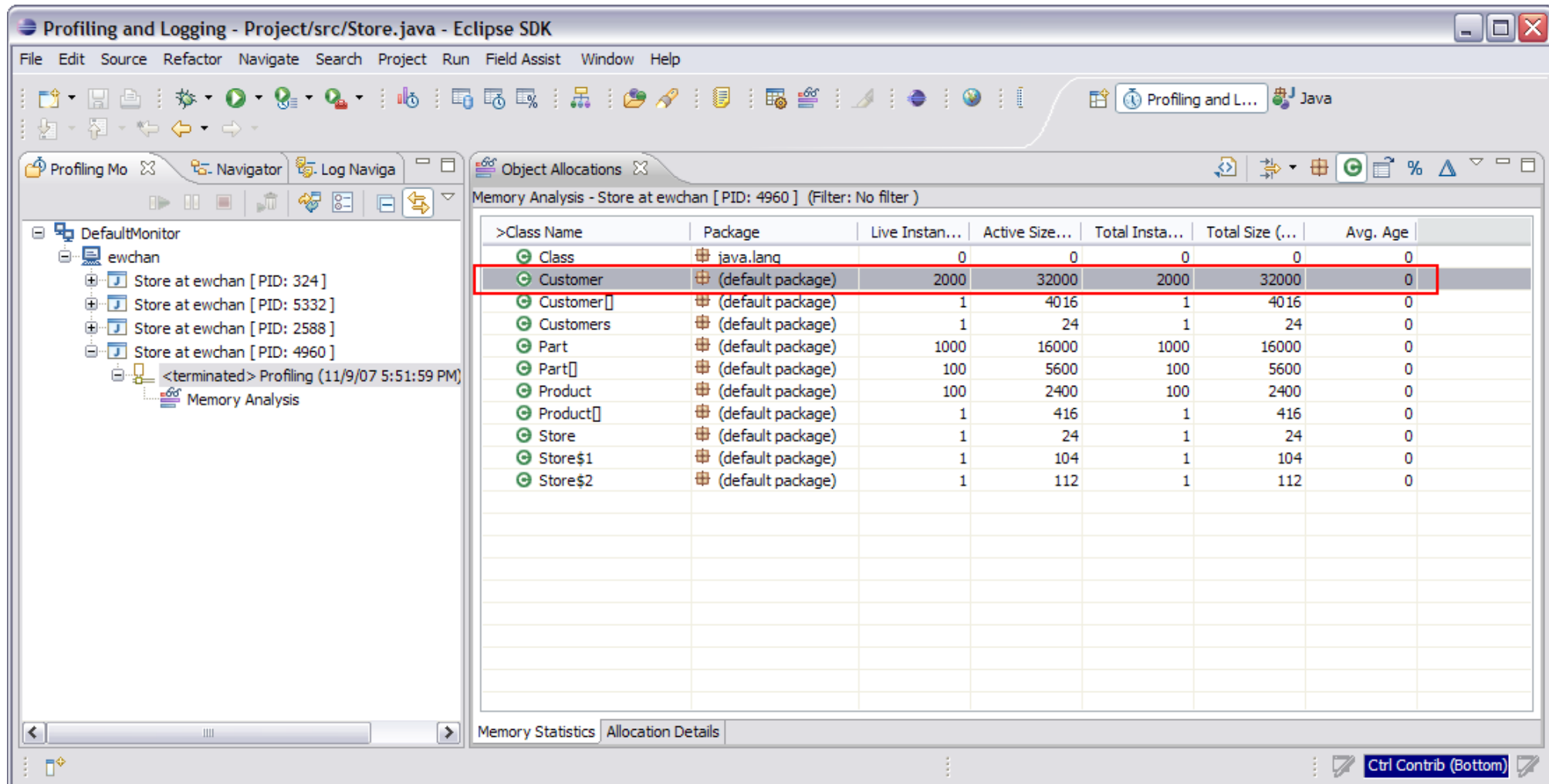
>Class Name	Package	Live Instan...	Active Size...	Total Insta...	Total Size (...)	Avg. Age
Class	java.lang	0	0	0	0	0
Customer	(default package)	2000	32000	2000	32000	0
Customer[]	(default package)	1	4016	1	4016	0
Customers	(default package)	1	24	1	24	0
Product[]	(default package)	1	416	1	416	0
Store	(default package)	1	24	1	24	0
	(default package)	1	104	1	104	0

Allocation Details

Ctrl Contrib (Bottom)

Profiling Tools

➤ Profiling View - Object Allocations view



Profiling and Logging - Project/src/Store.java - Eclipse SDK

File Edit Source Refactor Navigate Search Project Run Field Assist Window Help

Profiling Mo Navigator Log Naviga

Object Allocations

Memory Analysis - Store at ewchan [PID: 4960] (Filter: No filter)

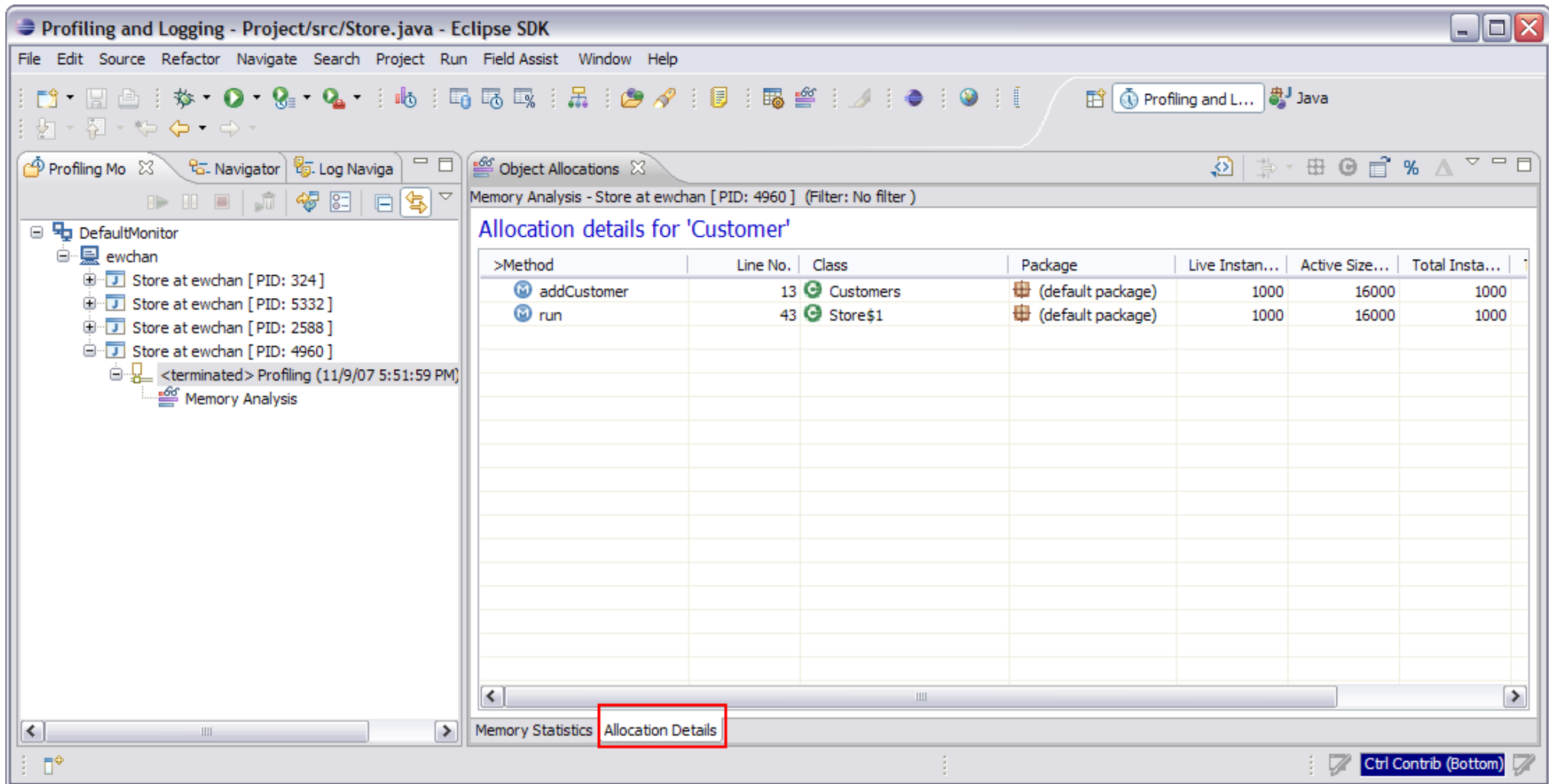
>Class Name	Package	Live Instan...	Active Size...	Total Insta...	Total Size (...)	Avg. Age
Class	java.lang	0	0	0	0	0
Customer	(default package)	2000	32000	2000	32000	0
Customer[]	(default package)	1	4016	1	4016	0
Customers	(default package)	1	24	1	24	0
Part	(default package)	1000	16000	1000	16000	0
Part[]	(default package)	100	5600	100	5600	0
Product	(default package)	100	2400	100	2400	0
Product[]	(default package)	1	416	1	416	0
Store	(default package)	1	24	1	24	0
Store\$1	(default package)	1	104	1	104	0
Store\$2	(default package)	1	112	1	112	0

Memory Statistics Allocation Details

Ctrl Contrib (Bottom)

Profiling Tools

➤ Profiling View – Object Allocation View



Profiling and Logging - Project/src/Store.java - Eclipse SDK

File Edit Source Refactor Navigate Search Project Run Field Assist Window Help

Profiling Mo Navigator Log Naviga

Object Allocations

Memory Analysis - Store at ewchan [PID: 4960] (Filter: No filter)

Allocation details for 'Customer'

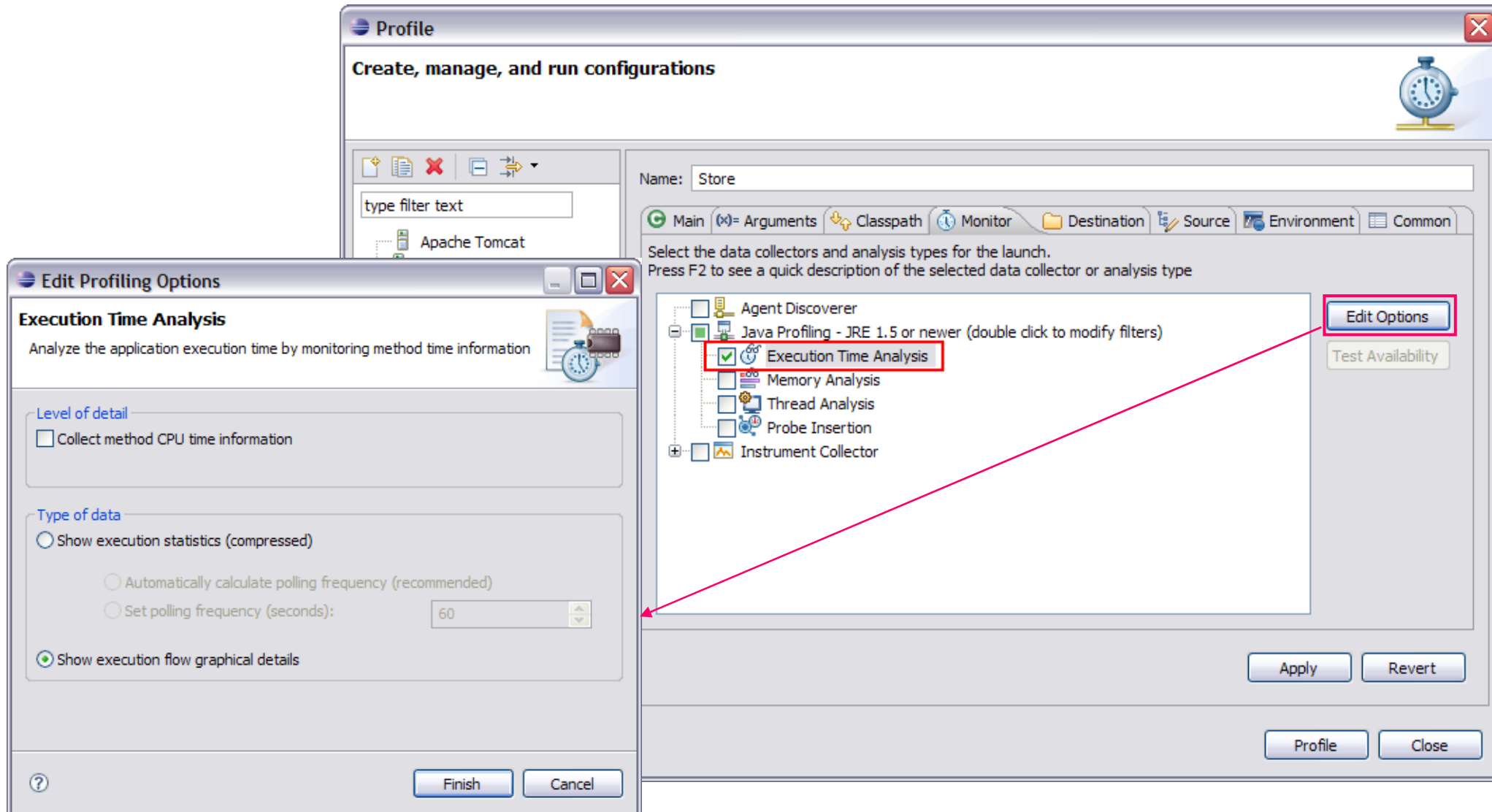
>Method	Line No.	Class	Package	Live Instan...	Active Size...	Total Insta...
addCustomer	13	Customers	(default package)	1000	16000	1000
run	43	Store\$1	(default package)	1000	16000	1000

