

Lab 17314

**IBM PD Tools Hands-On Lab: Dive into Increased
Programmer Productivity**

**IBM Application Performance Analyzer
for z/OS**

Eclipse interface

Hands-on Lab Exercises

IBM Application Performance Analyzer for z/OS V13

Lab Exercises

© Copyright International Business Machines Corporation, 2015. All rights reserved.

This document is intended for IBM internal use only. The contents of the document may be discussed with existing and/or potential customers. The document itself, however, may not be distributed outside of IBM.

Contents

	<u>Page</u>
Overview	2
Lab 1: Getting started with the Application Performance Analyzer perspective	5
Lab 2: Analyzing the performance of an application	11

Overview

Application Performance Analyzer

IBM® Application Performance Analyzer for z/OS® is a tool that helps you perform application performance tuning for applications running on z/OS systems.

Application performance tuning is the process of minimizing the amount of time and system resources that your application uses when it runs. By system resources, it is typically referring to CPU time, and IO operations (such as reading and writing to files or databases).

When should you perform application performance tuning?

Most often, you will do application tuning when you have noticed a problem. For example, online transactions are running too long, and users are complaining about long response times. Or maybe a batch job that runs every day starts to run longer than usual, and does not complete in its scheduled time window.

It is probably most typical to do application performance tuning when you notice that production applications have problems, that is slow response times or batch jobs running too long. And the focus of the tuning effort should be to fix performance bottlenecks.

But it can also be advantageous to do performance tuning earlier in the application life cycle, during system testing. At this stage, you might be able to identify potential problems before they get to production. You can ensure that new applications meet your performance goals, and that changes that you have made will not have an adverse performance impact.

So who does performance tuning? There might be a lot of people who have roles in the application tuning process. And each can have a different perspective on the problem. An application developer might be more focused on the performance of a single application, or a set of applications. He or she might be interested in questions like, “Will a change make the program run slower?”. A Database Administrator might approach the issue from the perspective of how the database management system performs, and might be more interested in questions like “Can the database performance be improved?”. And a systems programmer might approach it from the perspective of the entire system, rather than individual applications, and might have interest in questions such as “Why is the entire system running slowly?”. As this chart implies, all of these people have overlapping interests. Questions like “Can the application be changed to run faster?” and “Can the job be changed to use less CPU and I/O” can be of interest to all. This is often best resolved through a collaborative effort.

APA’s focus is on the application tuning process. There are other monitors, Omegamon for example, that are designed to provide information across multiple applications or for an entire system. APA is designed to help you understand the detailed performance characteristics of a single application.

Some of the main features of Application Performance Analyzer are:

- APA can collect detailed performance information about an application while it runs. After it has observed an application running, it processes the collected information, and presents it in formatted displays and reports. Most people use APA by accessing it through its online interface from TSO.
- How does APA help you tune your application? The most important thing that APA does is to help you answer the question: “Where is the application spending its time?” For example, is it spending most of its time, using CPU cycles, reading or writing to files, or waiting on other

resources. And specifically, which programs or modules are causing the CPU or wait time. Based on the answer, you are better equipped to answer the question: “Can a change be made to the application or the system to make it run faster?”

- Here is how it works. First, you request an observation session. The easiest way to do it is to use the online interface to tell APA to monitor your application while it runs. You can either monitor an application that is running right now, or schedule an observation session or sessions at some time in the future. During an observation session, APA collects performance information while the application runs, and stores the data in one or more measurement data sets. The observation request itself is stored in a file called the checkpoint file.
- APA collects performance information using a sampling technique. When APA’s observation session begins, it takes one sample. That means that APA takes something like a snapshot of what the application is doing at one moment in time. It determines whether the application is actively using the processor or waiting for system resources, and it gets detailed information about the program, and even the instruction that is either using CPU time or waiting. APA retains the detailed information for that sample, and then waits for a fixed interval of time. Then it takes another sample, to find out what the application is doing at that moment, and collects information. For example, in this chart during the first two samples, the application was executing instructions in program A. During the third sample, it was executing instructions in a system module. During the next three samples, it was waiting for an I/O operation to complete. At any point in time, an application can only be in one of three states. It can be executing instructions in a program, which means it is using CPU time. It can be waiting for a system resource, such as for a file I/O operation to complete. Or, it can be waiting because the system CPU is busy servicing other processes. Over time, APA will collect a statistically significant set of samples. And when aggregated, it gives you detailed information about where the application spends most of its time.
- When you are using a monitor to collect performance information about an application, there is sometimes a concern about how much overhead the monitor has. First, be aware that APA runs in its own address space, so most of APA’s processing does not increase the resource usage of the application being monitored. Of course, APA does use some CPU cycles to do its work, although most typically it will have no measurable impact on the application. However, APA reports exactly how much CPU time it uses. System overhead from a single APA session is typically insignificant and does not skew measurement statistics. But if it is still a concern, you can control APA’s sampling rate.
- After an APA observation session has completed, the information is presented to you using analysis reports. You can view and print these reports. Most people view the reports directly using the APA online interface in TSO. Or you can print reports to SYSOUT, or even generate PDF files that you can download to your workstation and print from there.. APA generates its reports by aggregating data from the measurement data sets. Sample file data can also be written to a file in XML format and viewed in an XML reader.
- You can view a wide range of performance reports. Here are just a few. There are reports that describe CPU usage, and these reports are broken down by program or module, and you can even drill right down to the program statements that use the most CPU time. There are similar reports for wait time. Using these reports, you can identify which programs or modules caused the most wait time, and drill right down to the program statements. For DB2 applications, there are reports containing various metrics, broken down by SQL statement, by the plan, by the thread, and so on. For CICS applications, the information is broken down by transaction, by task, by program, and

much more. For IMS applications, there are reports broken down by CPU time and by call service time. And there are reports that provide information about disk usage and storage.

- One of the best features of APA is that you can drill down to program statement level, to identify exactly which statements caused the most CPU time and wait time. APA can perform this source mapping for the programs using the IBM compilers that is indicated on the slide. They include COBOL, PLI, C, C++, Java, and Assembler. To enable source mapping in APA, source information must be saved when your programs are compiled. Depending on the compiler, source information can be saved in the form of compiler listings, SYSDEBUG files, LANGX files, or assembler ADATA files.
- When you request an observation session, the entry is stored in the checkpoint file. The checkpoint entry contains information about the session, and is updated as the session starts, and completes.
- In a SYSPLEX environment, the checkpoint file can be on shared DASD. That way, you can request an observation session from one LPAR for an application that is running on another LPAR. Also, information in the checkpoint file is retained across IPLs. So you can request an observation session to run at some future time, and the request is retained even if the system is IPLed.
- Sampling data from observation sessions are stored in measurement data sets. Measurement data sets are allocated automatically by APA when an observation session starts. The names of the files are controlled by the person who installs APA. If you delete an observation session, the measurement data sets are also deleted automatically.

This workbook

This workbook contains instructions for lab exercises that are designed to give you hands-on experience for the eclipse interface of IBM Application Performance Analyzer for z/OS.

Reference

Product manuals and other information about IBM Application Performance Analyzer for z/OS, and other IBM problem determination tools, are available on the Web at URL:

<http://www.ibm.com/software/awdtools/deployment>

Lab Exercise 1

Getting started with the Application Performance Analyzer perspective

In this exercise you will:

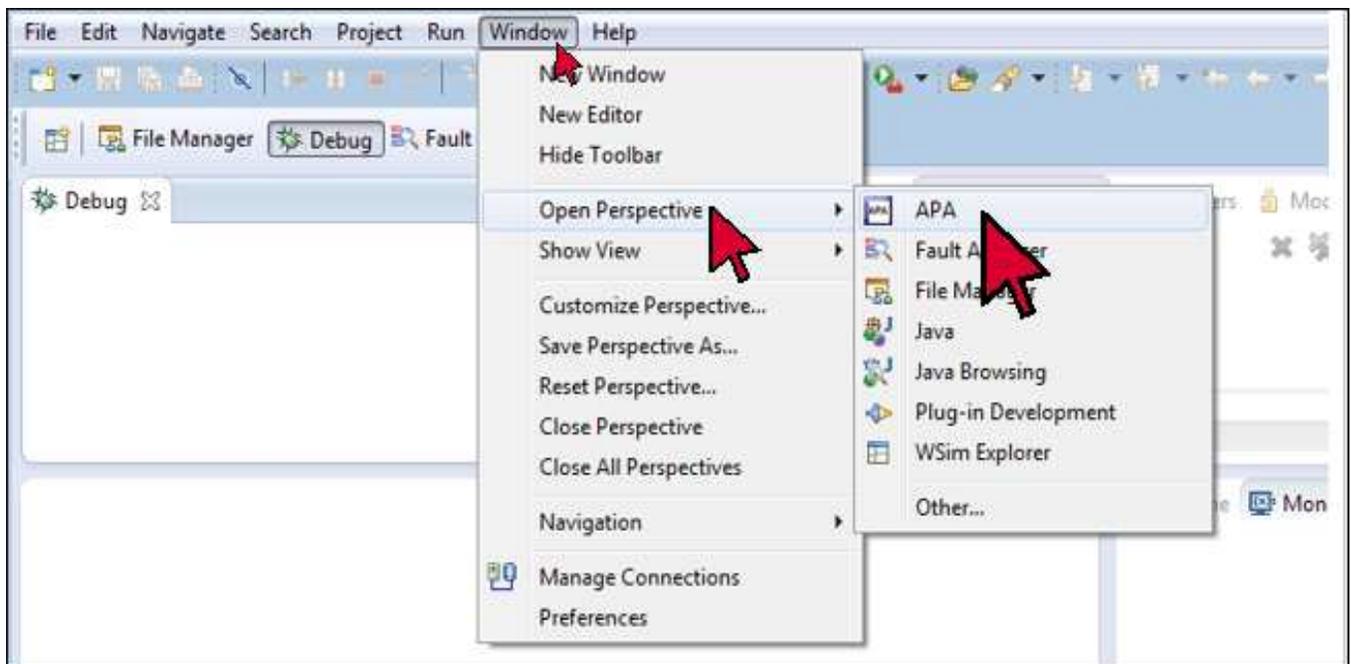
- Open the Application Performance Analyzer perspective in the Eclipse workbench.
- See the simple process of downloading reports from the host.
- Learn how to display help information for Application Performance Analyzer.
- Map source listing to see the lines of code that are suspect to performance.

1. Before you begin, you must have the Eclipse interface open on your workstation. If you aren't sure how to open it, please contact your instructor.

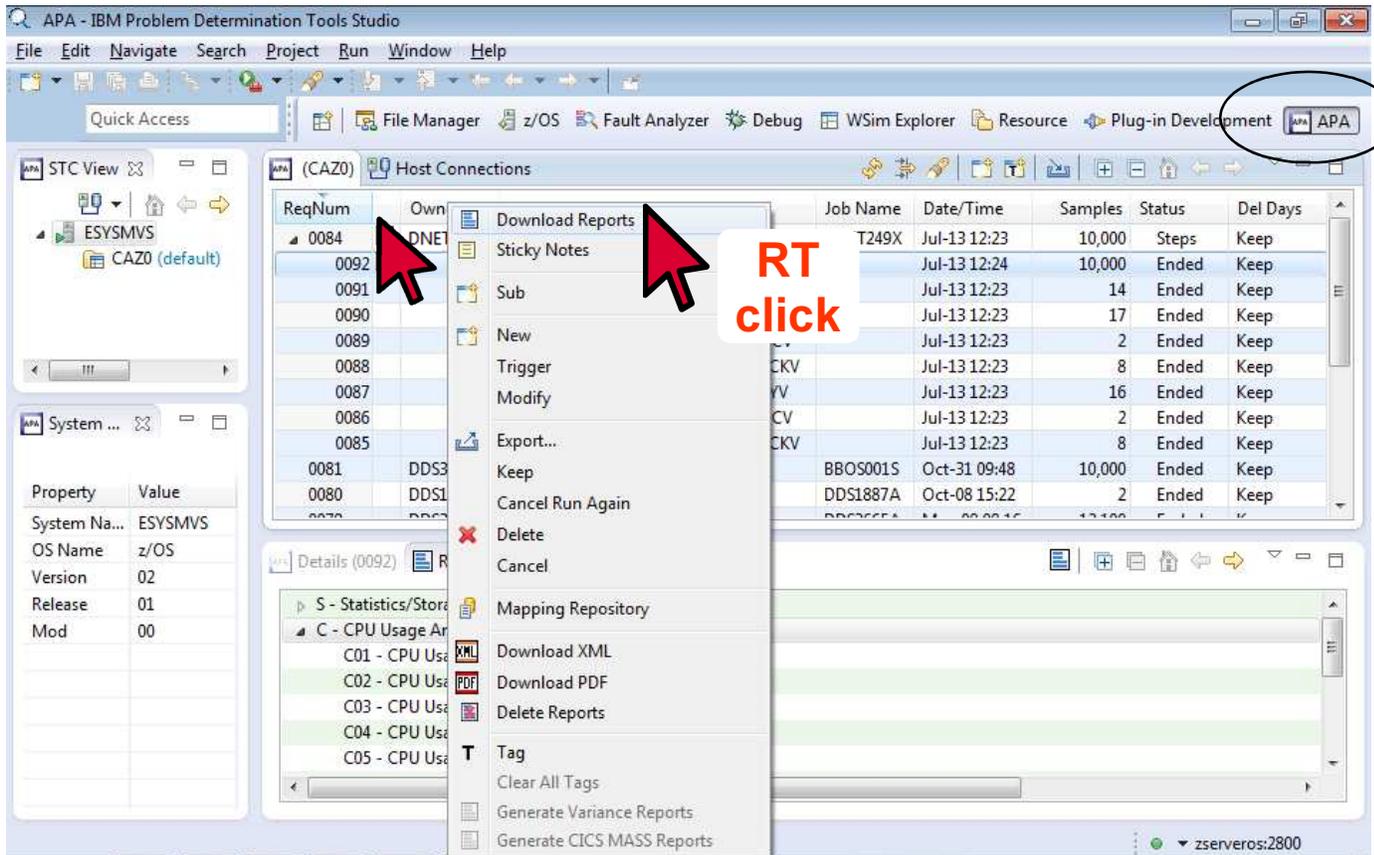
In eclipse, a perspective is a set of views (windows), menus, and options that provide a set of functions. The File Manager interface is a perspective. Before you can use File Manager, you need to open it's perspective.

2. Opening the Application Performance Analyzer perspective.

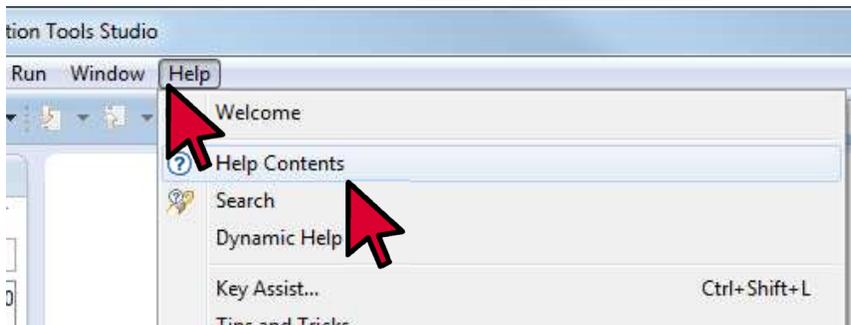
- a. From the menu bar near the top of the eclipse workbench, select **Window** > **Open Perspective** > **Application Performance Analyzer**.
 - i. Note: If "Application Performance Analyzer" is not shown as a selection, then the Application Performance Analyzer perspective is already open and you can proceed to the next step.



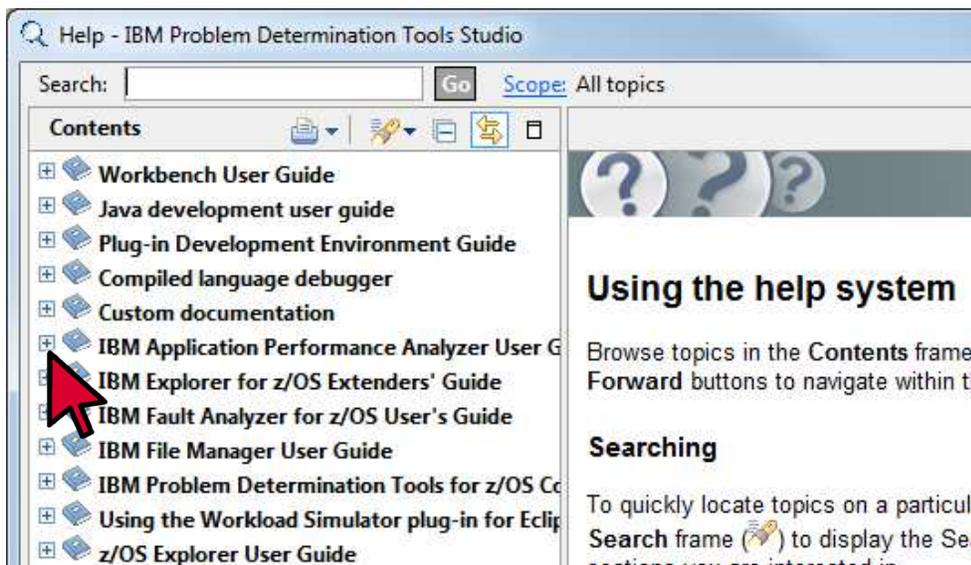
The Application Performance Analyzer perspective is displayed. Note: the contents of the Systems Information view may be different or even blank on your system.



