IBM Application Performance Analyzer for z/OS

Customization Guide

Version 12 Release 1

IBM Application Performance Analyzer for z/OS

Customization Guide

Version 12 Release 1

> **Note**
>
> Before using this information and the product it supports, be sure to read the general information under Notices.

This edition applies to IBM Application Performance Analyzer for z/OS Version 12 Release 1 (5697-Q03) and to any subsequent releases until otherwise indicated in new editions. Make sure you are using the correct edition for the level of Application Performance Analyzer.

You can order publications online at www.ibm.com/shop/publications/order, or order by phone or fax. IBM Software Manufacturing Solutions takes publication orders between 8:30 a.m. and 7:00 p.m. Eastern Standard Time (EST). The phone number is (800) 879-2755. The fax number is (800) 445-9269.

You can find out more about IBM Application Performance Analyzer for z/OS by visiting the IBM Web site for Application Performance Analyzer at: www.ibm.com/software/awdtools/apa/

Contents

About this document

This document describes IBM® Application Performance Analyzer for z/OS®, Version 12 Release 1.

For the latest Application Performance Analyzer PTF information, see http://www.ibm.com/software/awdtools/apa/support/.

Appendix A contains information about IBM Web sites that can help you answer questions and solve problems.

IBM Application Performance Analyzer for z/OS (also referred to as Application Performance Analyzer) is a tool you can use to analyze the performance of user applications throughout the design, development, and maintenance cycle.

If you need to install Application Performance Analyzer, refer to the Application Performance Analyzer *Program Directory* for installation instructions.

Who might use this document

This document is intended for system programmers and system administrators who plan for, customize, and maintain Application Performance Analyzer.

To use this document, you need to be familiar with the z/OS operating system, the publications that describe your system, and job control language (JCL) and REXX exec processing.

The following release levels are compatible with Application Performance Analyzer:
- For z/OS:
  - z/OS R1.8
  - z/OS R1.9
  - z/OS R1.10
  - z/OS R1.11
  - z/OS R1.12
- For CICS®:
  - CICS/TS 2.3
  - CICS/TS 3.1
  - CICS/TS 3.2
  - CICS/TS 4.1
  - CICS/TS 4.2
  - CICS/TS 5.1
- For IMS™:
  - IMS 9.1
  - IMS 10
  - IMS 11
  - IMS 12
- For DB2®:

- DB2 9.1
- DB2 10.1
- For MQSeries®:
  - MQSeries 6.0
  - MQSeries 7.0
- For Java:
  - Java 5 - SR10 and above
  - Java 6 - SR6 and above
  - Java 7
- For Adabas:
  - Adabas 8.2.2
- For Natural:
  - Natural 4.2.6
- For WebSphere® Application Server:
  - WebSphere Application Server V7
  - WebSphere Application Server V8

Other documentation that you might need

For the installation of IBM Application Performance Analyzer you will need to refer to the *IBM Application Performance Analyzer for z/OS Program Directory*.

For information about how to use the product, see the publication *IBM Application Performance Analyzer for z/OS User's Guide*.

For details about messages, see the publication *IBM Application Performance Analyzer for z/OS Messages Guide*.

For further information about how to do specific configuration tasks for DB2, IMS, and CICS (as might be described in this book), see the product documentation for these subsystems.

IBM Application Performance Analyzer publications, related product publications, and their order numbers are listed in the "Bibliography" on page 131.

Document conventions

The following typographical conventions are used throughout this document:

Category	Convention	Examples
Command names	ALL CAPITAL LETTERS	RDEFINE
Member names	ALL CAPITAL LETTERS	CAZ@SPF
Variables	*italics*	*xxx* OPERAND invalid *hlq*.SCAZSAMP
JCL code	`monospace type`	`//jobname` JOB
JCL lines that need to be altered for your system	**Bold**	**DSN=xxxxxx**
High-level qualifiers that you chose	*hlq*	*hlq*.SCAZSAMP

Some basic concepts and terminology

Sampling

The process by which Application Performance Analyzer measures activity – recording data observed during repeated, timed "snapshots"– is called "sampling." Application Performance Analyzer saves the observed data in a "measurement file," sometimes referred to as a "sample file."

Non-intrusive technology

Application Performance Analyzer's sampling mechanism is efficient, effective, and, above all, non-intrusive. Application Performance Analyzer's measurement engine will never schedule any task to execute in the address space being measured.

The Application Performance Analyzer started task

This started task performs all of Application Performance Analyzer's measurements. Normally, a single Application Performance Analyzer started task is active in a single MVS™ instance. In a multiple-MVS instance installation (including a sysplex), an Application Performance Analyzer started task needs to be active in each MVS instance in which measurements are to be performed.

The Application Performance Analyzer Listener started task

This started task manages the communication between the Application Performance Analyzer GUI and the Application Performance Analyzer started tasks. Multiple workstation GUI users communicate through one Listener to any compatible Application Performance Analyzer started task defined on a single MVS instance.

Multiple started tasks

In special situations you might want to run multiple Application Performance Analyzer started tasks in a single operating system instance. For example, you might want to do this when making a transition from one product version to the next. Each Application Performance Analyzer instance is identified by a unique four-character instance identifier. For simplicity, this manual discusses Application Performance Analyzer in the context of a single-instance configuration.

The Application Performance Analyzer product code and Application Performance Analyzer instance ID

The module name prefix (product code) used by Application Performance Analyzer is CAZ. We have chosen CAZ0 as the default Application Performance Analyzer instance identifier, and it is used for examples in this document.

The ISPF component

Application Performance Analyzer's ISPF component provides the user with full, interactive access to the product's functionality. A single, gateway dialog serves as the entry point to all interactive services. The interactive services fall into two basic categories: scheduling and managing measurement requests; and analyzing measurement data.

The Graphical User Interface

The Application Performance Analyzer graphical user interface (GUI) is a CICS Explorer plug-in that provides an alternative to the Application Performance Analyzer ISPF interface. It communicates with the Application Performance Analyzer started task through the Application Performance Analyzer Listener started task.

Common data store

An optional data set unique to each Application Performance Analyzer started task (except in a sysplex environment), used to store lists of data set and directory names containing source program mapping files, and source program mapping associations. It includes one common list of data set names that can be shared amongst all users, and personal source mapping file lists for individual users. Individual users maintain their personal source mapping file lists using the ISPF A03 and A04 panels or the GUI Mapping Repository feature. Administrators maintain the common source mapping file list using the ISPF A05 panel or the GUI Mapping Repository feature.

IEFUSI exit routine

One Application Performance Analyzer module will receive control during the initiation of all job steps and started tasks, via the Step Initiation Exit (IEFUSI). During execution of this exit routine, Application Performance Analyzer determines whether a measurement has been scheduled for the job step, or started task. If it has, the exit routine will initiate sampling for the target address space. Application Performance Analyzer requires that the exit module (CAZ00990) in library *hlq*.SCAZLINK be added to the LNKLST.

Post-process initiation exit

One Application Performance Analyzer module will receive control during the initiation of all USS processes via the Post-process initiation exit (BPX_POSPROC_INIT). During execution of this exit routine, Application Performance Analyzer determines whether the process is a child process of one that is currently being measured by Application Performance Analyzer. If it is, the exit routine will initiate sampling for this target process. Application Performance Analyzer requires that the exit module (CAZ00991) in library hlq.SCAZLINK be added to the LNKLST.

IEFU83 exit routine

This is an optional SMF Record exit that Application Performance Analyzer will enable when measuring WebSphere applications. The exit will receive control whenever the SMFWTM or SMFEWTM macro is invoked by WAS. It captures SMF type 120 subtype 9 records for active sampling requests that have the WAS data extractor turned on. All other SMF records are ignored. Application Performance Analyzer requires that the exit module (CAZ00993) in library *hlq*.SCAZLINK be added to the LNKLST. When this exit is enabled, you must determine the amount of above-the-bar common storage that a WAS sampling session is permitted to allocate. This value is specified in the MemObjSize in the Application Performance Analyzer-started task configuration.

IEFU84 exit routine

This is an optional SMF Record exit that Application Performance Analyzer will enable when the DB2 SMF configuration option has been enabled. The exit will receive control whenever the SMFWTM or SMFEWTM macro is invoked by DB2. It captures SMF type 101 records for active sampling requests that have the DB2+ data extractor turned on. All other SMF records are ignored. Application Performance Analyzer requires that the exit module (CAZ00994) in library *hlq*.SCAZLINK be added to the LNKLST. When this exit is enabled, you must determine the amount of above-the-bar common storage that a DB2+ sampling session are permitted to allocate. This value is specified in the MemObjSize in the Application Performance Analyzer-started task configuration.

J9VM sampling module

When sampling Java jobs, one Application Performance Analyzer module is dynamically loaded and invoked by the JVMTI agent. This module processes requests to record Java sample data in the sampling data space. Which module is loaded depends on whether Application Performance Analyzer has been configured to use a dynamically loaded JVMTI agent or a preloaded JVMTI agent, and whether the target Java application is 31-bit or 64-bit, as shown by the table below:

	31-bit	64-bit
Dynamically loaded JVMTI agent	CAZ00102	CAZ00104
Preloaded JVMTI agent	CAZ00111	CAZ00114

These modules in library *hlq*.SCAZLINK must be added to the LNKLST.

How to send your comments

Your feedback is important in helping us to provide accurate, high-quality information. If you have comments about this book or any other IBM Application Performance Analyzer documentation, contact us in one of these ways:

- Use the Online Readers' Comment Form at www.ibm.com/software/awdtools/rcf/. Be sure to include the name of the book, the publication number of the book, the version of IBM Application Performance Analyzer, and, if applicable, the specific location (for example, page number) of the text that you are commenting on.
- Fill out the Readers' Comment Form at the back of this book, and return it by mail or give it to an IBM representative. If the form has been removed, address your comments to:

  IBM Corporation
  H150/090
  555 Bailey Avenue
  San Jose, CA 95141-1003
  USA

- Fax your comments to this U.S. number: (800) 426-7773.

When you send information to IBM, you grant IBM a nonexclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

Chapter 1. Preparing to customize Application Performance Analyzer

This chapter provides planning information to help you prepare to customize Application Performance Analyzer. You can use the checklist to get an overview of the tasks and to note your progress.

If you need additional documentation for Application Performance Analyzer, refer to the Application Performance Analyzer library page at Application Performance Analyzer library page or the IBM System z Enterprise Development Tools and Compilers information center at System z Enterprise Development Tools & Compilers information center.

Checklists for installing and customizing Application Performance Analyzer

To customize Application Performance Analyzer, complete the tasks in the following checklist.

___ 1. If you ordered Application Performance Analyzer using the Custom-Built Product Delivery Offering (CBPDO), follow the steps in the *IBM Application Performance Analyzer for z/OS Program Directory* for installation. (If Application Performance Analyzer is delivered through a ServerPac, the installation is completed for you.)

___ 2. Ensure that security access is configured.

___ 3. Place SCAZLINK in the LNKLST.

___ 4. Enable the Application Performance Analyzer system exits.

___ 5. APF-authorize libraries *hlq*.SCAZAUTH and *hlq*.SCAZLINK.

___ 6. Modify the TSO/E logon procedure.

___ 7. Choose a started task instance identifier.

___ 8. Make the Application Performance Analyzer-started task available.

___ 9. Create installation-level settings and settings for the started task.

___ 10. Pre-allocate a checkpoint file and common data store file (optional).

___ 11. Create your own descriptive program attribution table (optional).

___ 12. Modify Application Performance Analyzer procedure to provide access to third party repository products for source program mapping support (optional).

___ 13. Start the Application Performance Analyzer-started task.

___ 14. Start Application Performance Analyzer.

___ 15. Run an IVP to verify the customization of Application Performance Analyzer (optional).

J9VM Checklist

If you plan to use Application Performance Analyzer to measure Java applications, you must choose between using a dynamically loaded JVMTI agent and using a preloaded JVMTI agent. Depending on which one you choose, you might have to perform some or all of the following customization tasks:

___ 1. APF authorize the Application Performance Analyzer USS files.

1

__ 2. Assign USS permissions to the Application Performance Analyzer started task.

__ 3. Set up J9VM related security rules.

__ 4. Update the J9VM related started task configuration settings.

__ 5. Enable the Java Attach API (for a dynamically loaded JVMTI agent) or specify a preloaded JVMTI agent.

Further details are found in Chapter 5, "Customizing Application Performance Analyzer for use with J9VM," on page 71.

Listener Checklist

If the Application Performance Analyzer Graphical User Interface (GUI) is to be used in your organization, you must install and customize the Application Performance Analyzer Listener started task, and each user must install the GUI desktop application.

Perform the following steps to customize the Application Performance Analyzer Listener started task:

__ 1. Choose a Listener started task identifier.

__ 2. Choose a high-level qualifier for the Listener data sets. This can be the same high-level qualifier as the Application Performance Analyzer started task.

__ 3. Choose a TCP/IP port number on which the Listener will listen for client connections.

__ 4. Create security access rules for the Listener data sets.

__ 5. Assign priority to the Listener started task.

__ 6. Make the Listener started task available.

__ 7. If Application Performance Analyzer uses internal security to secure access to functions, update the installation-level configuration settings to allow use of the Listener.

__ 8. Create the Listener started task configuration settings.

__ 9. If Application Performance Analyzer uses an external security product to secure access to functions, add the new rule to allow use of the Listener.

__ 10. Start the Application Performance Analyzer Listener started task.

Further details are found in Chapter 6, "Customizing Application Performance Analyzer Listener," on page 77.

Application Performance Analyzer Plug-in for Eclipse Checklist

Perform the following steps to install the Application Performance Analyzer GUI plug-in on each client:

__ 1. Ensure all software prerequisites are in place.

__ 2. Obtain the Application Performance Analyzer Listener host address and port from your systems programmer.

__ 3. Determine the default Application Performance Analyzer started task to which you will connect.

__ 4. Obtain the Readme PDF and Application Performance Analyzer Plug-in for Eclipse .zip file from the IBM Problem Determination Tools Plug-ins website.

___ 5. Use the software installation function of your existing Eclipse-based desktop product to install the Application Performance Analyzer Plug-in for Eclipse.

Further details are found in Chapter 7, "About Application Performance Analyzer Plug-ins for Eclipse," on page 89.

DB2 Checklist

If you plan to use the DB2 EXPLAIN feature, want to report plan and package BIND timestamps, or want to display actual host variable names in SQL statements, you must complete the following steps:

___ 1. Choose a unique authorization ID.

___ 2. Create and bind a DB2 access package and plan.

___ 3. Create a PLAN_TABLE.

___ 4. Provide DB2 authorization.

___ 5. Update the started task JCL.

___ 6. Update the started task configuration settings.

Further details are found in Chapter 4, "Customizing Application Performance Analyzer for use with DB2 Universal Database," on page 67.

WebSphere Application Services (WAS) Checklist

Application Performance Analyzer provides a feature to measure applications running under WebSphere Application Services by capturing SMF type 120 subtype 9 records. If you plan to use Application Performance Analyzer to measure applications running under WebSphere Application Services, you must also perform the following additional customization tasks:

___ 1. Add SAF FACILITY class authorities for CAZ00993

___ 2. Add RACF® authority for EMCS consoles

___ 3. Add SMF exit CAZ00993 to LNKLST

___ 4. Enable IEFU83 SMF record exit

___ 5. Calculate above-the-bar common storage required

___ 6. Determine who can measure applications running under WebSphere Application Services and then assign the internal access rule UseWAS or RACF external access rule USE.WAS to the appropriate users.

___ 7. Determine how recording of SMF type 120 subtype 9 records is controlled. You may turn this trace on permanently, or you may direct Application Performance Analyzer to start and stop the trace as needed.

___ 8. Update CONFIG SAMPLE setting to turn on the WAS extractor and enable SMF capture and/or recording.

___ 9. Update CONFIG SAMPLE setting to assign a value to the MemObjSize keyword.

Further details are found in this chapter and in Chapter 2, "Customizing Application Performance Analyzer," on page 11

DB2 SMF Accounting Trace Checklist

Application Performance Analyzer provides a feature to capture DB2 accounting data from SMF type 101 records during measurement of applications with the DB2+ data extractor selected. This data will be reported in the Application Performance Analyzer reports, for example, the F20 DB2 Class 3 Wait Times report.

If you want Application Performance Analyzer to capture and report SMF DB2 accounting data, you must also complete the following additional customization tasks:

__ 1. Add SAF FACILITY class authorities for CAZ00994.

__ 2. Add RACF authority for EMCS consoles.

__ 3. Add SMF exit CAZ00994 to LNKLST.

__ 4. Enable IEFU84 SMF record exit.

__ 5. Calculate above-the-bar common storage required.

__ 6. Determine how the DB2 accounting trace is controlled. You may turn this trace on permanently or you may direct Application Performance Analyzer to start and stop the trace as needed. If you want Application Performance Analyzer to stop and start the trace, ensure the Application Performance Analyzer started task has DB2 TRACE privileges or DB2 authorization to issue TRACE commands.

__ 7. Update CONFIG SAMPLE setting to turn on the DB2I extractor and enable SMF capture and/or recording.

__ 8. Update CONFIG SAMPLE setting to assign a value to the MemObjSize keyword.

__ 9. Set the DSNZPARM ACCUMACC parameter to NO to improve the accuracy of DB2 Class 3 Wait Times report for DDF and RRSAF threads.

Further details are found in this chapter and in Chapter 2, "Customizing Application Performance Analyzer," on page 11

Migration considerations

If you are migrating from an earlier version of Application Performance Analyzer to Version 12.1, you must include the following tasks in your customization.

Checklist for migrating Application Performance Analyzer

To migrate Application Performance Analyzer, complete the tasks in the following checklist.

__ 1. If you ordered Application Performance Analyzer using the Custom-Built Product Delivery Offering (CBPDO), follow the steps in the *IBM Application Performance Analyzer for z/OS Program Directory* for installation. (If Application Performance Analyzer is delivered through a ServerPac, the installation is completed for you.)

__ 2. Obtain and apply applicable PTFs as described in Appendix F, "Applying service APAR or PTF," on page 121. For a list of available PTFs, refer to Latest Application Performance Analyzer for z/OS service information.

__ 3. Replace the following libraries:
  - *hlq*.SCAZEXEC
  - *hlq*.SCAZAUTH
  - *hlq*.SCAZLINK
  - *hlq*.SCAZMENU
  - *hlq*.SCAZPENU
  - *hlq*.SCAZTENU
  - *hlq*.SCAZDBRM
  - *hlq*.SCAZPAX

__ 4. Migrate any changed members from the existing *hlq*.SCAZSAMP to the new *hlq*.SCAZSAMP.

__ 5. Customize *hlq*.SCAZEXEC members. Refer to "Modifying the TSO/E logon procedure to include Application Performance Analyzer user interface" on page 17 and "Modifying Application Performance Analyzer procedure for third party products" on page 58 for more information.

__ 6. APF-authorize libraries *hlq*.SCAZAUTH and *hlq*.SCAZLINK.

__ 7. Change the *hlq*.SCAZAUTH and *hlq*.SCAZSAMP library names in the Application Performance Analyzer-started task.

__ 8. Start the Application Performance Analyzer-started task.

__ 9. Start Application Performance Analyzer.

__ 10. Run an IVP to verify the customization of Application Performance Analyzer (optional).

The following changes are needed for your DB2 Universal Database™ when migrating to the latest version of IBM Application Performance Analyzer:

__ 1. Run CAZBIND with the new *hlq*.SCAZDBRM library.

__ 2. If the PLAN_TABLE was changed in your current version of IBM Application Performance Analyzer, then you need to recreate the PLAN_TABLE.

Running earlier versions simultaneously with Version 12.1

You can run a previous version of Application Performance Analyzer in the same image as Version 12.1. However, each version must use unique started task names, unique STCID, a unique checkpoint data set, a unique SPXGroupName and a unique common data store dataset. Also, use the appropriate version of the Application Performance Analyzer frontend (ISPF and GUI) for any Application Performance Analyzer-started tasks to which you are connecting. When running multiple versions of Application Performance Analyzer in a sysplex, the newest version should be started first.

Library names after you finish installing

After you install Application Performance Analyzer, you have the data sets available as listed in Table 1. If you changed the names during installation, you can add the current names to the table for easy reference.

Table 1. Application Performance Analyzer library names

Default name	Contents	Site name if different
hlq.SCAZLINK	LNKLST module	
hlq.SCAZAUTH	Load library modules	
hlq.SCAZEXEC	REXX exec library	
hlq.SCAZMENU	ISPF messages (English)	
hlq.SCAZPENU	ISPF panels (English)	
hlq.SCAZTENU	ISPF tables (English)	
hlq.SCAZDBRM	Data	
hlq.SCAZSAMP	Sample JCL	
hlq.SCAZPAX	Packed USS files for J9VM support	

Note: The library name prefix *hlq* above should not be confused with the *yourhlq* value described below for data sets created by Application Performance Analyzer started task when you run your applications. The same library prefix *hlq* can be used by all Application Performance Analyzer started task instances, but the *yourhlq* prefix for the checkpoint, log, and sample data sets must be unique for each Application Performance Analyzer instance.

Library names created by Application Performance Analyzer

The Application Performance Analyzer started task creates various data sets as part of its normal operation. The table shown below lists these data sets and indicates the names assigned by Application Performance Analyzer.

Table 2. Application Performance Analyzer data set names and descriptions

Data set name	Description
yourhlq.*CheckpointDSN*	Application Performance Analyzer checkpoint data set
yourhlq.**LOG***xx*.*stcid*.**D***yyddd*.**T***hhmm###* or *yourhlq*.**LOG***xx*.**G***nnnn***V***nn*	Optional log data set
yourhlq.CDS	Common data store for source program mapping
yourhlq.*userid*.**R***nnnn*.*jobname*.**SF**	Measurement data set or sample data set (when SampleDSN is not specified in CONFIG SAMPLE)
tsoprefix.*userid*.**R***nnnn*.XMIT.	Exported measurement data set or sample data set if the ExportDSN configuration parameter is not specified.

With the exception of the checkpoint file, the rightmost qualifiers are determined by Application Performance Analyzer. You choose a prefix that replaces the *yourhlq* symbol shown above; however, it must comply with the JCL rule for specifying a data set name. In this case your prefix can have a maximum of 17 characters.

Note: In the event that you execute multiple Application Performance Analyzer started tasks, each one must be assigned its own unique *yourhlq* value; each creates its own set of data sets. It is a useful convention to include the started task instance identifier as part of this prefix.

The log data set will have the "xx" identifier suffix on the second qualifier in sysplex environments to make sure the file is unique. The "###" at the end of the log data set is a sequence number assigned when each new log file is created.

By default, the sample data set and log data set will use the value in the **HLQ** parameter of the CONFIG BASIC statement. Optionally, a different sample *yourhlq* and log *yourhlq* can be specified on the CONFIG SAMPLE and CONFIG LOGFILE statements, and the checkpoint data set can be a fully qualified name. You also have the option of overriding the Application Performance Analyzer generated sample data set name by specifying the SampleDSN setting in CONFIG SAMPLE.

When the EXP (a single measurement) or EXPH (a hierarchy of measurements) command is issued against an observation request, a new data set is created. The data set in XMIT format contains the measurement data that Application Performance Analyzer uses to produce the performance analysis reports, and is used as input to the Application Performance Analyzer IMPORT command and

batch utility. If the ExportDSN configuration setting is specified, this data set is created using the values specified in that setting. Otherwise, this data set is created under the tsoprefix of the individual issuing the EXP or EXPH command. If tsoprefix is not used, this qualifier is excluded. The data set is named in the format: *tsoprefix.***R***nnnn.XMIT*.

Planning for security

Application Performance Analyzer runs on a z/OS system as a started task. The user ID assigned to the started task must have READ access or higher to the resources used by monitored systems and applications.

Establishing external security system access rules

The following applies to your External Security System (such as RACF). Application Performance Analyzer also provides its own access rules facility to complement the external security system. The Application Performance Analyzer access rules are described in "Configuring Application Performance Analyzer" on page 19.

Load library (SCAZAUTH)
You need to grant Application Performance Analyzer users execute access to the product load library SCAZAUTH.

Other libraries
Users should be granted read-only access to all other Application Performance Analyzer libraries.

Checkpoint file
The Application Performance Analyzer started task will allocate a checkpoint file. Application Performance Analyzer uses this data set to record the status of all measurement requests. The checkpoint file is named in the format *yourhlq.CheckpointDSN*. The Application Performance Analyzer started task must have full access to this data set. Additionally, you should grant full access to the appropriate product support personnel, and read-only access to all other users of Application Performance Analyzer.

Note: Some installations might need to pre-allocate the checkpoint file. See "Pre-allocating a checkpoint data set" on page 54 to determine if this applies to your installation.

Log files
If you choose to activate the logging option, the Application Performance Analyzer started task will allocate and write to log data sets. These data sets record activity within the started task. The Application Performance Analyzer started task must have full access to these data sets. Additionally, you should grant full access to the appropriate product support personnel and deny access to all other users. These data sets have a name in the form:

*yourhlq.***LOG.D***yyyyddd.***T***hhmmsst*

Common data store file
If you choose to enable the common data store option, the Application Performance Analyzer started task allocates the common data store file if one does not already exist. Application Performance Analyzer uses the common data store file, which is unique to each started task, to record and maintain a list of data set names and directories that contain source

program mapping files. It includes source mapping file lists for individual users, as well as those that are common to all users. Individual users maintain their personal lists using the ISPF A03 and A04 panels or the GUI Mapping Repository feature. Administrators maintain the common list using the ISPF A05 panel or the GUI Mapping Repository feature. The common data store file is typically named in the format *yourhlq*.CDS. The Application Performance Analyzer started task must have full access to this data set. Additionally, you should grant full access to the appropriate product support personnel.

Note: Some installations might need to preallocate the common data store file. See "Pre-allocating a common data store file" on page 55 to determine whether this applies to your installation.

Sample (measurement) files

The Application Performance Analyzer started task creates a measurement file (or "sample" file) for each measurement request that has completed. These data sets contain the measurement data that Application Performance Analyzer uses to produce the performance analysis reports. The Application Performance Analyzer started task must have full access to these data sets. Additionally, you should grant full access to the appropriate product support personnel, and read-only access to all other Application Performance Analyzer users. These data sets are named in the format:

*yourhlq.userid.***R***nnnn.jobname.***SF

You also have the option of overriding the Application Performance Analyzer generated sample data set name by specifying the SampleDSN setting in CONFIG SAMPLE.

SAF FACILITY class authorities

If your installation has activated the SAF FACILITY class for CSVDYNL and or CSVDYNEX, you must authorize the Application Performance Analyzer started task as described in the table below:

Table 3. Class authorities required for SAF FACILITY

Class entity	Access authority
CSVDYNL.*linklist*.TEST **1**	READ
CSVDYNEX.LIST	READ
CSVDYNEX.SYS.IEFUSI.CAZ00990	UPDATE
CSVDYNEX.SYSJES2.IEFUSI.CAZ00990 (for JES2 environments)	UPDATE
CSVDYNEX.SYSJES3.IEFUSI.CAZ00990 (for JES3 environments)	UPDATE
CSVDYNEX.SYSSTC.IEFUSI.CAZ00990	UPDATE
CSVDYNEX.SYSTSO.IEFUSI.CAZ00990	UPDATE
CSVDYNEX.BPX_POSPROC_INIT.CAZ00991	UPDATE
CSVDYNEX.SYS.IEFU83.CAZ00993	UPDATE
CSVDYNEX.SYSSTC.IEFU83.CAZ00993	UPDATE
CSVDYNEX.SYS.IEFU84.CAZ00994	UPDATE
CSVDYNEX.SYSSTC.IEFU84.CAZ00994	UPDATE

Note: **1** Where *linklist* is the name of the active LNKLST. This can be displayed on the console with the command:

```
D PROG,LNKLST,NAME=CURRENT
```

RACF Authority for EMCS Consoles

Application Performance Analyzer collects data from SMF for WebSphere measurements and additional DB2 accounting data for DB2+ measurements. In order to collect this data from SMF, the Application Performance Analyzer started task must have RACF authority to use an Extended MCS console. The following RACF General Resource Profile must be defined in the RACF OPERCMDS class:

```
MVS.MCSOPER.consname
```

Where *consname* is eight characters long, left justified and padded with zeros. It begins with CAZ followed by the Application Performance Analyzer started task ID and ends with zeros. For example, if the Application Performance Analyzer started task ID is CZ12, *consname* would be CAZCZ120. The user ID associated with the Application Performance Analyzer started task must have READ access to this profile.