Memory Statistics Allocation Details

Ctrl Contrib (Bottom)

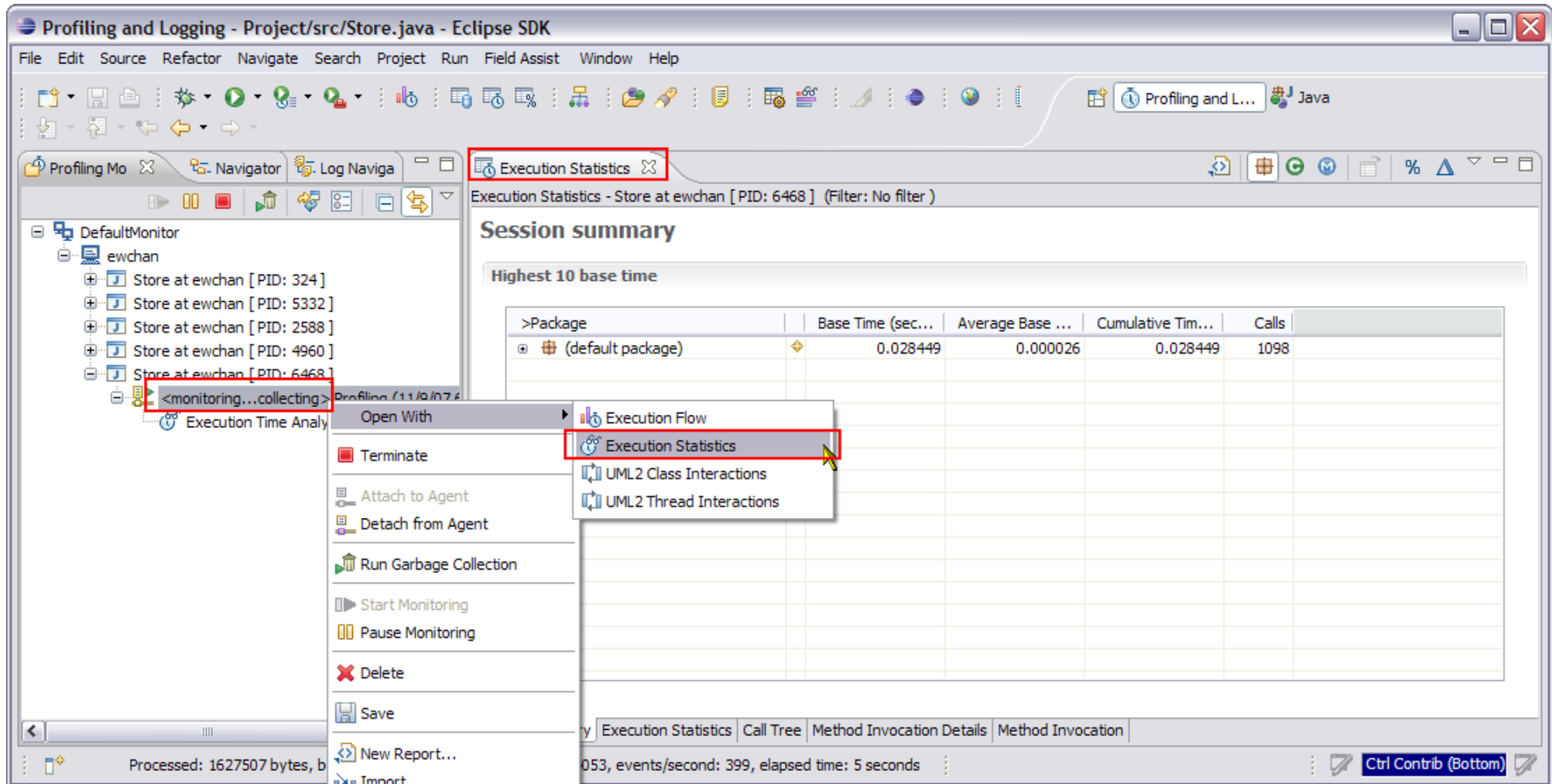
Profiling Tools

➤ Profiling Option – Execution Time Analysis



Profiling Tools

➤ Profiling View – Execution Statistic View



Profiling and Logging - Project/src/Store.java - Eclipse SDK

File Edit Source Refactor Navigate Search Project Run Field Assist Window Help

Profiling Mo Navigator Log Naviga Execution Statistics

Execution Statistics - Store at ewchan [PID: 6468] (Filter: No filter)

Session summary

Highest 10 base time

>Package	Base Time (sec...)	Average Base ...	Cumulative Tim...	Calls
(default package)	0.028449	0.000026	0.028449	1098

Context menu options:

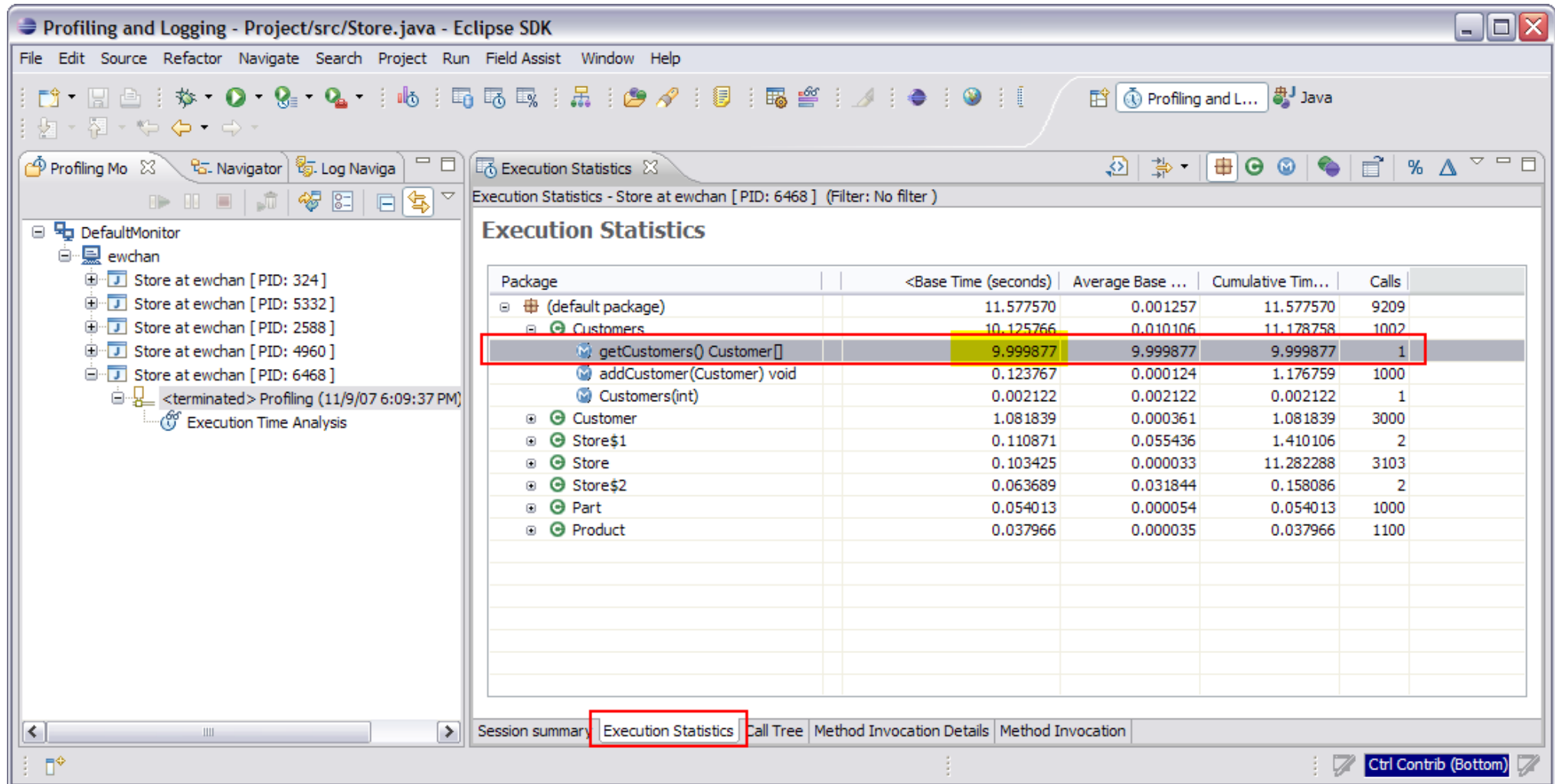
- Open With
 - Execution Flow
 - Execution Statistics**
 - UML2 Class Interactions
 - UML2 Thread Interactions
- Terminate
- Attach to Agent
- Detach from Agent
- Run Garbage Collection
- Start Monitoring
- Pause Monitoring
- Delete
- Save
- New Report...
- Import...

Processed: 1627507 bytes, b

Ctrl Contrib (Bottom)

Profiling Tools

➤ Profiling View – Execution Statistics View



Profiling and Logging - Project/src/Store.java - Eclipse SDK

File Edit Source Refactor Navigate Search Project Run Field Assist Window Help

Profiling Mo Navigator Log Naviga

Execution Statistics - Store at ewchan [PID: 6468] (Filter: No filter)