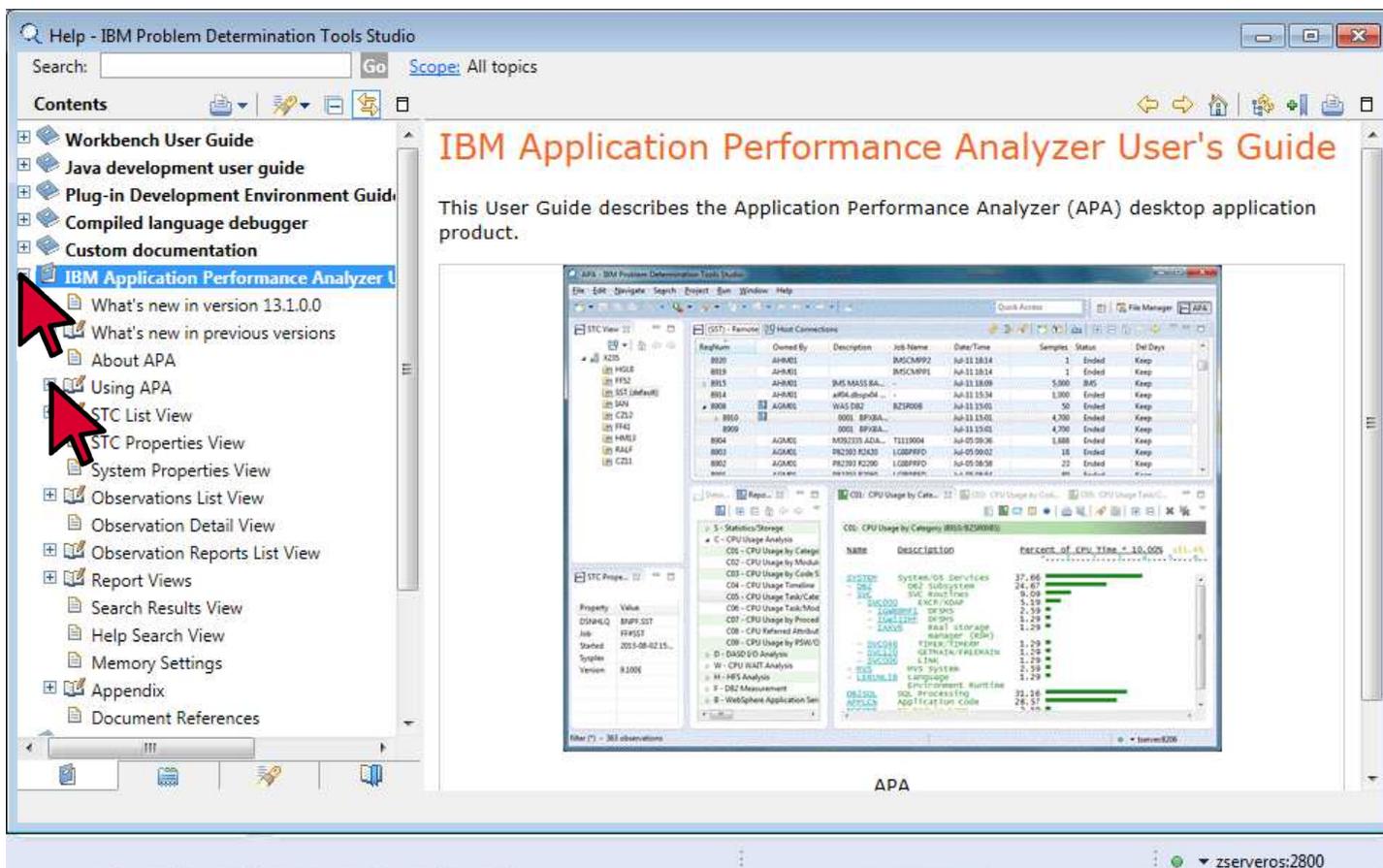
3. Notice that ReqNum 0084 was highlighted and expanded then a right mouse click reveals a pop-up window for downloading reports to your workstation.
4. You can display help information to assist with various features of Application Performance Analyzer.
 - a. To open the help dialog, select **Help > Help Contents**.



- b. The Help dialog is displayed. To expand the Application Performance Analyzer help topics, click the plus (+) next to IBM Application Performance Analyzer for z/OS User's Guide.

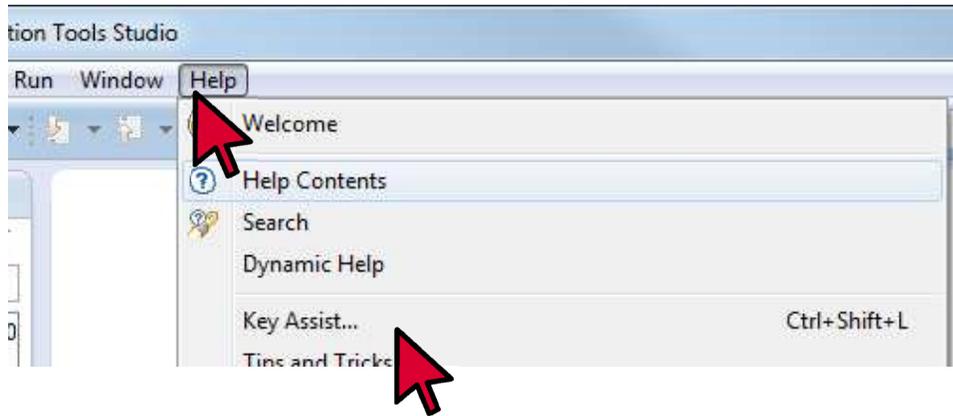


Under **IBM Application Performance Analyzer for z/OS User's Guide**, you can expand the **Using APA** section which covers several topics.

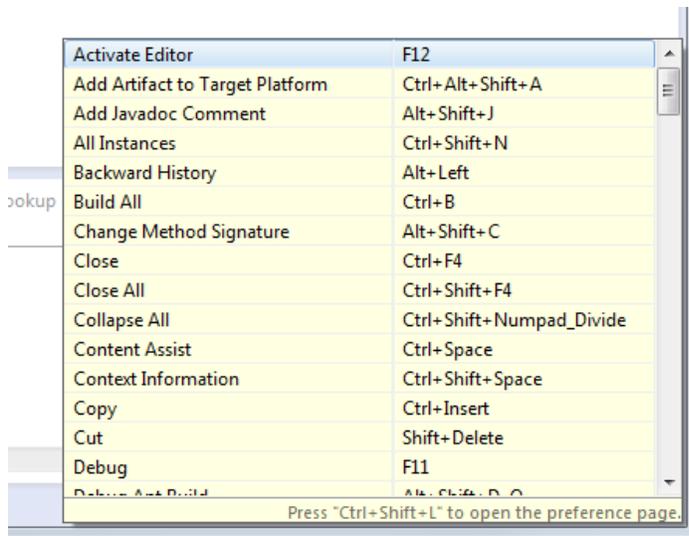


- You should now have a good understanding how to display general help information.
- Close the help dialog: click the **X** (close) icon in the upper right corner of the help window.

5. You can also display a list of keyboard commands.
 - a. In the eclipse workbench Select **Help** > **Key Assist**.



- b. A list of keyboard commands is displayed.



6. Note: As you work through the exercises, you may accidentally close or hide a view (a window) or other parts of the Application Performance Analyzer perspective. Remember that if that happens, you can easily reset the perspective to its default views. Try it now, so you know how to perform the reset:
 - From the menu, select **Window** > **Reset perspective**.
 - The perspective is reset. The default views are displayed in their default positions and with their default sizes.
7. Now that the Application Performance Analyzer perspective is open, and you know where to find help, you are ready to start using Application Performance Analyzer in the following exercises.