Non-RACF security

If you are running a security system other than RACF, you might need to give Application Performance Analyzer read access to any protected load libraries it might need to search. Application Performance Analyzer requires access to load libraries in order to gather information about all modules it encounters during an observation session. If the Application Performance Analyzer started task gets S913 abends in a non-RACF environment, protected loadlibs are a likely cause.

Deciding on the Application Performance Analyzer configuration settings

Application Performance Analyzer allows you to customize the product through the use of configuration settings. You must review these settings and decide on appropriate values for your installation. For more information, see "Installation-level configuration settings" on page 20.

Deciding on the Application Performance Analyzer access rules

Application Performance Analyzer allows you to set up access rules for your user community. These rules enable you to control levels of access to the product on a user-by-user basis. The product is shipped with some sample rules, but these rules limit access to some functionality, and are of course not specific to any of your users. To provide access suitable for your installation, you will need to specify your own access rules.

You can use either the internal security scheme, or a RACF-compatible external security product. For internal rules, see "Installation-level configuration settings" on page 20. For external rules, see "Configuring external security (RACF)" on page 47.

Assigning priority

If your installation is running Workload Manager (WLM) in GOAL mode, Application Performance Analyzer should be assigned to the IBM-supplied SYSSTC class. You must configure this in WLM.

If running in COMPAT mode, put Application Performance Analyzer in the same performance group as other high priority system started tasks such as VTAM® and TCP/IP. You add Application Performance Analyzer to the performance group by updating member IEAICS00 in SYS1.PARMLIB.

The dispatch priority and service class assigned should be verified using primary command "DA" in SDSF. Column "DP" shows the dispatch priority and column "SrvClass" shows the WLM service class.

Using the sysplex feature

This feature provides the ability to measure jobs, started tasks and TSO address spaces, and to view the resultant reports on any system in a sysplex, regardless of the system on which the user is currently logged. To accommodate this feature during customization, Application Performance Analyzer must be defined with the ability to communicate with other instances of Application Performance Analyzer on different systems in a sysplex.

The communication between Application Performance Analyzer instances on different systems is facilitated by the use of an Application Performance Analyzer sysplex group. A unique instance of Application Performance Analyzer must be installed on every system in a sysplex and each of these instances must be defined to the same Application Performance Analyzer sysplex group. Only one instance of Application Performance Analyzer on the same system can be a member in the same Application Performance Analyzer sysplex group.

An Application Performance Analyzer sysplex group is defined during customization using the "SPXGroupName" configuration setting on the CONFIG BASIC statement. This parameter is coded as "SPXGroupName=*apaspxgp*". All Application Performance Analyzer instances that you want to communicate with each other must code an identical "SPXGroupName=*apaspxgp*". The value of *apaspxgp* must be unique to Application Performance Analyzer and must not be the sysplex name of the system.

All Application Performance Analyzer instances defined in the same Application Performance Analyzer sysplex group must share a common checkpoint file and Common Data Store (CDS). Therefore, it is critical that the HLQ, UNIT, CheckpointDSN, and CDS parameters are identical. Failure to do so causes the started task to fail upon startup with an appropriate error message. The Application Performance Analyzer sysplex feature requires that all systems in a sysplex participate in DASD sharing. It is important that you choose an HLQ, UNIT, CheckpointDSN, and CDS that will allow Application Performance Analyzer to allocate all new data sets to volumes that are shared across the systems in the sysplex. For more information, see CONFIG BASIC statement.

In summary, it is a good practice to define identical CONFIG BASIC statements for every Application Performance Analyzer instance participating in the same Application Performance Analyzer sysplex group, with the exception of the STCID, which might be unique.

Chapter 2. Customizing Application Performance Analyzer

Perform the following tasks to set up the basic product after installing it:

- "Adding Application Performance Analyzer modules to the LNKLST"
- "Enabling the Application Performance Analyzer system exits" on page 12
- "Authorizing Application Performance Analyzer modules" on page 17
- "Modifying the TSO/E logon procedure to include Application Performance Analyzer user interface" on page 17
- "Choosing an Application Performance Analyzer-started task identifier" on page 18
- "Making the Application Performance Analyzer-started task available" on page 18
- "Creating the installation-level and started task configuration settings" on page 19
- "Configuring external security (RACF)" on page 47
- "Pre-allocating a checkpoint data set" on page 54 (optional)
- "Pre-allocating a common data store file" on page 55 (optional)
- "Updating the descriptive program attribution tables" on page 55 (optional)
- "Modifying Application Performance Analyzer procedure for third party products" on page 58 (optional)
- "Starting the Application Performance Analyzer-started task" on page 59
- "Starting the IBM Application Performance Analyzer user interface" on page 59

Adding Application Performance Analyzer modules to the LNKLST

The library *hlq*.SCAZLINK contains Application Performance Analyzer's exit routines and modules that must be included in the LNKLST. This includes:

- The step initiation exit (CAZ00990), which receives control at step initiation.
- The USS Post-process initiation exit (CAZ00991), which receives control at new process initiation.
- The SMF exit (CAZ00993), which captures SMF type 120 subtype 9 records for WAS data.
- The SMF exit (CAZ00994), which captures SMF type 101 records for DB2 accounting data.
- The J9VM sampling engine modules (CAZ00102 and CAZ00111), which measure 31-bit Java applications running under the IBM J9VM.
- The J9VM sampling engine modules (CAZ00104 and CAZ00114), which measure 64-bit Java applications running under the IBM J9VM.

hlq.SCAZLINK must be added to the LNKLST for Application Performance Analyzer to execute successfully. Add hlq.SCAZLINK to the LNKLST by performing one of the following steps.

- Update the PROGxx member in PARMLIB with one of the following statements, where *hlq* is the high-level qualifier that you used to install Application Performance Analyzer:
  - If the library is cataloged in the master catalog, use:
    ```
    LNKLST ADD NAME(lnklst.name) DSNAME(hlq.SCAZLINK)
    ```

– If the library is cataloged in a user catalog, use:

```
LNKLST ADD NAME(lnklst.name) DSNAME(hlq.SCAZLINK) VOLUME(volser)
```

To make the change take effect, IPL the system.

- Update the current PROGxx member or create a new PROGxx member that defines a new LNKLST with the addition of the SCAZLINK dataset.

  – If the library is cataloged in the master catalog, use:

```
LNKLST ADD NAME(lnklst.name) DSNAME(hlq.SCAZLINK)
```

  – If the library is cataloged in a user catalog, use:

```
LNKLST ADD NAME(lnklst.name) DSNAME(hlq.SCAZLINK) VOLUME(volser)
```

To make the change take effect, issue the following command:

```
'SET PROG=xx' where xx is the suffix of the PROG member that was just updated.
```

Enabling the Application Performance Analyzer system exits

Application Performance Analyzer requires several system exits to be enabled depending on the types of workloads on which Application Performance Analyzer will report. These exits include both SMF exits and a USS exit. They are described here.

IEFUSI

This is the SMF Step Initiation Exit and it must be enabled for the desired address space types on which Application Performance Analyzer will report. The IEFUSI exit point can be defined to SMF so that SMF will be called for the entire system, batch jobs, started tasks, and TSO logons. Enable IEFUSI by performing the following steps:

1. Create or update a SMFPRMxx member in SYS1.PARMLIB (or any parmlib that will be used for IPL). Specify IEFUSI on the EXITS option of the SYS or SUBSYS parameters. For example:

```
SYS(...,
EXITS(IEFUSI,...)) /* enable IEFUSI for the entire system   */

SUBSYS(TSO,
EXITS(IEFUSI,...)) /* enable IEFUSI for TSO workloads        */

SUBSYS(JES2,
EXITS(IEFUSI,...)) /* enable IEFUSI for JES2 Batch workloads */

SUBSYS(JES3,
EXITS(IEFUSI,...)) /* enable IEFUSI for JES3 Batch workloads */

SUBSYS(STC,
EXITS(IEFUSI,...)) /* enable IEFUSI for Started Tasks        */
```

2. To make the changes effective, issue the 'SET SMF=xx' console command.

When you set up SMFPRMxx correctly, the Application Performance Analyzer-started task will inform you of each exit point that Application Performance Analyzer is using. For example, if you set up SMFPRMxx to take SMF exit IEFUSI for TSO users, then you would see a message similar to the following for SYSTSO:

```
CAZ1221I CAZ0-MAIN  Module CAZ00990 added to exit SYSTSO.IEFUSI
```

After applying any maintenance or installing a new release of Application Performance Analyzer, do one of the following actions:

- IPL the system and restart the Application Performance Analyzer-started task, which will load the new Application Performance Analyzer exits.

- Or to load the new exits without IPLing the system, issue the following commands so that the system deletes the Application Performance Analyzer exits at the IEFUSI exit point:

  For the entire system...
  ```
  SETPROG EXIT,DELETE,EXITNAME=SYS.IEFUSI,MODNAME=CAZ00990
  ```
  For a JES2 environment...
  ```
  SETPROG EXIT,DELETE,EXITNAME=SYSJES2.IEFUSI,MODNAME=CAZ00990
  ```
  For a JES3 environment...
  ```
  SETPROG EXIT,DELETE,EXITNAME=SYSJES3.IEFUSI,MODNAME=CAZ00990
  ```
  ForTSO logons...
  ```
  SETPROG EXIT,DELETE,EXITNAME=SYSTSO.IEFUSI,MODNAME=CAZ00990
  ```
  For started tasks...
  ```
  SETPROG EXIT,DELETE,EXITNAME=SYSSTC.IEFUSI,MODNAME=CAZ00990
  ```

Then restart the Application Performance Analyzer-started task, which will load the new Application Performance Analyzer exits.

For more information about SMFPRMxx and its parameters, see *z/OS Initialization and Tuning Reference*.

IEFU83

This SMF Record Exit must be enabled and the recording of SMF record type 120 subtype 9 must be activated, for Application Performance Analyzer to report on WebSphere activity. The IEFU83 exit point can be defined to SMF so that SMF will be called for the entire system, batch jobs, started tasks, and TSO logons. Application Performance Analyzer only requires that IEFU83 be defined to the SYSSTC exit point. Application Performance Analyzer will define the exit point when WAS=(YES...) is specified on the CONFIG SAMPLE statement.

Enable IEFU83 and activate SMF record type 120 subtype 9 recording by performing the following steps:

1. Create or update a SMFPRMxx member in SYS1.PARMLIB (or any parmlib that will be used for IPL). Specify 120(9) on the TYPE option, and IEFU83 on the EXITS option of the SYS or SUBSYS(STC) parameters. For example:
   ```
   SYS(TYPE(120(9))),...,
   EXITS(IEFU83,...)) /* enable IEFU83 for the entire system */

   SUBSYS(STC,TYPE(120(9))
   EXITS(IEFU83,...)) /* enable IEFU83 for started tasks     */
   ```
2. To make the changes effective, issue the 'SET SMF=xx' console command.

When you set up SMFPRMxx correctly, the Application Performance Analyzer-started task will inform you of each exit point that Application Performance Analyzer is using. For example, if you set up SMFPRMxx to take SMF exit IEFU83 for started tasks then you would see a message similar to the following for SYSSTC:

```
CAZ1221I CAZ0-MAIN  Module CAZ00993 added to exit SYSSTC.IEFU83
```

After applying any maintenance or installing a new release of Application Performance Analyzer, do one of the following actions:

- IPL the system and restart the Application Performance Analyzer-started task, which will load the new Application Performance Analyzer exits.

- Or to load the new exits without IPLing the system, issue the following commands so that the system deletes the Application Performance Analyzer exits at the IEFU83 exit point:

  For the entire system...
  ```
  SETPROG EXIT,DELETE,EXITNAME=SYS.IEFU83,MODNAME=CAZ00993
  ```
  For started tasks...
  ```
  SETPROG EXIT,DELETE,EXITNAME=SYSSTC.IEFU83,MODNAME=CAZ00993
  ```

Then restart the Application Performance Analyzer-started task, which will load the new Application Performance Analyzer exits. For more information about SMFPRMxx and its parameters, see *z/OS Initialization and Tuning Reference*.

IEFU84

This SMF Record Exit must be enabled and the recording of SMF record type 101 must be activated, for Application Performance Analyzer to report on DB2 accounting activity. The IEFU84 exit point can be defined to SMF so that SMF will be called for the entire system, batch jobs, started tasks, and TSO logons. Application Performance Analyzer only requires that IEFU84 be defined to the SYSSTC exit point. Application Performance Analyzer will define this exit point when DB2I=(YES,DB2SMFEnable=YES...) is specified on the CONFIG SAMPLE statement.

Enable IEFU84 and activate SMF record type 101 recording by performing the following steps:

1. Create or update a SMFPRMxx member in SYS1.PARMLIB (or any parmlib that will be used for IPL). Specify 101 on the TYPE option, and IEFU84 on the EXITS option of the SYS or SUBSYS(STC) parameters. For example:
   ```
   SYS(TYPE(101),...,
   EXITS(IEFU84,...)) /* enable IEFU84 for the entire system */

   SUBSYS(STC,TYPE(101)
   EXITS(IEFU84,...)) /* enable IEFU84 for started tasks    */
   ```
2. To make the changes effective, issue the 'SET SMF=xx' console command.

When you set up SMFPRMxx correctly, the Application Performance Analyzer-started task will inform you of each exit point that Application Performance Analyzer is using. For example, if you set up SMFPRMxx to take SMF exit IEFU84 for started tasks then you would see a message similar to the following for SYSSTC:

```
CAZ1221I CAZ0-MAIN  Module CAZ00994 added to exit SYSSTC.IEFU84
```

After applying any maintenance or installing a new release of Application Performance Analyzer, do one of the following actions:

- IPL the system and restart the Application Performance Analyzer-started task, which will load the new Application Performance Analyzer exits.
- Or to load the new exits without IPLing the system, issue the following commands so that the system deletes the Application Performance Analyzer exits at the IEFU84 exit point:

  For the entire system...
  ```
  SETPROG EXIT,DELETE,EXITNAME=SYS.IEFU84,MODNAME=CAZ00994
  ```
  For started tasks...
  ```
  SETPROG EXIT,DELETE,EXITNAME=SYSSTC.IEFU84,MODNAME=CAZ00994
  ```

Then restart the Application Performance Analyzer-started task, which will load the new Application Performance Analyzer exits.

For more information about SMFPRMxx and its parameters, see *z/OS Initialization and Tuning Reference*.

BPX_POSPROC_INIT

This is the UNIX System Services post-process initiation exit, and it must be enabled for Application Performance Analyzer to measure USS child processes of a process that is currently being measured by Application Performance Analyzer. The exit requires no PARMLIB changes. Application Performance Analyzer will dynamically define the exit when USSSpwnEnable=YES is specified on the CONFIG SAMPLE statement.

The following message will be issued if the exit has been successfully defined:
```
CAZ1221I CAZ0-MAIN  Module CAZ00991 added to exit BPX_POSPROC_INIT
```

After applying any maintenance or installing a new release of Application Performance Analyzer, do one of the following actions:

- IPL the system and restart the Application Performance Analyzer-started task, which will load the new Application Performance Analyzer exits.
- Or to load the new exits without IPLing the system, issue the following commands so that the system deletes the Application Performance Analyzer exits at the exit point:

    For the entire system...
```
SETPROG EXIT,DELETE,EXITNAME=BPX_POSPROC_INIT,MODNAME=CAZ00991
```

Then restart the Application Performance Analyzer-started task, which will load the new Application Performance Analyzer exits.

For more information about this exit see *z/OS UNIX System Services Programming: Assembler Callable Services Reference*.

Defining the data spaces for the IEFUSI exit

The recommended size for data spaces in the IEFUSI exit is 3GB with 20 data spaces. Data space limitations can prevent Application Performance Analyzer from producing on-line reports or batch printed reports. These recommendations allow for very large sample files.

Determining the amount of above-the-bar common storage for the IEFU83 and IEFU84 exits

The optional Application Performance Analyzer IEFU83 and IEFU84 exit routines are used to record SMF data for WebSphere (WAS) activity, and DB2 accounting data which is captured for all DB2+ requests, respectively. They require above-the-bar common storage for each sampling request that captures these SMF records. You must specify the total amount of storage in the Application Performance Analyzer started task CONFIG SAMPLE keyword MemObjSize. This section provides guidelines to determine the amount of above-the-bar common storage required by Application Performance Analyzer.

Calculating the Storage for WebSphere (WAS) Requests

The following table shows the SMF record type 120 subtype 9 sections that are recorded by Application Performance Analyzer for each request processed by WAS.

Name	Description	Length	Minimum Repetitions	Maximum Repetitions	Maximum Total Length
APA	Memory Object Header	8	1	1	8
SM120ST9	Header	204	1	1	204
SM120S91	Platform Neutral Server Info Section	76	1	1	76
SM120S92	z/OS Server Info Section	68	1	1	68
SM120S93	Platform Neutral Request Info Section	68	1	1	68
SM120S94	z/OS Request Info Section	428	1	1	428
SM120S96	Network Data Section	188	0	1	188
SM120S97	Classification Data Section	144	0	5	720
SM120S99	CPU Usage breakdown section	548	0	30	16440
Total		**852**			**18200**

The minimum size for a WAS request is 852 bytes. The maximum size for a WAS request is 18200 bytes. The actual size required for a WAS request is dependent on the type of request. For example, the Network Data Section is not used for Message Driven Bean (MDB) requests.

To calculate the amount of APA memory object storage required, multiply the average request size by the number of WAS requests that will be processed during a WAS sampling session.

> For example, if the average WAS request has a Network Data Section, 3 Classification Data Sections and 3 CPU Usage Breakdown Sections, then the size of each request is 852 + 188 + (3 x 144) + (3 x 548) = 3116 bytes.

It will be up to each installation to determine the average request size based on their WAS processing profile and the number of WAS requests that would be created during a typical sampling session. Multiply the size of the average WAS request by the number of WAS requests per typical sampling session to determine the WAS component of memory object size (MemObjSize).

Calculating the Storage for DB2+ Requests

During a sampling session, each DB2 thread will require 208 bytes of memory object storage. Multiply the anticipated number of threads during a sampling session by 208 bytes to determine the DB2 component of the memory object. The number of threads will depend on the DB2 attach type that's being used in the sampled address space. Typically, TSO, CAF, and RRSAF attachments will have one thread. The number of threads for IMS and CICS attach types will depend on thread reuse. For the IMS attach type, the number of threads will correspond to the number of transactions processed in the sampled region. The number of threads for the CICS attach will depend on the ACCOUNTREC parameter on the CICS DB2CONN definition.

Authorizing Application Performance Analyzer modules

The libraries *hlq*.SCAZAUTH and *hlq*.SCAZLINK contain all of the Application Performance Analyzer modules that must be authorized for the product to execute successfully. Add *hlq*.SCAZAUTH and *hlq*.SCAZLINK to the system APF list by performing the following steps:

1. Create or update a PROGxx member in SYS1.PARMLIB (or any parmlib that will be used for IPL). Add one of the following statements, where *hlq* is the high-level qualifier that you used to install Application Performance Analyzer:

   - If the libraries are managed by SMS, use:

     ```
     APF ADD DSN(hlq.SCAZAUTH) SMS
     ```

     and

     ```
     APF ADD DSN(hlq.SCAZLINK) SMS
     ```

   - If the libraries are not managed by SMS, use:

     ```
     APF ADD DSN(hlq.SCAZAUTH) VOLUME(parm)
     ```

     and

     ```
     APF ADD DSN(hlq.SCAZLINK) VOLUME(parm)
     ```

     The variable *parm* is one of the following

     - A six-character identifier for the volume on which *hlq*.SCAZAUTH and *hlq*.SCAZLINK reside
     - ******, which indicates that *hlq*.SCAZAUTH and *hlq*.SCAZLINK are located on the current SYSRES volume
     - *MCAT*, which indicates that *hlq*.SCAZAUTH and *hlq*.SCAZLINK are located on the volume that contains the master catalog

2. To make the change effective, either issue the 'SET PROG=xx' command or IPL.

Modifying the TSO/E logon procedure to include Application Performance Analyzer user interface

The functional routines of Application Performance Analyzer are executed under ISPF in a TSO/E user's session. To allow users to start the Application Performance Analyzer user interface, you must make these ISPF libraries available to them.

To make the ISPF libraries available to users, do the following steps:

1. In members CAZ@SPF and CAZISPS in your REXX exec library *hlq*.SCAZEXEC, change the high-level qualifier as appropriate.

2. In member CAZISPS, if your *hlq*.SCAZAUTH library is not link listed, change IPCALOAD to your SCAZAUTH library.

3. In member CAZISPS, if you are not using the standard data set names created by Application Performance Analyzer, customize ISPPLIB, ISPMLIB, and ISPTLIB.

4. In member CAZISPS, if LE is not link listed, set the data set name of your SCEERUN library in parameter IPCACEE

5. Add the Application Performance Analyzer user interface to your ISPF menu by inserting the additional lines (`NEW`) as shown in the following example:

```
   ...
) BODY  CMD(ZCMD)

   ...
  9   IBM Products    IBM program development products
  10  SCLM            SW Configuration Library Manager
  11  Workplace       ISPF Object/Action Workplace
  APA AppPerfAnalyzer Application Performance Analyzer for z/OS ( NEW )
...
) PROC
   ...
  &ZSEL = TRANS( TRUNC  (&ZCMD,'.')
   ...
  9,'PANEL(ISRDIIS) ADDPOP'
 10,'PGM(ISRSCLM) SCRNAME(SCLM) NOCHECK'
 11,'PGM(ISRUDA) PARM(ISRWORK) SCRNAME(WORK)'
 APA,'CMD(EX ''hlq.SCAZEXEC(CAZISPS)'') +
 MODE(FSCR) NEWAPPL(CAZ3)' /*Application Performance Analyzer*/ ( NEW )
...
```

For more information about configuring your ISPF Primary Option menu, see *z/OS ISPF Planning and Customizing*.

Choosing an Application Performance Analyzer-started task identifier

An Application Performance Analyzer-started task is identified internally by a four-character identifier called the STCID. The IBM-default value for the started task identifier shipped with the product is CAZ0. Similarly, the IBM-default value for the Application Performance Analyzer-started task procedure name shipped with the product is CAZ0. You can choose a different four-character started task identifier and a different started task procedure name to meet your site's requirements. They do not have to match.

Making the Application Performance Analyzer-started task available

To make the Application Performance Analyzer-started task available, complete the following steps:

1. Edit the procedure CAZ0 in your library *hlq*.SCAZSAMP. See the instructions in the sample job for more information.

2. Copy this procedure to a library in the JES2 or JES3 procedure library concatenation for started tasks. You can rename this procedure to any valid procedure name to meet your site's requirements.

3. Define a user ID for the Application Performance Analyzer-started task procedure.

4. Grant the user ID you created READ access to Application Performance Analyzer libraries.

5. Define a profile in the RACF STARTED class for the Application Performance Analyzer-started task procedure so that it can access protected resources. To do

this, update your security definitions by using the RDEFINE command to associate the Application Performance Analyzer-started task procedure with user IDs as follows:

```
RDEFINE STARTED procname.* STDATA(USER(userid) GROUP(group))
```

Where *procname* is the Application Performance Analyzer-started task procedure name, *userid* is the user ID, and *group* is the group on your system that has access to the RACF-protected resources.

You can issue the above command under TSO/E or on the system console log.

For more information about using the STARTED class, see *z/OS Security Server (RACF) Security Administrator's Guide*.

6. Create or update a COMMND*xx* parmlib member that will be used for IPL so that it contains the following command:

```
COM='START procname'
```

Where *procname* is the Application Performance Analyzer-started task procedure name.

Note: The Application Performance Analyzer-started task should run at the same or higher priority than any other task you will measure. For more information about assigning priority, see "Assigning priority" on page 9.

Configuring Application Performance Analyzer

This section describes how to configure Application Performance Analyzer to suit your installation. There are two types of configuration: global installation-level settings, and settings for the started task.

Creating the installation-level and started task configuration settings

Application Performance Analyzer provides for two types of configuration:

Installation-level configuration settings
This is global, installation-level configuration information. You specify the settings in assembly macro statements and generate these into the load module CAZCNFG1 in *hlq*.SCAZAUTH. The key settings stored in this module are the Application Performance Analyzer access rules.

The member CAZCNFG1 in *hlq*.SCAZSAMP contains model source code. Modify and add settings to suit your installation. The member CAZASMCF in *hlq*.SCAZSAMP contains the assembly and linkedit JCL necessary to create the CAZCNFG1 module. Refer to the instructions in CAZASMCF for details of the necessary changes.

Started task configuration settings
You specify these settings in a SYSIN-type file which the Application Performance Analyzer started task processes when it begins execution. You can change these settings at any time by editing the member and then restarting the started task. The supplied member CAZCNFG0 in *hlq*.SCAZSAMP is an example of this file which you can copy and modify to suit your requirements. See "Started task configuration settings" on page 25 for more information.

Installation-level configuration settings

These settings are generated into the load module CAZCNFG1. You specify these in assembly macro statements then edit and submit sample job CAZASMCF in *hlq*.SCAZSAMP to generate the CAZCNFG1 module. Source member CAZCNFG1 in *hlq*.SCAZSAMP contains model source code. The following example illustrates the syntax of the statements in the CAZCNFG1 source member.

```
* - - - - - - - - - - - - - - - - - - - - - - - - - - - - - *
*     Configuration Section.
*
CAZCNFG1      CAZCVSET START
              CAZCVSET END


* - - - - - - - - - - - - - - - - - - - - - - - - - - - - - *
*    Access Rules (Security) Section.
*
01 CAZRULE   *,allow,ViewRequestsOwnedBy,========
02 CAZRULE   *,allow,DeleteRequestsOwnedBy,========
03 CAZRULE   *,allow,UpdateRequestsOwnedBy,========
04 CAZRULE   *,allow,MeasureJOB,*
05 CAZRULE   *,allow,MeasureSTC,*
06 CAZRULE   *,allow,MeasureTSU,========
07 CAZRULE   *,allow,UseDB2Plus,*
08 CAZRULE   *,allow,UseIMSPlus,*
09 CAZRULE   *,allow,MonitorJOB,*
10 CAZRULE   *,allow,MonitorSTC,*
11 CAZRULE   *,allow,MonitorTSU,========
12 CAZRULE   *,allow,UseWLMI,*
13 CAZRULE   *,allow,UseCICSPlus,*
15 CAZRULE   ADMIN01,allow,AdministerProduct,*
16 CAZRULE *,allow,UseWAS,*
   CAZRULE END
* - - - - - - - - - - - - - - - - - - - - - - - - - - - - - *
```

Specifying access rules

You specify access rules – rules controlling the type of access to Application Performance Analyzer functions granted by User ID – in the access rules section of CAZCNFG1. You define each access rule in the CAZCNFG1 source member using the CAZRULE statement. In each rule you specify:

- a User ID value, or pattern
- ALLOW or DISALLOW indicating if the rule specifies that the access is allowed or disallowed
- the type of access
- and the name, or name pattern, of an Object to be accessed.

The CAZRULE statement is an assembly macro statement and must follow the syntax rules for assembler macros. You specify the four values, noted above, in the first through fourth positional macro parameters. You might also specify a value in the name field – starting in column 1. This is ignored by the CAZRULE; you can use this for your own purposes.

CAZRULE syntax:

```
►►─CAZRULE─Userid─,──────────────────,─AccessType─,─Object─,──────────────►
                        ├─ALLOW────┤
                        └─DISALLOW─┘
```

```
      ┌─ID=*────┐
      └─ID=STCID─┘
```
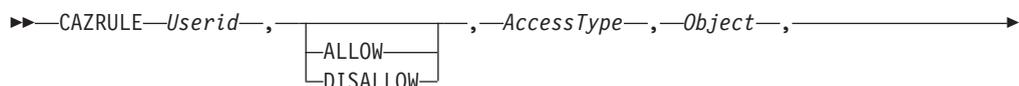
CAZRULE Statement Parameters:

Userid

Specify the User ID to which the rule applies in the first positional parameter. You can specify either a full User ID or a User ID pattern. The value is not case sensitive.