Execution Statistics

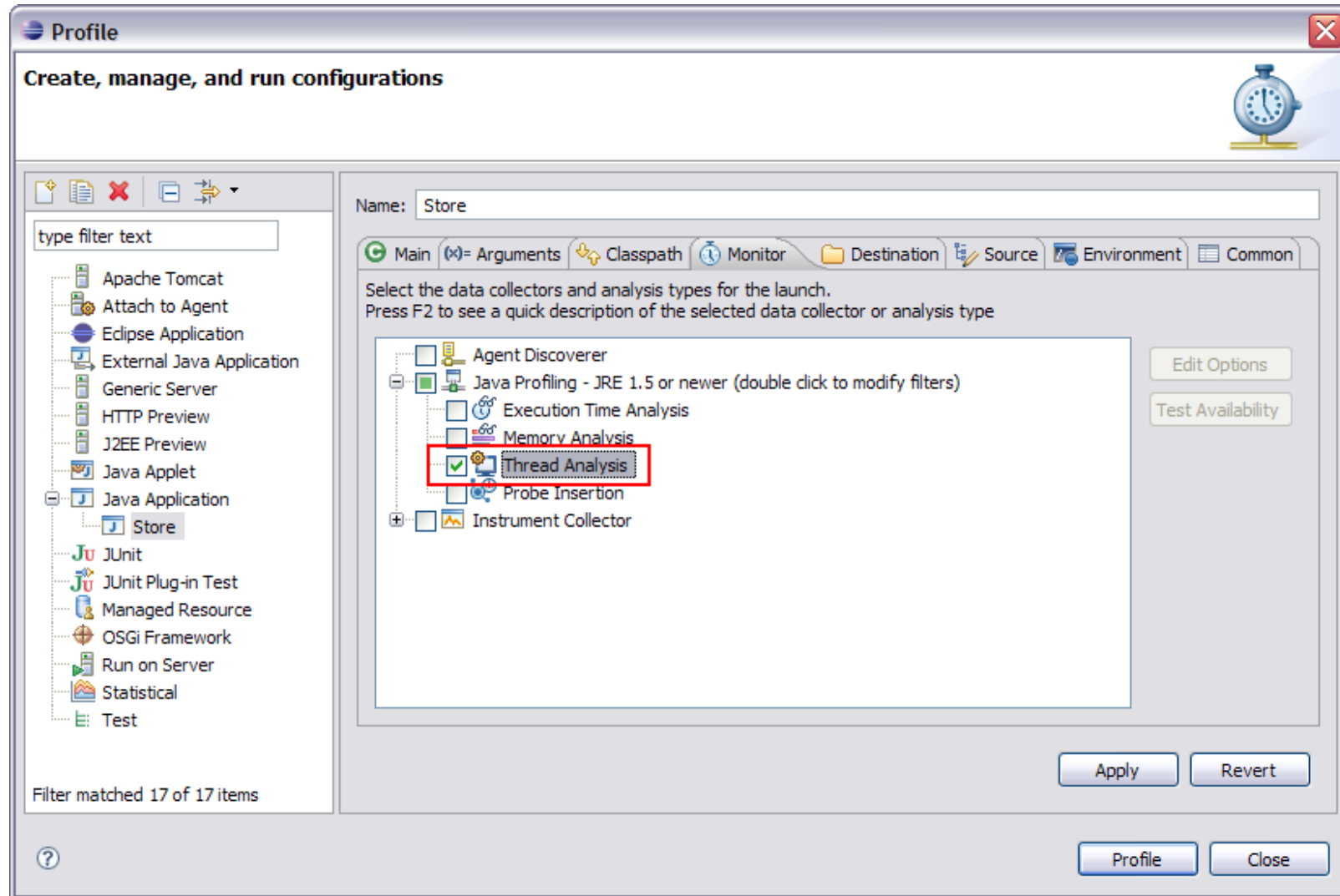
Package	<Base Time (seconds)	Average Base ...	Cumulative Tim...	Calls
(default package)	11.577570	0.001257	11.577570	9209
Customers	10.125766	0.010106	11.178758	1002
getCustomers() Customer[]	9.999877	9.999877	9.999877	1
addCustomer(Customer) void	0.123767	0.000124	1.176759	1000
Customers(int)	0.002122	0.002122	0.002122	1
Customer	1.081839	0.000361	1.081839	3000
Store\$1	0.110871	0.055436	1.410106	2
Store	0.103425	0.000033	11.282288	3103
Store\$2	0.063689	0.031844	0.158086	2
Part	0.054013	0.000054	0.054013	1000
Product	0.037966	0.000035	0.037966	1100

Session summary Execution Statistics Call Tree Method Invocation Details Method Invocation

Ctrl Contrib (Bottom)