Note: In the next exercise you will be reviewing reports generated from a batch application. The APA observation was set-up to start when a specific batch application appeared active in an initiator. The observation included all steps for one CPU minute and collected 10,000 samples. The GUI APA made a request to the host to download all the reports to the workstation. Observations can be set-up for batch, online, etc. and can specify the rate of sampling, when to begin and on what step. Threshold observations can be used too. The next lab will focus on the reports that are generated.

You have completed this exercise.

Lab Exercise 2

Analyzing the performance of an application

In this exercise you will:

- See the CAZ0 report of reports which contain the following reporting areas:
 - a. S - Statistics/Storage
 - b. C - CPU Usage Analysis
 - c. D - DASD I/O Analysis
 - d. W - CPU WAIT Analysis
 - e. G - Coupling Facility
 - f. J – Java Measurement
- Each category can be drilled down into sub categories.
- You will see the program source mapping against the reports.
- The program source mapping will provide indicators for the category and the source the program spent the most time in the lines highlighted.

Introduction to Application Performance Analyzer Reports

Overview of reports.

Overview of categories.

Overview of source mapping and reporting.

1. In the Systems Information Tab, under the z/OS system that you will use during this exercise, select **Application Performance Analyzer for z/OS > ESYSMVS > CAZO**



2. Here's an example of a typical APA Observations List (CAZO) – Local. The notes below point out color coding and status states.

Legend:

- Red: waiting to monitor
- Yellow: monitoring is active or did not collect the requested number of samples
- Black: monitoring ended

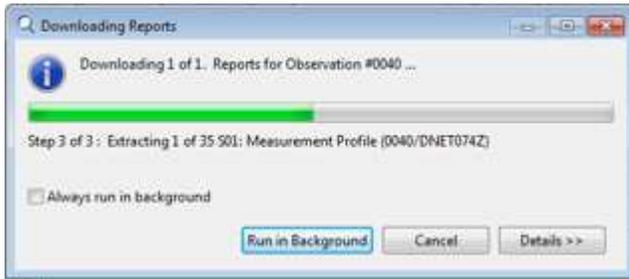
Status Definitions:

- Active:** currently monitoring
- Cancel:** cancelled by a user
- Ended:** monitoring has completed
- Ended+:** monitoring has completed, but may repeat
- Repeat:** repeating schedule request
- Sched:** scheduled, but measurement has not yet started
- Steps:** multi-step request
- Thresh:** threshold monitor request
- MultJb:** multiple job request

Job Name	Samples	Status	Date/Time	Del Days
PRO055	10,000	Thresh	Nov-21 19:56	Keep
PRO005	10,000	REPEAT	Nov-21 19:48	Keep
DNET074Z	8,537	Ended	Nov-21 19:31	Keep
DNET074Y	8,540	Ended	Nov-21 19:31	Keep
DNET074H	10,000	Steps	Nov-21 19:11	Keep
DNET074G	10,000	Ended	Nov-21 19:07	Keep
DNET074F	10,000	Steps	Nov-21 18:45	Keep
DNET074E	10,000	Steps	Nov-21 18:40	Keep
QCSCAORS	10,000	Ended	Nov-21 16:40	Keep
DNET074B	3,844	Ended	Nov-15 08:54	Keep
DNET074A	10,000	Steps	Nov-15 07:26	Keep
HRKDREP2	163	Ended	Oct-28 22:09	Keep
DNET187C	20,000	Ended	Oct-25 13:02	Keep
DNET187C	20,000	Steps	Oct-25 12:51	Keep
HRKDREP2	220	Ended	Oct-23 10:13	Keep
DNET461X	10,000	Steps	Oct-21 14:52	Keep
QCSCAORS	25,000	Ended	Oct-21 13:10	Keep

filter (*) - 34 observations

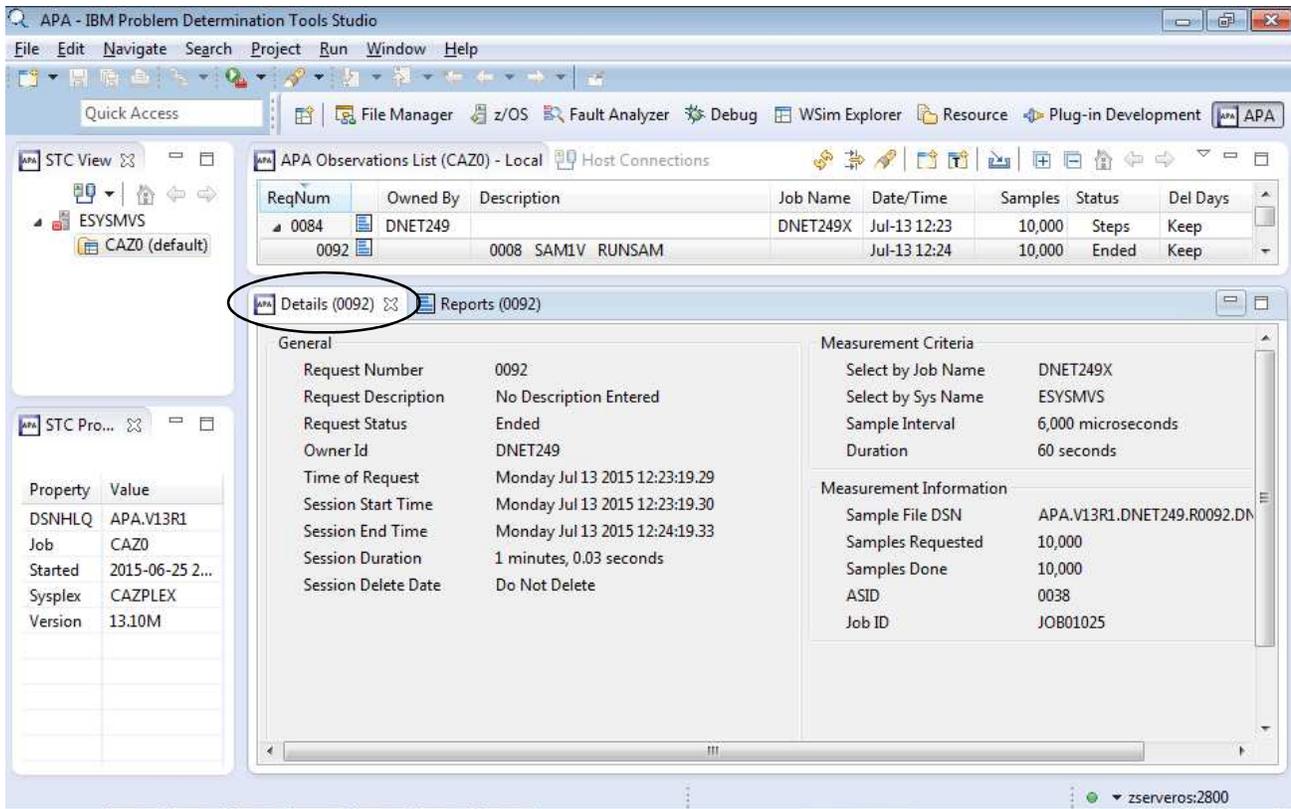
3. We already set-up an observation in APA. Once the specific batch job runs, reports are available.
 - a. We discussed how in the GUI APA you can download reports from the host. This is a typical pop-up screen showing the progress of the download.