Specify patterns using the following special characters:

* (asterisk)

To specify a wildcard value. An asterisk must be the last character in the value. The characters up to the asterisk are compared for an exact match. For example, ABC* will match any User ID beginning with ABC.

? (question mark)

To specify any character match. A question mark specifies that any User ID character, in the same position, will be matched. For example, AB??EF will result in the match of any User ID characters in the third and fourth positions. The User ID ABCDEF will be matched, as will ABXYEF.

ALLOW or DISALLOW

The second positional parameter determines whether the CAZRULE statement specifies that the access is to be permitted (ALLOW) or denied (DISALLOW). The value is not case sensitive.

AccessType

The third positional parameter in the CAZRULE statement specifies the type of access. See "Access rule descriptions" on page 22 for a summary of all AccessType values. The value is not case sensitive.

Object

The fourth positional parameter specifies the Object to which the rule applies. Refer to "Access rule descriptions" on page 22 for information about which objects apply to which AccessType values.

The types of objects are:
- User ID name or pattern
- Job Name or pattern
- Application Performance Analyzer ID or asterisk (*) for any Application Performance Analyzer started task

User ID Object pattern

When the object type is a User ID, an additional special pattern character, an equal sign (=), is permitted. Specifying '=' has the effect of substituting the character from the actual User ID (the User ID for which the rule is being checked) before the comparison is made.

ID   The CAZRULE accepts one keyword parameter: ID=STCID. You can use this parameter to specify the Application Performance Analyzer ID to which the rule applies. Typically you omit the ID= parameter for a default value of '*'. The rule applies to all Application Performance Analyzer started tasks.

Examples: Some examples of CAZRULE statements are illustrated here. The examples specify a name field that is ignored. We assigned reference numbers 01, 02, 03 and 04 to the rules in the name field for easier identification.

```
01 CAZRULE FRANK01,Allow,ViewRequestsOwnedBy,*
02 CAZRULE *,Allow,MeasureJob,*
03 CAZRULE *,Allow,MeasureTSU,========
04 CAZRULE ADMIN01,allow,AdministerProduct,*
```

Notes about CAZRULE statement examples:

1. The first example specifies an access rule that applies to User ID FRANK01. The second positional parameter, Allow, specifies that access is granted to FRANK01. The third positional parameter specifies the type of access. In this case, ViewRequestsOwnedBy specifies that FRANK01 is allowed to view reports for measurement requests. The fourth positional parameter specifies *. FRANK01 is allowed to view reports for measurement requests owned by any user.

2. The second example specifies an access rule that applies to all users; the wildcard pattern * in the first positional parameter indicates this. The second positional parameter, Allow, specifies that the access is granted. The third positional parameter specifies the type of access. In this case, MeasureJob; it specifies those jobs are allowed to be measured. The fourth positional parameter specifies *. This grants all users permission to measure all job names.

3. The third example specifies an asterisk for the User ID, hence, applies to all users. The value '========' is specified for the Object. In the case of MeasureTSU the Object is the User ID of the user's TSO address space. Since each character of the Object field is an equal sign, the corresponding characters of the actual User ID are substituted. The effect is that the user is only allowed to measure his own TSO address space.

4. The fourth example specifies that only the User ID ADMIN01 has Application Performance Analyzer administrator authority. ADMIN01 is allowed to update the common list of data sets in the common data store for source program mapping. When expiry days warning is enabled, ADMIN01 will see the expiry days warning for all users' observations.

Access rule descriptions:

Code this	For this type of access	Object type
ViewRequestsOwnedBy	Controls which measurement requests the user is allowed to view reports for.	User ID of user that created the request.
DeleteRequestsOwnedBy	Controls which measurement requests the user is allowed to delete.	User ID of user that created the request.
UpdateRequestOwnedBy	Controls which measurement requests the user is allowed to modify.	User ID of user that created the request.
MeasureJOB	Controls what job names the user is allowed to measure.	Job name of job to be measured.
MeasureSTC	Controls what STCs (started tasks) the user is allowed to measure.	Name of started task procedure to be measured.
MeasureTSU	Controls what TSUs (TSO regions) the user is allowed to measure.	Name of TSO region (the TSO User ID).

Code this	For this type of access	Object type
UseDB2Plus	Controls whether the user is allowed to use the DB2+ data extractor.	Always '*'
UseIMSPlus	Controls whether the user is allowed to use the IMS+ data extractor.	Always '*'
MonitorJOB	Controls what job names the user is allowed to threshold-monitor.	Job name of job to be monitored.
MonitorSTC	Controls what STCs (started tasks) the user is allowed to threshold-monitor.	Name of started task procedure to be monitored.
MonitorTSU	Controls what TSUs (TSO regions) the user is allowed to threshold-monitor.	Name of TSO region (the TSO User ID).
UseWLMI	Controls whether the user is allowed to use the WLM Intercept to measure a specific DB2 stored procedure or user-defined function and/or use the collateral DB2 (CDB2) data extractor	Always '*'
UseCICSPlus	Controls whether the user is allowed to use the CICS+ data extractor	Always '*'
AdministerProduct	Controls whether the user has Application Performance Analyzer administrator authority. Administrator authority allows a user to update the common list of data sets in the CDS for source program mapping. When expiry days warning is enabled, users with administrator authority will see the expiry days warning for all users'observations. By default, this access is disallowed.	Always '*'
UseWAS	Controls whether a user is allowed to use the WAS data extractor.	Always '*'

Precedence of rules: Whenever separate CAZRULE statements specify conflicting effects for the same User ID, the statements specified later in the source member take precedence over earlier ones.

For example:

```
CAZRULE *,allow,ViewRequestsOwnedBy,========
CAZRULE FRANK01,allow,ViewRequestsOwnedBy,*
```

The first statement above specifies that all users are allowed to view reports for only their own requests. The second statement, however, specifies that User ID FRANK01 is allowed to view reports for all requests. This statement overrides the previous one for FRANK01.

The approach is to first code statements for the most inclusive rules and then follow these with less inclusive rules.

Suggested Application Performance Analyzer rules: Member CAZCNFG1 in *hlq*.SCAZSAMP contains suggested Application Performance Analyzer rules. You need to carefully review these, and make changes for your own installation. For example, rule number 06 allows all users to measure stated tasks. However, you might want to control who is able to run measurements of, for example, your CICS regions, in which case you would have to specify more specific rules controlling started task measurement. Measurement of CICS regions can either be permitted or restricted, more detailed options are not available.

The rules supplied in CAZCNFG1 leave access fairly open, and their equivalent RACF profiles are shown below:

```
01 CAZRULE *,allow,ViewRequestsOwnedBy,========
     /* This is a default, no RACF profile required.  */
02 CAZRULE *,allow,DeleteRequestsOwnedBy,========
     /* This is a default, no RACF profile required.  */
03 CAZRULE *,allow,UpdateRequestsOwnedBy,========
     /* This is a default, no RACF profile required.  */
04 CAZRULE *,allow,MeasureJOB,*
     /* RDEFINE FACILITY CAZ0.MEASURE.JOB.* UACC(READ)  */
05 CAZRULE *,allow,MeasureSTC,*
     /* RDEFINE FACILITY CAZ0.MEASURE.STC.* UACC(READ)  */
06 CAZRULE *,allow,MeasureTSU,========
     /* This is a default, no RACF profile required.  */
07 CAZRULE *,allow,UseDB2Plus,*
     /* RDEFINE FACILITY CAZ0.USE.DB2PLUS UACC(READ)  */
08 CAZRULE *,allow,UseIMSPlus,*
     /* RDEFINE FACILITY CAZ0.USE.IMSPLUS UACC(READ)  */
09 CAZRULE *,allow,MonitorJOB,*
     /* RDEFINE FACILITY CAZ0.MONITOR.JOB.* UACC(READ) */
10 CAZRULE *,allow,MonitorSTC,*
     /* RDEFINE FACILITY CAZ0.MONITOR.STC.* UACC(READ) */
11 CAZRULE  *,allow,MonitorTSU,========
     /* This is a default, no RACF profile required. */
12 CAZRULE  *,allow,UseWLMI,*
     /* RDEFINE FACILITY CAZ0.USE.WLMI UACC(READ) */
13 CAZRULE *,allow,UseCICSPlus,*
     /* RDEFINE FACILITY CAZ0.USE.CICSPlus UACC(READ) */
14 CAZRULE ADMIN01,allow,AdministerProduct,*
     /* RDEFINE FACILITY CAZ0.ADMINISTERPRODUCT UACC(NONE) */
     /* PERMIT CAZ0.ADMINISTERPRODUCT ID(ADMIN01) */
15 CAZRULE *,allow,UseWAS,*
     /* RDEFINE FACILITY CAZ0.USE.WAS UACC(READ) */
```

Rule number	Explanation
01	Users can view reports for only their own measurement requests. Access to reports for other users' requests is denied.
02	Users can only delete their own measurement requests.
03	Users can only modify/cancel their own measurement requests.
04	Any user can measure jobs with any jobname.
05	Any user is allowed to measure a started task.
06	Any user is allowed to measure his/her own TSO region.

Rule number	Explanation
07	Any user is allowed to turn on the DB2+ data extractor.
08	Any user is allowed to turn on the IMS+ data extractor.
09	Any user is allowed to threshold-monitor jobs with any jobname.
10	Any user is allowed to threshold-monitor a started task.
11	Any user is allowed to threshold-monitor his or her own TSO region.
12	Any user is allowed to measure a specific DB2 stored procedure or user-defined function and/or turn on the collateral DB2 (CDB2) data extractor.
13	Any user is allowed to turn on the CICS+ data extractor.
14	Allow user ADMIN01 to update the common list of data sets in the common data store. Disallow update access to all other users. When expiry days warning is enabled, ADMIN01 will see the expiry days warning for all users' observations. All other users will see the expiry days warning for their own observations only.
15	Any user is allowed to turn on the WAS data extractor.

Started task configuration settings

The Application Performance Analyzer-started task obtains these settings from a file allocated to DDname CONFIG. The file is a SYSIN-type member which you edit in order to change the settings. The member CAZCNFG0 in `hlq.SCAZSAMP` contains the initial settings. A group of configuration settings is specified by a CONFIG statement. Settings are organized into categories or classes. Each CONFIG statement will define settings for a particular class.

The general syntax for a CONFIG statement is illustrated here:

```
CONFIG class
keyword1=value1
keyword2=value2    // this is a comment
..
keywordn=valuen ;  // end with semicolon
```

You must code a single keyword/value pair expression within a single line; it cannot span two lines. Separate keyword/value pair expressions from each other using one or more blanks. A CONFIG statement, including all its keyword/value pair expressions must be terminated by a semicolon.

A CONFIG statement might be coded anywhere within columns 1 to 72 inclusive. An asterisk in column 1 denotes a comment line. A double slash (//) anywhere in the line indicates the remaining text in the line is a comment.

The following CONFIG BASIC statement, which is supplied in the predefined CAZCNFG0 member, illustrates the CONFIG syntax.

```
CONFIG BASIC                  // Basic settings:
   STCid=CAZ0                 // . Application Performance Analyzer instance ID
   HLQ=CAZ0                   // . DSN high level qualifier
   UNIT=SYSDA                 // . unit name for new data sets
   CheckpointDSN='.CHKP';     // . checkpoint file DSN
                              // (appended to HLQ)
```

CONFIG statement parameters

Class   Specifies the category of settings that the CONFIG statement is to modify. The allowable values for class are:

- BASIC
- SAMPLE
- LOGFILE
- OPER

keywordn
>   This specifies the name of the setting variable and is paired with a value expression in *keyword=value* format. The name is not case sensitive.

valuen   This is paired with a keyword name and specifies the value to be set.

CONFIG BASIC statement

BASIC settings allow you specify values for basic characteristics of the Application Performance Analyzer installation. The following settings can be specified:

Note: Required parameters are listed first. The alphabetical order specified below is not the order of the parameters in the configuration file.

CDS   Required.

>   This setting indicates whether the common data store is used for source program mapping, and if so specifies the name given to the Application Performance Analyzer common data store file.

>   The syntax for the CDS keyword is CDS=(*status,dsn*) where

>   - status = ENAbled | DISabled. Specify ENAbled to enable the common data store for source program mapping. Status must be ENAbled to allow source program mapping from the Application Performance Analyzer GUI.
>   - dsn = The name of the common data store partitioned data set extended. *dsn* is required only when status=ENAbled. The recommended value for *dsn* is '.CDS'. When using this recommended value or any *dsn* value that begins with a period ('.'), the value you specify for HLQ will be prefixed to the value you specify for *dsn*. In this case, the length of the value specified in *dsn* must be 44 characters, less the length of the value of HLQ. Otherwise, *dsn* must contain the fully qualified data set name. It can be up to 44 characters long. The Application Performance Analyzer started task automatically allocates the common data store file (as specified in the *dsn*) if one does not already exist.

>   For further information on the common data store for source program mapping, refer to Chapter 10 in the *Application Performance Analyzer for z/OS User's Guide*.

>   The following example enables the common data store with the file name of IBMAPA.CDS:

>   ```
>   CDS=(ENA,'.CDS')
>   HLQ=IBMAPA
>   ```

>   The following example disables the common data store:

>   ```
>   CDS=(DIS)
>   ```

>   Default: none

CheckpointDSN
>   Required.

>   Specifies the name to be given to the Application Performance Analyzer checkpoint file. You have two options when specifying this value:

- If *CheckpointDSN* begins with a period ('.') then the value that you specify for HLQ above will be prefixed to the value you specify in *CheckpointDSN*. In this case the length of the value specified in *CheckpointDSN* must be 44 characters less the length of the value of HLQ.
- Otherwise, *CheckpointDSN* must contain the fully qualified data set name. It can be up to 44 characters long.

Default: None

HLQ   Required.

Specifies the high-level qualifier for data sets created by the Application Performance Analyzer started task as part of its normal operation. The value that you specify for HLQ must comply with the JCL rule for specifying a data set name. It can be no more than 17 characters long.

Note: If this Application Performance Analyzer instance is a member of a sysplex group, you must select a high-level qualifier that will allow Application Performance Analyzer to allocate all new data sets to volumes that are shared across the sysplex. These data sets include the checkpoint file, sample files, and optional log files. See `SampleHLQ` in "CONFIG SAMPLE statement" on page 32, and `LogHLQ` in "CONFIG LOGFILE statement" on page 45 for descriptions of how to set a unique high-level qualifier for the sample files and log files.

Default: None.

STCID   Required.

This is the Application Performance Analyzer internal identifier, a four-character value that uniquely identifies the Application Performance Analyzer-started task. This may be, but does not have to be the same as the Application Performance Analyzer-started task procedure name.

Note: If you intend to run Application Performance Analyzer in a SYSPLEX group, you must never change the value of STCID once it has been set. The STCID forms part of the Application Performance Analyzer SYSPLEX group name. Application Performance Analyzer compares the group name recorded in each observation record, with the active Application Performance Analyzer group name to determine if the observation is local or remote to that instance of Application Performance Analyzer. Changing the STCID will cause observations once local to a Application Performance Analyzer instance to be treated as remote. Under rare conditions, this may potentially trigger a high volume of unnecessary XCF messages.

DefaultSMSClass

Optional.

This setting is used to assign SMS class values to the data sets created by Application Performance Analyzer. DefaultSMSClass specifies the default SMS class values used when no other SMSClass keyword from any other customization section is provided, and is used to allocate the following data sets:

- Checkpoint
- Common Data Store (CDS)
- Sample files
- Export files

- Log files
- PDF and XML reports

The syntax for this setting is DefaultSMSClass=(*mgmt,stg,data*) where the positional parameters are:

- *mgmt* = The management class name. It must be one to eight alphanumeric or national characters, as defined by the storage administrator at your installation.
- *stg* = The storage class name. It must be one to eight alphanumeric or national characters, as defined by the storage administrator at your installation.
- *data* = The data class name. It must be one to eight alphanumeric or national characters, as defined by the storage administrator at your installation.

DeleteOnJCLError
> Optional.
>
> Specifies whether multiple-step requests that get a JCL error in one of the steps should delete all the sample files and entries for the remaining steps. Specify `DeleteOnJCLError=Y` to have the entries for steps with a JCL error automatically deleted.
>
> Default: DeleteOnJCLError=N

ExpiryDays
> Optional.
>
> This setting indicates how Application Performance Analyzer is to manage the retention of completed observations. It accepts 3 variables using the syntax: ExpiryDays=(*days,ExpiryProt*,WARN=*nnn*) where

> *days* = **0-9999**
>> This specifies the number of days an observation is to be retained, before it is automatically deleted by Application Performance Analyzer. The user can override this value when entering an observation request by overtyping the `Retain File for (days)` field on the Job Information NEW panel. Specify 0 to retain the observation indefinitely. 0 is the default.

> *ExpiryProt* = **Y|N**
>> This specifies whether the `Retain File for (days)` field on the Job Information NEW panel is protected or unprotected. Specify Y to protect the field from changes. Specify N to leave the field unprotected. N is the default.

> *nnn* = **0-255.**
>> This enables the expiry days warning feature and specifies the number of days prior to observation automatic deletion that Application Performance Analyzer is to start warning the user. When this feature is enabled, the request number in the observation list might be displayed in red, yellow or green based on the expiry date. It is displayed in red when an observation will be automatically deleted within 24 hours. It will be displayed in yellow when an observation will be automatically deleted within the number of days specified for WARN. It will be displayed in green when the observation will never be deleted. Users with administrator authority will see the colors for all users'

observations. All other users will see the colors for their own observations only. A value of 255 disables this feature. 255 is the default.

Examples:

- ExpiryDays=20 (observation will be automatically deleted in 20 days, Retain File for (days) field is unprotected, expiry days warning is disabled)
- ExpiryDays=(20,Y) (observation will be automatically deleted in 20 days, Retain File for (days) field is protected, expiry days warning is disabled)
- ExpiryDays=(20,N,WARN=5) (observation will be automatically deleted in 20 days, Retain File for (days) field is unprotected, expiry days warning is enabled and user will be warned when 5 days are remaining before automatic deletion)

Default: ExpiryDays=(0,N,WARN=255) to indicate the observation file is to be retained indefinitely, the Retain File for (days) field is unprotected, and the expiry day warning feature is disabled.

MsgLevel

Optional.

Specifies the severity level of messages to be written to the log data sets. `MsgLevel=I` will write all messages, `MsgLevel=W` will write warning level and higher severity messages only. No SYSOUT data sets will be created for a sampling session with `MsgLevel=W` when only informational messages are produced.

Default: MsgLevel=I

SecureImportSys

Optional.

Specify Y (yes) to use the originating system name of imported sample files in Application Performance Analyzer security requests. Specify N (no) to use the local system name, which is the system into which the sample file is imported. This applies only when Security=External is specified. When Y, external security rules must be written to accommodate the use of the originating system name.

This keyword is applicable in cases where sample files are imported from an originating system with restricted access into a local system that has less restrictive access. If SecureImportSys=Y is not specified, users that are not authorized to view sample files on the originating system could potentially view the same sample file once imported into the local system.

Default: N

Example: System PROD allows only users USERA and USERB to view Application Performance Analyzer reports. System TEST allows all users to view all Application Performance Analyzer reports. To ensure that only USERA and USERB can view reports in TEST for sample files imported from PROD to TEST, the following must be specified in the TEST system:

- On the CONFIG BASIC statement:
  ```
  SecureImportSys=Y
  Security=External
  ```
- In RACF:
  ```
  RDEFINE PROD.VIEW.* UACC(NONE)
  PERMIT PROD.VIEW.* ID(USERA) ACCESS(READ)
  PERMIT PROD.VIEW.* ID(USERB) ACCESS(READ)
  ```

Security

Optional. (But `Security=External` is highly recommended.)

Specifies whether this started task should use Application Performance Analyzer's internal security scheme or use an external product, such as RACF. Specify `Security=External` if you use an external security product. To use Application Performance Analyzer's internal security scheme, you can specify `Security=Internal` or allow the default. (Application Performance Analyzer defaults to internal security.) You can also specify `Security=None` to run without security.

Default: Internal.

Note: The `Security=Internal` option is supported for backwards compatibility, but is not recommended. Similarly, you can specify the keyword `COMPAT` along with `Security=External` for backwards compatibility with an early version of the external security feature. The list below explains the available security options:

Security=Internal

Specifies that no external security will be used. This option is not recommended, particularly in sysplex environments.

Security=External

Specifies that external security will be used. This is the recommended option. It fully supports sysplex environments.

Security=(External,Compat)

Specifies that external security will be used. This option is for Application Performance Analyzer, Version 1.1 with PTF UK07430 applied for APAR PK10866 only. Do not use this option if you have not applied PTF UK07430, or if you have applied subsequent PTFs. (For external security with subsequent PTFs, use `Security=External`.) In this mode, the first node of a resource name is the STCID instead of the system name. For improved performance, migrate to the `Security=External` option.

Security=None

Specifies that no security will be used. This option is useful in test environments, in which security is not required.

Default: Internal

SecurityClassName

Optional.

Specifies an alternate class name will be used instead of `FACILITY`. This keyword is only used when `Security=External` is also specified; otherwise it is ignored. This is mandatory when SecurityHLQ is specified.

Default: `FACILITY`

Rules for class name:

1. Must be between 1-8 characters, consisting of:
   - A-Z
   - 0-9
   - #, @, $
2. Must contain at least one character from the following:
   0-9, #, @, $

If SecurityClassName is coded as `FACILITY` or allowed to default, then parameter SecurityHLQ may not be used. The FACILITY class is limited to 39 characters for the access rule that can be exceeded if SecurityHLQ is used. An alternative class is XFACILIT, a predefined class in RACF that allows up to 246 characters.

SecurityHLQ
Optional.

Specifies an optional qualifier for security profiles to be placed in front of the sysplex system name when a SecurityClassName has been specified other than `FACILITY`. This can be a maximum of 17 characters and must follow the rules established for the class in the Class Descriptor Table (CDT). The contents of this field are not checked during startup. This HLQ can be used in a sysplex environment to provide common access rules for all images in the sysplex.

If specified, it will be concatenated in front of the sysplex system name during security checks.

When SecuirtyHLQ is specified, SecurityClassName must also be specified, and it must not be `FACILITY`. See SecurityClassName for more details.

Default: None.

SecuritySuppMsg
Optional.

This option causes Application Performance Analyzer to either suppress or display messages issued as a result of `RACROUTE` calls made by Application Performance Analyzer. If not suppressed, messages will be displayed by `RACROUTE` via WTO.

Default: `Y` to suppress these messages.

SPXGroupName
Optional.

Specifies the Application Performance Analyzer sysplex group of which this Application Performance Analyzer instance will be a member within a sysplex environment. All Application Performance Analyzer instances that will communicate with one another in a sysplex must code an identical SPXGroupName=*apaspxgp*. The value of *apaspxgp* can be a maximum of eight alphanumeric characters, must be unique to Application Performance Analyzer, and must not be the sysplex name of the system.

All Application Performance Analyzer instances with the same SPXGroupName=*apaspxgp* must run on different systems in the sysplex and participate in DASD sharing. This allows Application Performance Analyzer to allocate all new data sets to volumes that are shared across the systems in the sysplex. Because they share a common checkpoint file and Common Data Store (CDS), it is important that the HLQ, UNIT, CheckpointDSN, and CDS are also identical.

In summary, it is a good practice to define identical CONFIG BASIC statements for every Application Performance Analyzer instance participating in the same Application Performance Analyzer sysplex group, with the exception of the STCID, which might be unique.

Default: None.

TZ Optional (only required for USS source mapping).

Specifies the TZ environment variable, which conforms to the POSIX standard. This must be coded if USS programs are to be source mapped using debug files in ELF/DWARF format. Otherwise, it is not required. Details on the format and content of a TZ environment variable can be found at:

`http://www.opengroup.org/onlinepubs/009695399/basedefs/xbd_chap08.html`

`TZ="EST5EDT,M3.2.0/02:00,M11.1.0/02:00"`

UNIT Optional.

Specifies the unit name to be used when allocating all new data sets of the following type:
- Application Performance Analyzer checkpoint file
- Application Performance Analyzer log data set
- Application Performance Analyzer sample files

Default: SYSDA

Unit can be either a device type such as 3390, or an esoteric such as SYSDA. A mount status of STORAGE or PUBLIC is nonspecific and does not require a VOLSER. (This release of Application Performance Analyzer does not allow a VOLSER to be specified.)

On a non-SMS-managed system, you must ensure that the UNIT name you specified contains one or more volumes mounted as either STORAGE or PUBLIC. If there are not such volumes mounted, then Application Performance Analyzer will be unable to allocate critical files and will terminate.

On an SMS-managed system, SMS will ignore the unit parameter. To ensure SMS control over all data sets allocated by Application Performance Analyzer, specify an HLQ value that is recognized and managed by SMS.

CONFIG SAMPLE statement

These settings pertain to the Sampling (measurement) process. The following settings might be specified:

Note: The alphabetical order specified below is not the order of the parameters in the configuration file.

Adabas Optional.

Specify YES to make the Adabas data extractor available.

Default: NO

AllExtractorsOn
Optional.

Specify YES to force all new measurement requests to have all the data extractors selected by default. This parameter should be used in special cases in which a particular instance of Application Performance Analyzer will be used to measure an address space by someone who does not know what activity types are running in that address space. Using AllExtractorsOn ensures that all types of activity will be recorded without the user manually selecting all the data extractors.

Default: NO

When YES is specified, Application Performance Analyzer will not select the WAS extractor by default because it is incompatible with some other extractors.

AltCICSStepNm

Optional.

Specifies an alternate load module name for the CICS region. By default, a step name of DFHSIP is used to determine if the region being sampled is a CICS region. If this step name is not found, CICS sampling is turned off to reduce overhead.

Default: DFHSIP

BufSize

Optional.

Specifies the size that Application Performance Analyzer is to use as the blocksize when allocating a new sample file.

Default: 16,384 bytes.

CICSI   Optional.

This is used to specify that the CICS Intercept should be turned on. Specify CICSI=YES to turn on this feature and to make the CICS+ data extractor available. When this feature is turned on and the CICS+ data extractor is selected for a measurement, Application Performance Analyzer will count the precise number of CICS transactions executed during a measurement and record the exact service time and CPU time for each transaction.

Default: NO

Compress

Optional.

Specify YES to turn sample file compression on. When sample file compression is activated, Application Performance Analyzer will compress all sample files prior to writing them. When a sample file is opened for reporting, Application Performance Analyzer will automatically decompress the file for the duration of the reporting session.

Default: NO

DB2EXPLAIN

Optional. Multiple DB2EXPLAIN entries are allowed.

The DB2EXPLAIN parameter enables the Application Performance Analyzer EXPLAIN feature for a specified DB2 subsystem. The DB2 Intercept must be enabled if EXPLAIN is specified. (See DB2I below.)

The syntax for the DB2EXPLAIN keyword is as follows:

DB2EXPLAIN=(ssid,D|Q|A,Yes|No)

This entry is made up of from one to three positional parameters. The first parameter specifies a DB2 subsystem name. The next two positional parameters are optional.

The second parameter indicates that the SQL STRING DELIMITER value specified in the application programming defaults panel (DSNTIPF) for the specified DB2 subsystem at installation. The values are as follows:
- D - Default
- Q - Quote (")

- A - Apostrophe (')

This value must match the DB2 subsystem installation value. The default value is D.

The third parameter indicates that the USE FOR DYNAMICRULES value specified in the application programming defaults panel (DSNTIPF) for the specified DB2 subsystem at installation. The values are as follows:

- Y or Yes

- N or No

This value must match the DB2 subsystem installation value. The default value is Yes.

The DB2EXPLAIN keyword can be repeated for each DB2 subsystem that you want to enable for EXPLAIN requests. To make it easier to enable EXPLAIN requests on all DB2 subsystems, an asterisk can be specified for the subsystem name. In its simplest form, a single entry can be specified as follows:

```
DB2EXPLAIN=(*)
```

This specifies that EXPLAIN requests can be run on all DB2 subsystems, with default values for SQL STRING DELIMITER and USE FOR DYNAMICRULES.