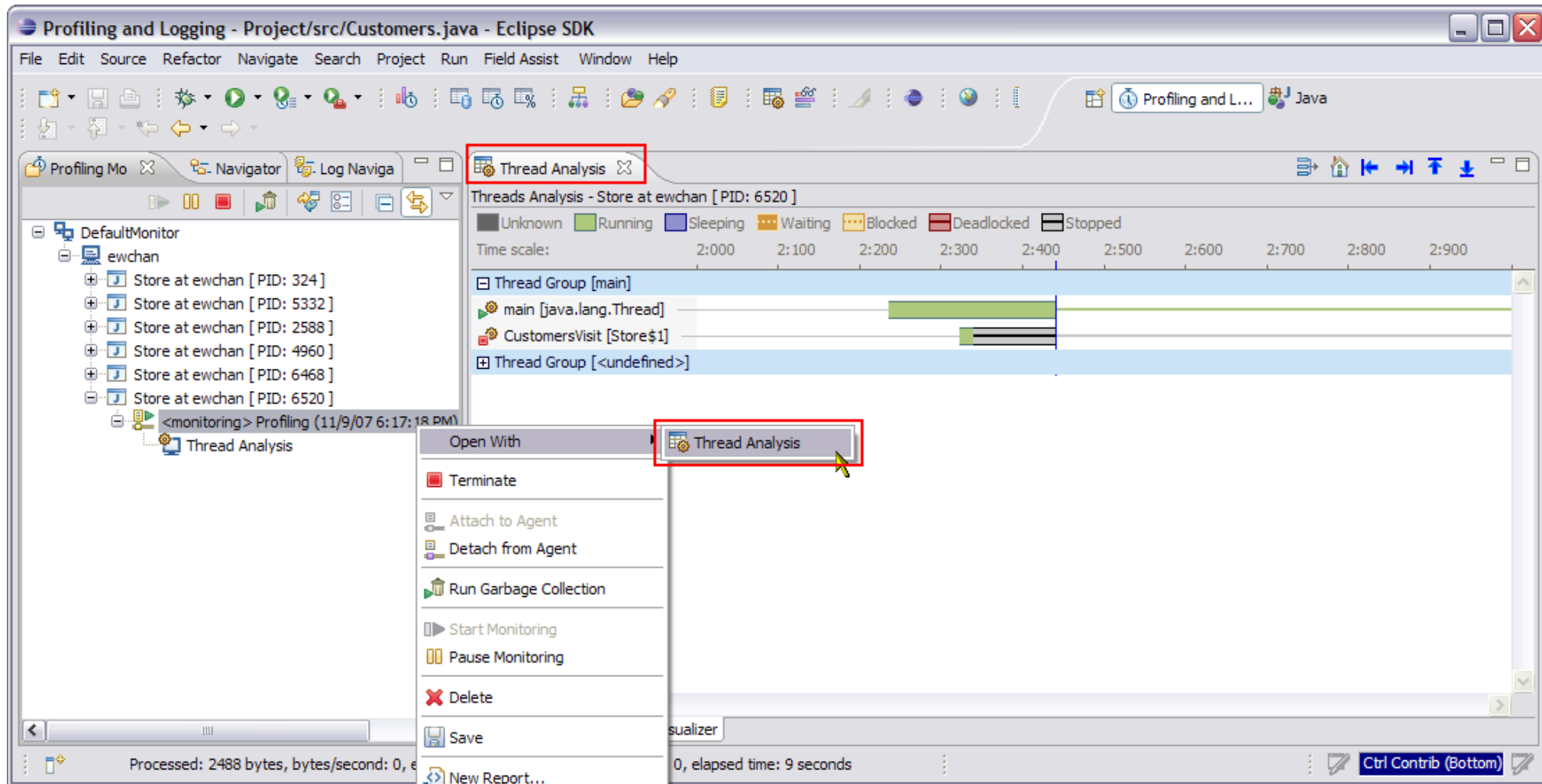
Profiling Tools

➤ Profiling Option – Thread Analysis



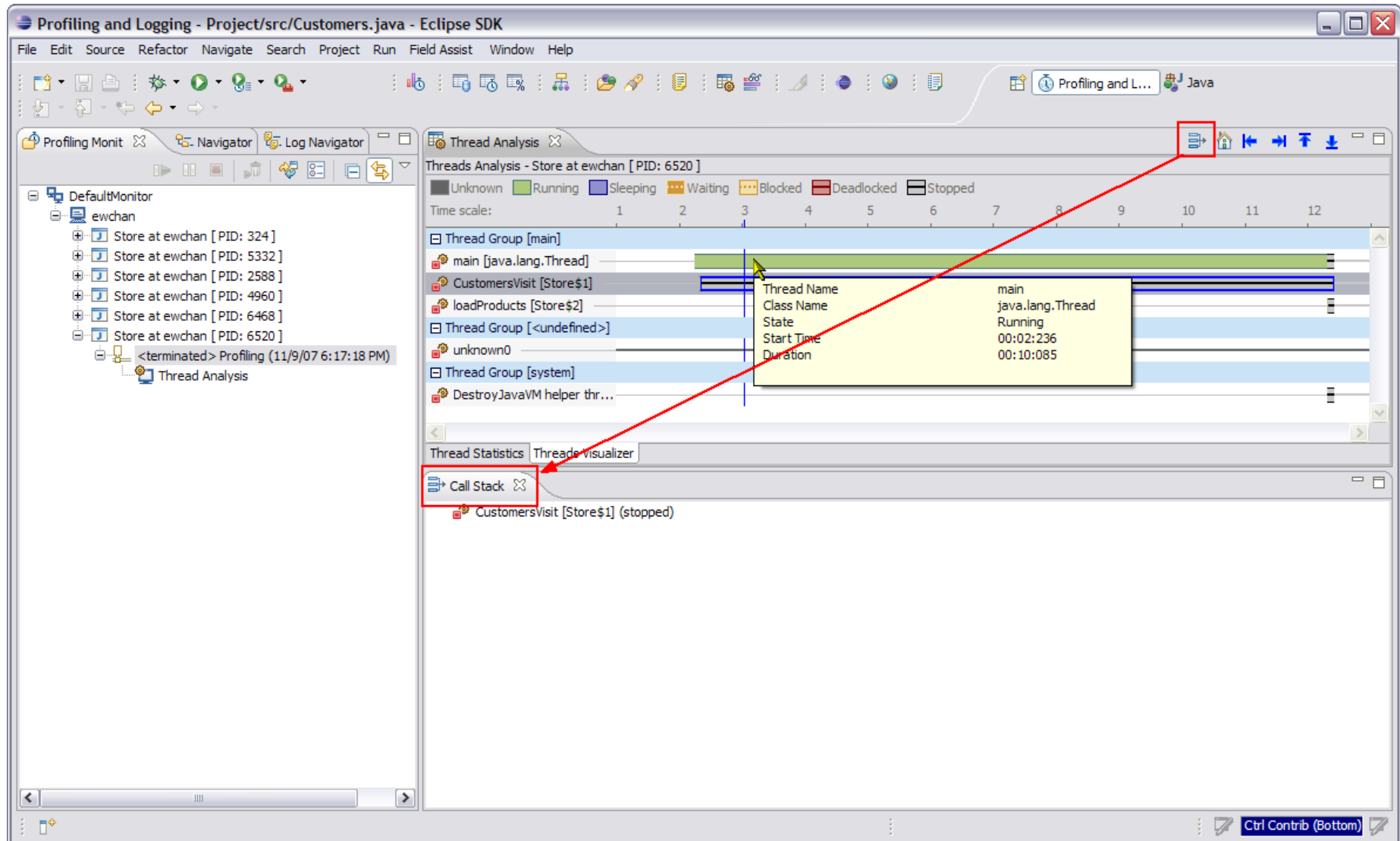
Profiling Tool

➤ Profiling View – Thread Analysis View



Profiling Tool

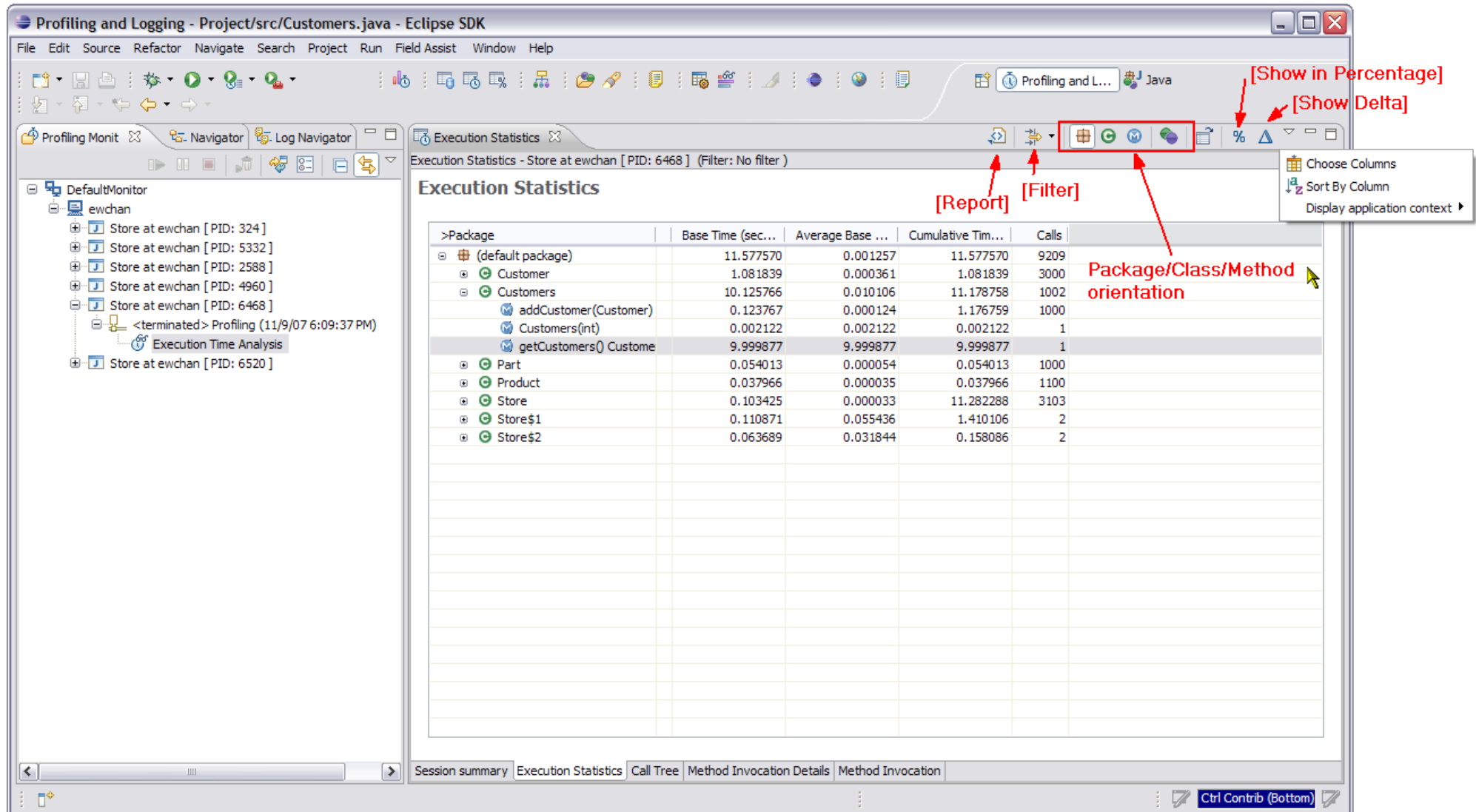
➤ Profiling View – Call Stack



The screenshot shows the Eclipse Profiling and Logging tool interface. The main window is titled "Profiling and Logging - Project/src/Customers.java - Eclipse SDK". The interface includes a menu bar (File, Edit, Source, Refactor, Navigate, Search, Project, Run, Field Assist, Window, Help) and a toolbar with various icons. The left sidebar contains a "Profiling Monitor" view showing a tree of monitoring elements, including "DefaultMonitor", "ewchan", and several "Store at ewchan" entries with their respective PIDs. The main area is divided into two panes. The top pane, titled "Thread Analysis - Store at ewchan [PID: 6520]", displays a "Threads Analysis" view with a time scale from 1 to 12. It shows a list of threads, including "main [java.lang.Thread]", "CustomersVisit [Store\$1]", "loadProducts [Store\$2]", "unknown0", and "DestroyJavaVM helper thr...". A tooltip is visible over the "main" thread, showing details: Thread Name (main), Class Name (java.lang.Thread), State (Running), Start Time (00:02:236), and Duration (00:10:085). The bottom pane, titled "Thread Statistics" and "Threads visualizer", shows a "Call Stack" view for the "CustomersVisit [Store\$1] (stopped)" thread. A red arrow points from the "Call Stack" icon in the bottom pane to the "main" thread in the top pane.

Profiling Tool

➤ Profiling Views – Toolbar Actions



Execution Statistics - Store at ewchan [PID: 6468] (Filter: No filter)

Execution Statistics

>Package	Base Time (sec...)	Average Base ...	Cumulative Tim...	Calls
(default package)	11.577570	0.001257	11.577570	9209
Customer	1.081839	0.000361	1.081839	3000
Customers	10.125766	0.010106	11.178758	1002
addCustomer(Customer)	0.123767	0.000124	1.176759	1000
Customers(int)	0.002122	0.002122	0.002122	1
getCustomers() Custome	9.999877	9.999877	9.999877	1
Part	0.054013	0.000054	0.054013	1000
Product	0.037966	0.000035	0.037966	1100
Store	0.103425	0.000033	11.282288	3103
Store\$1	0.110871	0.055436	1.410106	2
Store\$2	0.063689	0.031844	0.158086	2

Toolbar actions:

- [Report]
- [Filter]
- [Show in Percentage]
- [Show Delta]
- Choose Columns
- Sort By Column
- Display application context

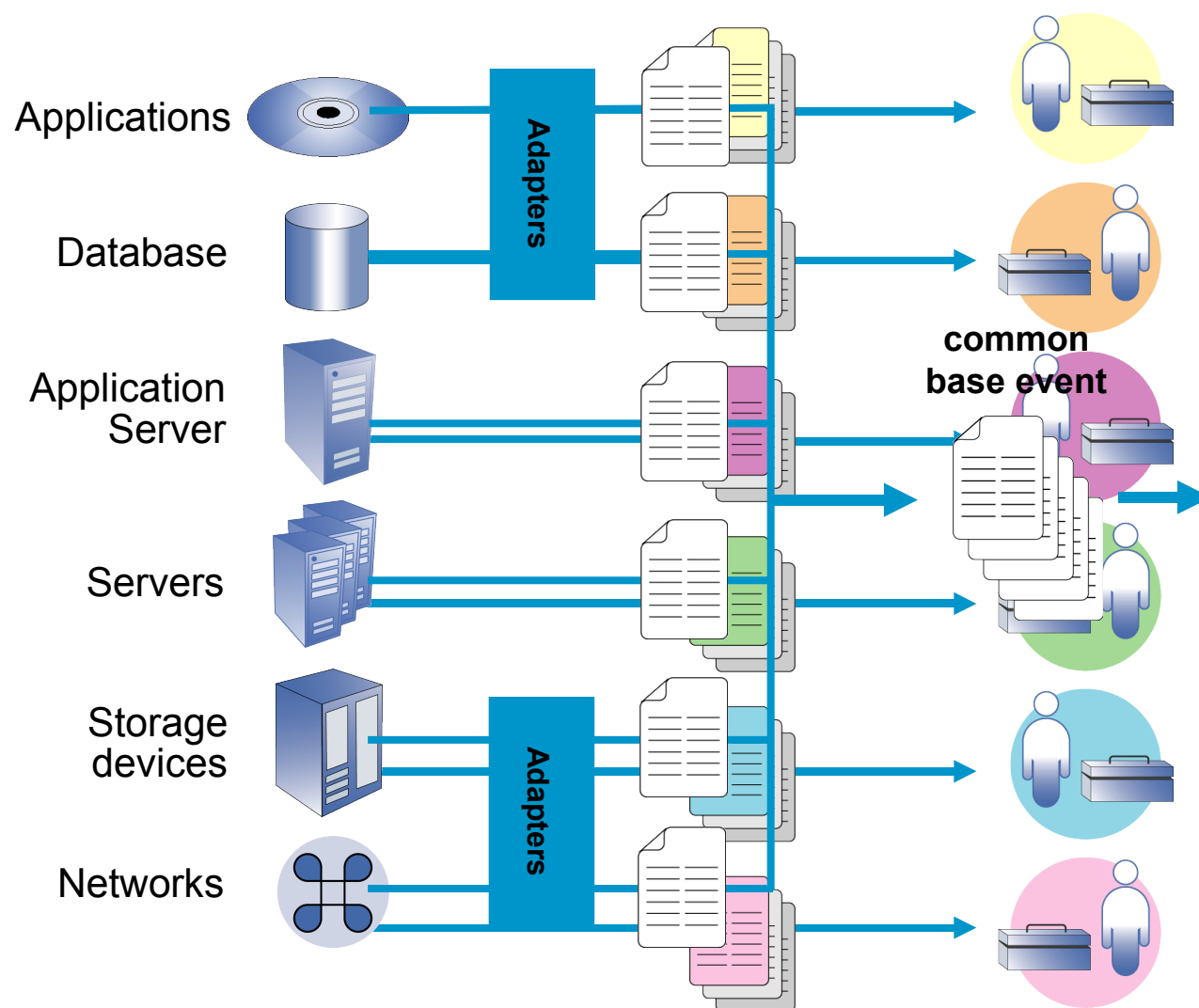
Package/Class/Method orientation

Agenda

- ✓ Overview of TPTP
- ✓ TPTP Profiling tools
- **TPTP Monitoring**
- Extending TPTP
- Challenges
- What's new?



Monitoring - Overview



- Disparate pieces and parts
- Tools focused on individual products
- No synergies in creating log entries
- Common format for log files
- Common set of tools
- Common interfaces among tools

Monitoring - Overview

➤ Logging Agent

- XML message-based agent for real-time monitoring of message generating facilities (e.g. loggers).
- Extensible architecture for crafting proprietary Logging Agents:
 - Defined APIs and programming model.
- C implementation with a Java wrapper for supporting C/C++ and Java run-times.
- TPTP provides Logging Agent support for several popular Java-based logging and tracing facilities.

➤ Common Base Event

- Open-source specification (OASIS) to provide a *common* and standardized taxonomy for *events* occurring in hardware and software
- Unified format and terminology for the standardized exchange and consistent interpretation of problem determination data to circumvent varying vendor, product and version representations.
- Common Base Event XML schema defines the overall structure of the event, format of each property and all mandatory properties for completeness.
- TPTP provides EMF consumer (e.g. model artifacts) Java, EMF producer (e.g. native logging) Java, non-EMF Java and C/C++ implementations



Monitoring – Generic Log Adapter

➤ Generic Log Adapter

- A component of TPTP
- In TPTP, Generic Log Adapter transforms log messages into Common Base Event objects. Common Base Event is the common format for messages in TPTP.
- Has flexible architecture that is easily extendable.
- Uses a configuration XML file to describe the transformation
 - Includes Standalone run-time
 - Adapter configuration files and sample execution scripts
 - Editor for creating, modifying and testing configuration files



Monitoring – GLA adapters

➤ GLA

- Transforms are defined in adapter files:
- Contains one or more contexts and the components for each context.
- Stored as XML based on a defined schema.
- Parsing component incorporates mapping proprietary log and trace record properties to Common Base Event properties using mapping rules.