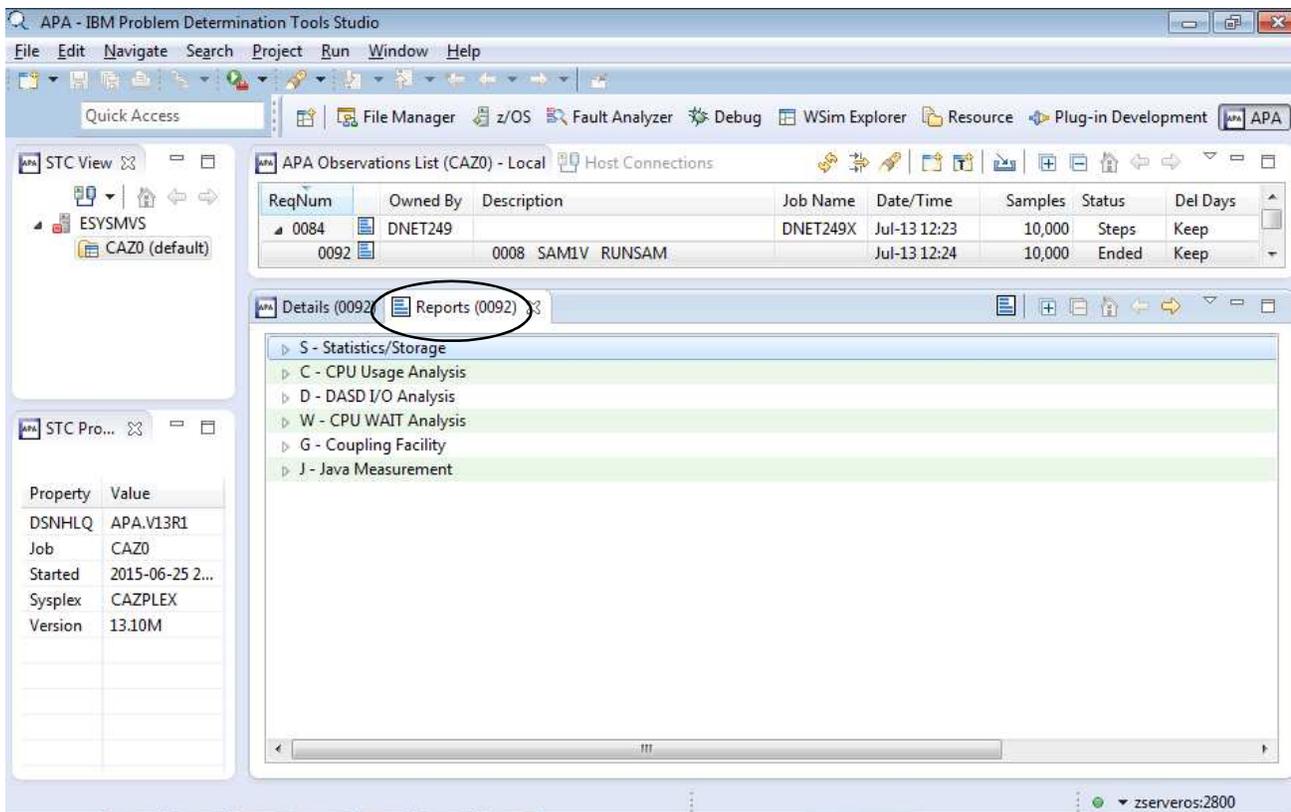
- b. The job we observed had multiple steps. Our concern is in step 8. Below, you can see the ReqNum is 0084. The owner, jobname and date/time are also available. ReqNum 0092 is the multi-step reference.

ReqNum	Owned By	Description	Job Name	Date/Time
0084	DNET249		DNET249X	Jul-13 12:23
0092		0008 SAM1V RUNSAM		Jul-13 12:24
0091		0007 IDCAMS VERIFY		Jul-13 12:23
0090		0006 IDCAMS CUSTKSDS COPYV		Jul-13 12:23
0089		0005 IEFBR14 CUSTKSDS ALLOCV		Jul-13 12:23
0088		0004 IKJEFT01 CUSTKSDS CHECKV		Jul-13 12:23
0087		0003 IDCAMS CUSTKSDS COPYV		Jul-13 12:23
0086		0002 IEFBR14 CUSTKSDS ALLOCV		Jul-13 12:23
0085		0001 IKJEFT01 CUSTKSDS CHECKV		Jul-13 12:23

4. Details (0092) tab shows General observation information.



5. Reports (0092) shows the breakdown of the report categories.



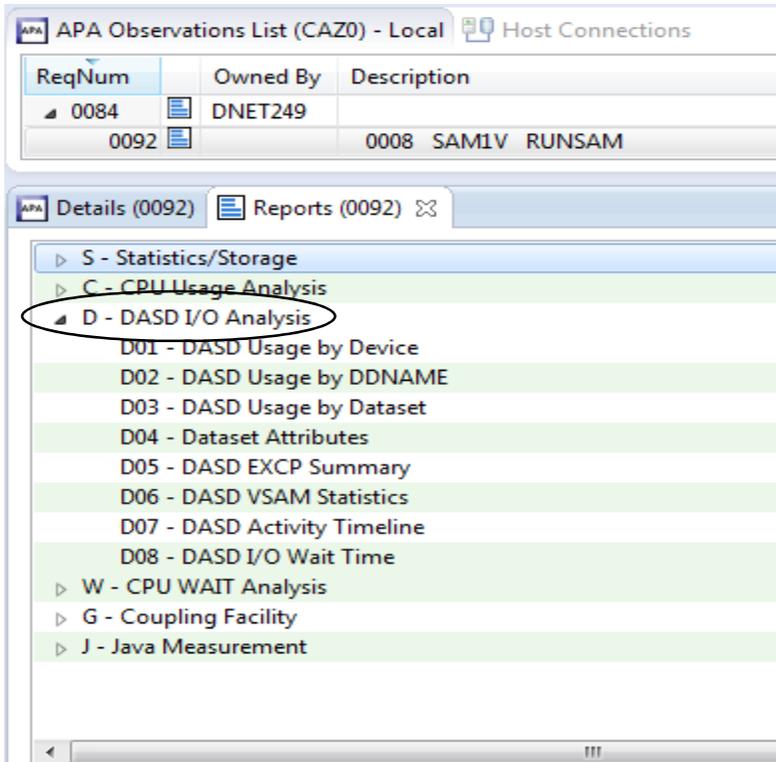
6. Reports (0092) shows the breakdown of the report categories and sub-categories for Statistics/Storage.

The screenshot shows the 'APA Observations List (CAZ0) - Local' window. At the top, there is a table with columns 'ReqNum', 'Owned By', and 'Description'. The table contains two rows: one for ReqNum 0084 (Owned By DNET249) and one for ReqNum 0092 (Description 0008 SAM1V RUNSAM). Below the table, there are two tabs: 'Details (0092)' and 'Reports (0092)'. The 'Reports (0092)' tab is active, showing a tree view of report categories. The 'S - Statistics/Storage' category is selected and circled in red. Underneath it, a list of sub-categories is shown: S01 - Measurement Profile, S02 - Load Module Attributes, S03 - Load Module Summary, S04 - TCB Summary, S05 - Memory Usage Timeline, S06 - Data Space Usage Timeline, S07 - TCB Execution Summary, S08 - Processor Utilization Summary, S09 - Measurement Analysis, C - CPU Usage Analysis, D - DASD I/O Analysis, W - CPU WAIT Analysis, G - Coupling Facility, and J - Java Measurement.

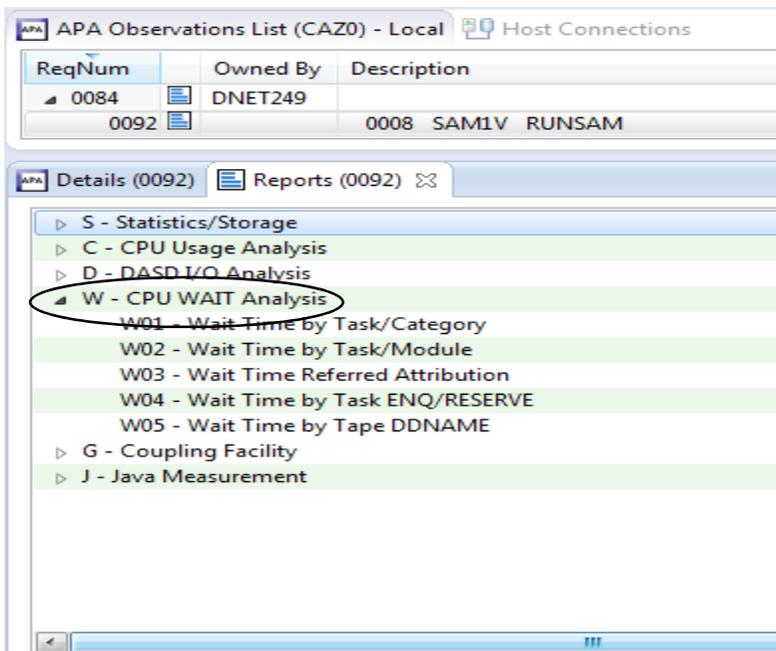
7. Reports (0092) shows the breakdown of the report categories and sub-categories for CPU Usage Analysis.