A maximum of 22 DB2 subsystems can be identified by 22 separate DB2EXPLAIN entries. If more than one entry is specified, and one of those has an asterisk for the DB2 subsystem name, then the Application Performance Analyzer started task will reject the configuration and fail to start.

DB2I   Optional.

This setting is used to specify that the DB2 Intercept and DB2 SMF data collection are to be enabled. The syntax is as follows:

```
DB2I=(YES|NO,SMF=YES|NO,CONTROLSMF=YES|NO).
```

The first positional parameter specifies whether the Application Performance Analyzer DB2 Intercept is to be enabled. The default value is NO. Specify YES to enable the DB2 Intercept. Enabling this feature makes the DB2+ data extractor available, which allows additional DB2 data to be collected.

Note: Running Application Performance Analyzer measurements with the DB2+ data extractor turned on causes each DB2 call to be intercepted to collect additional data. This might have a small impact on the performance of the target address space. Care should be taken when using this feature with other products that also intercept DB2 calls because unpredictable results might occur. You might want to restrict its use in production environments. You can use Access Rules to restrict the DB2+ feature to specific users. See `UseDB2Plus` in "Access rule descriptions" on page 22.

The SMF keyword parameter specifies whether Application Performance Analyzer support for collecting DB2 SMF data is to be enabled. The default value is NO. If YES is specified, Application Performance Analyzer will collect DB2 SMF data for all measurements that have the DB2+ data extractor selected. In this case Application Performance Analyzer will produce additional DB2 reports using data from the DB2 SMF record type 101 accounting records (refer to the User's Guide for report descriptions). When YES is specified, a non-zero value is required for the MemObjSize

keyword parameter. If NO is specified for DB2I in the first positional parameter or it is allowed to default, then this parameter is ignored.

Note: The DB2 DSNZPARM ACCUMACC parameter is used to roll up distributed and RRSAF thread information for efficiency. The IBM-provided default is 10. When ACCUMACC is specified with a value other than 'NO', the data in the F20 DB2 Class 3 Wait Times report may be inaccurate (or missing) for DDF and RRSAF threads. To improve the accuracy of this report, when SMF=YES is specified for DB2I, assign a value of NO to the DB2 DSNZPARM ACCUMACC.

The CONTROLSMF keyword parameter specifies whether Application Performance Analyzer is to control the DB2 Accounting Trace required for SMF data collection. The DB2 Accounting trace (class 1 and 3) must be started with a destination of SMF to enable Application Performance Analyzer to collect the required SMF record type 101 records. The default value is NO. If NO is specified or allowed to default, then it is up to the system administrator to ensure that the DB2 Accounting trace is on for the applicable DB2 subsystems when the measurement starts. If YES is specified, Application Performance Analyzer will start the DB2 Accounting trace (class 1 and 3) on the applicable DB2 subsystem when a DB2+ measurement begins and stop the trace when the measurement has completed. If NO is specified for DB2I in the first positional parameter or in the SMF keyword parameter, or either is allowed to default, then this parameter is ignored.

Note: When CONTROLSMF=YES is specified for DB2I, the Application Performance Analyzer started task must be granted the DB2 TRACE privilege or must have DB2 authorization to issue TRACE commands.

`Default: DB2I=(NO,SMF=NO,CONTROLSMF=NO)`

DB2IMaxTraceSize
Optional.

Specifies the number of DB2+ SQL call interceptions for which full details will be written to the sample file. Collecting full details on every interception allows the F02 Timeline report to report exact times for all SQL calls. The F02 report will be truncated at the number of calls specified in DB2IMaxTraceSize. The DB2+ data extractor continues to collect the data it requires for the other reports for the duration of the measurement. Use this parameter if you need more or less than the default 100000 entries on the F02 report. It has a range of 0 to 9999000, and must be specified in increments of 1000. Larger values will increase report load time and can cause dataspace overflows. This parameter does not affect DDF tracing.

This parameter can also specify whether the maximum trace size field is to be displayed on Panel 2 - Options, where users can change the value when creating or modifying observation requests.

Examples:
- To collect full details for 50,000 DB2+ SQL call interceptions where this value cannot be changed by the user on Panel 2, code the following:
  `DB2IMaxTraceSize=(50000,N)`
- To collect full details for 100,000 DB2+ SQL call interceptions where this value can be changed by the user on Panel 2, code the following:
  `DB2IMaxTraceSize=(100000,Y)`

Default: (100000,N)

DB2PlanName
Optional.

The plan name must be specified if you want Application Performance Analyzer to gather the necessary data to report plan and package BIND timestamps. You must also bind the Application Performance Analyzer DB2 plan to use this feature.

DB2Qualifier
Optional.

This entry specifies the qualifier to be used for the PLAN_TABLE that is dedicated to the Application Performance Analyzer started task. If omitted, it defaults to the authorization ID of the started task. It should be used only if more than one Application Performance Analyzer started task is running in the same MVS image.

The primary purpose for DB2Qualifier is to prevent using the same PLAN_TABLE for more than one started task. If two started tasks are each processing an `EXPLAIN` request at the same time using the same PLAN_TABLE, then one will have to wait while the other executes. Giving each started task its own qualifier prevents serializing the `EXPLAIN` requests among the started tasks.

DB2X Optional.

Specifies whether the Application Performance Analyzer static EXPLAIN feature is to be enabled. Enabling this feature makes the DB2X data extractor available for new measurement requests. If YES is specified, then the DB2 explain (DB2EXPLAIN) and DB2 Intercept (DB2I) must also be enabled.

When enabled for static EXPLAIN and when the DB2X data extractor is active for a measurement, Application Performance Analyzer will attempt to collect the EXPLAIN data for observed SQL statements that were bound with the EXPLAIN(YES) option at bind time.

Default: NO

DefLocSys
Optional.

Specifies to Application Performance Analyzer whether or not the local system id is to be used as the default for NEW observation requests. The acceptable values for this parameter are YES and NO. A value of YES directs Application Performance Analyzer to populate the System Name field on NEW observation requests with the local system id. A value of NO populates the field with *. When the System Id field is set to * and a non-active job is specified, Application Performance Analyzer looks at all systems in the Sysplex Group and starts measuring the first one it encounters. When the System Id field is set to * and an active job is specified, the active job list displays jobs from all systems in the Sysplex Group. In any case, the user can type over the field. This parameter specifies how it should initially be populated.

Default: NO

DescRequired
Optional.

Specifies whether the user is required to complete the description field on NEW, TNEW and MOD requests. YES and NO are the only valid values. YES indicates a description of at least 8 characters is required.

Default: NO

DEXMR   Optional.

Specifies whether the Data Extractor Module Resolution User Exit (CAZDEXMR) is called to resolve PSW addresses that were unresolved by the sampling engine. Specify YES to activate. A PSW address might be unresolved because the address was sampled in a foreign address space, or the address belongs to a module that is not managed by Content Supervision Services (i.e. LOAD, LINK, XTCL, ATTACH).

Default: NO

DSPMaxSize
Optional.

Specifies the maximum amount of storage in 4K blocks that the sampling data space is permitted to use. This storage limit applies to each sampling session that is running. The DSPMaxSize parameter will not directly limit the data space size used for reporting, but will affect reporting by limiting the amount of data that can be collected for reporting.

The minimum amount that can be specified is 30,000, the maximum is 514,288.

Default: 50,000 blocks (approximately 200MB)

DuplicateDSN
Optional.

Specifies the action Application Performance Analyzer is to take when the data set name of the sample file it is attempting to create already exists. This parameter is applicable only when the SampleDSN setting has been specified in CONFIG SAMPLE. The acceptable values are REP, GEN and DIS.

- REP – Replace an existing sample data set. This will delete the sample data set with the same data set name and create the new one. The observation for the deleted sample data set is also deleted from the R02 Observation List.
- GEN – Generate a sample data set name using the HLQ setting specified in CONFIG BASIC, and following the standard Application Performance Analyzer naming convention.
- DIS – Discard the sample for the active measurement. The status of the active measurement is marked as 'Failed'.

Default: GEN

ExportDSN
Optional.

Specifies up to 50 characters to be used to name export data sets. The value in this setting overrides the default generated export dataset name. The name may consist of system symbols and Application Performance Analyzer defined user symbols, which are substituted with values at the time the export dataset is allocated. The total length is restricted to 44 characters after substitution. If symbols are concatenated with other symbols or hard coded characters, a period must separate the items. The Application Performance Analyzer defined user symbols are:

- &USERID - the user ID of the user requesting the export.
- &JOB - the name of the address space associated with the observation request selected by the user for export.

  If the job name is a dash (-) as it would be for DB2 or IMS Multiple Address Space Support, then $MASS$ will be substituted for the job name.

  If the job name contains a percent sign (%) for Multi-job support, then $MJOB$ will be substituted for the job name.
- &SAMP - the observation request number of the request selected by the user for export. The value substituted will be in the form **R**nnnn where nnnn is the request number.

Note: System symbol &JOBNAME will substitute the name of the Application Performance Analyzer started task, not the observation request's job name.

Example:

ExportDSN=&SYSNAME..CAZ0.&JOB..D&YYMMDD..&SAMP..XMIT

Default: none.

ExportSMSClass
Optional.

This setting is used to assign the SMS class values that Application Performance Analyzer is to use to allocate the export file data sets. It overrides any values specified in the DefaultSMSClass setting in CONFIG BASIC. Omitted positional parameters use the corresponding positional parameter values from DefaultSMSClass.

The syntax for this setting is ExportSMSClass=(*mgmt,stg,data*) where the positional parameters are:
- *mgmt* = The management class name. It must be one to eight alphanumeric or national characters, as defined by the storage administrator at your installation.
- *stg* = The storage class name. It must be one to eight alphanumeric or national characters, as defined by the storage administrator at your installation.
- *data* = The data class name. It must be one to eight alphanumeric or national characters, as defined by the storage administrator at your installation.

ExtractorsOnByDefault
Optional.

Specifies defaults for specific data extractors that are on for a NEW command line measurement request. (NEW entered as a line command on an existing request will ignore this parameter, and clone the target request's extractors as usual). The data extractor names are listed separated by commas. Valid values are: CICS, CICS+, DB2, DB2+, DB2V, CDB2, IMS, IMS+, JAVA, MQ, and WAS. For example, to have CICS, DB2 and DB2+ turned on by default, code the following:

ExtractorsOnByDefault=(CICS,DB2,DB2+)

The WAS extractor is mutually exclusive with the following extractors: CICS+, IMS, IMS+, MQ, ADA, and NAT. An error will occur if you attempt to include WAS with these extractors.

The parameter AllExtractorsOn, if set to YES, will override this parameter and turn on all extractors.

Default: NO.

FillDefaults

Optional.

Specifies whether the installation default values for Number of Samples (NSamples) and Duration (SampleDur) are automatically completed on all NEW and TNEW requests. YES and NO are the only valid values.

Default: NO.

HFSReqLim

Optional.

Specifies the limit for the number of HFS files to be observed in each observation request. HFSReqLim specifies the maximum number of HFS files for the entire observation request. The value must be a number from 1 to 64000. Use this parm if you wish to either limit the number of HFS files to be observed, or increase the number above the default maximum of 500. Measuring more HFS file affects the performance of the measurement.

Default: 500

IMSI Optional.

This is used to specify that the IMS Intercept should be turned on. Specify IMSI=YES to turn on this feature. Turning this feature on makes the IMS+ data extractor available, which allows additional IMS data to be collected.

Default: NO.

Note: Running Application Performance Analyzer measurements with the IMS+ data extractor turned on causes each IMS call to be intercepted to collect additional data. This might have a small impact on the performance of the target address space. Care should be taken when using this feature with other products that also intercept IMS calls because unpredictable results might occur. You might want to restrict its use in production environments. You can use Access Rules to restrict the IMS+ feature to specific users. See UseIMSPlus in "Access rule descriptions" on page 22.

IMSIMaxTraceSize

Optional.

Specifies the number of IMS+ DLI call interceptions for which full details will be written to the sample file. Collecting full details on every interception allows the I02 and I03 Timeline reports to report exact times for all DLI calls and IMS transactions. The I02 and I03 reports will be truncated at the number of calls specified in IMSIMaxTraceSize. The IMS+ data extractor continues to collect the data it requires for the other reports for the duration of the measurement. Use this parameter if you need more or less than the default 100000 entries on the I02 and I03 reports. It has a range of 0 to 9999000, and must be specified in increments of 1000. Larger values will increase report load time and can cause dataspace overflows. This parameter can also specify whether the maximum trace size field is to be displayed on Panel 2 - Options, where users can change the value when creating or modifying observation requests.

Examples:

- To collect full details for 50,000 IMS+ DLI call interceptions where this value cannot be changed by the user on Panel 2, code the following:

  `IMSIMaxTraceSize=(50000,N)`

- To collect full details for 100,000 IMS+ DLI call interceptions where this value can be changed by the user on Panel 2, code the following:

  `IMSIMaxTraceSize=(100000,Y)`

Default: (100000,N)

J9VMEnable

Optional.

Specifies whether programs running under Java are to be sampled using a dynamically loaded JVMTI agent. Specify YES or 64 if you require this ability. A value of YES launches the 31-bit version of Java in the Application Performance Analyzer started task. A value of 64 launches the 64-bit version of Java in the Application Performance Analyzer started task. If YES or 64 is specified, then both JavaHome and USSHome must also be specified. Refer to Chapter 5, "Customizing Application Performance Analyzer for use with J9VM," on page 71 for further details.

Note: Regardless of the whether YES or 64 is specified, Application Performance Analyzer can sample both 31-bit and 64-bit Java applications. If you have multiple versions of Java installed, only one version can be specified for the Application Performance Analyzer started task at start up, but any version of Java supported by Application Performance Analyzer can be run in the target application.

A value of NO means that programs running under Java are to be sampled using a preloaded JVMTI agent.

Default: NO.

JavaHome

Optional.

Specifies the path name to the home directory of the most recent Java installation that supports attaching to a target J9VM. If specified, this should look similar to the following:

`JavaHome=/usr/lpp/java/J5.0`

If your system is different, specify the directory path to Java on your system. This applies only if J9VMEnable=YES or 64. Refer to Chapter 5, "Customizing Application Performance Analyzer for use with J9VM," on page 71 for further details.

Note: Ensure you have the correct Java PTF installed on your system to enable Java monitoring for Application Performance Analyzer. Refer to Chapter 5, "Customizing Application Performance Analyzer for use with J9VM," on page 71 for further details.

Default: none.

MaxMIPPercent

Optional.

Specifies the maximum percentage of MIPS (or machine instructions that a computer can execute in one second) that Application Performance Analyzer will be allowed to consume. The MaxMIPPercent parameter is used to ensure that the total number of MIPS used concurrently by

Application Performance Analyzer observation requests does not exceed the specified percentage. When this percentage is reached, new samples created will run at a reduced rate for the entire sample. The reduced rate is shown in S01. If it is already over the limit, new requests will be denied due to a resource shortage. It is recommended that this value be set somewhere between 10 and 50 percent. Typically, a lower value is used on a production machine and a higher value on a development machine.

Default: 25 percent.

MaxSampleRate
> Optional.
>
> Specifies the maximum rate at which a measurement session is permitted to sample. When the user submits a measurement request, the user's requested sampling rate is compared to this value for validity. The value is specified in samples per second.
>
> Default: 1,000 samples per second.

MaxSampleSize
> Optional.
>
> Specifies the maximum number of samples that a measurement session can record. When the user submits a measurement request, the requested number of samples is compared to this value for validity. The value is specified in samples.
>
> The maximum value you can specify is 175,000.
>
> Default: 30,000 samples.

MemObjSize
> Optional.
>
> This setting specifies the amount of above-the-bar common storage that a sampling session is permitted to allocate. This storage applies to each sampling session that captures SMF records. Refer to the WAS and DB2I CONFIG SAMPLE settings. The memory is allocated when sampling begins and is freed when sampling completes. The value is specified in megabytes and ranges from 0 to 9999.
>
> A value of 0 (zero) indicates that a shared memory object is not to be allocated.
>
> Default: 0.
>
> Refer to "Determining the amount of above-the-bar common storage for the IEFU83 and IEFU84 exits" on page 15 for guidelines on determining the value.

MLPALibs
> Optional.
>
> Application Performance Analyzer can find PLPA library names, but not MPLA library names. This parameter is used to specify up to 10 MPLA loadlib DSNs to be searched for MLPA modules. Add a new MLPALibs parameter for each data set name in the format:
> `MLPALibs=(your.dataset.name)`

MultiJobLimit
> Optional.
>
> Specifies the limit for the number of jobs allowed in a multiple job request.

Default: 20

Natural

Optional.

Specify YES to make the Natural data extractor available.

Default: NO.

NSamples

Optional.

Specifies the default number of samples that a measurement session collects during the measurement session. This value is used if the user omits a value when he creates a measurement request. Normally, the user would enter this value.

Default: 1,000 samples.

SampleDSN

Optional.

Specifies up to 50 characters to be used to name sample data sets. The value in this setting overrides the Application Performance Analyzer generated sample data set name. The name may consist of system symbols and Application Performance Analyzer defined user symbols, which are substituted with values at sample data set allocation time. The total length is restricted to 44 characters after substitution.

The Application Performance Analyzer defined user symbols are:

- &USERID - the user ID of the user requesting the sample
- &JOB - the name of the address space being sampled
- &SAMP - the observation request number being created, this will be in the form Rnnnn where nnnn is the sample file number.

Note: System symbol &JOBNAME will substitute the name of the Application Performance Analyzer started task, not the job being sampled.

Example: `SampleDSN=SYS2.CAZ0.&SYSNAME..&JOB..D&LYYMMDD..&SAMP`

Default: none.

SampleDur

Optional.

Specifies the default length of time that a measurement session collects sampling data. This value is used if the user omits a value when he creates measurement request. The value is specified in seconds. Normally, the user would enter this value.

Default: 60 seconds.

SampleHLQ

Optional.

Specifies the high-level qualifier to be used when creating sample data sets.

SampleHLQ can include system symbols for any qualifier. At sample data set allocation time, the current value of the system symbol will be substituted.

Example when the value of &SYSNAME is SYS1:

```
SampleHLQ=&SYSNAME..CAZ0     resolves to SYS1.CAZ0
SampleHLQ=CAZ0.&SYSNAME      resolves to CAZ0.SYS1
```

SampleHLQ can also include the keyword &USERID as the first qualifier. At sample data set allocation time the TSO userid of the user requesting the sample will be substituted for this variable.

SampleHLQ must comply with the JCL rule for specifying a data set name. The length of SampleHLQ is still restricted to 17 characters, including the substitutions.

If this parameter is not specified, the value that you specify for HLQ on the CONFIG BASIC statement will be used.

Default: none.

SMSClass

Optional.

This setting is used to assign the SMS class values that Application Performance Analyzer is to use to allocate the sample file data sets. It overrides the any values specified in the DefaultSMSClass setting in CONFIG BASIC. Omitted positional parameters use the corresponding positional parameter values from DefaultSMSClass.

The syntax for this setting is SMSClass=(*mgmt,stg,data*) where the positional parameters are:

- *mgmt* = The management class name. It must be one to eight alphanumeric or national characters, as defined by the storage administrator at your installation.

- *stg* = The storage class name. It must be one to eight alphanumeric or national characters, as defined by the storage administrator at your installation.

- *data* = The data class name. It must be one to eight alphanumeric or national characters, as defined by the storage administrator at your installation.

USSHome

Optional.

Specifies the USS directory in which the Application Performance Analyzer USS components have been installed. If specified, this should look similar to the following:

```
USSHome=#PathPrefix/usr/lpp/apa/v11r1
```

where `#PathPrefix` is the appropriate high level directory name. This `#PathPrefix` string must match the specification for `#PathPrefix` in the CAZWISMK installation job found in hlq.SCAZSAMP. This applies only if J9VMEnable=YES or 64. Refer to Chapter 5, "Customizing Application Performance Analyzer for use with J9VM," on page 71 for further details.

Default: none.

USSMaxCM

Optional.

This setting limits the number of USS address spaces that can be measured concurrently. It is intended to control the number of observations that would be initiated due to measured parent processes spawning or forking new processes. It has a range of 1 to 255. It is ignored if `USSSpwnEnable=NO`.

This is a system-wide limit for all requests initiated by all users.

This setting causes a control block to be allocated in ECSA. The size can be computed as follows: 80 + (MaxUSSCM x 48).

The control block is created once only and remains for the duration of the IPL. Its size can only be altered by re-IPLing the system.

If you change this value and recycle the STC, message CAZ0080W will be issued if the value has been increased. The increased value is ignored.

Default: 48

USSMaxObsPReq
Optional.

This parameter limits the number of *child* address spaces originating from the target address space that can be measured by a request. In this sense *child* address spaces are those created using the spawn, fork or exec functions.

A value of zero prevents any child address space from being measured. The maximum value is 255. This value cannot exceed the value specified in USSMaxCM.

The parameter is ignored if USSSpwnEnable=NO.

Default: 10

USSSpwnEnable
Optional.

If set to YES, this setting will enable Application Performance Analyzer to measure USS address spaces that are spun off or substeps that are executed by the original measured (parent) address space.

Default: NO.

Note: USS applications spawn many address spaces. You might need to set USSSpawnEnable to NO if address spaces that are created are not being sampled.

WAS Optional.

This setting controls Application Performance Analyzer's WebSphere Application Server (WAS) support. Application Performance Analyzer collects WebSphere activity data from the SMF record type 120 subtype 9 records. The syntax is as follows:

WAS=(YES|NO,CONTROLSMF=YES|NO)

The first positional parameter specifies whether Application Performance Analyzer WAS support is to be enabled. The default value is NO. If YES is specified, WAS support is enabled and the WAS data extractor is available to collect WebSphere information during a measurement. If YES is specified, a non-zero value is required for the MemObjSize keyword parameter.

The CONTROLSMF keyword parameter specifies whether Application Performance Analyzer is to control recording SMF record type 120 subtype 9 records. The default is NO. If NO is specified or allowed to default, then it is up to the system administrator to ensure that WAS SMF recording is enabled for the WebSphere Server(s) that are to be sampled. If YES is specified, Application Performance Analyzer will enable SMF recording when a WAS sample session begins and disable SMF recording when sampling has completed. If NO is specified for WAS in the first positional parameter or it is allowed to default, then this parameter is ignored.

Default:

```
WAS=(NO,CONTROLSMF=NO)
```

WLMI   Optional.

This is used to specify whether the Application Performance Analyzer WLM Intercept is to be turned on or off. YES and NO are the only valid values. Specify WLMI=YES to turn on this feature. WLMI=YES allows you to measure specific DB2 stored procedures and user-defined functions. It also makes the collateral DB2 (CDB2) data extractor available. When the CDB2 data extractor is selected, Application Performance Analyzer can measure DB2 activity in stored procedures and user-defined functions that are invoked by the application being measured.

When WLMI=YES is specified, Application Performance Analyzer ensures that no other third party product is already intercepting WLM calls. If one is found, a diagnostic error message is displayed, identifying the job name of the third party product that is already intercepting WLM and ignore the WLMI option. To avoid this, Application Performance Analyzer should always be started before any other product that intercepts WLM calls.

Running Application Performance Analyzer measurements with the WLMI data extractor turned on causes each WLM call to be intercepted to collect additional data. This can have a small impact on the performance of the target address space. You might wish to restrict its use in production environments. You can use Access Rules to restrict the measurement of specific stored procedures and user-defined functions, and the use of the CDB2 data extractor to specific users. For additional information, refer to "Suggested Application Performance Analyzer rules" on page 24.

Default: NO.

CONFIG LOGFILE statement

Logging records activity within the Application Performance Analyzer started task. Log data will always be written to SYSOUT, but you have the option of also writing this data to log data sets. LOGFILE settings allow you manage these data sets. The following settings might be specified:

Note: The alphabetical order specified below is not the order of the parameters in the configuration file.

LogGDG   Optional.

If you have specified LogToFile=YES, then LogGDG allows you to specify whether the log data sets will be allocated as GDGs or normal sequential data sets. If you specify LogGDG=YES, you must first create the Generation Data Group Index using an index name of *yourhlq*.LOG (or LogHLQ.LOG if you choose to use LogHLQ). YES and NO are the only valid values.

Default: NO.

Note: A GDG log file cannot be shared by multiple instances of Application Performance Analyzer. A second started task with the same GDG log file will fail on start up with messages IKJ56225I and CAZ1030S. A unique GDG log file must be defined for each instance of Application Performance Analyzer.

LogHLQ   Optional.

Specifies the high-level qualifier to be used when creating log data sets. LogHLQ must comply with the JCL rule for specifying a data set name. It can be no more than 17 characters long. A data set will only be created if LogToFile=YES is specified.

If this parameter is not specified, the value that you specify for HLQ on the CONFIG BASIC statement will be used.

Default: none.

LogPRIM

Optional.

If you have specified LogToFile=YES, then LogPRIM allows you to specify the size of each log data set that is allocated. No secondary specification is permitted; therefore, when the log data set has filled, it will be closed and a new log data set will be allocated.

Default: 15.

LogToFile

Optional.

Specifies whether you want log data to be written to permanent data sets. YES and NO are the only valid values.

Default: NO.

Note: It is recommended that LogToFile be set to YES. The log can be used to help diagnose problems.

LogUNIT

Optional.

If you have specified LogToFile=YES, then LogUNIT allows you to specify whether you want the log data set to be allocated in Tracks or Cylinders. T and C are the only valid values.

Default: T.

SMSClass

Optional.

This setting is used to assign the SMS class values that Application Performance Analyzer is to use to allocate the log file data sets. It overrides the any values specified in the DefaultSMSClass setting in CONFIG BASIC. Omitted positional parameters use the corresponding positional parameter values from DefaultSMSClass.

The syntax for this setting is SMSClass=(*mgmt,stg,data*) where the positional parameters are:

- *mgmt* = The management class name. It must be one to eight alphanumeric or national characters, as defined by the storage administrator at your installation.
- *stg* = The storage class name. It must be one to eight alphanumeric or national characters, as defined by the storage administrator at your installation.
- *data* = The data class name. It must be one to eight alphanumeric or national characters, as defined by the storage administrator at your installation.

CONFIG OPER statement

These settings pertain to operational parameters and should not be changed by the user. These should only be changed on the advice of IBM. Application Performance Analyzer ships with only one parameter specified, which is DiagLevel=1.

Configuring external security (RACF)

This section describes how to use RACF, or a RACF-compatible external security product. When using an external security product, you must code `Security=External` in the CONFIG BASIC section in CAZCNFG0, see "Security" parameter in "CONFIG BASIC statement" on page 26.

Defining RACF profiles

When using external security, Application Performance Analyzer functions are protected using general resource profiles – defined in the default `FACILITY` class, or in a user-specified class that has been previously defined (it can be an existing class or a new class defined specifically for Application Performance Analyzer).

The profile name is specified as:

HLQ.MVS_system_name.action.object

- `HLQ` is an optional high-level qualifier for the resource profile
- `MVS_System_name` is the target or servicing system
- `Action` is the Application Performance Analyzer action or function to be performed
- `Object` identifies the entity to be accessed

HLQ

The optional high-level qualifier can be used to define generic rules for all members of a sysplex.

MVS system name

The MVS system name, as it applies to an observation, represents the system on which the observation was finally serviced. In a non-sysplex environment the system name is the same as the name of the system where the observation request was made.

In a sysplex environment, the intent of the system name is to control an observation request based on the target system or servicing system name, which prevents a user from adding an observation request that targets a job that will execute on a system to which that the user does not have access.

In determining the resource profile to be used for authorization, Application Performance Analyzer derives the system name based on the status of the observation, the action to be performed, and the sysplex options specified on the observation request. For observation requests that are to be scheduled or are in schedule status in which the target system is specified, the target system name is used. If the target system is specified as `*` or ALL, an attempt is made to evaluate the user's authority based on the current system name and the system names of the systems in the sysplex group – if authority cannot be determined definitively, the request is provisionally accepted and user authorization will be performed again when the request runs. For completed or active observations, the system name used will be the servicing system name. See "Examples" on page 50 for specific examples of how Application Performance Analyzer derives the system name.