- TPTP provides Common Base Event and Logging Agent standalone and plug-in support for the following popular logging facilities:
 - Jakarta Apache Commons
 - Java Logging (JSR-047)
 - Jakarta Apache Log4J



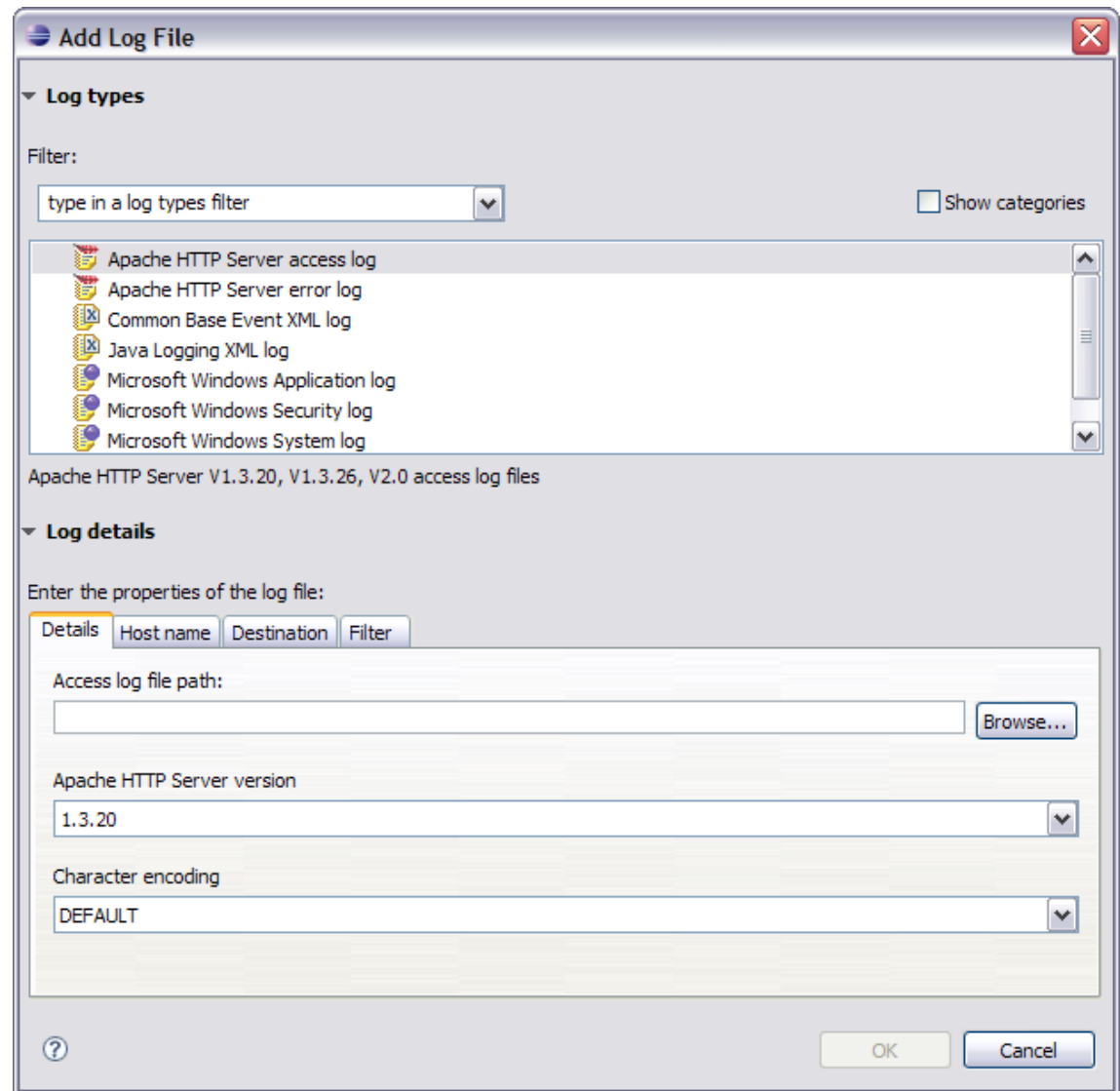
Monitoring – Symptom Catalog

➤ Symptom Catalog

- Symptoms are the sign of a malfunction, a problem or just simply a change of state in a resource that is the subject of our problem determination analysis.
- A *database* consisting of matching patterns indicating known problems, explanations and resolution steps.
- Persisted as an XML file with a defined schema.
- Crafted using the Symptom Editor of TPTP
- Vendors and organizations may provide multiple hierarchal symptom catalog based on logical or business divisions.
- Local and remote (FTP/HTTP) importing functionality to ensure content freshness.

Monitoring – Import Log

- Import Log Wizard
- Log Type (Adapter)
- Log Location
 - Local
 - Remote
- Log details



Monitoring – Log View

Profiling and Logging - Eclipse SDK

File Edit Navigate Search Project Run Window Help

Profiling Monitor Navigator Log Navigator

Log View - Common Base Event XML log activity_v5_small.xml

(Filter: No filter)

Log Records (Page 1 of 2 Filter matched 122 of 122 records)

Creation Time	Severity	Message Text	Priority	Situation Type	Source Component...	Source Sub-compon...	Sequence...
Jan 27, 2003 6:54:14.001000 PM	50	J2CA0020E: The Connection Pool M...	0	ReportSituation	IBM WebSphere Ap...	Application Server:c...	0
Jan 27, 2003 6:54:14.000000 PM	50	J2CA0046E: Method createManaged...	0	ReportSituation	IBM WebSphere Ap...	Application Server:c...	0
Jan 27, 2003 6:01:53.001000 PM	50	J2CA0020E: The Connection Pool M...	0	ReportSituation	IBM WebSphere Ap...	Application Server:c...	0
Jan 27, 2003 6:01:53.000000 PM	50	J2CA0046E: Method createManaged...	0	ReportSituation	IBM WebSphere Ap...	Application Server:c...	0
Jan 27, 2003 3:31:10.000000 PM	10	SRVE0169I: Loading Web Module: M...	0	ReportSituation	IBM WebSphere Ap...	Application Server:c...	0
Jan 27, 2003 2:57:41.000000 PM	10	SRVE0169I: Loading Web Module: Si...	0	ReportSituation	IBM WebSphere Ap...	Application Server:c...	0
Jan 27, 2003 10:31:04.000000 AM	10	SRVE0169I: Loading Web Module: Pl...	0	ReportSituation	IBM WebSphere Ap...	Application Server:c...	0
Jan 27, 2003 9:41:02.000000 AM	10	SRVE0169I: Loading Web Module: T...	0	ReportSituation	IBM WebSphere Ap...	Application Server:c...	0
Jan 27, 2003 9:24:07.000000 AM	10	WSVR0200I: Starting application: De...	0	StartSituation	IBM WebSphere Ap...	Application Server:c...	0
Jan 27, 2003 4:24:30.000000 AM	10	SRVE0169I: Loading Web Module: M...	0	ReportSituation	IBM WebSphere Ap...	Application Server:c...	0
Jan 27, 2003 1:36:15.000000 AM	10	ADMIN0015I: AdminService initialized	0	ReportSituation	IBM WebSphere Ap...	Application Server:c...	0
Jan 26, 2003 11:07:29.001000 PM	10	SRVE0163I: Supported JSP Specifica...	0	ReportSituation	IBM WebSphere Ap...	Application Server:c...	0
Jan 26, 2003 11:07:29.000000 PM	10	SRVE0162I: Servlet Specification Le...	0	ReportSituation	IBM WebSphere Ap...	Application Server:c...	0
Jan 26, 2003 6:24:11.000000 PM	10	SRVE0169I: Loading Web Module: IV...	0	ReportSituation	IBM WebSphere Ap...	Application Server:c...	0
Jan 26, 2003 5:17:50.000000 PM	10	SRVE0169I: Loading Web Module: M...	0	ReportSituation	IBM WebSphere Ap...	Application Server:c...	0
Jan 26, 2003 12:17:44.000000 PM	10	SRVE0169I: Loading Web Module: Pl...	0	ReportSituation	IBM WebSphere Ap...	Application Server:c...	0
Jan 26, 2003 9:47:45.001000 AM	10	WSVR0200I: Starting application: pe...	0	StartSituation	IBM WebSphere Ap...	Application Server:c...	0
Jan 26, 2003 9:47:45.000000 AM	10	WSVR0221I: Application started: Pla...	0	StartSituation	IBM WebSphere Ap...	Application Server:c...	0
Jan 26, 2003 6:27:36.000000 AM	10	SRVE0169I: Loading Web Module: S...	0	ReportSituation	IBM WebSphere Ap...	Application Server:c...	0

Properties

Event Details

Additional Data Attributes

Correlation Data Attributes

Situation

Message Information

Source Component

Reporting Component

Associated Event

CommonBaseEvent XML

Analysis Result

Message Text

J2CA0020E: The Connection Pool Manager could not allocate a Managed Connection: javax.resource.spi.ResourceAllocationException: Caught ResourceException thrown during creation of the ManagedConnection.
at com.ibm.ejs.j2c.poolmanager.FreePool.createManagedConnectionWithMCWrapper(FreePool.java:1138)
at com.ibm.ejs.j2c.poolmanager.FreePool.createOrWaitForConnection(FreePool.java:897)
at com.ibm.ejs.j2c.poolmanager.PoolManager.reserve(PoolManager.java:1065)
at com.ibm.ejs.j2c.ConnectionManager.allocateMCWrapper(ConnectionManager.java:560)

Creation Time: Jan 27, 2003 6:54:14.001000 PM

Severity: 50

Global Instance ID: A1DB11E68241C420C4739A1DF6586C1

Local Instance ID:

Number of Repeated Events: 0

Version: 1.0.1

Priority: 0

Event Type: CBECommonBaseEvent

Sequence Number: 0

Duration of Repeated Events: 0

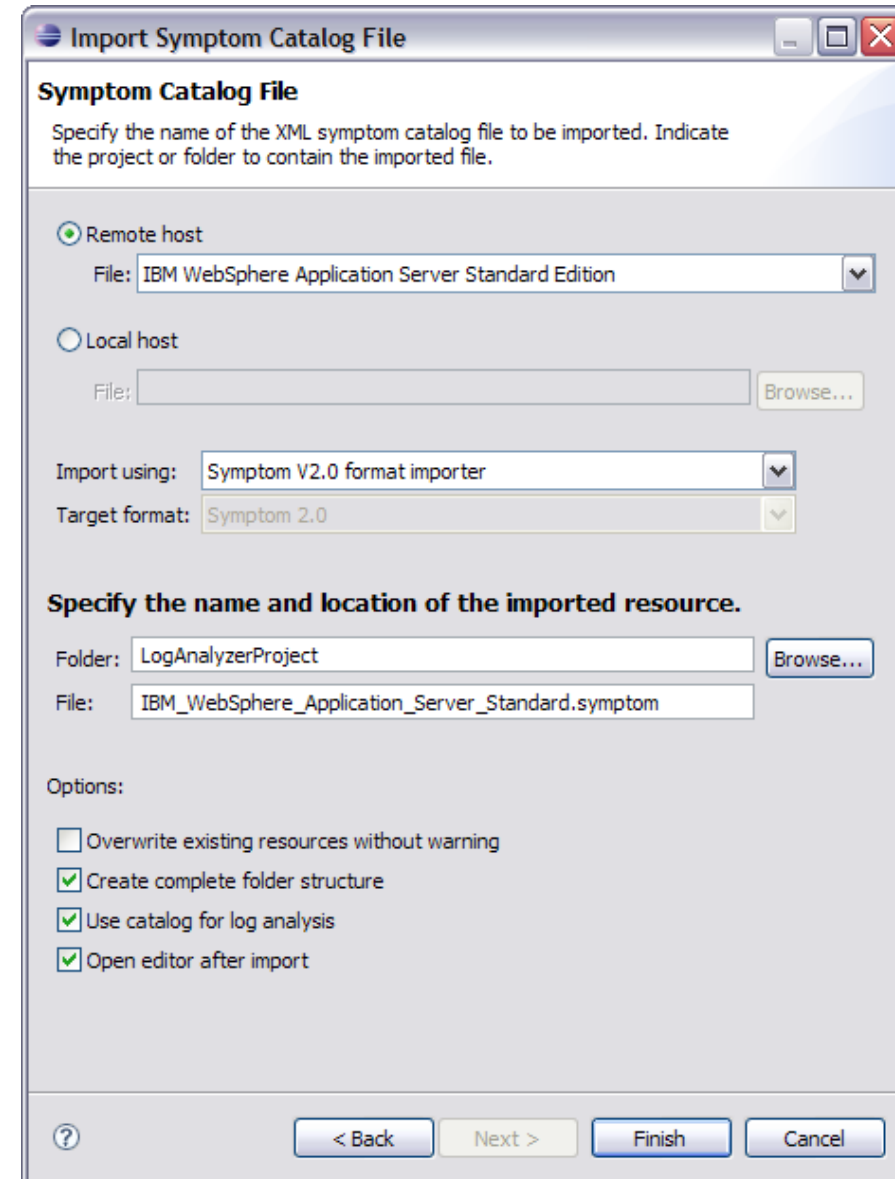
Monitoring – Import Symptom

➤ Import Symptom Wizard

- Local
- Remote

➤ Log Analysis

- Allows users to easily detect and solve problems that have already been previously encountered, and persisted in symptom catalog
- Analysis consists of lexicographically comparing varying Common Base Event properties with match patterns in one or more symptom databases.
- Extensible architecture to allow users to define vendor and product specific analysis engines.