The screenshot shows the 'APA Observations List (CAZ0) - Local' window, identical to the one above. The 'Reports (0092)' tab is active, and the 'C - CPU Usage Analysis' category is selected and circled in red. Underneath it, a list of sub-categories is shown: C01 - CPU Usage by Category, C02 - CPU Usage by Module, C03 - CPU Usage by Code Slice, C04 - CPU Usage Timeline, C05 - CPU Usage Task/Category, C06 - CPU Usage Task/Module, C07 - CPU Usage by Procedure, C08 - CPU Referred Attribution, C09 - CPU Usage by PSW/ObjCode, D - DASD I/O Analysis, W - CPU WAIT Analysis, G - Coupling Facility, and J - Java Measurement.

8. Reports (0092) shows the breakdown of the report categories and sub-categories for DASD I/O Analysis.



9. Reports (0092) shows the breakdown of the report categories and sub-categories for CPU WAIT Analysis.



10. Reports (0092) shows the breakdown of the report categories and sub-categories for Coupling Facility.

ReqNum	Owned By	Description
0084	DNET249	
0092		0008 SAM1V RUNSAM

APA Details (0092) Reports (0092)

- ▷ S - Statistics/Storage
- ▷ C - CPU Usage Analysis
- ▷ D - DASD I/O Analysis
- ▷ W - CPU WAIT Analysis
- ▷ **G - Coupling Facility**
 - G01 - Coupling Facility Summary
 - G02 - Coupling Facility Mean Times
 - G03 - Coupling Facility Total Times
- ▷ J - Java Measurement

11. Reports (0092) shows the breakdown of the report categories and sub-categories for Java Measurement.

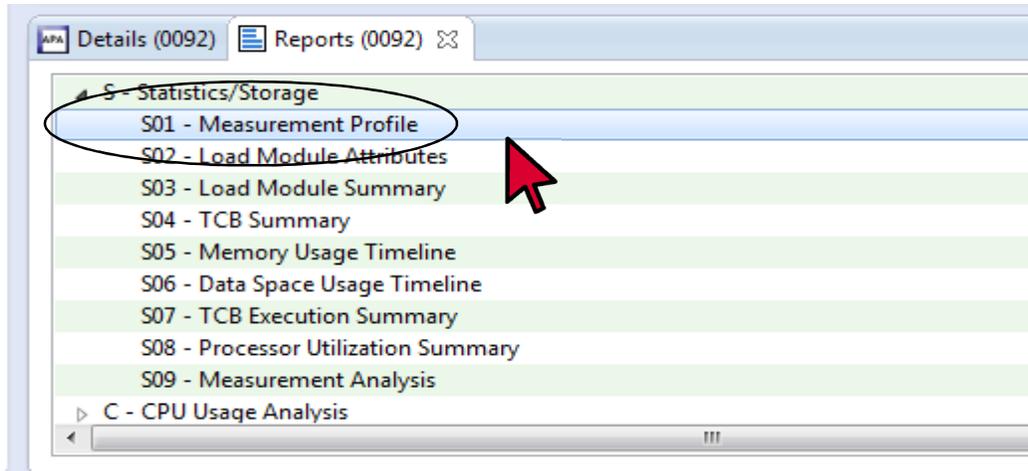
ReqNum	Owned By	Description	Job Name
0084	DNET249		DNET249X
0092		0008 SAM1V RU...	
0091		0007 IDCAMS V...	
0090		0006 IDCAMS C...	
0089		0005 IEFBR14 C...	
0088		0004 IKJEFT01 C...	
0087		0003 IDCAMS C...	
0086		0002 IEFBR14 C...	
0085		0001 IKJEFT01 C...	
0081	DDS3665		BBOS001S
0080	DDC1997		DDC1997A

APA (CAZ0) - Remote Host Connections

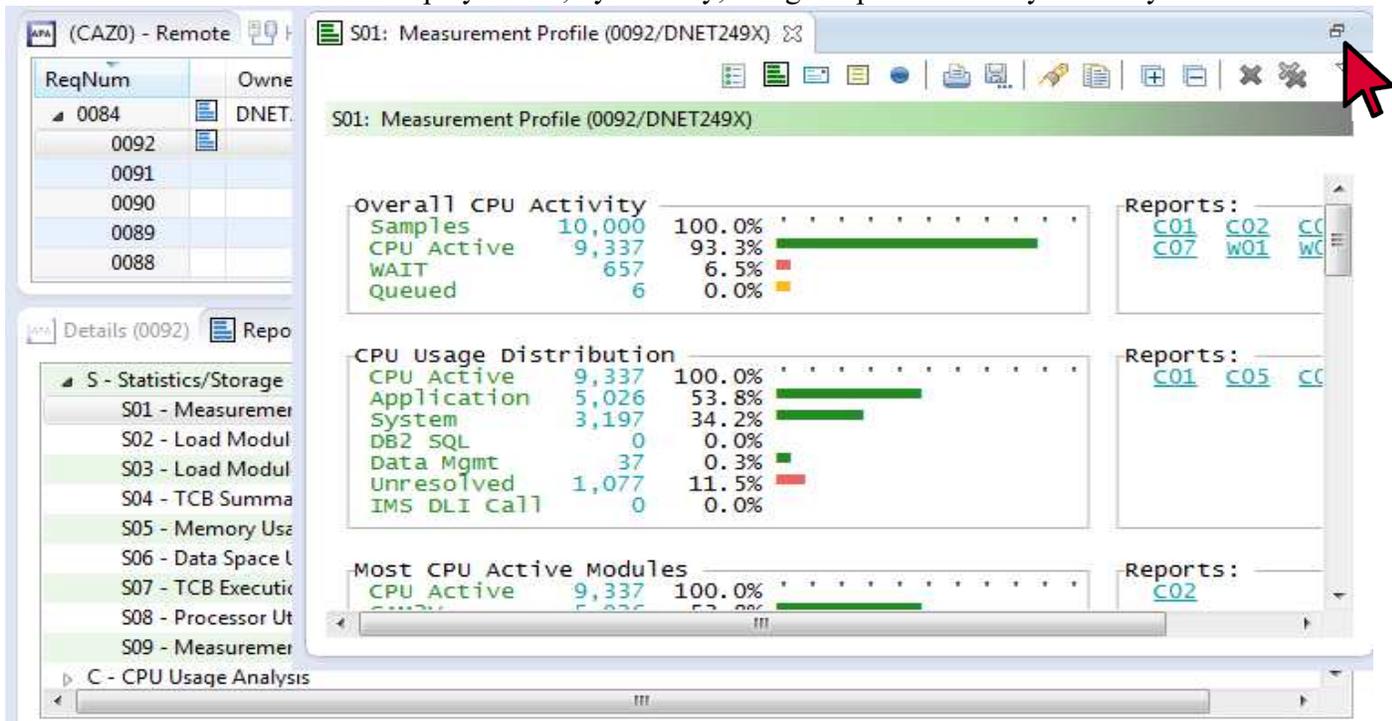
APA Details (0092) Reports (0092)

- ▷ S - Statistics/Storage
- ▷ C - CPU Usage Analysis
- ▷ D - DASD I/O Analysis
- ▷ W - CPU WAIT Analysis
- ▷ G - Coupling Facility
- ▷ **J - Java Measurement**
 - J01 - Java Summary/Attributes

12. Returning to the Statistics/Storage report categories and then clicking on the S01 report...



13. The S01: Measurement Profile is displayed and, by the way, is a good place to start your analysis.



- a. In the upper right hand of the S01 report there is a full screen button. You can simply left mouse button click the full screen and then use the page up/down buttons or the scrolling slide bar to navigate the reports. The blocks on the left side present the measurement information gathered and the blocks on the right side contain a list of related reports. You can simply select a report and left mouse click it to drill down into the associated report. Let's scroll all the way to the bottom of the S01 report to see what's available and associated.
- b. The Overall CPU Activity gives you APA's analysis of where the application spent time:
 - i. CPU Active.
 - ii. WAIT.
 - iii. QUEUED.

- c. With the S01 report filling the screen, use the mouse to hold and scroll the page.
- d. Notice much of the time is in application modules

The screenshot displays the APA - IBM Problem Determination Tools Studio interface. The main window shows a report titled "S01: Measurement Profile (0092/DNET249X)". The report is organized into several sections, each with a table of data and a corresponding bar chart. A red mouse cursor is pointing at the vertical scrollbar on the right side of the report area, indicating that the report is being scrolled.