Action

The action represents the type of access involved in the function being performed. The types of access are:

- `VIEW` to be able to view reports for observations
- `UPDATE` to be able to update a scheduled observation
- `DELETE` to be able to delete an observation
- `MEASURE.JOB` for measuring a batch job
- `MEASURE.STC` for measuring a started task
- `MEASURE.TSU` for measuring TSO users
- `USE.DB2PLUS` to be able to request a DB2 PLUS measurement
- `USE.IMSPLUS` to be able to request an IMS PLUS measurement
- `MONITOR.JOB` for threshold-monitoring a batch job
- `MONITOR.STC` for threshold-monitoring a started task
- `MONITOR.TSU` for threshold-monitoring TSO users
- `USE.WLMI` for using the WLM Intercept to request a specific DB2 stored procedure or user-defined function measurement and/or turn on the collateral DB2 (CDB2) data extractor
- `USE.CICSPlus` to be able to request a CICS PLUS measurement
- `ADMINISTERPRODUCT` to be able to perform administrator functions. This includes updating the common list of source files in the common data store and seeing expiry days warning for all users' observations. This authority is needed to update the ISPF A05 Source Mapping Common List and the GUI Source Program Mapping Repository Common list. It is not necessary to give this authority to users to maintain their personal list of source files through the ISPF A04 Source Mapping Dataset List and the GUI Source Program Mapping Repository Personal list. Without this authority, when expiry days warning is enabled, users will see the warning for their own observations only.
- USE.WAS to be able to request a WAS measurement

By default, all users have access to VIEW, UPDATE and DELETE any observation they create, as well as monitor their own TSO userid. Therefore, profiles do not need to be specified for a user to have access to their own observations. All other types of access are protected by default and must have a profile written for any user to have access to them. Access is denied if RACF denies access or if RACF cannot find an applicable profile.

Objects

The types of objects are:

- User ID name or pattern
- Job Name or pattern

The User ID or Job Name pattern follows standard RACF syntax and can be a full name, a name pattern, a partial name with an *, or just an * to indicate any name is acceptable. See the *IBM z/OS Security Server RACF Security Administrator's Guide*, (SA22-7683), for coding details on generic profiles.

Profile names

For any authorization call, Application Performance Analyzer determines the profile name to be used based on the derived MVS system name, status, and action. The following table lists some of the values for these criteria and the resulting profile name in a sysplex group.

Status	System name	Action	RACF profile names
Complete	Servicing system name	New (line command)	*<hlq.>service_system_name*.VIEW.*userid* *<hlq.>service_system_name*.UPDATE.*userid* *<hlq.>service_system_name*.MEASURE.*xxx.jobname* *<hlq.>service_system_name*.MONITOR.*xxx.jobname* *<hlq.>service_system_name*.USE.*xxx*
		Submit	*<hlq.>service_system_name*.VIEW.*userid* *<hlq.>service_system_name*.UPDATE.*userid* *<hlq.>service_system_name*.MONITOR.*xxx.jobname* *<hlq.>service_system_name*.MEASURE.*xxx.jobname* *<hlq.>service_system_name*.USE.*xxx*
		Report/ Select	*<hlq.>service_system_name*.VIEW.*userid*
		Detail	*<hlq.>service_system_name*.VIEW.*userid*
		Delete	*<hlq.>service_system_name*.DELETE.*userid*
		Keep	*<hlq.>service_system_name*.UPDATE.*userid*
		Administer Product	*<hlq.>service_system_name*.ADMINISTERPRODUCT
Scheduled	Target system name specified	New (line command)	*<hlq.>target_system_name*.VIEW.*userid* *<hlq.>target_system_name*.UPDATE.*userid* *<hlq.>target_system_name*.MEASURE.*xxx.jobname* *<hlq.>target_system_name*.MONITOR.*xxx.jobname* *<hlq.>target_system_name*.USE.*xxx*
		Modify	*<hlq.>target_system_name*.VIEW.*userid* *<hlq.>target_system_name*.UPDATE.*userid* *<hlq.>target_system_name*.MEASURE.*xxx.jobname* *<hlq.>target_system_name*.MONITOR.*xxx.jobname* *<hlq.>target_system_name*.USE.*xxx*
		Detail	*<hlq.>target_system_name*.VIEW.*userid*
		Delete	*<hlq.>target_system_name*.DELETE.*userid*
Scheduled	* or all specified	New (line command)	*<hlq.>current_system_name*.VIEW.*userid* *<hlq.>all_sysplex_system_names*.MEASURE.*xxx.jobname* *<hlq.>all_sysplex_system_names*.MONITOR.*xxx.jobname* *<hlq.>all_sysplex_system_names*.USE.*xxx*
		Modify	*<hlq.>current_system_name*.VIEW.*userid* *<hlq.>current_system_name*.UPDATE.*userid* *<hlq.>all_sysplex_system_names*.MEASURE.*xxx.jobname* *<hlq.>all_sysplex_system_names*.MONITOR.*xxx.jobname* *<hlq.>all_sysplex_system_names*.USE.*xxx*
		Detail	*<hlq.>originating_system_name*.VIEW.*userid*
		Delete	*<hlq.>originating_system_name*.DELETE.*userid*
Active	Servicing system name	New (line command)	*<hlq.>service_system_name*.VIEW.*userid* *<hlq.>service_system_name*.MEASURE.*xxx.jobname* *<hlq.>service_system_name*.MONITOR.*xxx.jobname* *<hlq.>service_system_name*.USE.*xxx*

Status	System name	Action	RACF profile names

		Submit	*<hlq.>service_system_name*.VIEW.*userid*
			<hlq.>service_system_name.MEASURE.
			      xxx.jobname
			<hlq.>service_system_name.MONITOR.
			      xxx.jobname
			<hlq.>service_system_name.USE.*xxx*

		Detail	*<hlq.>service_system_name*.VIEW.*userid*
		Cancel	*<hlq.>service_system_name*.UPDATE.*userid*
		RTM	*<hlq.>service_system_name*.VIEW.*userid*

Examples

The following table provides examples of how a system name is derived. The examples assume the following:

- SYSTEMD is a development system
- SYSTEMP is a production system
- SYSTEMD and SYSTEMP are part of the same sysplex group
- There is one instance of Application Performance Analyzer running on SYSTEMD and one running on SYSTEMP, and both instances are part of the same Application Performance Analyzer sysplex group
- Users DEV1 and DEV2 have full access to all observations owned by DEV* users whose servicing system was SYSTEMD
- DEV1 and DEV2 have full access to creating observations for address spaces that execute on SYSTEMD
- When signed on to SYSTEMP, users DEV1 and DEV2 are limited to sampling their own TSO session and updating those observations
- Users DEV1 and DEV2 can view, keep, or delete any observation that was serviced by SYSTEMD
- User ADM1 is an administrator who has full access to all observations on all systems

User ID	Action	Logon system	Target system	Servicing system	Comments
DEV1	Add new observation	SYSTEMD	SYSTEMD	n/a	Access granted based on target system.
DEV1	Add new observation	SYSTEMD	SYSTEMP	n/a	Access denied based on target system.
DEV1	Add new observation	SYSTEMD	* (any)	n/a	Access provisionally granted. Access will be checked again based on the system where the job finally executes.
DEV1	Delete completed observation	SYSTEMD	SYSTEMD	SYSTEMD	Access granted based on servicing system.
DEV1	Delete completed observation	SYSTEMD	* (any)	SYSTEMP	Access denied based on servicing system.
DEV1	Delete completed observation of DEV1 TSO session	SYSTEMD	SYSTEMP	SYSTEMP	Access granted. User is allowed to update observation that he owns.

User ID	Action	Logon system	Target system	Servicing system	Comments
DEV1	Delete scheduled observation owned by DEV2	SYSTEMD	* (any)	n/a	Access granted. Since the system name is not known, SYSTEMD (the logon system) will be used to determine the resource profile name.
DEV1	Delete scheduled observation owned by DEV2	SYSTEMD	SYSTEMD	n/a	Access granted. Target system used to determine the resource profile name.
DEV1	Modify scheduled observation	SYSTEMD	SYSTEMD	n/a	Access granted based on target system.
DEV1	Modify scheduled observation	SYSTEMD	* (any)	n/a	Access denied based on logon system and owner ADM1.
DEV1	Add new observation	SYSTEMP	SYSTEMD	n/a	Access granted based on target system.
DEV1	Add new observation	SYSTEMP	SYSTEMP	n/a	Access denied based on target system.
DEV1	Add new observation to measure DEV1's active TSO session	SYSTEMP	SYSTEMP	SYSTEMP	Access granted. User is allowed to measure his own TSO session.
DEV1	Delete completed observation owned by DEV2	SYSTEMP	* (any)	SYSTEMP	Access granted based on servicing system.

The following examples illustrate how to define RACF profiles based on the examples in the table above. The examples assume the following:

- The RACF group, ADMGRP, contains user IDs ADM1 and ADM2
- The RACF group, DEVGRP, contains user IDs DEV1 and DEV2
- SYSTEMD and SYSTEMP are part of the same sysplex group
- The RACF database is shared by SYSTEMD and SYSTEMP
- There is one instance of Application Performance Analyzer running on SYSTEMD and one instance of Application Performance Analyzer running on SYSTEMP

The following RACF commands define the resource profiles that are dependant on the MVS system name.

```
RDEFINE FACILITY SYSTEMD.VIEW.* UACC(NONE)
RDEFINE FACILITY SYSTEMD.UPDATE.* UACC(NONE)
RDEFINE FACILITY SYSTEMD.DELETE.* UACC(NONE)
RDEFINE FACILITY SYSTEMD.MEASURE.** UACC(NONE)
RDEFINE FACILITY SYSTEMD.MONITOR.** UACC(NONE)

RDEFINE FACILITY SYSTEMD.VIEW.DEV% UACC(NONE)
RDEFINE FACILITY SYSTEMD.UPDATE.DEV% UACC(NONE)
RDEFINE FACILITY SYSTEMD.DELETE.DEV% UACC(NONE)
```

```
RDEFINE FACILITY SYSTEMD.USE.DB2PLUS UACC(NONE)
RDEFINE FACILITY SYSTEMD.USE.IMSPLUS UACC(NONE)
RDEFINE FACILITY SYSTEMD.USE.WLMI UACC(NONE)
RDEFINE FACILITY SYSTEMD.USE.CICSPlus UACC(NONE)
RDEFINE FACILITY SYSTEMD.USE.WAS UACC(NONE)

RDEFINE FACILITY SYSTEMP.VIEW.* UACC(NONE)
RDEFINE FACILITY SYSTEMP.UPDATE.* UACC(NONE)
RDEFINE FACILITY SYSTEMP.DELETE.* UACC(NONE)
RDEFINE FACILITY SYSTEMP.MEASURE.** UACC(NONE)
RDEFINE FACILITY SYSTEMP.MONITOR.** UACC(NONE)

RDEFINE FACILITY SYSTEMP.VIEW.DEV1 UACC(NONE)
RDEFINE FACILITY SYSTEMP.VIEW.DEV2 UACC(NONE)
RDEFINE FACILITY SYSTEMP.UPDATE.DEV1 UACC(NONE)
RDEFINE FACILITY SYSTEMP.UPDATE.DEV2 UACC(NONE)
RDEFINE FACILITY SYSTEMP.DELETE.DEV1 UACC(NONE)
RDEFINE FACILITY SYSTEMP.DELETE.DEV2 UACC(NONE)

RDEFINE FACILITY SYSTEMP.USE.DB2PLUS UACC(NONE)
RDEFINE FACILITY SYSTEMP.USE.IMSPLUS UACC(NONE)
RDEFINE FACILITY SYSTEMP.USE.WLMI UACC(NONE)
RDEFINE FACILITY SYSTEMP.USE.CICSPlus UACC(NONE)
RDEFINE FACILITY SYSTEMP.USE.WAS UACC(NONE)
```

The following RACF commands permit users ADM1 and ADM2 to view, update, or delete any observations. The RACF commands also allow users ADM1 and ADM2 to measure or threshold-monitor any address space type in which SYSTEMD is the servicing system.

```
PERMIT SYSTEMD.VIEW.* CLASS(FACILITY) ID(ADMGRP) ACCESS(READ)
PERMIT SYSTEMD.UPDATE.* CLASS(FACILITY) ID(ADMGRP) ACCESS(READ)
PERMIT SYSTEMD.DELETE.* CLASS(FACILITY) ID(ADMGRP) ACCESS(READ)
PERMIT SYSTEMD.MEASURE.** CLASS(FACILITY) ID(ADMGRP) ACCESS(READ)
PERMIT SYSTEMD.MONITOR.** CLASS(FACILITY) ID(ADMGRP)ACCESS(READ)
```

The following RACF commands permit users DEV1 and DEV2 to view, update, or delete any observations owned by either DEV1 or DEV2 in which SYSTEMD is the servicing system. The RACF commands also allow users DEV1 and DEV2 to measure or threshold-monitor any address space type in which the target system or servicing system is SYSTEMD.

```
PERMIT SYSTEMD.VIEW.DEV% CLASS(FACILITY) ID(DEVGRP) ACCESS(READ)
PERMIT SYSTEMD.UPDATE.DEV% CLASS(FACILITY) ID(DEVGRP) ACCESS(READ)
PERMIT SYSTEMD.DELETE.DEV% CLASS(FACILITY) ID(DEVGRP) ACCESS(READ)
PERMIT SYSTEMD.MEASURE.** CLASS(FACILITY) ID(DEVGRP) ACCESS(READ)
PERMIT SYSTEMD.MONITOR.** CLASS(FACILITY) ID(DEVGRP) ACCESS(READ)
```

The following RACF commands permit all users to activate DB2Plus, IMSPlus, WLMI, CICSI and WAS for observations in which the target system or servicing system is SYSTEMD.

```
PERMIT SYSTEMD.USE.DB2PLUS CLASS(FACILITY) ACCESS(READ)
PERMIT SYSTEMD.USE.IMSPLUS CLASS(FACILITY) ACCESS(READ)
PERMIT SYSTEMD.USE.WLMI CLASS(FACILITY) ACCESS(READ)
PERMIT SYSTEMD.USE.CICSPlus CLASS(FACILITY) ACCESS(READ)
PERMIT SYSTEMD.USE.WAS CLASS(FACILITY) ACCESS(READ)
```

The following RACF commands permit users ADM1 and ADM2 to view, update, or delete any observations. The RACF commands also allow users ADM1 and ADM2 to measure or threshold-monitor any address space type in which the servicing system is SYSTEMP.

Note: By default, all users can measure their own TSO sessions.

```
PERMIT SYSTEMP.VIEW.* CLASS(FACILITY) ID(ADMGRP) ACCESS(READ)
PERMIT SYSTEMP.UPDATE.* CLASS(FACILITY) ID(ADMGRP) ACCESS(READ)
PERMIT SYSTEMP.DELETE.* CLASS(FACILITY) ID(ADMGRP) ACCESS(READ)
PERMIT SYSTEMP.MEASURE.** CLASS(FACILITY) ID(ADMGRP) ACCESS(READ)
PERMIT SYSTEMP.MONITOR.** CLASS(FACILITY) ID(ADMGRP) ACCESS(READ)
```

The following RACF commands permit users ADM1 and ADM2 to activate DB2Plus,
IMSPlus, WLMI, CICSI, and WAS for observation in which the target system or the
servicing system is SYSTEMD.

```
PERMIT SYSTEMD.USE.DB2PLUS CLASS(FACILITY) ID(ADMGRP) ACCESS(READ)
PERMIT SYSTEMD.USE.IMSPLUS CLASS(FACILITY) ID(ADMGRP) ACCESS(READ)
PERMIT SYSTEMD.USE.WLMI CLASS(FACILITY) ID(ADMGRP) ACCESS(READ)
PERMIT SYSTEMD.USE.CICSPlus CLASS(FACILITY) ID(ADMGRP) ACCESS(READ)
PERMIT SYSTEMD.USE.WAS CLASS(FACILITY) ID(ADMGRP) ACCESS(READ)
```

The following RACF commands permit users ADM1 and ADM2 to maintain the
common list of source files in the Common Data Store and to see the expiry days
warning for all users' observations on SYSTEMD and SYSTEMP:

```
PERMIT SYSTEMD.ADMINISTERPRODUCT CLASS(FACILITY) ID(ADMGRP) ACCESS(READ)
PERMIT SYSTEMP.ADMINISTERPRODUCT CLASS(FACILITY) ID(ADMGRP) ACCESS(READ)
```

Suggested profiles
Some basic RACF profiles and explanations are shown here:

1. `RDEFINE FACILITY SYSTEMD.MEASURE.JOB.* UACC(READ)`

   Any user can measure jobs with any jobname when the servicing system is
   SYSTEMD.

2. `RDEFINE FACILITY SYSTEMD.MEASURE.STC.* UACC(READ)`

   Any user is allowed to measure a started task when the servicing system is
   SYSTEMD.

3. `RDEFINE FACILITY SYSTEMD.USE.DB2PLUS UACC(READ)`

   Any user is allowed to turn on the DB2+ data extractor when the servicing
   system is SYSTEMD.

4. `RDEFINE FACILITY SYSTEMD.USE.IMSPLUS UACC(READ)`

   Any user is allowed to turn on the IMS+ data extractor when the servicing
   system is SYSTEMD.

5. `RDEFINE FACILITY SYSTEMD.USE.WLMI UACC(READ)`

   Any user is allowed to measure a specific DB2 stored procedure or user-defined
   function and/or to turn on the collateral (CDB2) data extractor when the
   servicing system is SYSTEMD.

6. `RDEFINE FACILITY SYSTEMD.USE.CICSPlus UACC(READ)`

   Any user is allowed to turn on the CICS+ data extractor when the servicing
   system is SYSTEMD.

7. `RDEFINE FACILITY SYSTEMD.USE.WAS UACC(READ)`

   Any user is allowed to turn on the WAS data extractor when the servicing
   system is SYSTEMD.

External security when Application Performance Analyzer is in a sysplex group
When Application Performance Analyzer operates as a member of a sysplex group,
it must use the same security method (Internal, External, or External(Compat)) as
the other members of the group. The first Application Performance Analyzer
member to initialize sets the security method, all other Application Performance
Analyzer members that start must use the same method or they will fail. In
addition, it is highly recommended that the External security method be used by

all members. `Internal` and `External(Compat)` will not produce desired results and are only supported for backward compatibility.

For more information about external security, see Appendix C, "Sysplex External Security Conversion Aid," on page 101.

Using the SecurityHLQ and SecurityClassName parameters

When using the SecurityHLQ parameter, the SecurityClassName parameter must also be specified. For example, using the parameters:

```
SecurityHLQ=CAZ
SecurityClassName=CACLASS1
```

The VIEW rules for the parameters above would be written as:

```
RDEFINE CACLASS1 CAZ.SYSTEMD.VIEW.* UACC(NONE)
PERMIT CAZ.SYSTEMD.VIEW.* CLASS(CACLASS1) ID(ADMGRP) ACCESS(READ)
```

To allow VIEW access on all systems in the sysplex, the rule could be written as:

```
PERMIT CAZ.*.VIEW.* CLASS(CACLASS1) ID(ADMGRP) ACCESS(READ)
```

The SecurityHLQ allows the rules to be written for all members of the sysplex rather than having to code a rule for each member.

For more information about external security, see Appendix C, "Sysplex External Security Conversion Aid," on page 101.

Pre-allocating a checkpoint data set

In most cases, you can just let Application Performance Analyzer dynamically allocate and initialize the checkpoint file, which Application Performance Analyzer will try to do the first time the Application Performance Analyzer started task runs.

Your installation might have special requirements such as to what volume the checkpoint file is allocated, or under what SMS rules to allocate the checkpoint file.

Under these conditions, use the JCL provided in sample job CAZCHKPT in *hlq*.SCAZSAMP to pre-allocate a new checkpoint file prior to starting the Application Performance Analyzer started task for the first time. Application Performance Analyzer will detect the file and initialize it. To pre-allocate a checkpoint file, edit and submit member CAZCHKPT in `hlq.`SCAZSAMP. Consult the instructions in the sample job for more information.

The following rules apply to allocating the checkpoint file:
1. The file must reside on a direct access device.
2. It must not be allocated as a compressed format data set.
3. It must not be allocated as an extended-format data set.
4. The data set name (*yourhlq*.`checkpointDSN`) must conform to the rules specified in the "CheckpointDSN" parameter described in "CONFIG BASIC statement." For more information, see "CONFIG BASIC statement" on page 26.

Pre-allocating a common data store file

In most cases, you can allow Application Performance Analyzer to dynamically allocate and initialize the common data store file, which Application Performance Analyzer will try to do the first time the Application Performance Analyzer started task runs.

Your installation might have special requirements such as to what volume the common data store file is allocated, or under what SMS rules to allocate the common data store file. Under these conditions, use the JCL provided in sample job CAZCDS in *hlq*.SCAZSAMP to pre-allocate a new common data store file before starting the Application Performance Analyzer started task for the first time. Application Performance Analyzer will detect the file and initialize it. To pre-allocate a common data store file, edit and submit member CAZCDS in *hlq*.SCAZSAMP. See the instructions in the sample job for more information.

The following rules apply to allocating the common data store file:
1. It must be allocated as a partitioned data set extended (PDSE).
2. The data set name (*yourhlq*.CDS) must conform to the rules specified in the "CDS" DSN sub-parameter described in "CONFIG BASIC statement" on page 26.

Updating the descriptive program attribution tables

Application Performance Analyzer comes with the descriptive program attribution (DPA) tables containing descriptions of known IBM modules. You can also create your own user DPA table to enter custom descriptions of load modules, control section names, and CICS transactions. This feature allows Application Performance Analyzer to report descriptions of customer programs or non-IBM system software modules.

DPA table entry elements

Each DPA table entry contains the following elements:

Program Name (keyword: NAME)
> This can be specified as either a control section (CSECT) name or a load module name. In most cases, load module name is specified. Support for control section name provides for additional granularity. Wildcard matching of this field is supported. For example, DSN* could be specified to provide a single description for all DB2 system components. The matching algorithm will resort to a wildcard match only if it does not find a more exact name match.

Program Group (keyword: GRP)
> A unique eight-character name defines a program group to which the program belongs. The supplied table uses a group name of CUSTOMER. The GRP keyword only needs to be specified in a CAZDPAGN SUBGROUP definition. It must match a previously defined CAZDPAGN GROUP definition, and it specifies which GROUP the SUBGROUP belongs to.

Sub Group (keyword: SUBGRP)
> A unique (within the sub groups) eight-character name defines a program sub group to which the program belongs. The supplied table uses a sub group name of CUSTOMER. SUBGRP must be specified on all CAZDPAGN entries for CSECTS, load modules, and CICS transactions.

Description (keyword: DESCRP)

> This field contains a description of the load module or control section function, or CICS transaction. In CAZDPAGN GROUP or CAZDPAGN SUBGROUP entries, it contains the description for the GROUP or SUBGROUP.

CAZDPAGN macro

The CAZDPAGN macro is used to generate assembly statements which, when assembled, produce an object module defining a DPA table. Several keyword parameters are specified in each CAZDPAGN statement. There are also a few special CAZDPAGN statements at the beginning and end of a table, which use positional parameters. The positional parameter specifies the function code and is one of the following values: PROLOG, GROUP, SUBGROUP, and END. These positional parameters are described below:

Special CAZDPAGN statements with positional parameters

PROLOG

> Generates prologue logic at the beginning of the CAZDPAGN module. This must be the first statement in the assembly source. The label field specifies the DPA table CSECT and load module name. This must be CAZUSR01. No keyword parameters are required.

GROUP

> Defines a Program Group. Placement of these statements is immediately after the PROLOG statement. The Program Group name must be coded in the label field. One keyword parameter must be specified: DESCRP. The supplied sample table uses CUSTOMER for the group name.

> Additional GROUPs can be defined, and reporting will include these GROUPs in the SYSTEM category of the CPU Usage reports. These GROUPs will only affect reporting if there are DPA entries belonging to the groups that have TYPE=SYSTEM coded.

SUBGROUP

> Defines a sub group within the Program Group. Placement of these statements is immediately after the GROUP CAZDPAGN statement. The SUBGROUP name must be coded in the label field. One keyword parameter must be specified: DESCRP. The supplied sample table uses CUSTOMER for the sub group name.

> Additional SUBGROUPs can be defined, and reporting will include these SUBGROUPs in the SYSTEM category of CPU Usage reports. These SUBGROUPs will only affect reporting if there are DPA entries belonging to the groups that have TYPE=SYSTEM coded.

END Used to mark the end of the CAZDPAGN statements in module CAZUSR01.

Standard CAZDPAGN keyword parameters

The standard CAZDPAGN entries are described here. These are entries with no positional parameter specified, and are used to define individual entries for load modules, CSECTS, or CICS transactions.

NAME

> Defines a program (load module or CSECT) name and its descriptive information. NAME is used to specify a program name (or wildcard pattern), a range of program names or a CICS Transaction. A range of

program names is specified in the format NAME=*(pgm1,pgm2)* where *pgm1* is the name (or wildcard pattern) at the start of the range and *pgm2* is the name (or wildcard pattern) at the end of the range. A CICS Transaction is specified in the same format, by specifying a TYPE=CICSTXN keyword. CICS Transactions can also be specified as a range. The keyword parameter DESCRP specifies the functional description.

SUBGRP

The SUBGRP keyword specifies the associated SUBGROUP name. For entries with TYPE=USER, this will not affect reporting, so coding SUBGRP=CUSTOMER is recommended. (Assuming the CUSTOMER SUBGROUP has been created as supplied in the sample CAZUSR01 module).

For system modules, where TYPE=SYSTEM has been coded, the SUBGRP keyword can specify a custom SUBGROUP, as long as it has been defined in a CAZDPAGN SUBGROUP statement previously.

TYPE Specifies the type. Valid values are: USER, SYSTEM, and CICSTXN. If this keyword is omitted, it will default to SYSTEM. Specify TYPE=USER if the module is an application module; it will be reported under the Application category.

Specify TYPE=SYSTEM if this is a system module; it will be reported under the System category.

Example of CAZDPAGN Macros

The following illustrates the structure of an assembly source member which defines a DPA table:

```
1 CAZUSR01   CAZDPAGN    PROLOG
2 CUSTOMER   CAZDPAGN    GROUP,DESCRP='Customer Programs'
3 CUSTOMER   CAZDPAGN     SUBGROUP,GRP=CUSTOMER,DESCRP='Customer Programs'
4            CAZDPAGN  NAME=SAMP*,DESCRP='CICS Samples',SUBGRP=CUSTOMER
           CAZDPAGN  NAME=(APPL1*,APPL4*),                          X
              DESCRP='Batch Applications',SUBGRP=CUSTOMER,TYPE=USER
5          CAZDPAGN NAME=CICSDB2,SUBGRP=CUSTOMER,                   X
              DESCRP='DB2 Test Program',TYPE=USER
           CAZDPAGN NAME=MQSAMP1,SUBGRP=CUSTOMER,                   X
              DESCRP='MQ Test Program',TYPE=USER
6          CAZDPAGN NAME=TDB2,SUBGRP=CUSTOMER,                      X
              DESCRP='DB2 Test Trans',TYPE=CICSTXN
           CAZDPAGN NAME=MQS1,SUBGRP=CUSTOMER,                      X
              DESCRP='MQ Test Trans',TYPE=CICSTXN
7          CAZDPAGN END
8          END CAZUSR01
```

Note:

1. This generates prologue code.
2. These statements define Program Groups.
3. These statements define the Sub Groups within the Program Group entries.
4. This is an example of a wildcard name. Any program name beginning with SAMP will be described as "CICS samples." In the event that a more exact name match is found, the description associated with that match will prevail.
5. This is an example of an exact name match entry.
6. This is an example of a CICS transaction definition.
7. This marks the end of the table of program entries.
8. This marks the end of the assembly file.

Implementation

A sample DPA table is supplied in the *hlq*.SCAZSAMP library as member CAZUSR01. Modify this member as required with your program entries then assemble and link this member using the JCL provided in *hlq*.SCAZSAMP in member CAZASMDP. The load module name must be CAZUSR01. If this load module is found in *hlq*.SCAZAUTH, it will be used first to look up program descriptions.