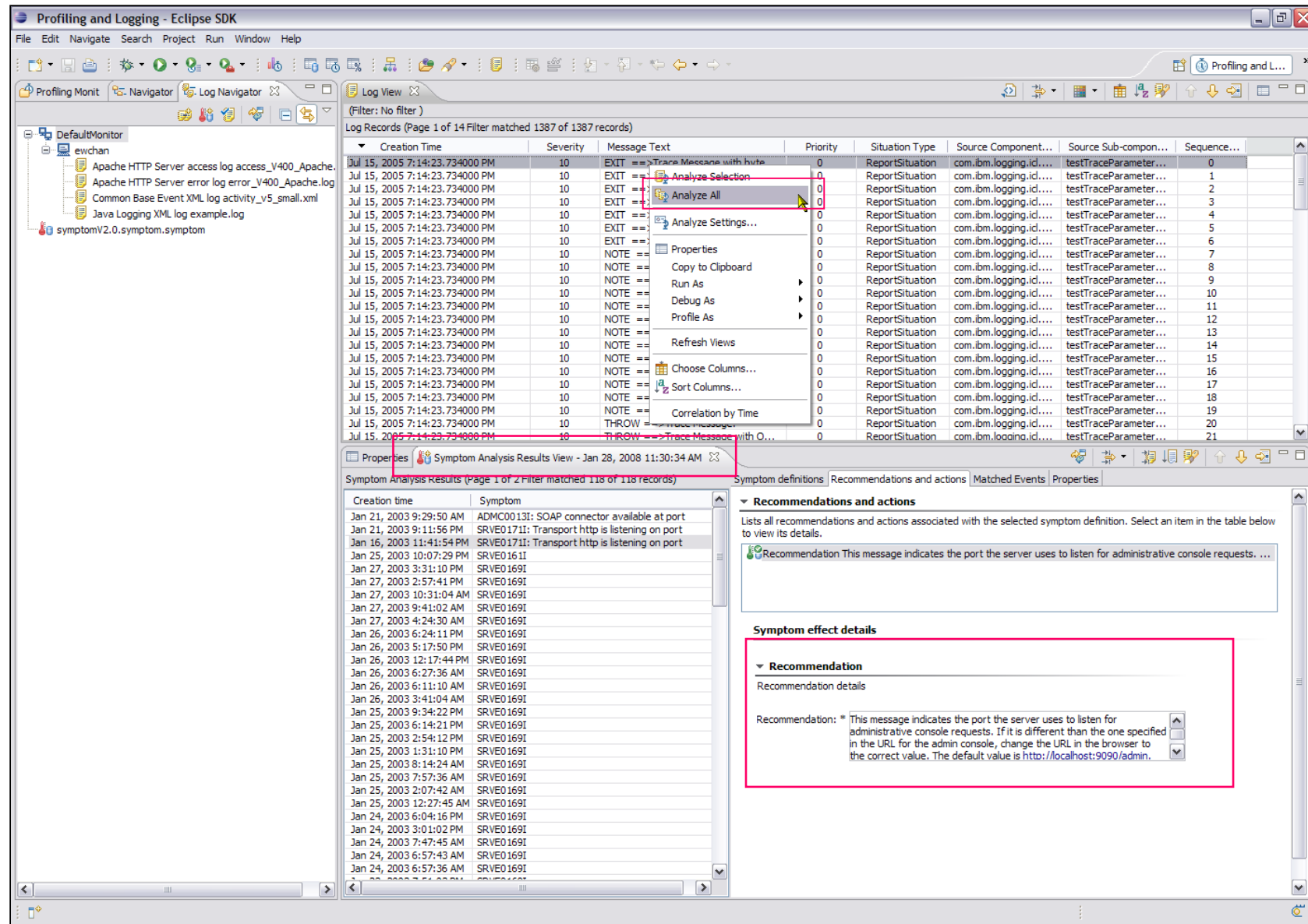


The dialog box is titled "Import Symptom Catalog File". It contains the following sections:

- Symptom Catalog File**: A text box for the file name, currently containing "IBM WebSphere Application Server Standard Edition".
- Remote host**: A radio button is selected. Below it is a dropdown menu showing "IBM WebSphere Application Server Standard Edition".
- Local host**: A radio button is unselected. Below it is a text box for the file name and a "Browse..." button.
- Import using**: A dropdown menu showing "Symptom V2.0 format importer".
- Target format**: A dropdown menu showing "Symptom 2.0".
- Specify the name and location of the imported resource.**: A section with a "Folder" text box containing "LogAnalyzerProject" and a "Browse..." button, and a "File" text box containing "IBM_WebSphere_Application_Server_Standard.symptom".
- Options**: A section with four checkboxes:
 - ☐ Overwrite existing resources without warning
 - ☒ Create complete folder structure
 - ☒ Use catalog for log analysis
 - ☒ Open editor after import

At the bottom, there are four buttons: "< Back", "Next >", "Finish", and "Cancel".

Monitoring – Analysis Results View



Log View
(Filter: No filter)
Log Records (Page 1 of 14 Filter matched 1387 of 1387 records)

Creation Time	Severity	Message Text	Priority	Situation Type	Source Component...	Source Sub-compon...	Sequence...
Jul 15, 2005 7:14:23.734000 PM	10	EXIT ==> Trace Message with huta	0	ReportSituation	com.ibm.logging.id....	testTraceParameter...	0
Jul 15, 2005 7:14:23.734000 PM	10	EXIT ==> Trace Message with huta	0	ReportSituation	com.ibm.logging.id....	testTraceParameter...	1
Jul 15, 2005 7:14:23.734000 PM	10	EXIT ==> Trace Message with huta	0	ReportSituation	com.ibm.logging.id....	testTraceParameter...	2
Jul 15, 2005 7:14:23.734000 PM	10	EXIT ==> Trace Message with huta	0	ReportSituation	com.ibm.logging.id....	testTraceParameter...	3
Jul 15, 2005 7:14:23.734000 PM	10	EXIT ==> Trace Message with huta	0	ReportSituation	com.ibm.logging.id....	testTraceParameter...	4
Jul 15, 2005 7:14:23.734000 PM	10	EXIT ==> Trace Message with huta	0	ReportSituation	com.ibm.logging.id....	testTraceParameter...	5
Jul 15, 2005 7:14:23.734000 PM	10	EXIT ==> Trace Message with huta	0	ReportSituation	com.ibm.logging.id....	testTraceParameter...	6
Jul 15, 2005 7:14:23.734000 PM	10	EXIT ==> Trace Message with huta	0	ReportSituation	com.ibm.logging.id....	testTraceParameter...	7
Jul 15, 2005 7:14:23.734000 PM	10	EXIT ==> Trace Message with huta	0	ReportSituation	com.ibm.logging.id....	testTraceParameter...	8
Jul 15, 2005 7:14:23.734000 PM	10	EXIT ==> Trace Message with huta	0	ReportSituation	com.ibm.logging.id....	testTraceParameter...	9
Jul 15, 2005 7:14:23.734000 PM	10	EXIT ==> Trace Message with huta	0	ReportSituation	com.ibm.logging.id....	testTraceParameter...	10
Jul 15, 2005 7:14:23.734000 PM	10	EXIT ==> Trace Message with huta	0	ReportSituation	com.ibm.logging.id....	testTraceParameter...	11
Jul 15, 2005 7:14:23.734000 PM	10	EXIT ==> Trace Message with huta	0	ReportSituation	com.ibm.logging.id....	testTraceParameter...	12
Jul 15, 2005 7:14:23.734000 PM	10	EXIT ==> Trace Message with huta	0	ReportSituation	com.ibm.logging.id....	testTraceParameter...	13
Jul 15, 2005 7:14:23.734000 PM	10	EXIT ==> Trace Message with huta	0	ReportSituation	com.ibm.logging.id....	testTraceParameter...	14
Jul 15, 2005 7:14:23.734000 PM	10	EXIT ==> Trace Message with huta	0	ReportSituation	com.ibm.logging.id....	testTraceParameter...	15
Jul 15, 2005 7:14:23.734000 PM	10	EXIT ==> Trace Message with huta	0	ReportSituation	com.ibm.logging.id....	testTraceParameter...	16
Jul 15, 2005 7:14:23.734000 PM	10	EXIT ==> Trace Message with huta	0	ReportSituation	com.ibm.logging.id....	testTraceParameter...	17
Jul 15, 2005 7:14:23.734000 PM	10	EXIT ==> Trace Message with huta	0	ReportSituation	com.ibm.logging.id....	testTraceParameter...	18
Jul 15, 2005 7:14:23.734000 PM	10	EXIT ==> Trace Message with huta	0	ReportSituation	com.ibm.logging.id....	testTraceParameter...	19
Jul 15, 2005 7:14:23.734000 PM	10	EXIT ==> Trace Message with huta	0	ReportSituation	com.ibm.logging.id....	testTraceParameter...	20
Jul 15, 2005 7:14:23.734000 PM	10	EXIT ==> Trace Message with huta	0	ReportSituation	com.ibm.logging.id....	testTraceParameter...	21

Symptom Analysis Results View - Jan 28, 2008 11:30:34 AM
Symptom Analysis Results (Page 1 of 2 Filter matched 118 of 118 records)

Creation time	Symptom
Jan 21, 2003 9:29:50 AM	ADMC0013I: SOAP connector available at port
Jan 21, 2003 9:11:56 PM	SRVE017II: Transport http is listening on port
Jan 16, 2003 11:41:54 PM	SRVE017II: Transport http is listening on port
Jan 25, 2003 10:07:29 PM	SRVE016II
Jan 27, 2003 3:31:10 PM	SRVE0169I
Jan 27, 2003 2:57:41 PM	SRVE0169I
Jan 27, 2003 10:31:04 AM	SRVE0169I
Jan 27, 2003 9:41:02 AM	SRVE0169I
Jan 27, 2003 4:24:30 AM	SRVE0169I
Jan 26, 2003 6:24:11 PM	SRVE0169I
Jan 26, 2003 5:17:50 PM	SRVE0169I
Jan 26, 2003 12:17:44 PM	SRVE0169I
Jan 26, 2003 6:27:36 AM	SRVE0169I
Jan 26, 2003 6:11:10 AM	SRVE0169I
Jan 26, 2003 3:41:04 AM	SRVE0169I
Jan 25, 2003 9:34:22 PM	SRVE0169I
Jan 25, 2003 6:14:21 PM	SRVE0169I
Jan 25, 2003 2:54:12 PM	SRVE0169I
Jan 25, 2003 1:31:10 PM	SRVE0169I
Jan 25, 2003 8:14:24 AM	SRVE0169I
Jan 25, 2003 7:57:36 AM	SRVE0169I
Jan 25, 2003 2:07:42 AM	SRVE0169I
Jan 25, 2003 12:27:45 AM	SRVE0169I
Jan 24, 2003 6:04:16 PM	SRVE0169I
Jan 24, 2003 3:01:02 PM	SRVE0169I
Jan 24, 2003 7:47:45 AM	SRVE0169I
Jan 24, 2003 6:57:43 AM	SRVE0169I
Jan 24, 2003 6:57:36 AM	SRVE0169I

Recommendations and actions
Lists all recommendations and actions associated with the selected symptom definition. Select an item in the table below to view its details.