Overall CPU Activity

Samples	10,000	100.0%
CPU Active	9,337	93.3%
WAIT	657	6.5%
Queued	6	0.0%

Reports: [C01](#) [C02](#) [C03](#) [C05](#)
[C07](#) [W01](#) [W02](#)

CPU Usage Distribution

CPU Active	9,337	100.0%
Application	5,026	53.8%
System	3,197	34.2%
DB2 SQL	0	0.0%
Data Mgmt	37	0.3%
Unresolved	1,077	11.5%
TMS DLI Call	0	0.0%

Reports: [C01](#) [C05](#) [C08](#) [W01](#)

Most CPU Active Modules

CPU Active	9,337	100.0%
SAM2V	5,026	53.8%
IGZCPAC	3,173	33.9%
IDA019L1	23	0.2%
ICYRDWR	16	0.1%
IGG0CLHA	8	0.0%

Reports: [C02](#)

Most CPU Active CSECTS

CPU Active	9,337	100.0%
SAM2V in SAM2V	5,026	53.8%
IGZCXDI in IGZCPAC	3,173	33.9%
PTFRDWR in ICYRDWR	15	0.1%
IDA019C1 in IDA019L1	9	0.0%
IDA019R3 in IDA019L1	7	0.0%

Reports: [C02](#)

CPU Modes

Active CPU	9,337	100.0%
------------	-------	--------

Reports: [S08](#)

e. After scrolling, the complete section on CPU Modes is available.

APA - IBM Problem Determination Tools Studio

File Edit Navigate Search Project Run Window Help

File Manager z/OS Fault Analyzer Debug APA WSim Explorer Resource Plug-in Development

S01: Measurement Profile (0092/DNET249X)

S01: Measurement Profile (0092/DNET249X)

CPU Modes

Active CPU	9,337	100.0%	
Supv Mode	36	0.3%	
Prob Mode	9,301	99.6%	
In SVC	36	0.3%	
AMODE 24	0	0.0%	
AMODE 31	9,333	99.9%	
AMODE 64	4	0.0%	
User Key	9,301	99.6%	
System Key	36	0.3%	

Reports:

[S08](#)

Request parameters

Request number	0092
Description	Unspecified
Sample file DSN	APA.V13R1.DNET249.R0092.DNET249X.SF
Retention	Retain until deleted by user
Data extractors	CICS,IMS,DB2,MQ,DB2+,IMS+,JAVA,DB2V,ADA,NAT,CDB2,...

Requesting user	DNET249	Nbr of samples	10,000
Time of request	12:23:19	Duration	60 sec
Date of request	Mon Jul-13-2015	Active/pending	Pending
Job name	DNET249X	Proc step name	n/a
Step name/number	n/a	Delay time	none
Step program	n/a		

Measurement environment

Job name	DNET249X	Region size <16MB	4,160K
Job number	JOB01025	Region size >16MB	32,768K
Step name	RUNSAM	Step program	SAM1V
Proc step name		Region type	Batch
ASID	56	DB2 Attach type	n/a

System ID	ESYSMVS	IBM APA Version	13.10M
SMFID	MVSE	DB2 subsystem name	
O/S level	z/OS 02.01.00	IMS system id	

0:0

f. And the last page of the S01 report is displayed.

The screenshot shows the IBM Problem Determination Tools Studio (APA) interface. The main window displays the S01 Measurement Profile report for job DNET249X. The report is organized into three main sections: Measurement environment, Measurement statistics, and CPU consumption. A red mouse cursor is pointing to the right side of the window.

Measurement environment

Job name	DNET249X	Region size <16MB	4,160K
Job number	JOB01025	Region size >16MB	32,768K
Step name	RUNSAM	Step program	SAM1V
Proc step name		Region type	Batch
ASID	56	DB2 Attach type	n/a
System ID	ESYSMVS	IBM APA Version	13.10M
SMFID	MVSE	DB2 subsystem name	
O/S level	z/OS 02.01.00	IMS system id	
General CPUs	5	CPU model	2817
Specialty CPUs	1	CPU rate factor	294
MIPS per CPU	1,122	SUS per second	54421.7

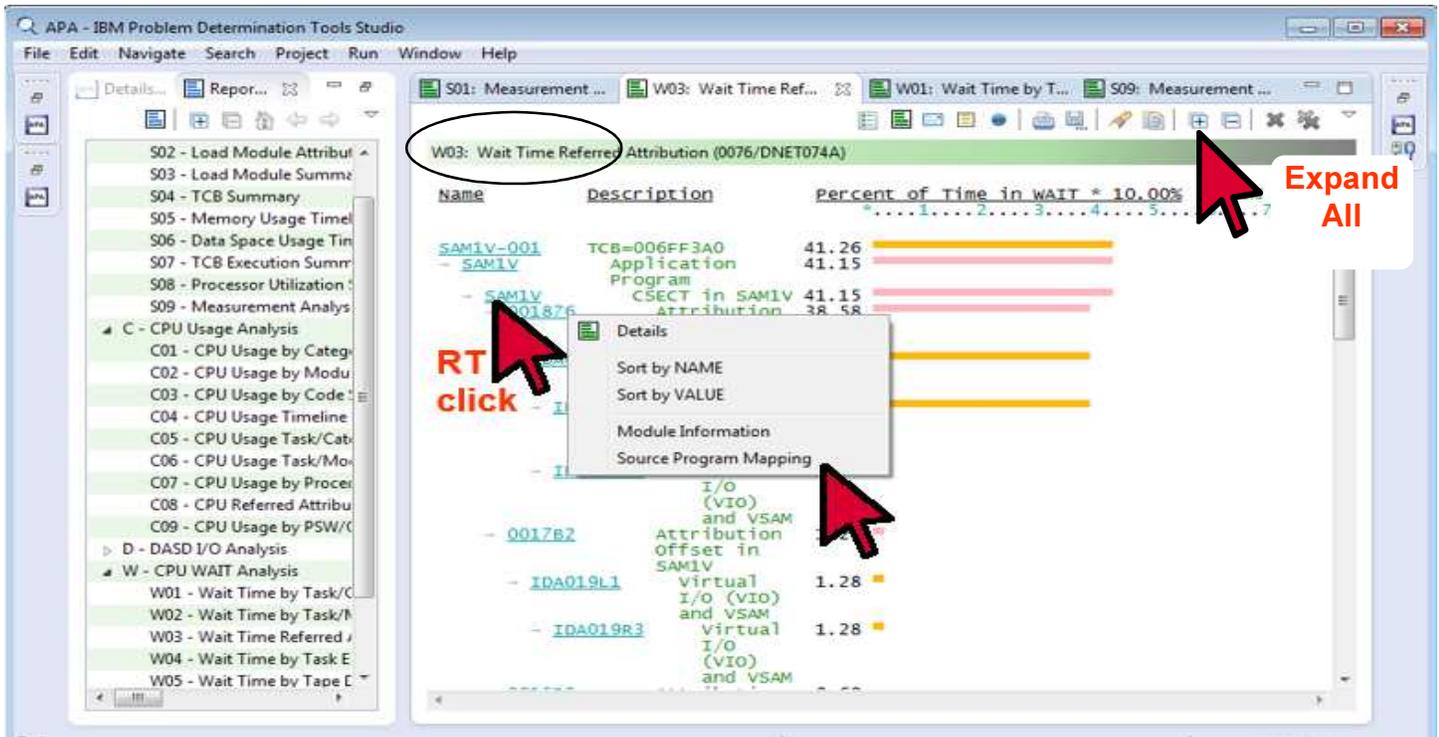
Measurement statistics

Start time	12:23:19	Start date	Mon Jul-13-2015
End time	12:24:19	End date	Mon Jul-13-2015
Total samples	10,000	Duration	1 min 0.02 sec
Sampling rate	166.61 per sec	Report dataspace	3.19MB
CPU/WAIT samples	9,994	Sample dataspace	6.00MB
TCB samples	10,000	Meas significance	99.94%
CPU queued samples	6		
Pages in	0	Overall CPU	23.24%
Pages out	0	Overall ZAAP CPU	0.00%
EXCPs	5,923	Overall ZIIP CPU	3.29%