Modifying Application Performance Analyzer procedure for third party products

The REXX procedure CAZRXOEM in `hlq.SCAZEXEC` must be customized to provide access to third party repository products for source program mapping support. This requires detailed knowledge of the utility used to extract listings from the repository into a fixed block file with a 132 or 133 character record length. CAZRXOEM is invoked whenever a user specifies a repository of T for "Third party" on the Source Program Mapping panel, A01.

Note: CAZRXOEM may only specify standard TSO commands such as: ALLOCATE and CALL. ISPEXEC commands are not permitted.

Edit the member CAZRXOEM in the `hlq.SCAZEXEC` and make the following changes:

1. Comment out the line:
   ```
   QUEUE "CAZRXOEM"
   ```
2. Replace "SYS1.LINKLIB(OEMUTIL)" in the statement:
   ```
   QUEUE "CALL 'SYS1.LINKLIB(OEMUTIL)'"
   ```

   with the data set name of the library that contains the utility, and the name of the utility to be run.
3. Increase the space to more than the default allocation of one track, if needed, in the statement:
   ```
   QUEUE "INPUT SPACE(1) TRACKS"
   ```
4. Modify the ALLOC statements as required for space and format. The following statements might require changes:
   ```
   QUEUE "ALLOC SYSUT1 DSN(@repository@) SHR REUSE"
   QUEUE "ALLOC SYSUT2 DSN(@extract@)" ,
       "NEW REUSE SPACE(10 10) TRACKS" ,
       "RECFM(F B A) LRECL(133) BLKSIZE(13300) DSORG(PS)"
   QUEUE "ALLOC SYSIN DSN(@parms@) OLD"
   QUEUE "ALLOC SYSPRINT DSN(@sysprint@)" ,
       "NEW REUSE SPACE(1 1) TRACKS" ,
       "RECFM(F B A) LRECL(121) BLKSIZE(12100)"
   ```

   Change only the SPACE, RECFM, LRECL, and BLKSIZE parameters. The DSN will be filled in from the input supplied by the user when requesting a listing.
   - @repository@ is the input data set that is the repository for the listings.
   - @extract@ is the output of the utility and used as input to source program mapping.
   - @parms@ contains any input parms required.
   - @sysprint@ contains messages and control statement output from the utility, normally the SYSPRINT data set.

Starting the Application Performance Analyzer-started task

The Application Performance Analyzer-started task must be started for it to become operational.

If you have not re-IPLed after making changes to parmlib as described in step 6 on page 19 of the task for making Application Performance Analyzer available, you can start Application Performance Analyzer by entering the command 'S *procname*' on the system console, where *procname* is the Application Performance Analyzer-started task procedure name.

When you re-IPL your system, the Application Performance Analyzer started task is automatically started.

Starting the IBM Application Performance Analyzer user interface

To start IBM Application Performance Analyzer, do the following steps:

1. Enter 'EX '*hlq*.SCAZEXEC(CAZ@SPF)'' at the TSO/E command prompt (not the TSO Ready prompt).

```
-------------------- TSO COMMAND PROCESSOR -----------------
ENTER TSO COMMAND, CLIST, OR REXX EXEC BELOW:
===> EX 'hlq.SCAZEXEC(CAZ@SPF)'
```

Alternatively, you can select APA from your ISPF Primary Option menu, as shown below:

```
  Menu  Utilities  Compilers  Options  Status  Help
 -------------------------------------------------------------------------------
                         ISPF Primary Option Menu

Option ===>

0        Settings          Terminal and user parameters
1        View              Display source data or listings
2        Edit              Create or change source data
3        Utilities         Perform utility functions
4        Foreground        Interactive language processing
5        Batch             Submit job for language processing
6        Command           Enter TSO or Workstation commands
7        Dialog Test       Perform dialog testing
8        LM Facility       Library administrator functions
9        IBM Products      IBM program development products
10       SCLM              SW Configuration Library Manager
11       Workplace         ISPF Object/Action Workplace
APA      AppPerfAnalyzer   Application Performance Analyzer for z/OS
```

You will see the R02: IBM Application Performance Analyzer for z/OS Observation List panel.

```
   File  View  Navigate  Help
 --------------------------------------------------------------------------------
 R02: IBM APA for z/OS Observation Session List (CAZ0)        Row 00001 of 00245
 Command ===> _____  Scroll ===> CSR

  NEW     To define a new measurement
  TNEW    To define a threshold measurement
  CONNECT To connect to another instance of the measurement task
  VERSION To display version information for all instances
  IMPORT  To IMPORT a previously Exported sample file or hierarchy
  HIDE    To remove these commands from the display (recommended)
  /       On top of any ReqNum to get a list of the line commands

 Reqnum  Owned By  Description     Job Name  Date/Time        Samples  Status

  0882    USER1 id                 MACHIND   Jan-3  18:12      50,000  Ended
  0880 +  USER1 ims                DONDRVRN  Jan-3  18:11      50,000  STEPS
  0873 +  USER2                    MQGET     Jan-3  18:11      60,000  STEPS
  0872 +  USER2                    MQPUT     Jan-3  18:10      60,000  STEPS
  0871 +  USER2                    CICSC22F  Jan-3  18:07      60,000  REPEAT
  0868 +  USER2                    VSAMJOB   Jan-3  18:08      60,000  STEPS
  0867    USER2                    CICSC22F  Jan-3  17:55      60,000  Ended
  0866    USER2                    DBJOB8    Jan-3  17:56     100,001  Cancel
  0865    USER2                    AMENT1    Jan-3  17:54       4,567  Ended
  0864    USER2                    AMENT1    Jan-3  17:54           1  Failed
  0863    USER2                    AMENTPL1  Jan-3  17:52           1  Failed
 ._____.
 | Welcome to IBM APA for z/OS ISPF Version 8.1. You are currently          |
 | connected to measurement task id CAZ0. Enter CONNECT for and alternate   |
 | connection, VERSION for version information, NEW to start a measurement. |
 ._____.

 --------------------------------------------------------------------------------
```

Chapter 3. Verifying the installation

In this section, you activate the Application Performance Analyzer started task. You then perform a measurement using the supplied IVP program CAZIVP01. This job is designed to run for exactly 30 seconds. CAZIVP01 loops to consume CPU time, and it incrementally obtains blocks of memory as its execution progresses.

Beginning the started task

After you complete all of the customization tasks described in previous chapters, you can initiate the Application Performance Analyzer started task. Stop any Application Performance Analyzer started task currently running, then issue a START command on the system console specifying the name of the procedure you created.

For example:

```
S CAZ0
```

The Application Performance Analyzer started task initializes quickly. It should only be a few seconds before it is up and ready to accept measurement requests. Various messages are written to the JES message log during initialization. One of the first should appear as shown here:

```
CAZ0001I CAZ0-MAIN IBM APA for z/OS Version 8.100 starting.
```

The above message indicates execution has begun. The message in the following format is the one you need to see:

```
CAZ0100I CAZ0-MAIN (PLXF@67) IBM APA for z/OS operating.
```

This indicates the Application Performance Analyzer started task has initialized successfully.

Logged messages

The Application Performance Analyzer started task logs informational messages to various destinations managed by JES. The following SDSF panel illustrates these output files:

```
NP    DDNAME    StepName    ProcStep   DSID
      JESMSGLG  JES2                       2
      JESJCL    JESYSMSG                   3
      JESYSMSG  JES4                       4
      COMMANDS  SCAZ0                    102
      LOGR      CAZ0                     103
      MAIN      CAZ0                     104
      LOAD      CAZ0                     105
      CMDP      CAZ0                     106
      URSS      CAZ0                     107
```

All critical messages will appear in JESMSGLG. If the started task fails to start successfully, look in JESMSGLG for messages reporting the failure. Messages written to other destinations are described here:

COMMANDS

Any commands processed by the started task (external or internal) are logged to this file.

LOGR, MAIN, LOAD, CMDP, URSS
Each of these represents a subtask attached by the Application Performance
Analyzer started task and contains messages logged by that task. These are
primarily informational messages and during normal operations you need not be
concerned with the contents of these files. They can be useful during problem
diagnosis.

Starting the ISPF dialog

The next step is to start the ISPF user interface. Start Application Performance
Analyzer by selecting APA from the ISPF Primary Option menu, or by typing 'TSO
CAZ@SPF' at the TSO/E command prompt.

You should see a panel like the one shown below.

```
  File View Navigate Help
 --------------------------------------------------------------------------
 R02: IBM APA for z/OS Observation Session List    (CAZ0) Row 00001 of 00002
 Command ===> _____ Scroll ===> CSR

  NEW     To define a new measurement
  TNEW    To define a threshold measurement
  CONNECT To connect to another instance of the measurement task
  VERSION To display version information for all instances
  IMPORT  To IMPORT a previously Exported sample file or hierarchy
  HIDE    To remove these commands from the display (recommended)
  /       On top of any ReqNum to get a list of the line commands

 ReqNum  Owned By Description    Job Name  Date/Time  Samples  Status




   +------------------------------------------------------------------------+
   | Welcome to IBM APA for z/OS Version 8.1. You are                       |
   | currently connected to measurement task id CAZ0.                       |
   | Enter CONNECT for an alternate connection, VERSION                     |
   | for additional version information, NEW to start a measurement.        |
   +------------------------------------------------------------------------+
```

A list of commonly used primary commands is displayed at the top of the screen
to assist those users new to Application Performance Analyzer. Once you become
familiar with the Observation List panel, these commands can be hidden from
view by entering the HIDE primary command. They can be displayed again by
entering the SHOW primary command.

The screen should have displayed a message indicating that you are connected to
the measurement task. If this is not displayed, Application Performance Analyzer
assumes you are a new user and displays a different message, with a list of
available measurement tasks. Enter a CONNECT command specifying your
Application Performance Analyzer instance identifier. For example:

  ===> CONNECT CAZ0

If you have configured Application Performance Analyzer to enable the expiry
days warning feature, the welcome message may be replaced with a warning
message indicating that some observations are about to be deleted. You might see a
message like the following.

```
+------------------------------------------------------------------------------------+
|  Warning: one or more observations are approaching their expiry dates. Enter the SW |
|  line command on the ReqNum heading to view all observations that are approaching   |
|  their expiry date.                                                                 |
+------------------------------------------------------------------------------------+
```

Entering a measurement request

Assuming this worked, you have finished the installation. Application Performance Analyzer is ready to accept measurement requests. The panel illustrated above shows a list of Application Performance Analyzer observation sessions and is the dialog from which you access all other Application Performance Analyzer dialogs and reports. Since no measurement requests have been made yet, the list is empty. The next step is to perform the first measurement. From this screen, enter the NEW command.

```
   ===> NEW
```

Application Performance Analyzer will open a dialog screen into which you specify the measurement parameters. This is shown here:

```
 File View Navigate Help
 -------------------------------------------------------------------------------
 R03: Schedule New Measurement            Row 00001 of 00019
 Command ===>                   Scroll ===> CSR
    1. Job Information    3. Multi Steps     5. Subsystems   7. Schedule
    2. Options           4. Active Jobs      6. Sysplex      8. Sched Options
 _____
 Panel 1. Job Information

  Job Name/Pattern . .  _____
                        (Inactive)

  Step Specification
   Step No. . . . . . ___    Specify step number, program name,
   Program Name . . . _____    step name or step name + Proc step
   Step Name  . . . . _____    name. Use panel 3 to specify more
   ProcStepName . . . _____    than one step.

 Description     . . . _____
 Number of Samples    . _____   Measure to step end   . . . N
 Duration (min:sec)   . _____   Delay by (secs) . . . . . __
 Notify TSO User    . . _____   Retain file for (days) . 20
                                USS observations . . . . _____   Max. 20
```

Now enter the following input to this panel. This will create a measurement request for a supplied installation verification job named CAZIVP01.

- Job Name/Pattern, enter: CAZIVP01
- Step No, enter: 1
- Description, enter: IVP 01 (or any description you want)
- Number of Samples, enter: 6000
- Duration of session, enter: 30

Now press the ENTER key. Assuming you did not make any errors in the input, you will see the following message:

```
Input more data or ENTER to submit
```

Press the ENTER key again and your request will be accepted. You will be returned to the observation session list screen. The list will now contain the request you just made.

```
 File View Navigate Help
 -----------------------------------------------------------------------------------------
 R02: IBM APA for z/OS Observation Session List               (CAZ0) Request submitted OK
 Command ===> _____ Scroll ===> CSR
 ReqNum  Owned By   Description    Job Name   Date/Time   Samples    Status
 0001    USER1      IVP 01         CAZIVP01   Jul-20 9:35 6,000      Sched
```

This shows that the request is scheduled. The Application Performance Analyzer
started task will automatically begin the measurement when it detects execution of
the job named CAZIVP01. Now we will submit the job. The JCL for CAZIVP01 is
in *hlq*.SCAZSAMP in the member CAZIVPJ. Submit this JCL (after making any
necessary modifications).

When the job begins execution, the status shown in the observation session list
screen will show *Active*, and the value under the *Samples* heading will display the
actual number of samples completed. This will refresh dynamically each time you
press the ENTER key.

```
  File  View  Navigate  Help
 -------------------------------------------------------------------------------
 R02: IBM APA for z/OS Observation Session List (CAZ0)         Row 00001 of 00246
 Command ===> _____ Scroll ===> CSR

 Reqnum  Owned By  Description    Job Name  Date/Time      Samples  Status

  0001    USER1    my userid      CAZIVP01  Jun-11  6:51        43  Active


 -------------------------------------------------------------------------------
```

Upon completion of the job, the status will change to *Ended*.

```
  File  View  Navigate  Help
 -------------------------------------------------------------------------------
 R02: IBM APA for z/OS Observation Session List (CAZ0)         Row 00001 of 00246
 Command ===> _____ Scroll ===> CSR

 Reqnum  Owned By  Description    Job Name  Date/Time      Samples  Status

  0001    USER1    my userid      CAZIVP01  Jun-11 19:03     6,000  Ended


 -------------------------------------------------------------------------------
```

Displaying performance analysis reports

With completion of a measurement, you are now ready to display performance
analysis reports. The subject of performance analysis reports and their
interpretation is beyond the scope of this manual, but we will demonstrate this
here just to complete the installation verification.

To enter the reporting facility, enter the R line command on the measurement
request you would like to report on. You will then see a report selection screen like
the one shown here.

```
 File  View  Navigate  Help
 -------------------------------------------------------------------------------
 R01: IBM APA for z/OS Performance Reports (0001/CAZIVP01)   Row 00001 of 00007
 Command ===>_____ Scroll ===> CSR

       Select a category from the list below to view the available reports
   A Admin/Miscellaneous    I IMS Measurement       E CICS Measurement
   S Statistics/Storage     F DB2 Measurement       Q MQ Measurement
   C CPU Usage Analysis     D DASD I/O Analysis     G Coupling Facility
   W CPU WAIT Analysis      V Variance Reports      X Multi Address Space
   J Java Measurement       B WebSphere App Server  H HFS Analysis
 _____

 Enter S to make a selection or enter the report code on the command line

     __S01 Session Statistics        __S07 TCB Execution Summary
     __S02 Load Module Attributes    __S08 Processor Utilization Summary
     __S03 Load Module Summary       __S09 Measurement Analysis
     __S04 TCB Summary
     __S05 Memory Usage Timeline
     __S06 Data Space Usage Timeline
```

From this panel, you can select from an array of performance analysis reports. As an example, select report S05 (Memory Usage Timeline). You can position the cursor on the S05 item and press enter, or you can type S05 directly on the command line. You should see a screen like the one shown here:

```
 File  View  Navigate  Help
 -------------------------------------------------------------------------------
 S05: Memory Usage Timeline (0001/CAZIVP01)       Row 00001 of 00015
 Command ===> _____ Scroll ===> CSR
 SEQN  Seconds  Storage   <----2650K-----------------------------5830K-->
          *....+....+....+....+....+....+....+....+....+....+....*
 0001   1.857   2924K   ****
 0002   1.935   3124K   *******
 0003   1.935   3324K   **********
 0004   1.944   3524K   *************
 0005   1.927   3724K   ****************
 0006   1.935   3928K   ********************
 0007   1.934   4128K   **********************
 0008   1.942   4328K   **************************
 0009   1.929   4528K   ****************************
 0010   1.936   4728K   ******************************
 0011   1.935   4932K   *********************************
 0012   1.934   5132K   ************************************
 0013   1.934   5332K   ******************************************
 0014   1.934   5532K   *******************************************
 0015   1.905   5732K   **********************************************
```

Ending the Application Performance Analyzer started task

During normal operation of Application Performance Analyzer, you will typically initiate one or more started tasks after an IPL and they will remain active. Stopping (and restarting) the started task is necessary, however, to apply any configuration setting changes. Stop a task simply by entering the STOP command on the operator console.

At termination, you will see the following messages.

```
 CAZ0490I CAZO-MAIN Application Performance Analyzer is shutting down.
 CAZ1190I CAZO-CMDP Command processor stopping.
 CAZ1199I CAZO-CMDP Command processor terminated.
 CAZ0499I CAZO-MAIN  Application Performance Analyzer is terminating.
 CAZ1090I CAZO-LOGR Logger stopping.
```

```
CAZ1099I CAZO-LOGR Logger terminated.
IEF404I CAZO - ENDED - TIME=14.11.45
IEF352I ADDRESS SPACE UNAVAILABLE
$HASP395 CAZO ENDED
```

The IEF352I message is reported because MVS marks the address space as
unavailable. This is because Application Performance Analyzer creates a global
linkage entry. This is normal for any program that creates a global linkage entry.

Chapter 4. Customizing Application Performance Analyzer for use with DB2 Universal Database

This chapter describes the tasks for configuring the optional DB2 component. These tasks are required only if you plan to use any of the following Application Performance Analyzer features:

- DB2 EXPLAIN
- Report plan and package BIND timestamps
- Display actual host variable names in SQL statements instead of :H.

We recommend that you work closely with a DB2 administrator to complete the following tasks:

1. "Deciding which authorization ID to use"
2. "Binding a DB2 package and plan"
3. "Enabling DB2 EXPLAIN" on page 68
4. "Providing DB2 authorization" on page 69
5. "Updating the Application Performance Analyzer-started task" on page 70
6. "Updating the started task configuration settings" on page 70

Deciding which authorization ID to use

Before running the tasks described in the following sections, you must choose a unique *authid*. The *authid* is the same *authid* that you defined for the Application Performance Analyzer RACF STARTED class. You must use the same *authid* to bind a DB2 plan (see "Binding a DB2 package and plan") and authorize the plan name (see "Providing DB2 authorization" on page 69).

Binding a DB2 package and plan

To bind a DB2 package and plan, do the following steps:

1. Edit member CAZBIND in your *hlq*.SCAZSAMP library. This sample job requires you to have knowledge of the DB2 subsystem. See the instructions in the sample job for more information.
2. The default package name used in the first BIND statement above is CAZPACK1. You can change this name to match your site requirements. If you do so then you must also change this value in the BIND PLAN statement.

   The default plan name used in the statement is CAZPLAN1. You can change this name to match your site's requirements. If you do so then you must also change the plan name specified in the keyword DB2PlanName, in member CAZCNFG0 in the *hlq*.SCAZSAMP library.

   Repeat these BIND operations if you are binding this plan to multiple DB2 subsystems.

   Note: You must use the same plan name on all DB2 subsystems.
3. Submit the sample job. You will get a return code of 0 if the job runs correctly.

Enabling DB2 EXPLAIN

Application Performance Analyzer supports both static and dynamic EXPLAIN options. These need to be configured separately, but both options require that the DB2 Intercept be enabled.

Static EXPLAIN

Static EXPLAIN is enabled in Application Performance Analyzer by specifying DB2X=Yes in the CONFIG SAMPLE statement during customization. This makes the DB2X data extractor available for DB2 measurements. The DB2EXPLAIN keyword in the CONFIG SAMPLE statement is required to indicate which DB2 subsystems are to be accessed for explain information.

While sampling a DB2 application when the DB2X data extractor is active for the measurement, Application Performance Analyzer can collect EXPLAIN data created as a result of a DB2 BIND with the EXPLAIN(YES) option. This applies to static SQL statements only. The EXPLAIN data is retrieved from the PLAN_TABLE used by DB2 at the time of the bind.

Dynamic EXPLAIN

Dynamic EXPLAIN is enabled in Application Performance Analyzer by specifying the DB2EXPLAIN keyword in the CONFIG SAMPLE statement during customization.

To use dynamic EXPLAIN with the Application Performance Analyzer DB2 component, Application Performance Analyzer started tasks must have access to a PLAN_TABLE. Bypass this section if you do not use the dynamic EXPLAIN feature.

- For DB2 Version 9, edit and submit job CAZPLANT in *hlq*.SCAZSAMP to create a PLAN_TABLE. For DB2 Version 10, edit and submit job CAZPLANX in *hlq*.SCAZSAMP to create a PLAN_TABLE.
- The qualifier for this PLAN_TABLE defaults to the authorization ID of the started task. You can override this default by changing the DB2Qualifier setting in the configuration file. See *DB2Qualifier* under CONFIG SAMPLE statement.

The CREATE statement in job CAZPLANT is in the 49-column format compatible with DB2 Version 6. The 51-column format (for DB2 Version 7) and the 58-column format (for DB2 Version 8) can be used instead. For DB2 Version 10, the PLAN_TABLE must be created using job CAZPLANX. Whichever PLAN_TABLE is used, the dynamic EXPLAIN feature will only make use of the first 49 columns when reporting the results of an EXPLAIN.

You can specify the database and tablespace name in which the PLAN_TABLE will be created by adding the appropriate lines to the CREATE statement.

The CREATE statement assumes that the escape character for the DB2 subsystem is the quote character ("). If the DB2 subsystem has been installed with the quote character as the SQL string delimiter, then the escape character is the apostrophe ('). In this case, the quotes in the CREATE statement must be changed to apostrophes.

Considerations in using dynamic DB2 EXPLAIN

The DB2+ data extractor is required for Application Performance Analyzer to capture the required qualification information. If DB2+ is not selected, then dynamic EXPLAIN will not be available.

There is a limit of 15000 bytes on the target SQL statement size. The sample file LRECL dictates this limit.

A DB2 restriction prevents some SQL statements from being the subject of dynamic EXPLAIN. UPDATE or DELETE statements with a WHERE CURRENT OF clause are among the prevented SQL statements.

Application Performance Analyzer needs to access catalog information for the target SQL statement. If any of the information changed since the sample file was created, then the resulting dynamic EXPLAIN data might not be correct. In the case of a static SQL statement of 4000 bytes or greater, the statement itself might have changed and might no longer match the statement in the sample file.

Providing DB2 authorization

If you want to use the static EXPLAIN feature, the authorization ID of the Application Performance Analyzer started task must have SELECT authority on the following DB2 tables:
- SYSIBM.SYSPLAN
- SYSIBM.SYSPACKAGE
- SYSIBM.SYSSTMT
- SYSIBM.SYSPACKSTMT
- *authid*.PLAN_TABLE, where *authid* is the owner of the package or plan that was bound with the EXPLAIN(YES) option.

If you want to use the dynamic EXPLAIN feature, the authorization ID of the Application Performance Analyzer started task must have SYSADM authority. You can use the following SQL statement to grant the SYSADM privilege to the *authid* of the started task:

```
GRANT SYSADM TO authid
```

If you do not grant SYSADM to the started task, then it must be authorized as follows:
- SELECT privilege on the catalog tables:
  - SYSIBM.SYSPLAN
  - SYSIBM.SYSPACKAGE
  - SYSIBM.SYSROUTINES
  - SYSIBM.SYSDBRM
  - SYSIBM.SYSPACKSTMT
  - SYSIBM.SYSSTMT
- SELECT and DELETE privileges on *authid*.PLAN_TABLE, where *authid* is the authorization ID assigned to the started task.
- EXPLAIN *sqlstmt*, where *sqlstmt* is any SQL statement that needs to be explained. The started task must have all authorization required by dynamic execution of the SQL statement.

Updating the Application Performance Analyzer-started task

Add a DSN for the DB2 load library (for example, *db2hlq*.SDSNLOAD) to the STEPLIB in the Application Performance Analyzer-started task JCL (member CAZ0 in *hlq*.SCAZSAMP). This load library can be at version 9 or version 10, but must contain the following load modules:

- DSNALI
- DSNTIAR
- alias DSNHLI2

Updating the started task configuration settings

1. Edit member CAZCNFG0 in library *hlq*.SCAZSAMP.
2. In the CONFIG SAMPLE statement, complete the DB2EXPLAIN, DB2PlanName, and DB2X parameters as required by your installation.
3. Restart your Application Performance Analyzer-started task to make the change effective.

Chapter 5. Customizing Application Performance Analyzer for use with J9VM

This chapter describes the tasks for configuring Java support. Sampling a Java application requires loading a JVMTI agent to obtain Java samples. There are two ways that this can be accomplished:

- Use a dynamically loaded JVMTI agent.

  This option loads the JVMTI agent at the time that a sampling session is started. The JVMTI agent is loaded using the Java Attach API. The agent runs only for the duration of the sampling session.

- Use a preloaded JVMTI agent.

  This option loads the JVMTI agent when the target Java job is started. The agent remains alive until the Java job terminates. The agent sits in a wait state until a sampling session is requested. After the sampling session completes, the agent returns to a wait state.

Application Performance Analyzer can be configured to support one or the other of these options, not both.

Work closely with a security administrator and a USS administrator to complete the following tasks:

1. "Install USS files in an Application Performance Analyzer directory on your system"
2. "Provide APF authorization for the Application Performance Analyzer USS files" on page 72
3. "Assign USS permissions to the Application Performance Analyzer started task" on page 72
4. "Set up Security Access Rules" on page 72
5. "Update the started task configuration settings" on page 73
6. Enable Java attach API (for a dynamically loaded JVMTI agent) or Specify a preloaded JVMTI agent

Application Performance Analyzer can be configured to support either 31-bit or 64-bit Java in the started task. Regardless of how it is configured, Application Performance Analyzer can sample both 31-bit Java applications and 64-bit Java applications, with the appropriate Java PTFs installed on your system.

Install USS files in an Application Performance Analyzer directory on your system

Some USS files must be installed in an Application Performance Analyzer directory on your system. Refer to "CONFIG SAMPLE statement" on page 32 for more details.

Verify the directory contains the following files:
- CAZJNI01
- CAZJNI02.class
- CAZJNI03
- CAZJNI04.class

- libCAZJNI02.so
- libCAZJNI02.*x*
- libCAZJN104.so
- libCAZJNI04.x
- libCAZJVMTI.so
- libCAZJVMTI.*x*
- libCAZJ64TI.so
- libCAZ64TI.x
- jvmoptions.txt
- libCAZJTA11.so
- libCAZJTA14.so

If you are using a dynamically loaded JVMTI agent, the path name for this directory must be made known to Application Performance Analyzer through the configuration option USSHome.

Provide APF authorization for the Application Performance Analyzer USS files

CAZJNI01, CAZJNI03, libCAZJNI02.so, and libCAZJNI04.so must be APF authorized. If you will be sampling any address spaces that are running authorized, libCAZJVMTI.so, libCAZJ64TI.so, libCAZJTA11.so, and libCAZJTA14.so must also be APF authorized. This should be automatically handled during installation. If not, issue an EXTATTR command from an authorized user ID to set the APF extended attribute bit on for these files.

Assign USS permissions to the Application Performance Analyzer started task

If you are using a preloaded JVMTI agent, you can skip this step.

The Application Performance Analyzer started task should be given USS superuser privileges, with UID=0. An alternative to granting superuser privileges is to ensure that the Application Performance Analyzer started task runs under the same UID as the Java applications being measured. In this case, you can perform the tasks described in "Set up Security Access Rules" below.

Set up Security Access Rules

If you are using a preloaded JVMTI agent, or you have granted USS superuser privileges to the Application Performance Analyzer started task, you can skip this step.

For J9VM support, the Application Performance Analyzer started task must have an OMVS definition in RACF to access USS data and programs. It requires USS authorization for the following:

- Read and execute permission on the directories specified by the JavaHome and USSHome settings in the CONFIG SAMPLE statement. Refer to "CONFIG SAMPLE statement" on page 32 for more details on these settings.
- Read, write, and execute permission on the **/tmp** directory.