Symptom effect details

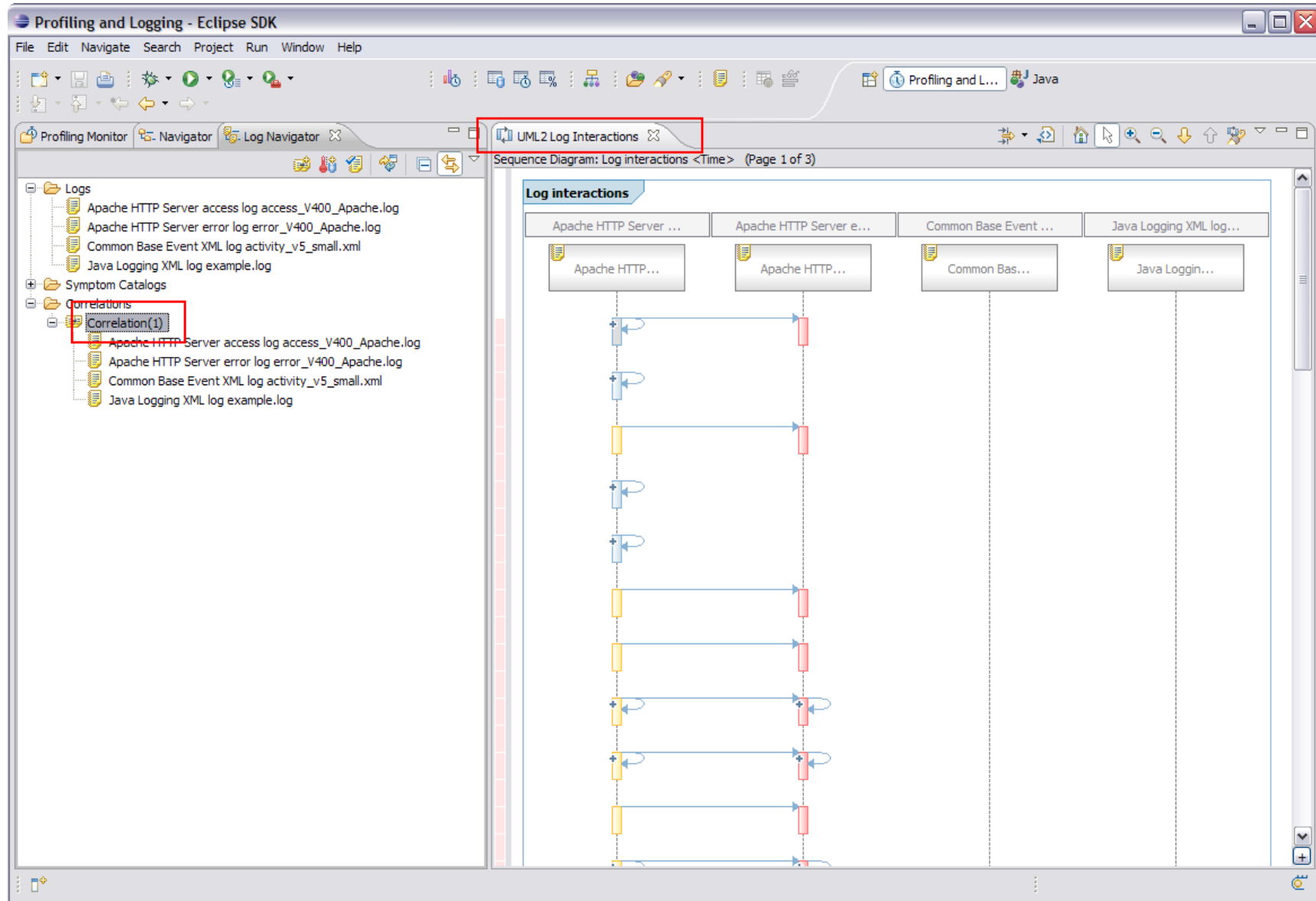
Recommendation
Recommendation details

Recommendation: * This message indicates the port the server uses to listen for administrative console requests. If it is different than the one specified in the URL for the admin console, change the URL in the browser to the correct value. The default value is <http://localhost:9090/admin>.

Monitoring - Correlation

- Computing system maintainers require a detailed understanding of an entire computing system in order to detect and resolve cascading problems.
- Correlation determines one or more sets of related events to visualize control flow within and between computing systems.
- A correlation engine or schema associates varying Common Base Event properties (log events) based on a predetermined criteria (e.g. time).
- Extensible architecture to allow users to define vendor and product specific correlation engines or schemas.

Monitoring - Correlation



Agenda

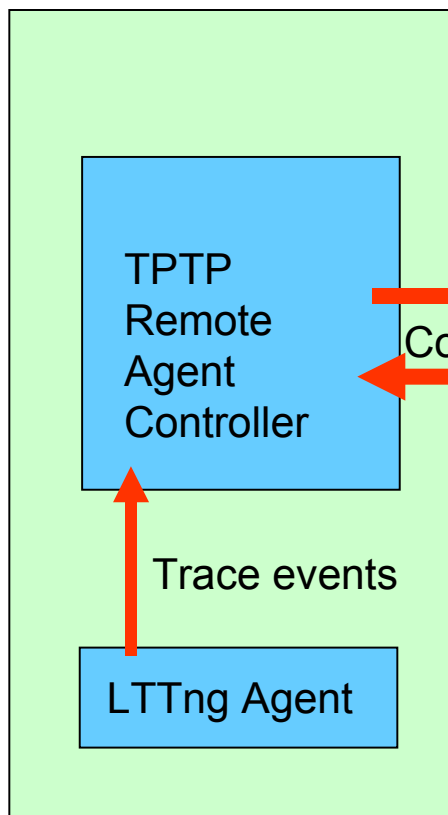
- ✓ Overview of TPTP
- ✓ TPTP Profiling tools
- ✓ TPTP Monitoring
- **Extending TPTP**
- Challenges
- What's new?



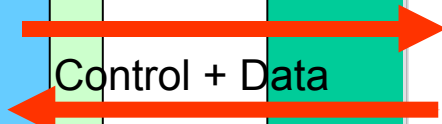
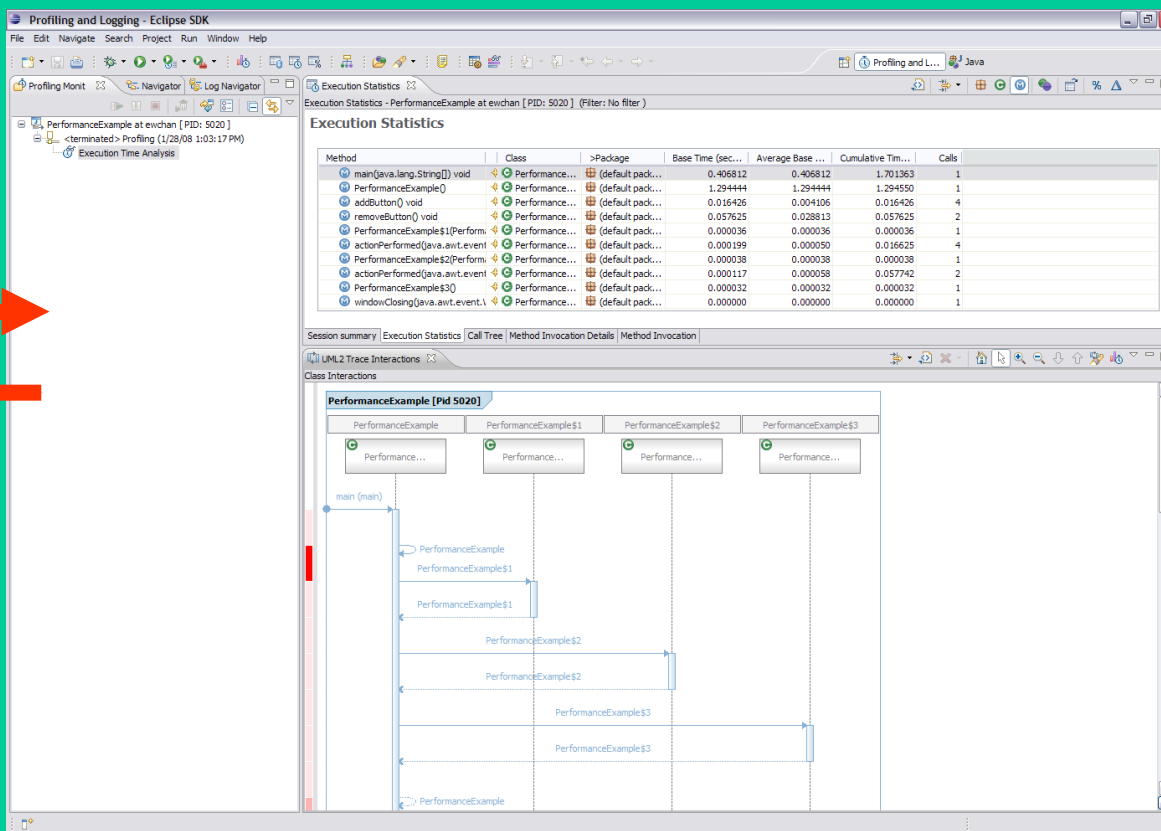
Extending TPTP - Profiling

Host B

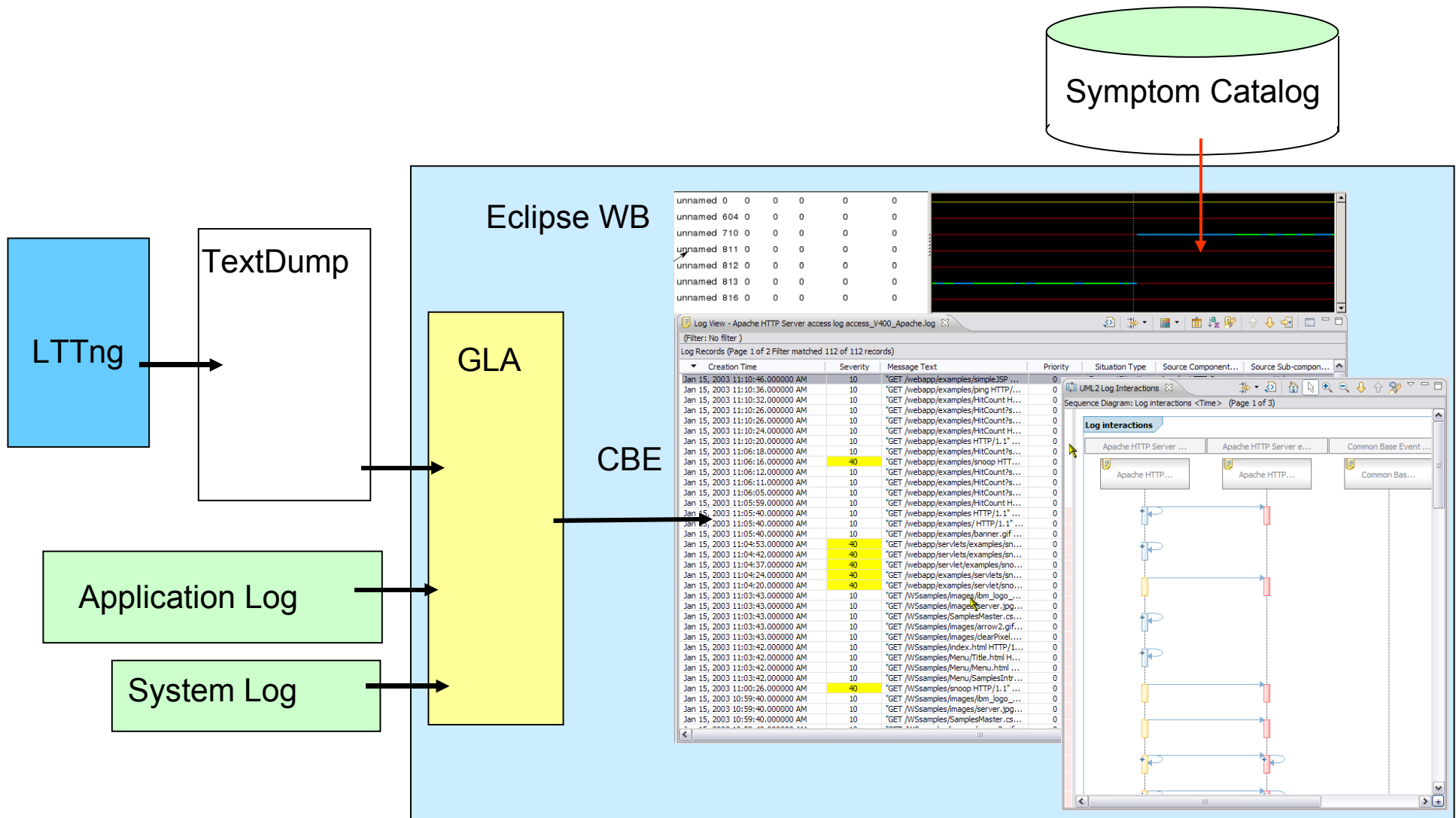
Host A



Control + Data

Extending TPTP - Monitoring



Extending TPTP

- TPTP defines extension points that allow consumer plug-ins to contribute behaviors to existing UI.