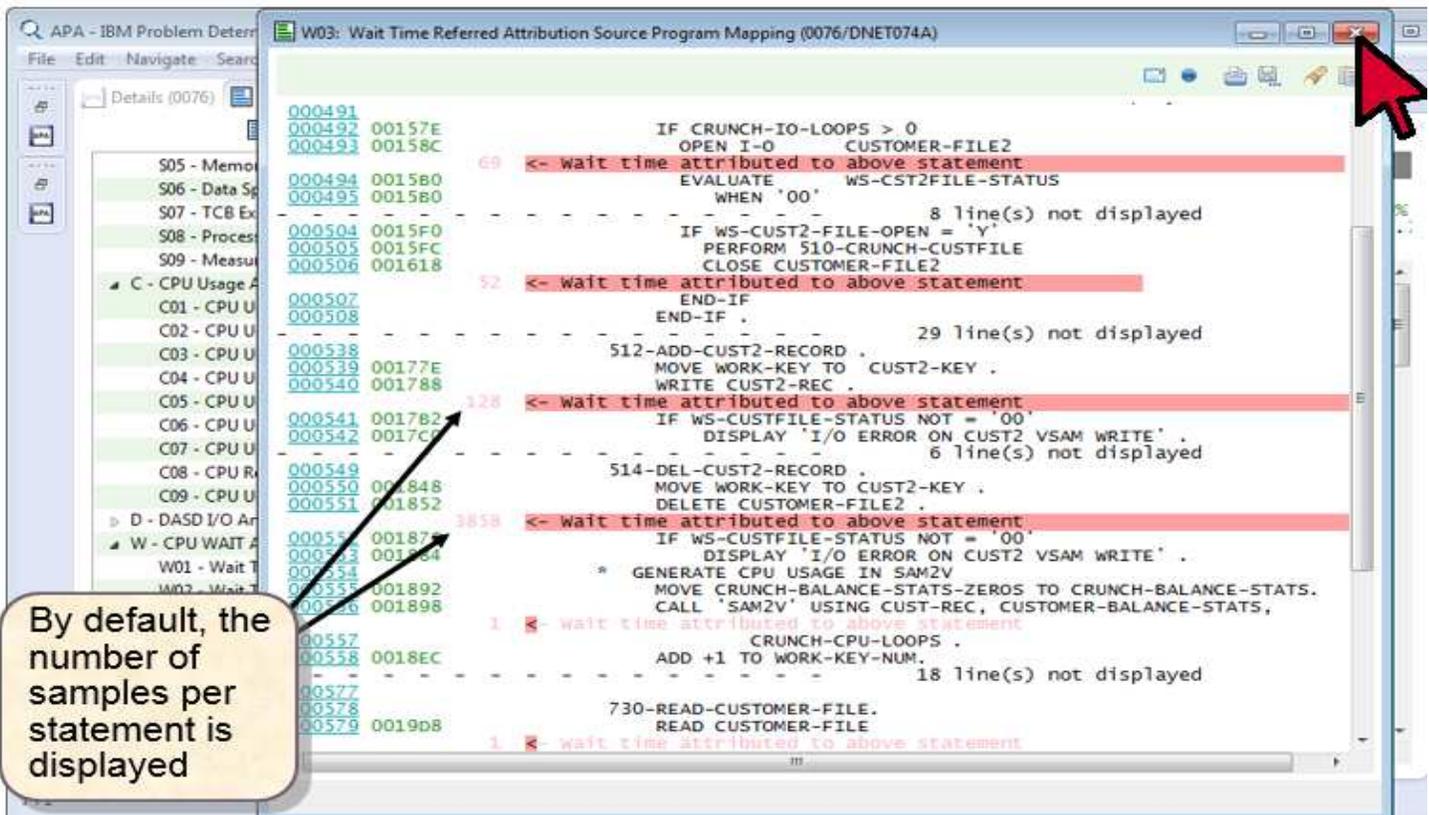
CPU consumption

CPU active samples	9,337	CPU time TCB	53.48 sec
CPU active time	93.37%	CPU time SRB	0.20 sec
CPU WAIT samples	657	Service Units	2,921,356
CPU WAIT time	6.57%	Measurement SRB	0.38 sec

- Open the W03 report and expand all the names with the “plus” icon. Right click the SAM1V CSECT. Then right click the Source Program Mapping.



- Source for the program is displayed on the W03 report now, showing which statements caused the most wait time.



16. The C01 report and source mapping shows the statements that caused the most CPU consumption.

LineNo	Offset	Prcnt	Source Statement
S02	000093		050-CALC-BALANCE-STATISTICS.
S03	000094		MOVE 0 TO LOOP-COUNT.
S04	000095	1.75	PERFORM 100-CRUNCH-LOOP
S05	000096		UNTIL LOOP-COUNT > CRUNCH-CPU-LOOPS .
S06	000097		
S07	000098		100-CRUNCH-LOOP.
S08	000099	.42	MOVE 'CALCULATING BALANCE STATS' TO WS-PROGRAM-STATUS.
S09	000100		*** Increment Record Count ***
S10	000101	5.39	ADD +1 TO BALANCE-COUNT
S11	000102		*** Add this customer's BALANCE to the grand total ***
S12	000103	13.86	COMPUTE BALANCE-TOTAL =
S13	000104		BALANCE-TOTAL + CUST-ACCT-BALANCE
S14	000105		*** Calculate Average ***
S15	000106	19.04	COMPUTE BALANCE-AVERAGE =
S16	000107	40.68	BALANCE-TOTAL / BALANCE-COUNT
S17	000108		*** Calculate Minimum ***
S18	000109	.26	IF WS-FIRST-TIME-SW = 'Y'
S19	000110		MOVE CUST-ACCT-BALANCE TO BALANCE-MIN.
S20	000111	.91	IF CUST-ACCT-BALANCE < BALANCE-MIN
S21	000112		MOVE CUST-ACCT-BALANCE TO BALANCE-MIN.
S22	000113		*** Calculate Maximum ***
S23	000114	.19	IF WS-FIRST-TIME-SW = 'y'
S24	000115		MOVE CUST-ACCT-BALANCE TO BALANCE-MAX.
S25	000116	.94	IF CUST-ACCT-BALANCE > BALANCE-MAX
S26	000117		MOVE CUST-ACCT-BALANCE TO BALANCE-MAX.
S27	000118		*** CALCULATE RANGE ***
S28	000119	2.93	COMPUTE BALANCE-RANGE = BALANCE-MAX - BALANCE-MIN.
S29	000120	4.62	ADD 1 TO LOOP-COUNT.
S30	000121		
S31	000122		500-INIT-STATISTICS.

17. C08 and source mapping provide referred attribution reporting.

Name	Description	Percent of CPU Time * 10.00%
SAM2V	Application Program	40.68
- SAM2V	CSECT in SAM2V	40.68
- 00044C	Attribution	40.68
- IGZCPAC	COBPACK	29.38
- IGZCXDI	double precision division	29.38
- 22891xxx	unresolved Address	11.30
SAM1V	Application Program	8.84
- SAM1V	CSECT in SAM1V	8.84
- 001876	Attribution	3.31
- IDA019L1	Virtual I/O (VIO) and VSAM	2.40
- IDA019C1	Virtual I/O (VIO) and VSAM	1.43
- IDA019R3	Virtual I/O (VIO) and VSAM	0.71
- IDA019R1	Virtual I/O (VIO) and VSAM	0.14
- IDA019R2	Virtual I/O (VIO) and VSAM	0.05
- IDA019R1	Virtual I/O (VIO) and VSAM	0.03
- IDA019R4	Virtual I/O	0.01

CPU time in a system routine is attributed up the call chain to an application program

Tip: If Application Performance Analyzer has captured new abends since you opened a fault history file, you can refresh it. In the Systems Information view, **right-click** the Fault History file and select **Reload history file from host**.

18. You have completed this exercise.

Summary

Congratulations, you have completed the exercises and have familiarized yourself with the Application Performance Analyzer eclipse interface.

In these two labs:

- You opened the APA perspective.
- You reviewed the report download process.
- You learned to get HELP screens and pop-ups.
- You learned the layout of the CAZ0 report.
- You learned about six of the report categories.
- You learned how to drill down into sub-categories of reports.
- You learned about the displaying the W03, C01, & C08 reports with source mapping.

If you have any questions about Application Performance Analyzer, please contact the instructor.



© Copyright IBM Corporation 2015. All rights reserved.

The information contained in these materials is provided for informational purposes only, and is provided AS IS without warranty of any kind, express or implied. IBM shall not be responsible for any damages arising out of the use of, or otherwise related to, these materials. Nothing contained in these materials is intended to, nor shall have the effect of, creating any warranties or representations from IBM or its suppliers or licensors, or altering the terms and conditions of the applicable license agreement governing the use of IBM software. References in these materials to IBM products, programs, or services do not imply that they will be available in all countries in which IBM operates.

IBM, the IBM logo, and other IBM products and services are trademarks of the International Business Machines Corporation, in the United States, other countries or both. Other company, product, or service names may be trademarks or service marks of others.