Application Performance Analyzer looks in the **/tmp** directory for the attach information for Java. If the default temporary directory is not **/tmp**, you must add the following Java run time parameter to the applications being measured:

`-Dcom.ibm.tools.attach.directory=/tmp/.com_ibm_tools_attach`

The J9VM creates work files in the directory **/tmp/.com_ibm_tools_attach**.

If the Application Performance Analyzer configuration setting DiagLevel is set to 7 or higher, Application Performance Analyzer creates a diagnostic file for each observation request in the directory **/tmp/CAZ**. The diagnostic file is named *sysid.nnnn.pppppppp*.JVMTI.txt, where *sysid* is the Application Performance Analyzer started task identifier, *nnnn* is the observation request number, and *pppppppp* is the process id of the target application.

Application Performance Analyzer also writes `stdout` and `stderr` files in the **/tmp/CAZ** directory if DiagLevel is set to 7 or higher. The files are named *sysid*.stdout and `sysid.stderr`, where *sysid* is the Application Performance Analyzer-started task identifier.

Update the started task configuration settings

If you are using a preloaded JVMTI agent, you can skip this step.

Edit member CAZCNFG0 in library hlq.SCAZSAMP in the CONFIG SAMPLE statement as shown below

- Set J9VMEnable=YES|64.
- Set the JavaHome setting.
- Set the USSHome setting.

Restart your Application Performance Analyzer started task to make the change effective.

Refer to "CONFIG SAMPLE statement" on page 32 for more details on these settings.

Enable Java attach API

If you are using a dynamically loaded JVMTI agent, the Java Attach API must be enabled in the application being sampled. The Java attach API is enabled by adding, the '-Dcom.ibm.tools.attach.enable=yes' run time parameter to the application being sampled. An example of how to enable the Java attach API in JCL that executes BPXBATCH is as follows:

```
//STEP EXEC PGM=BPXBATCH,
//PARM='sh java -Dcom.ibm.tools.attach.enable=yes'
```

Depending on the Java service release level you are running, the application being sampled may need to be run with UID=0 (the same UID as Application Performance Analyzer itself). See the chart below for the specific Java PTFs and service release levels that include the attach API, and for the associated job requirements for sampling.

Table 4. Java PTFs Including the Attach API

Java Version	Service Release	Java PTF	Job Requirements
V5 31-bit	SR10	UK48410	Add Dcom.ibm.tools.attach.enable=yes to the application being sampled. Run the application being sampled with UID=0.

Table 4. Java PTFs Including the Attach API (continued)

Java Version	Service Release	Java PTF	Job Requirements
V5 31-bit	SR11	UK51963	Add Dcom.ibm.tools.attach.enable=yes to the application being sampled. Run the application being sampled with UID=0.
V5 31-bit	SR12 and above	UK59131	Add Dcom.ibm.tools.attach.enable=yes to the application being sampled.
V5 64-bit	SR12 and above	UK59134	Add Dcom.ibm.tools.attach.enable=yes to the application being sampled.
V6 31-bit	SR6	UK50947	Add Dcom.ibm.tools.attach.enable=yes to the application being sampled. Run the application being sampled with UID=0.
V6 31-bit	SR7	UK53084	Add Dcom.ibm.tools.attach.enable=yes' to the application being sampled. Run the application being sampled with UID=0.
V6 31-bit	SR8 and above	UK56434	Add Dcom.ibm.tools.attach.enable=yes to the application being sampled.
V6 64-bit	SR8 and above	UK56435	Add Dcom.ibm.tools.attach.enable=yes to the application being sampled.

When measuring applications running with '-Dcom.ibm.tools.attach.enable=yes' and without 'UID=0', J9VM might write some OPEN access violations in the system log. These can be ignored, because J9VM is attempting to open tmp directories that require UID=0 access. These directories have no significance to Application Performance Analyzer Analyzer and will not affect the measurement.

Specify a preloaded JVMTI agent

If you are using a preloaded JVMTI agent, you must specify the agent in a runtime parameter of the target Java job. The agent is loaded using the Java -agentpath: runtime parameter. Specify the full path name of either the 31-bit JVMTI agent (libCAZJTA11.so) or the 64-bit JVMTI agent (libCAZJTA14.so) depending on whether your target Java job is 31-bit or 64-bit.

An example of how to specify a preloaded 64-bit JVMTI agent in JCL that executes BPXBATCH is as follows. The example assumes that the USS components of Application Performance Analyzer have been installed in a directory named **/usr/lpp/apa/v13r1**, and that there is only one Application Performance Analyzer started task in the system.

```
//STEP EXEC PGM=BPXBATCH,
//PARM='sh java —agentpath:/usr/lpp/apa/v13r1/libCAZJTA14.so'
```

If there is more than one Application Performance Analyzer started task in the system, the -agentpath: runtime parameter can specify the system id of the started task. The following example shows how to specify that the JVMTI agent is to be loaded for the started task with a system id of CAZ1.

```
//STEP EXEC PGM=BPXBATCH,
//PARM='sh java —agentpath:/usr/lpp/apa/v13r1/libCAZJTA14.so=CAZ1'
```

The JVMTI agent must locate the Application Performance Analyzer started task. By default, it will timeout after 1 minute if the started task is not found. This time limit can be overridden. The following example shows how to specify a time limit of 9 minutes.

```
//STEP EXEC PGM=BPXBATCH,
//PARM='sh java —agentpath:/usr/lpp/apa/v13r1/libCAZJTA14.so=CAZ1,9'
```

If the Application Performance Analyzer configuration setting DiagLevel is set to 7 or higher, Application Performance Analyzer creates a diagnostic file for each observation request in the directory **/tmp/CAZ**. The diagnostic file is named *sysid.nnnn.pppppppp*.JVMTI.txt, where *sysid* is the Application Performance Analyzer started task identifier, *nnnn* is the observation request number, and *pppppppp* is the process id of the target application.

Constraints and usage notes

Only a single J9VM in an address space is supported.

Only one observation at a time is supported for a particular J9VM in an address space.

IMS Java programs are not supported.

CICS Java programs are supported for CICS/TS 4.2 and above.

The Application Performance Analyzer JVMTI agent will stop sampling when the main thread ends, even if the specified number of samples has not been reached.

J9VM sampling can only be carried out if there is more than one CPU allocated to the system image on which it runs.

Depending on the processor speed of the system, Application Performance Analyzer might reduce the sampling rate of a J9VM observation.

The Application Performance Analyzer JVMTI agent uses about 2.5MB of heap storage. It might be more or less at your installation.

When J9VM sampling support is enabled for a dynamically loaded JVMTI agent, a J9VM is launched in the Application Performance Analyzer started task address space. This increases the amount of storage used by the started task. The real storage pages may increase to about 10 times the usual amount.

The J9VM launched in the started task is configured to use a 4MB heap. It is also configured not to use any zAAP processors.

For more information about using the Java Attach API, please see the *User Guide for IBM SDK for z/OS* for the Java release you are running.

Chapter 6. Customizing Application Performance Analyzer Listener

The Application Performance Analyzer Listener (APA/Listener) started task is an optional component of IBM Application Performance Analyzer for z/OS. It is required when workstation users wish to communicate with Application Performance Analyzer through the graphical user interface (APA/GUI).

This chapter explains how to configure the Application Performance Analyzer Listener started task after it has been installed.

Choosing an Application Performance Analyzer Listener-started task identifier

An Application Performance Analyzer Listener-started task is identified internally by a four-character identifier called the LISTENERID. The IBM-default value for the Listener-started task identifier shipped with the product is CAZL. Similarly, the IBM-default value for the Listener-started task procedure name shipped with the product is CAZL. You can choose a different four-character LISTENERID and a different started task procedure name to meet your site's requirements. They do not have to match. LISTENERID can be changed in the CONFIG BASIC statement found in the member CAZCNFGL in *hlq*.SCAZSAMP.

Choosing a high-level qualifier for the Application Performance Analyzer Listener data sets

Application Performance Analyzer Listener might create additional data sets as part of its normal operation. The table shown below lists these data sets and indicates the names assigned by Application Performance Analyzer Listener.

Table 5. Application Performance Analyzer Listener Additional Data Sets

Data set name	Description
yourhlq.LSRV####.@*yymmdd*.@*hhmms*	Trace data sets
yourhlq.LLG.*LiId*.D*yyddd*.T*hhmm*### or *yourhlq*.LLG..*gen*#	Optional log data sets
yourdatahlq.*userid*.R*nnnn*.XMIT or *userid*.R*nnnn*.XMIT	Data set name created by GUI export requests if the ExportDSN configuration parameter is not specified.

By default, the trace data sets and log data sets use the value in the HLQ parameter coded in the Application Performance Analyzer Listener CONFIG BASIC statement. Optionally, a different trace *yourhlq* (HLQ=) and log *yourhlq* (LogHLQ=) can be specified on the CONFIG BASIC and CONFIG LOGFILE statements, respectively.

Application Performance Analyzer Listener creates an export (XMIT) data set for each successful export request issued by an Application Performance Analyzer GUI plugin user. By default, the high-level qualifier for the GUI exported data set is set to the GUI user's user ID connected to z/OS. Optionally, you might choose to specify a single high-level qualifier for all GUI export data sets. To do this, you

choose a prefix that replaces the *yourdatalhq* symbol above by coding a value for the DataHLQ setting in the Application Performance Analyzer Listener CONFIG BASIC statement.

You choose a prefix that replaces the *yourhlq* symbol shown above. However, it must comply with the JCL rule for specifying a data set name. In this case, your prefix can have a maximum of 17 characters. This can be the same prefix as used by Application Performance Analyzer.

In the event that you execute multiple Application Performance Analyzer Listener started tasks, each one must be assigned its own unique *yourhlq* value. Each creates its own set of data sets. It is a useful convention to include the started task instance identifier as part of this prefix.

Trace data sets:

- #### represents the server instance number. There is one server instance number per client connection.
- *yymmdd* represents the year, month and day the trace data set is created.
- *hhmmsst* represents the hour, minute, second and 10th of a second the trace data set is created.

Log data sets:

- *LiId* represents the Listener ID. This is specified in the CONFIG BASIC statement.
- *yyddd* represents the year and julian day the log data set is created.
- *hhmm* represents the hour and minute the log data set is created.
- ### represents a sequence number from 000 to 999.
- *gen#* represents the GDG generation number.

GUI Export data sets:

- *Rnnnn* represents the request number of the observation that was exported

Choosing a TCP/IP port number

The Application Performance Analyzer GUI communicates with the Application Performance Analyzer Listener across TCP/IP. The Application Performance Analyzer Listener listens for client connections on any TCP/IP stacks and port numbers specified in the Application Performance Analyzer Listener started task configuration. Use the PORT and ListenTo settings of the CONFIG BASIC statement described further in this chapter.

Establishing external security system access rules

The data sets created by Application Performance Analyzer Listener as part of its normal operations must be secured against unauthorized access, while allowing Application Performance Analyzer Listener the authority it needs to create and manage the data sets. This section describes the necessary access rules that apply to your External Security System (such as RACF).

Trace files

As a result of certain diagnostic actions, Application Performance Analyzer Listener might allocate and write to trace data sets. The Application Performance

Analyzer Listener started task must have full access to these data sets. Additionally, you should grant full access to the appropriate product support personnel and deny access to all other users. These data sets have a name in the form:

yourhlq.LSRV####.@*yymmdd*.@*hhmms*

Log files

If you choose to activate the Listener logging option as recommended, Application Performance Analyzer Listener allocates and writes to log data sets. These data sets record activity within the started task. The Application Performance Analyzer Listener started task must have full access to these data sets. Additionally, you should grant full access to the appropriate product support personnel and deny access to all other users. These data sets have a name in the form:

yourhlq.LLG.*LiId*.D*yyddd*.T*hhmm*###

or

yourhlq.LLG.*gen#*

GUI Export data sets

The Listener requires full access to the GUI export data sets.

If you specify an ExportDSN parameter in the CONFIG BASIC statement of the Listener started task configuration settings, then you must ensure that the Listener started task has full access to the high level qualifier specified in the ExportDSN parameter.

If you do not specify an ExportDSN parameter and you specify a DataHLQ parameter in the CONFIG BASIC statement of the Listener started task configuration settings, then you must ensure that the Listener started task has full access to the high level qualifier specified in the DataHLQ parameter. The export data sets will have a name in the form:

yourdatahlq.*userid*.R*nnnn*.XMIT

If you do not to specify an ExportDSN parameter and you do not to specify a DataHLQ parameter, you must ensure that the Application Performance Analyzer Listener started task has full access to all data sets that begin with the user ID of every GUI user who will be performing export requests. In this case, the export data sets will have a name in the form:

userid.R*nnnn*.XMIT

Assigning priority

If your installation is running Workload Manager (WLM) in GOAL mode, Application Performance Analyzer Listener should be assigned to the same priority class as Application Performance Analyzer.

If running in COMPAT mode, put Application Performance Analyzer Listener in the same performance group as Application Performance Analyzer.

Making the Application Performance Analyzer Listener-started task available

To make the Application Performance Analyzer Listener-started task available, complete the following steps:

1. Edit the procedure CAZL in your library *hlq*.SCAZSAMP. See the instructions in the sample job for more information.

2. Copy this procedure to a library in the JES2 or JES3 procedure library concatenation for started tasks. You can rename this procedure to any valid procedure name to meet your site's requirements.

3. Define a user ID for the Application Performance Analyzer Listener-started task procedure

4. Grant the user ID you created READ access to Application Performance Analyzer libraries.

5. Define a profile in the RACF STARTED class for the Application Performance Analyzer Listener-started task procedure so that it can access protected resources. To do this, update your security definitions by using the RDEFINE command to associate the Application Performance Analyzer Listener-started task procedure with user IDs as follows:

   ```
   RDEFINE STARTED procname.* STDATA(USER(userid) GROUP(group))
   ```

   Where *procname* is the Application Performance Analyzer Listener-started task procedure name, *userid* is the user ID, and *group* is the group on your system that has access to the RACF-protected resources. You can issue the above command under TSO/E or on the system console log. For more information about using the STARTED class, see *z/OS Security Server (RACF) Security Administrator's Guide*.

6. Create or update a COMMND*xx* parmlib member that will be used for IPL so that it contains the following command:

   ```
   COM='START procname'
   ```

   Where *procname* is the Application Performance Analyzer Listener-started task procedure name.

Configuring Application Performance Analyzer Listener

This section describes how to configure Application Performance Analyzer Listener to suit your installation. There are two types of configuration: global installation-level settings, and settings for the started task.

Installation-level configuration settings

This step is part of the Application Performance Analyzer configuration, and must be completed only when your Application Performance Analyzer started tasks are configured to use Application Performance Analyzer's internal security, for example, its own access rules facility, to control access to functions. If your installation has chosen to control access to functions using RACF or a RACF-compatible external security product, you can bypass this step.

The installation-level configuration settings for Application Performance Analyzer Listener are generated into the Application Performance Analyzer load module CAZCNFG1. Refer to "Installation-level configuration settings" on page 20 in Chapter 2 for details on these settings and how they are processed.

An access rule is required to support workstation users that connect to an Application Performance Analyzer started task via the Application Performance

Analyzer Listener started task. The rule is called: 'UseListener'. An example of this rule is provided below as it would be displayed in the CAZCNFG1 SCAZSAMP member.

```
13  CAZRULE *,allow,UseListener,*
```

This rule allows all users to connect to the Application Performance Analyzer started task via the Application Performance Analyzer Listener started task.

Access rule description:

Table 6. Access Rule Description for UseListener Rule

Code this	For this type of access	Object type
UseListener	Controls which workstation users are allowed to connect to the Application Performance Analyzer started task via the Application Performance Analyzer Listener started task.	User ID of the user that has connected to the Application Performance Analyzer Listener started task.

Started task configuration settings

This section describes how to configure the Application Performance Analyzer Listener started task to suit your installation.

The Application Performance Analyzer Listener started task obtains the configuration settings from a file allocated to DDname CONFIG. The file is a SYSIN-type member that you edit in order to change the settings. The member CAZCNFGL in *hlq*.SCAZSAMP contains the initial settings. A group of configuration settings is specified by a CONFIG statement. Settings are organized into categories or classes. Each CONFIG statement defines settings for a particular class. The general syntax for a CONFIG statement is illustrated here:

```
CONFIG class
keyword1=value1
keyword2=value2 // this is a comment
...
keywordn=valuen
;    // end with semicolon
```

You must code a single keyword/value pair expression within a single line. It cannot span two lines. Separate keyword/value pair expressions from each other using one or more blanks. A CONFIG statement, including all its keyword/value pair expressions, must be terminated by a semicolon.

A CONFIG statement might be coded anywhere within columns 1 to 71 inclusive. An asterisk in column 1 denotes a comment line. A double slash (//) anywhere in the line indicates the remaining text in the line is a comment.

The following CONFIG BASIC statement, which is supplied in the predefined CAZCNFGL member, illustrates the CONFIG syntax.

```
  CONFIG BASIC              // Basic settings:
    ListenerID=CAZL         // . ID
    HLQ=#yourhlq            // . DSN HLQ for diagnostic
                            //    trace datasets
    PORT=8001               // . Listener's port#
    UNIT=SYSALLDA           // . unit name for new data set
```

```
SECURITY=DISABLED     //  . no security checking
MsgLevel=I            //  message level
SessionTimeOut=0000   //  Session time out (never)
KeepConnected=0000    //  ping interval    (never)
```

CONFIG statement parameters

Class

Specifies the category of settings that the CONFIG statement is to modify. The allowable values for class are:

- BASIC
- LOGFILE
- APPL_WSFE
- OPER

keywordn

Specifies the name of the setting variable and is paired with a value expression in *keyword=value* format. The name is not case sensitive.

valuen

This is paired with a keyword name and specifies the value to be set.

CONFIG BASIC statement

BASIC settings allow you specify values for basic characteristics of the Application Performance Analyzer Listener installation. The following settings can be specified:

LISTENERID

Required. This is the Application Performance Analyzer Listener ID, a four-character value that uniquely identifies the Application Performance Analyzer Listener started task.

HLQ

Required. Specifies the high-level qualifier for data sets created by the Application Performance Analyzer Listener started task as part of its normal operation. The value that you specify for HLQ must comply with the JCL rule for specifying a data set name. It can be no more than 17 characters long.

Currently HLQ is used to allocate trace data sets used for diagnostic purposes.

Default: None.

PORT

Required. Specifies the 1 to 5 digit TCP/IP port number on which the Application Performance Analyzer Listener listens for client connections. The port number can be between 1 and 65,535 and must be unique to a single instance of the Application Performance Analyzer Listener. No other product is allowed to use the same port number. Application Performance Analyzer supports both secure and unsecure ports. Secure ports are required to enforce SSL communication in the Application Performance Analyzer GUI.

Avoid using 'Well-Known' (privileged) port numbers as well as registered port numbers. Refer to the Internet Assigned Numbers Authority (IANA) at: www.iana.org for more information on assigned port numbers.

When the ListenTo setting is not specified, the Application Performance Analyzer Listener will listen on this port for every active TCP/IP stack that is defined on the z/OS image.

Default: None.

DefaultSMSClass

Optional.

This setting is used to assign SMS class values to the data sets created by Application Performance Analyzer through the Listener and the GUI plug-in. DefaultSMSClass specifies the default SMS class values used when no other SMSClass keyword from any other customization section is provided, and are used to allocate the following data sets:

- Listener Trace files
- Listener Log files
- GUI plug-in Export files

The syntax for this setting is DefaultSMSClass=(*mgmt,stg,data*) where the positional parameters are:

- *mgmt* = The management class name. It must be one to eight alphanumeric or national characters, as defined by the storage administrator at your installation.
- *stg* = The storage class name. It must be one to eight alphanumeric or national characters, as defined by the storage administrator at your installation.
- *data* = The data class name. It must be one to eight alphanumeric or national characters, as defined by the storage administrator at your installation.

ExportSMSClass
Optional.

This setting is used to assign the SMS class values that Application Performance Analyzer is to use to allocate the data sets created by the GUI plug-in Export actions. It overrides any values specified in the DefaultSMSClass setting in CONFIG BASIC. Omitted positional parameters use the corresponding positional parameter values from DefaultSMSClass.

The syntax for this setting is ExportSMSClass=(*mgmt,stg,data*) where the positional parameters are:

- *mgmt* = The management class name. It must be one to eight alphanumeric or national characters, as defined by the storage administrator at your installation.
- *stg* = The storage class name. It must be one to eight alphanumeric or national characters, as defined by the storage administrator at your installation.
- *data* = The data class name. It must be one to eight alphanumeric or national characters, as defined by the storage administrator at your installation.

ListenTo
Optional. Specifies 1 to 9 specific TCP/IP stacks and port numbers on which the Application Performance Analyzer Listener listens for client connections. When not specified, the Application Performance Analyzer Listener will listen on the port specified in the PORT setting, of every active TCP/IP stack defined on the z/OS image. Application Performance Analyzer supports both secure and unsecure ports. Secure ports are required to enforce SSL communication in the Application Performance Analyzer GUI.

The syntax is ListenTo=(STACK=(*procname,port*) ... STACK=(*procname,port*))

STACK=(*procname,port*) is a unique TCP/IP stack and port number combination that may be repeated up to 9 times where:

- *procname* is required and specifies the name of a member in a procedure library that is used to start a TCP/IP started task (stack).
- *port* is optional and specifies the 1-5 digit TCP/IP port number on which the Application Performance Analyzer Listener listens for client connections. The

default is the port number specified on the PORT setting. The port number can be between 1 and 65,535 and must be unique to a single instance of the Application Performance Analyzer Listener. No other product is allowed to use the same port number. Application Performance Analyzer supports both secure and unsecure ports. Secure ports are required to enforce SSL communication in the Application Performance Analyzer GUI.

Avoid using 'Well-known' (privileged) port numbers as well as registered port numbers. Refer to the Internet Assigned Numbers Authority (IANA) at: `http://www.iana.org` for more information on assigned port numbers.

Example:

```
PORT=8000
ListenTo=(STACK=TCPIP STACK=(TCPIP1,8001) STACK=(TCPIP2,8002) STACK=(TCPIP3,8003))
```

In this example, because the first STACK entry TCPIP has no *port* specified, the Application Performance Analyzer Listener will listen on port 8000, as specified on the PORT setting. TCPIP1 will listen on port 8001, TCPIP2 will listen on port 8002, and TCPIP3 will listen on port 8003.

UNIT

Optional. Specifies the unit name to be used when allocating all new trace data sets.

Default: SYSDA

Unit can be either a device type such as 3390, or esoteric such as SYSDA. A mount status of STORAGE or PUBLIC is non-specific and does not require a VOLSER. Application Performance Analyzer Listener does not allow a VOLSER to be specified.

On a non-SMS-managed system, you must ensure that the UNIT name you specified contains one or more volumes mounted as either STORAGE or PUBLIC. If there are no such volumes mounted, then Application Performance Analyzer Listener is unable to allocate critical files and terminates.

On an SMS-managed system, SMS ignores the unit parameter. To ensure SMS control over all data sets allocated by Application Performance Analyzer Listener, specify an HLQ value that is recognized and managed by SMS.

SECURITY

Optional. Specifies whether workstation users will have their user ID and password verified by the external security system, for example, RACF. ENABLED and DISABLED are the only two valid choices.

Default: ENABLED

MSGLEVEL

Optional. Specifies the severity level of messages to be written to the log data sets. MsgLevel=I writes all messages.

Default: I

SESSIONTIMEOUT

Optional.

Specifies the length of time a workstation session is allowed to remain connected to the Application Performance Analyzer Listener until it is disconnected. In order for the session to remain connected, the workstation user must cause Application Performance Analyzer GUI to initiate activity with the Application Performance Analyzer Listener within the SESSIONTIMEOUT interval. This is to prevent the Application Performance Analyzer GUI users from monopolizing Application Performance Analyzer Listener sessions.

It must be specified as a four digit number. The value is in the form of: *HHMM*. Where *HH* is the number hours in the range of 00-99, and *MM* is the number of minutes in the range of 00-59. The maximum allowed connection time is 9959 or 4 days 3 hours and 59 minutes. The minimum allowed value is 5 minutes, except for a value of 0000 which indicates that session timeout will not be applied.

Note: This value is not affected by the KEEPCONNECTED value if it also set. Default: 0000

KEEPCONNECTED
Optional.

Specifies an interval in minutes. This a heart beat or pulse interval used by the Application Performance Analyzer GUI to ping the Application Performance Analyzer Listener once per interval. The purpose of this value is to keep the connection active. It is common when you are behind a NAT proxy or a firewall to be disconnected due to inactivity.

The maximum interval is 99 minutes. The minimum allowed value is 4 minutes, except for a value of 00 which indicates that no pings will be sent.

Note: This value does not affect the SESSIONTIMEOUT value if it is also set. Default: 00

DataHLQ
Optional.

Specifies the high-level qualifier used for data sets created by GUI requests to export observation files. The value that you specify for DataHLQ must comply with the JCL rule for specifying a data set name. It can be no more than 17 characters long.

If no ExportDSN parameter is specified, it will be the high level qualifier on export data sets. If an ExportDSN is specified, the DataHLQ value can be specified in the export file data set name. See below.

Default: None.

ExportDSN
Optional.

Specifies up to 64 characters to be used to name export data sets. The value in this setting overrides the default generated export data set name. The name may consist of system symbols and Application Performance Analyzer defined user symbols, which are substituted with values at the time the export dataset is allocated. The total length is restricted to 44 characters after substitution. If symbols are concatenated with other symbols or hard coded characters, use a period to separate the items. The Application Performance Analyzer defined user symbols are:

- &DATAHLQ - the value specified in the DataHLQ configuration parameter, if any. This symbol may only be specified as the first symbol in the string. Do not use this symbol if there is no DataHLQ configuration parameter.
- &USERID - the user ID of the user requesting the export.
- &JOB - the name of the address space associated with the observation request selected by the user for export.

 If the job name is a dash (-) as it would be for DB2 or IMS Multiple Address Space Support, then $MASS$ will be substituted for the job name.

If the job name contains a percent sign (%) for Multi-job support, then $MJOB$ will be substituted for the job name.

- &SAMP - the observation request number of the request selected by the user for export. The value substituted will be in the form **R**_nnnn_ where _nnnn_ is the request number.

Note: System symbol &JOBNAME will substitute the name of the Listener started task, not the observation request's job name.

Example:

```
ExportDSN=&DATAHLQ..&JOB..D&LYYMMDD..&SAMP..XMIT
```

The following name would be generated:

```
MYDHLQ.MYJOBNM.D121129.R1234.XMIT
```

Default: none.

CONFIG LOGFILE statement

Logging records activity within the Application Performance Analyzer Listener started task. Log data is always written to SYSOUT, but you have the option of also writing this data to log data sets. LOGFILE settings allow you manage these data sets. The following settings might be specified:

LogGDG

Optional. If you have specified LogToFile=YES, then LogGDG allows you to specify whether the log data sets are allocated as GDGs or normal sequential data sets. If you specify LogGDG=YES, you must first create the Generation Data Group Index using an index name of _yourhlq_.LLG (or _LogHLQ_.LLG if you choose to use LogHLQ). YES and NO are the only valid values.

Default: NO.

Note: A GDG log file cannot be shared by multiple instances of Application Performance Analyzer Listener. A second started task with the same GDG log file fails on start up with messages IKJ56225I and CAZL034S. A unique GDG log file must be defined for each instance of Application Performance Analyzer Listener.

LogHLQ

Specifies the high-level qualifier to be used when creating log data sets. LogHLQ must comply with the JCL rule for specifying a data set name. It can be no more than 17 characters long. A data set is only created if LogToFile=YES is specified.

If this parameter is not specified, the value that you specify for HLQ on the CONFIG BASIC statement is used.

Default: None.

LogPRIM

Optional. If you have specified LogToFile=YES, then LogPRIM allows you to specify the size of each log data set that is allocated. No secondary specification is permitted. Therefore, when the log data set has filled, it is closed and a new log data set is allocated.

Default: 15.

LogToFile

Optional. Specifies whether you want log data to be written to permanent data sets. YES and NO are the only valid values.