- Context Language Extension
 - Define a user interaction scenario which is applied for specific languages.
 - The content, description and actions for a profiling view can be customized based on the type of the profiled application
 - Custom columns can be added to the existing views to show language specific information
 - Custom implementation for the 'Open source' action which opens the source code on a specified resource

Extending TPTP



Profiling and Logging - Eclipse SDK

File Edit Navigate Search Project Run Window Help

Profiling Monitor Navigator

Execution Statistics - unknown at TPF System [PID: 1] (Filter: Default Filter)

Execution Statistics

>Function/Method	Compile Unit/Class	Module/Namespace	Base Time (seconds)	Average Base Time (seconds)	Cumulative Time (seconds)	Calls
@@QPN0A	QPN0	QPN0	0.037460	0.037460	0.040090	1
cinc	CTAL	CTAL	0.055900	0.055900	0.055900	1
exit	CISO	CISO	0.148420	0.148420	0.148420	1
func1	qpm1a	QPM1	0.001310	0.000164	0.001310	8
IPRSE_bldprstr	CTBX	CTBX	0.018590	0.018590	0.018590	1
sqrt	CISO	CISO	0.026890	0.026890	0.026890	1
srand	CISO	CISO	0.012030	0.012030	0.012030	1

Profiling and Logging - Eclipse SDK

File Edit Navigate Search Project Run Window Help

Profiling Monitor Navigator

Execution Statistics - Unknown at Unknown (Filter: Default Filter)

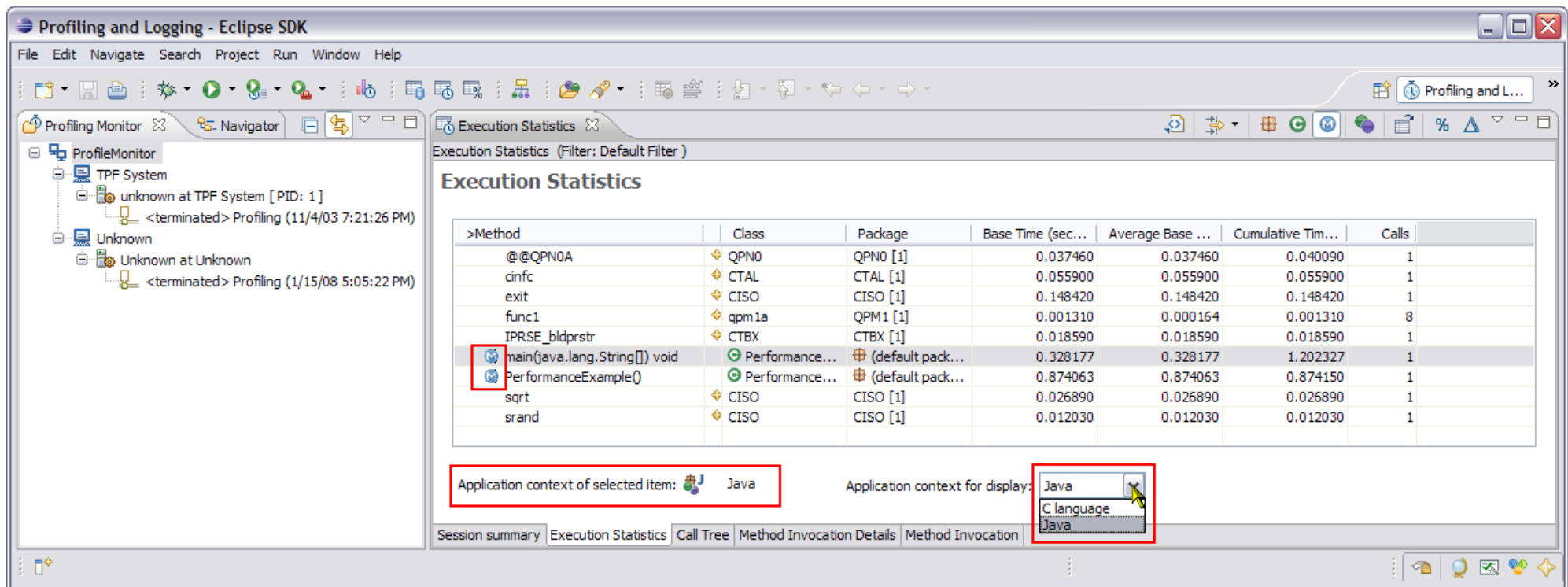
Execution Statistics

>Method	Class	Package	Base Time (sec...)	Average Base ...	Cumulative Tim...	Calls
main(java.lang.String[]) void	Performance...	(default pack...	0.328177	0.328177	1.202327	1
PerformanceExample()	Performance...	(default pack...	0.874063	0.874063	0.874150	1

Session summary Execution Statistics Call Tree Method Invocation Details Method Invocation

Extending TPTP

➤ Multiple contexts support



The screenshot shows the Eclipse Profiling and Logging window. The left pane displays the Profiling Monitor tree with a selected item. The main pane shows the Execution Statistics table, which lists methods, classes, packages, and timing data. A red box highlights the selected item in the tree, and another red box highlights the application context for display dropdown menu.

Execution Statistics (Filter: Default Filter)

>Method	Class	Package	Base Time (sec...)	Average Base ...	Cumulative Tim...	Calls
@@QPN0A	QPN0	QPN0 [1]	0.037460	0.037460	0.040090	1
cinc	CTAL	CTAL [1]	0.055900	0.055900	0.055900	1
exit	CISO	CISO [1]	0.148420	0.148420	0.148420	1
func1	qpm1a	QPM1 [1]	0.001310	0.000164	0.001310	8
IPRSE_bldprstr	CTBX	CTBX [1]	0.018590	0.018590	0.018590	1
main(java.lang.String[]) void	Performance...	(default pack...	0.328177	0.328177	1.202327	1
PerformanceExample()	Performance...	(default pack...	0.874063	0.874063	0.874150	1
sqr	CISO	CISO [1]	0.026890	0.026890	0.026890	1
srand	CISO	CISO [1]	0.012030	0.012030	0.012030	1

Application context of selected item: Java

Application context for display: Java

Extending TPTP

[org.eclipse.hyades.ui.contexts]

- Defines the language context so that custom content can be applied for this type of language.
- The contextKey attribute, which uniquely identifies the language type, is persisted into the data model.

```
<extension
  point="org.eclipse.hyades.ui.contexts">
  <context
    id="C/C++"
    contextKey="C/C++"
    name="%_1"
    description="%_2"
    icon="/icons/full/obj16/compileunit.gif">
  </context>
</extension>
```

[org.eclipse.hyades.ui.contextProviders]

- Defines the content to be displayed in the statistical views
- Icons, labels and views content can be customized using this extension point



Extending TPTP

[org.eclipse.hyades.ui.contextOpenSourceProviders]

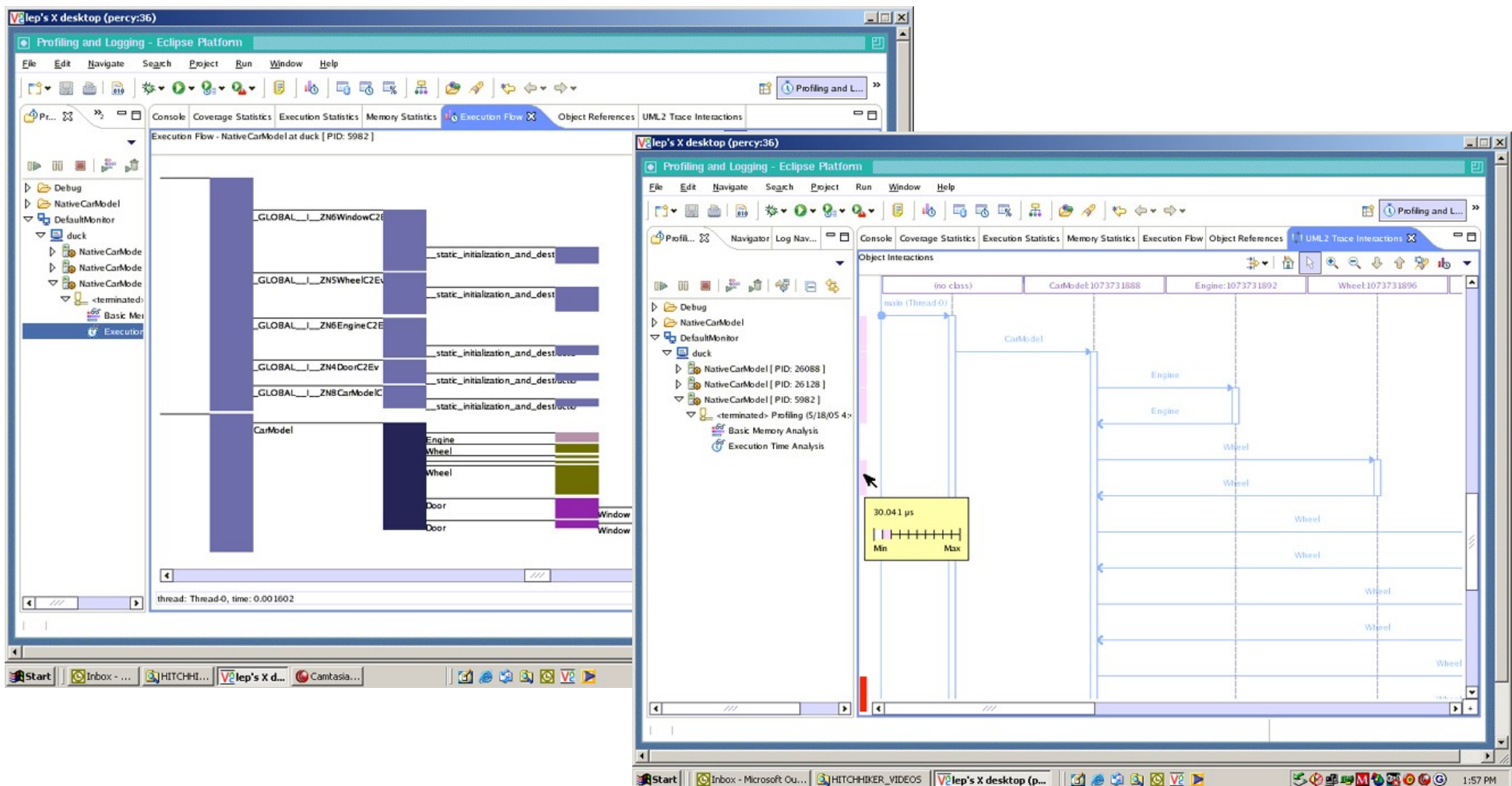
- Custom implementation for the 'Open source' action which opens the source code on a specified resource
- In the Web Services scenario, it can open the WSDL file defining the web service

➤ Trace event declares context language:

- `<traceStart traceId="1" agentIdRef="1" time="...." language="C/C++" />`
- Default language is Java.
- Mixture of languages is supported, language specified at object def level.

Extending TPTP

- Example: OC Systems Hitchhiker
 - Trace, profile or find memory leaks in any C/C++ application



Agenda

- ✓ Overview of TPTP
- ✓ TPTP Profiling tools
- ✓ TPTP Monitoring
- ✓ Extending TPTP
- **Challenges**
- What's new?



Challenges

- TPTP Profiler tool has been initially designed for profiling Java applications
 - Java centric data model
 - Host, Package, Class and Method hierarchy
 - Java centric visualization
 - new UI extensions since TPTP 4.1 for visualizing non-Java languages

- Size of Data
 - Filtering support – collection time and UI level
 - Profile to file

Agenda

- ✓ Overview of TPTP
- ✓ TPTP Profiling tools
- ✓ TPTP Monitoring
- ✓ Extending TPTP
- ✓ Challenges
- What's new?





What's New in TPTP 4.5

➤ Profiling Tools:

- Binary data format to increase profiler performance and scalability.
- Hover details for method invocations in the Method Statistics view.
- Improved profiling filters (filter sets).
- Secure profiling (JVMTI) including authentication and encryption.
- Simplified stand-alone profiling (JVMTI).
- Thread contention analysis for locating monitor and data contention.
- Support for Java 6 and IPv6 networks.

References

- TPTP home page
<http://www.eclipse.org/tptp/>
- TPTP download page
<http://www.eclipse.org/tptp/home/downloads/>
- TPTP documentation
<http://www.eclipse.org/tptp/home/documents/>
- TPTP tutorials and demos
http://www.eclipse.org/tptp/home/downloads/quicktour/v44/quick_tour/
- TPTP Eclipse corner article
<http://www.eclipse.org/articles/Article-TPTP-Profiling-Tool/tptpProfilingTool.html>