Default: NO.

Note: It is recommended that LogToFile be set to YES. The log can be used to help diagnose problems.

LogUNIT

Optional. If you have specified LogToFile=YES, then LogUNIT allows you to specify whether you want the log data set to be allocated in Tracks or Cylinders. T and C are the only valid values.

Default: T.

SMSClass

Optional.

This setting is used to assign the SMS class values that Application Performance Analyzer is to use to allocate the Listener log file data sets. It overrides any values specified in the DefaultSMSClass setting in CONFIG BASIC. Omitted positional parameters use the corresponding positional parameter values from DefaultSMSClass.

The syntax for this setting is SMSClass=(*mgmt,stg,data*) where the positional parameters are:

- *mgmt* = The management class name. It must be one to eight alphanumeric or national characters, as defined by the storage administrator at your installation.
- *stg* = The storage class name. It must be one to eight alphanumeric or national characters, as defined by the storage administrator at your installation.
- *data* = The data class name. It must be one to eight alphanumeric or national characters, as defined by the storage administrator at your installation.

CONFIG APPL_WSFE statement

This statement defines configurable parameters for the Application Performance Analyzer Workstation Front-End application (GUI). The following parameter can be specified:

EnableEmail

Optional. Specifies whether the Application Performance Analyzer GUI is to enable its report email feature or not. YES and NO are the only valid values.

Default: YES

MAXOPENOBS

Optional. Specifies the maximum number of concurrent observations that each workstation connection is permitted to have open at any one time. Acceptable values are in the range of 2-9.

Default: 5.

CONFIG OPER statement

These settings pertain to operational parameters and should not be changed by the user. These should only be changed on the advice of IBM. Application Performance Analyzer ships with only one parameter specified, which is DiagLevel=1.

Configuring external security (RACF)

This step is part of the Application Performance Analyzer configuration and must be completed only when your Application Performance Analyzer started tasks are configured to use an external security product to control access to Application Performance Analyzer functions. You can bypass this step if your installation has chosen to control access to functions using Application Performance Analyzer's internal security, for example, its own access rules facility.

Note: Configuring external security is suggested if internal security is used in a sysplex environment when running Application Performance Analyzer. This can prevent situations in which one user has set up an observation request for job *aaaaaaaa* to run in the sysplex environment, and a second user submits a job with the same name on one of the other machines, resulting in the second job capturing data from the first job submitted.

Access to Application Performance Analyzer functions can be controlled by RACF or a RACF-compatible external security product. Refer to "Defining RACF profiles" on page 47 in Chapter 2 for details on how to use RACF or a RACF-compatible external security product to control access to Application Performance Analyzer functions.

A RACF action is required to support workstation users that connect to an Application Performance Analyzer started task via the Application Performance Analyzer Listener started task. The action is called USE.LISTENER.

Below are examples of RACF commands based on the assumptions made in "Defining RACF profiles" on page 47:

Define the resource profile:

```
RDEFINE FACILITY SYSTEMD.USE.LISTENER UACC(NONE)
```

Define the permission to allow all users to connect to the Application Performance Analyzer started task that is running on SYSTEMD:

```
PERMIT SYSTEMD.USE.LISTENER CLASS(FACILITY) ACCESS(READ)
```

Starting the Application Performance Analyzer Listener-started task

The Application Performance Analyzer Listener-started task must be started for it to become operational. Once started, workstation users with the proper authority can connect to any active Application Performance Analyzer-started task using the Application Performance Analyzer GUI.

If you have not re-IPLed after making changes to parmlib as described in step 6 on page 80 you can start the Listener by entering the command 'S *procname*' on the system console, where *procname* is the Application Performance Analyzer Listener-started task procedure name.

When you re-IPL your system, the Application Performance Analyzer Listener-started task is automatically started.

Chapter 7. About Application Performance Analyzer Plug-ins for Eclipse

The Application Performance Analyzer Plug-in for Eclipse is a desktop version of the Application Performance Analyzer ISPF interface, which in addition to new features, provides a majority of parallel tools and functionality. It is used for submitting new observation requests, and for navigating the Performance Analysis Reports generated from observation requests. The Application Performance Analyzer Plug-in can be installed into existing Eclipse based desktop products such as RD/z, CICS Explorer, IMS Explorer and Eclipse.

The optional Application Performance Analyzer Bridge for CICS Explorer for Eclipse provides an interface for the Application Performance Analyzer Plug-in to use the CICS Explorer connection framework to setup and launch the remote z/OS connection to Application Performance Analyzer for z/OS. As it provides this limited set of functionality, it must be installed into your Eclipse based desktop product along with the Application Performance Analyzer Plug-in.

The Application Performance Analyzer Plug-in communicates with Application Performance Analyzer for z/OS through an Application Performance Analyzer Listener started task on z/OS. The Application Performance Analyzer Listener must be installed, configured and active on z/OS before you can connect and use the Application Performance Analyzer Plug-in to access Application Performance Analyzer for z/OS.

Multi-language support

The Application Performance Analyzer Plug-in can display menus and dialogs in English, Japanese or Korean. Input must be in English. The Application Performance Analyzer Plug-in will detect the regional and language options of your workstation and launch in the appropriate language.

Accessing the Application Performance Analyzer Plug-ins

The Application Performance Analyzer Plug-in and the Application Performance Analyzer Bridge for CICS Explorer are packaged in one zip file that is available and can be downloaded from the IBM Problem Determination Tools Plug-ins website.

The Readme PDF available on the Application Performance Analyzer Plug-in for Eclipse page of this website includes the system requirements, prerequisites, download, and installation instructions. Please read this prior to downloading and installing the plug-ins.

Appendix A. Support resources and problem solving information

This section shows you how to quickly locate information to help answer your questions and solve your problems. If you have to call IBM support, this section provides information that you need to provide to the IBM service representative to help diagnose and resolve the problem.

For a comprehensive multimedia overview of IBM software support resources, see the IBM Education Assistant presentation "IBM Software Support Resources for System z Enterprise Development Tools and Compilers products" at http://publib.boulder.ibm.com/infocenter/ieduasst/stgv1r0/index.jsp?topic=/com.ibm.iea.debugt/debugt/6.1z/TrainingEducation/SupportInfoADTools/player.html.

- "Searching knowledge bases"
- "Getting fixes" on page 93
- "Subscribing to support updates" on page 93
- "Contacting IBM Support" on page 95

Searching knowledge bases

You can search the available knowledge bases to determine whether your problem was already encountered and is already documented.

- "Searching the information center"
- "Searching product support documents"

Searching the information center

You can find this publication and documentation for many other products in the IBM System z Enterprise Development Tools & Compilers information center at http://publib.boulder.ibm.com/infocenter/pdthelp/v1r1/index.jsp. Using the information center, you can search product documentation in a variety of ways. You can search across the documentation for multiple products, search across a subset of the product documentation that you specify, or search a specific set of topics that you specify within a document. Search terms can include exact words or phrases, wild cards, and Boolean operators.

To learn more about how to use the search facility provided in the IBM System z Enterprise Development Tools & Compilers information center, you can view the multimedia presentation at http://publib.boulder.ibm.com/infocenter/pdthelp/v1r1/index.jsp?topic=/com.ibm.help.doc/InfoCenterTour800600.htm.

Searching product support documents

If you need to look beyond the information center to answer your question or resolve your problem, you can use one or more of the following approaches:

- Find the content that you need by using the IBM Support Portal at www.ibm.com/software/support or directly at www.ibm.com/support/entry/portal.

  The IBM Support Portal is a unified, centralized view of all technical support tools and information for all IBM systems, software, and services. The IBM

Support Portal lets you access the IBM electronic support portfolio from one place. You can tailor the pages to focus on the information and resources that you need for problem prevention and faster problem resolution.

Familiarize yourself with the IBM Support Portal by viewing the demo videos at https://www.ibm.com/blogs/SPNA/entry/the_ibm_support_portal_videos?lang=en_us about this tool. These videos introduce you to the IBM Support Portal, explore troubleshooting and other resources, and demonstrate how you can tailor the page by moving, adding, and deleting portlets.

Access a specific IBM Software Support site:
  – Application Performance Analyzer for z/OS Support
  – Debug Tool for z/OS Support
  – Enterprise COBOL for z/OS Support
  – Enterprise PL/I for z/OS Support
  – Fault Analyzer for z/OS Support
  – File Export for z/OS Support
  – File Manager for z/OS Support
  – WebSphere Developer Debugger for System z Support
  – WebSphere Studio Asset Analyzer for Multiplatforms Support
  – Workload Simulator for z/OS and OS/390 Support

• Search for content by using the IBM masthead search. You can can use the IBM masthead search by typing your search string into the Search field at the top of any ibm.com page.

• Search for content by using any external search engine, such as Google, Yahoo, or Bing. If you use an external search engine, your results are more likely to include information that is outside the ibm.com domain. However, sometimes you can find useful problem-solving information about IBM products in newsgroups, forums, and blogs that are not on ibm.com. Include "IBM" and the name of the product in your search if you are looking for information about an IBM product.

• The IBM Support Assistant (also referred to as ISA) is a free local software serviceability workbench that helps you resolve questions and problems with IBM software products. It provides quick access to support-related information. You can use the IBM Support Assistant to help you in the following ways:
  – Search through IBM and non-IBM knowledge and information sources across multiple IBM products to answer a question or solve a problem.
  – Find additional information through product and support pages, customer news groups and forums, skills and training resources and information about troubleshooting and commonly asked questions.

In addition, you can use the built in Updater facility in IBM Support Assistant to obtain IBM Support Assistant upgrades and new features to add support for additional software products and capabilities as they become available.

For more information, and to download and start using the IBM Support Assistant for IBM System z Enterprise Development Tools & Compilers products, please visit http://www.ibm.com/support/docview.wss?rs=2300 &context=SSFMHB&dc=D600&uid=swg21242707&loc=en_US&cs=UTF-8 &lang=en.

General information about the IBM Support Assistant can be found on the IBM Support Assistant home page at http://www.ibm.com/software/support/isa.

Getting fixes

A product fix might be available to resolve your problem. To determine what fixes and other updates are available, select a link from the following list:

- Latest PTFs for Application Performance Analyzer for z/OS
- Latest PTFs for Debug Tool for z/OS
- Latest PTFs for Fault Analyzer for z/OS
- Latest PTFs for File Export for z/OS
- Latest PTFs for File Manager for z/OS
- Latest PTFs for Optim Move for DB2
- Latest PTFs for WebSphere Studio Asset Analyzer for Multiplatforms
- Latest PTFs for Workload Simulator for z/OS and OS/390

When you find a fix that you are interested in, click the name of the fix to read its description and to optionally download the fix.

Subscribe to receive email notifications about fixes and other IBM Support information as described in Subscribing to Support updates..

Subscribing to support updates

To stay informed of important information about the IBM products that you use, you can subscribe to updates. By subscribing to receive updates, you can receive important technical information and updates for specific Support tools and resources. You can subscribe to updates by using the following:

- RSS feeds and social media subscriptions
- My Notifications

To subscribe to Support updates, follow the steps below. Additional information is provided at http://www.ibm.com/support/docview.wss?rs=615 &uid=swg21172598.

1. Go to the IBM software support site at http://www.ibm.com/software/ support.
2. Click the **My Notifications** link in the **Notifications** portlet on the page that is displayed.
3. If you have already registered for **My notifications**, sign in and skip to the next step. If you have not registered, click **register now**. Complete the registration form using your e-mail address as your IBM ID and click **Submit**.
4. In the **My notifications** tool, click the **Subscribe** tab to specify products for which you want to receive e-mail updates.
5. To specify Problem Determination Tools products, click **Other software** and then select the products for which you want to receive e-mail updates, for example, **Debug Tool for z/OS** and **File Manager for z/OS**.
6. To specify a COBOL or PL/I compiler, click **Rational** and then select the products for which you want to receive e-mail updates, for example, **Enterprise COBOL for z/OS**.
7. After selecting all products that are of interest to you, scroll to the bottom of the list and click **Continue**.
8. Determine how you want to save your subscription. You can use the default subscription name or create your own by entering a new name in the **Name** field. It is recommended that you create your own unique subscription name using something easily recognized by you. You can create a new folder by

entering a folder name in the **New** field or select an existing folder from the pulldown list. A folder is a container for multiple subscriptions.

9. Specify the types of documents you want and the e-mail notification frequency.

10. Scroll to the bottom of the page and click **Submit**.

To view your current subscriptions and subscription folders, click **My subscriptions**.

If you experience problems with the **My notifications** feature, click the **Feedback** link in the left navigation panel and follow the instructions provided.

RSS feeds and social media subscriptions

For general information about RSS, including steps for getting started and a list of RSS-enabled IBM web pages, visit the IBM Software Support RSS feeds site at http://www.ibm.com/software/support/rss/other/index.html. For information about the RSS feed for the IBM System z Enterprise Development Tools & Compilers information center, refer to the Subscribe to information center updates topic in the information center at http://publib.boulder.ibm.com/infocenter/pdthelp/v1r1/topic/com.ibm.help.doc/subscribe_info.html.

My Notifications

With My Notifications, you can subscribe to Support updates for any IBM product. You can specify that you want to receive daily or weekly email announcements. You can specify what type of information you want to receive (such as publications, hints and tips, product flashes (also known as alerts), downloads, and drivers). My Notifications enables you to customize and categorize the products about which you want to be informed and the delivery methods that best suit your needs.

To subscribe to Support updates, follow the steps below. Additional information is provided at http://www.ibm.com/support/docview.wss?rs=615 &uid=swg21172598.

1. Go to the IBM software support site at http://www.ibm.com/software/support.

2. Click the **My Notifications** link in the **Notifications** portlet on the page that is displayed.

3. If you have already registered for **My notifications**, sign in and skip to the next step. If you have not registered, click **register now**. Complete the registration form using your e-mail address as your IBM ID and click **Submit**.

4. In the **My notifications** tool, click the **Subscribe** tab to specify products for which you want to receive e-mail updates.

5. To specify Problem Determination Tools products, click **Other software** and then select the products for which you want to receive e-mail updates, for example, **Debug Tool for z/OS** and **File Manager for z/OS**.

6. To specify a COBOL or PL/I compiler, click **Rational** and then select the products for which you want to receive e-mail updates, for example, **Enterprise COBOL for z/OS**.

7. After selecting all products that are of interest to you, scroll to the bottom of the list and click **Continue**.

8. Determine how you want to save your subscription. You can use the default subscription name or create your own by entering a new name in the **Name**

field. It is recommended that you create your own unique subscription name using something easily recognized by you. You can create a new folder by entering a folder name in the **New** field or select an existing folder from the pulldown list. A folder is a container for multiple subscriptions.

9. Specify the types of documents you want and the e-mail notification frequency.

10. Scroll to the bottom of the page and click **Submit**.

To view your current subscriptions and subscription folders, click **My subscriptions**.

If you experience problems with the **My notifications** feature, click the **Feedback** link in the left navigation panel and follow the instructions provided.

Contacting IBM Support

IBM Support provides assistance with product defects, answering FAQs, and performing rediscovery.

After trying to find your answer or solution by using other self-help options such as technotes, you can contact IBM Support. Before contacting IBM Support, your company must have an active IBM maintenance contract, and you must be authorized to submit problems to IBM. For information about the types of available support, see the information below or refer to the Support portfolio topic in the Software Support Handbook at http://www14.software.ibm.com/webapp/set2/sas/f/handbook/offerings.html.

- For IBM distributed software products (including, but not limited to, Tivoli®, Lotus®, and Rational® products, as well as DB2 and WebSphere products that run on Windows, or UNIX operating systems), enroll in Passport Advantage® in one of the following ways:

    Online
    Go to the Passport Advantage Web site at http://www.lotus.com/services/passport.nsf/ WebDocs/Passport_Advantage_Home and click **How to Enroll**.

    By phone
    For the phone number to call in your country, go to the Contacts page of the *IBM Software Support Handbook* on the Web at http://www14.software.ibm.com/webapp/set2/sas/f/handbook/contacts.html and click the name of your geographic region.

- For customers with Subscription and Support (S & S) contracts, go to the Software Service Request Web site at http://www.ibm.com/support/servicerequest.

- For customers with IBMLink, CATIA, Linux, S/390®, iSeries®, pSeries®, zSeries®, and other support agreements, go to the IBM Support Line Web site at http://www.ibm.com/services/us/index.wss/so/its/a1000030/dt006.

- For IBM eServer™ software products (including, but not limited to, DB2 and WebSphere products that run in zSeries, pSeries, and iSeries environments), you can purchase a software maintenance agreement by working directly with an IBM sales representative or an IBM Business Partner. For more information about support for eServer software products, go to the IBM Technical Support Advantage Web site at http://www.ibm.com/servers/eserver/techsupport.html.

If you are not sure what type of software maintenance contract you need, call 1-800-IBMSERV (1-800-426-7378) in the United States. From other countries, go to the Contacts page of the *IBM Software Support Handbook* on the Web at http://www14.software.ibm.com/webapp/set2/sas/f/handbook/contacts.html and click the name of your geographic region for phone numbers of people who provide support for your location.

Complete the following steps to contact IBM Support with a problem:
1. "Define the problem and determine the severity of the problem"
2. "Gather diagnostic information"
3. "Submit the problem to IBM Support" on page 97

To contact IBM Software support, follow these steps:

Define the problem and determine the severity of the problem

Define the problem and determine severity of the problem When describing a problem to IBM, be as specific as possible. Include all relevant background information so that IBM Support can help you solve the problem efficiently.

IBM Support needs you to supply a severity level. Therefore, you need to understand and assess the business impact of the problem that you are reporting. Use the following criteria:

Severity 1
> The problem has a **critical** business impact. You are unable to use the program, resulting in a critical impact on operations. This condition requires an immediate solution.

Severity 2
> The problem has a **significant** business impact. The program is usable, but it is severely limited.

Severity 3
> The problem has **some** business impact. The program is usable, but less significant features (not critical to operations) are unavailable.

Severity 4
> The problem has **minimal** business impact. The problem causes little impact on operations, or a reasonable circumvention to the problem was implemented.

For more information, see the Getting IBM support topic in the Software Support Handbook at http://www14.software.ibm.com/webapp/set2/sas/f/handbook/getsupport.html.

Gather diagnostic information

To save time, if there is a Mustgather document available for the product, refer to the Mustgather document and gather the information specified. Mustgather documents contain specific instructions for submitting your problem to IBM and gathering information needed by the IBM support team to resolve your problem. To determine if there is a Mustgather document for this product, go to the product support page and search on the term Mustgather. At the time of this publication, the following Mustgather documents are available:

- Mustgather: Read first for problems encountered with Application Performance Analyzer for z/OS: http://www.ibm.com/support/docview.wss?rs=2300 &context=SSFMHB&q1=mustgather&uid=swg21265542&loc=en_US&cs=utf-8 ⟨=en
- Mustgather: Read first for problems encountered with Debug Tool for z/OS: http://www.ibm.com/support/docview.wss?rs=615&context=SSGTSD &q1=mustgather&uid=swg21254711&loc=en_US&cs=utf-8&lang=en
- Mustgather: Read first for problems encountered with Fault Analyzer for z/OS:http://www.ibm.com/support/docview.wss?rs=273&context=SSXJAJ &q1=mustgather&uid=swg21255056&loc=en_US&cs=utf-8&lang=en
- Mustgather: Read first for problems encountered with File Manager for z/OS: http://www.ibm.com/support/docview.wss?rs=274&context=SSXJAV &q1=mustgather&uid=swg21255514&loc=en_US&cs=utf-8&lang=en
- Mustgather: Read first for problems encountered with Enterprise COBOL for z/OS: http://www.ibm.com/support/docview.wss?rs=2231&context=SS6SG3 &q1=mustgather&uid=swg21249990&loc=en_US&cs=utf-8&lang=en
- Mustgather: Read first for problems encountered with Enterprise PL/I for z/OS: http://www.ibm.com/support/docview.wss?rs=619&context=SSY2V3 &q1=mustgather&uid=swg21260496&loc=en_US&cs=utf-8&lang=en

If the product does not have a Mustgather document, please provide answers to the following questions:

- What software versions were you running when the problem occurred?
- Do you have logs, traces, and messages that are related to the problem symptoms? IBM Software Support is likely to ask for this information.
- Can you re-create the problem? If so, what steps were performed to re-create the problem?
- Did you make any changes to the system? For example, did you make changes to the hardware, operating system, networking software, and so on.
- Are you currently using a workaround for the problem? If so, be prepared to explain the workaround when you report the problem.

Submit the problem to IBM Support

You can submit your problem to IBM Support in one of three ways:

Online using the IBM Support Portal
Click **Service request** on the IBM Software Support site at http://www.ibm.com/software/support. On the right side of the Service request page, expand the Product related links section. Click Software support (general) and select ServiceLink/IBMLink to open an Electronic Technical Response (ETR). Enter your information into the appropriate problem submission form.

Online using the Service Request tool
The Service Request tool can be found at http://www.ibm.com/software/ support/servicerequest.

By phone
Call 1-800-IBMSERV (1-800-426-7378) in the United States or, from other countries, go to the Contacts page of the *IBM Software Support Handbook* at http://www14.software.ibm.com/webapp/set2/sas/f/handbook/ contacts.html and click the name of your geographic region.

If the problem you submit is for a software defect or for missing or inaccurate documentation, IBM Support creates an Authorized Program Analysis Report (APAR). The APAR describes the problem in detail. Whenever possible, IBM Support provides a workaround that you can implement until the APAR is resolved and a fix is delivered. IBM publishes resolved APARs on the IBM Support website daily, so that other users who experience the same problem can benefit from the same resolution.

After a Problem Management Record (PMR) is open, you can submit diagnostic MustGather data to IBM using one of the following methods:

- FTP diagnostic data to IBM. For more information, refer to http://www.ibm.com/support/docview.wss?rs=615&uid=swg21154524.

- If FTP is not possible, email diagnostic data to techsupport@mainz.ibm.com. You must add PMR xxxxx bbb ccc in the subject line of your email. xxxxx is your PMR number, bbb is your branch office, and ccc is your IBM country code. Go to http://itcenter.mainz.de.ibm.com/ecurep/mail/subject.html for more details.

Always update your PMR to indicate that data has been sent. You can update your PMR online or by phone as described above.

Appendix B. Stopping the IBM Application Performance Analyzer-started tasks

To stop the Application Performance Analyzer-started tasks, complete the following steps:

1. Enter the command P *procname* from the system console. Where *procname* is the name of the Application Performance Analyzer-started task procedure.
2. If P *procname* fails to stop the started task, retry the command.

Appendix C. Sysplex External Security Conversion Aid

In order to use the sysplex security enhancements, you must select the Application Performance Analyzer External Security option. As part of these enhancements, the naming convention for the RACF (or equivalent) security profiles has changed.

Note: The changes apply to all External Security option users in both sysplex and non-sysplex environments.

Currently, the RACF general resource profiles use the Application Performance Analyzer started task identifier (STCID) for a prefix. The sysplex enhancements require that the profile names use the sysplex system names of the target or servicing systems as the prefix.

A REXX utility program, CAZRACON in library *hlq*.SCAZEXEC, is provided to aid in the conversion of the external security profiles. This utility supports RACF environments only. CAZRACON is a REXX program and can be run in TSO foreground or in a TSO batch job. Sample JCL is provided below. Using the existing RACF profiles as a base, CAZRACON generates the JCL and RACF commands needed to create new RACF profiles using the sysplex naming conventions. CAZRACON must be run once for each Application Performance Analyzer instance or STCID.

Sample JCL

```
//jobname   JOB <job parameters>
//*
//* Run CAZRACON to convert Application Performance
//* Anaylyzer RACF profiles for the sysplex
//* security enhancements.
//*
//TSO      EXEC PGM=IKJEFT01,DYNAMNBR=20
//SYSEXEC  DD  DISP=SHR,DSN=hlq.SCAZEXEC
//SYSTSPRT DD  SYSOUT=*
//SYSPRINT DD  SYSOUT=*
//SYSUDUMP DD  SYSOUT=*
//SYSTSIN  DD  *
 CAZRACON STCID=CAZ0 SYSNAMES=(SYSA,SYSB) CLASS=CAZCLASS
/*
//
```

CAZRACON syntax

CAZRACON STCID=*stcid*
        SYSNAMES=*system_name* | **SYSNAMES**=(*system_name,system_name,....*)
        DSN=*output_dataset_name*
        CLASS=*RACF_class*

Parameters

STCID=*stcid*

- Required.
- Specifies the Application Performance Analyzer started task identifier to be converted.

SYSNAMES=*system_name* | **SYSNAMES=(***system_name,system_name,....***)**

- Required.
- Specifies the target sysplex system name, or list of system names, for this Application Performance Analyzer instance. Up to 32 system names can be specified.
- In a non-sysplex environment, the system name is the same as the name of the system where the observation request was made.

DSN=`output_dataset_name`
- Optional.
- Default: `tsoprefix.`CAZRACF.COMMANDS
- Fully qualified data set names must be in quotes.
- If not specified, the default data set will be created.
- If specified and the specified data set does not exist, the data set will be created.
- If the specified data set is a PDS, it must exist and the member name must be included. The member can be an existing or a new member.

CLASS=`RACF_class`
- Optional.
- Default: FACILITY
- Specifies a RACF class for the new profiles. Application Performance Analyzer now provides the option for a user-specified class for the external security profiles (see the `SecurityClassName` parameter in the "CONFIG BASIC statement" on page 26). This parameter allows the user to migrate the new profiles from the FACILITY class to the user-specified class in the CONFIG settings.

The CAZRACON utility will extract all the RACF profiles that contain the specified STCID prefix. From the extracted information, RACF RDEFINE commands are generated for defining the corresponding sysplex profiles. More specifically, for each extracted profile, new profiles are created with the system names replacing the STCID as the first qualifier. A profile is created for each sysname specified.

The FROM parameter is used on the RDEF command to copy the attributes of the old profile, including the authorization list.

Note: Be aware that the RDEF command with the FROM parameter does not guarantee that the new profile is identical to the old profile, minor differences can occur. See the *IBM z/OS Security Server RACF Command Language Reference*, (SA22-7687), for details.

RACF RDELETE commands are also generated to delete the old profiles. The RDEL commands are written as comments. Remove the comment characters /* before you submit the job.

Depending on the sysplex installation, duplicate RDEF commands can be generated for a specific system name. This can happen when the utility is run for multiple Application Performance Analyzer instances in the sysplex, and two or more of the instances have common target systems. In this case, you must decide which RDEF commands best suit your sysplex environment. Additionally, you might need to alter the authorization lists.

Examples

Suppose the sysplex consists of two systems, SYSA and SYSB, with an Application Performance Analyzer instance on each system, CAZA and CAZB, respectively. Both systems are valid targets for both CAZA and CAZB.

The existing RACF profiles in the FACILITY class are:

```
CAZA.VIEW.USER123
CAZA.MEASURE.JOB.JOB123

CAZB.VIEW.USER123
CAZB.MEASURE.JOB.JOB123
```

When the CAZRACON utility is run for CAZA with the following command,

```
CAZRACON STCID=CAZA
         SYSNAMES=(SYSA,SYSB)
         CLASS=CAZCLASS
         DSN=CAZA.COMMANDS
```

the following output is written to data set *tsoprefix*.CAZA.COMMANDS:

```
        (INSERT JOB CARD)
//*
//* Create RACF Profiles for Sysplex security enhancements.
//*
//TSO      EXEC PGM=IKJEFT01,DYNAMNBR=20
//SYSTSPRT DD  SYSOUT=*
//SYSPRINT DD  SYSOUT=*
//SYSUDUMP DD  SYSOUT=*
//SYSTSIN  DD  *
 RDEF CAZCLASS SYSA.VIEW.USER123 FROM(CAZA.VIEW.USER123) -
      FCLASS(FACILITY)
 RDEF CAZCLASS SYSB.VIEW.USER123 FROM(CAZA.VIEW.USER123) —
      FCLASS(FACILITY)

 RDEF CAZCLASS SYSA.MEASURE.JOB.JOB123 FROM(CAZA.MEASURE.JOB.JOB123) -
      FCLASS(FACILITY)
 RDEF CAZCLASS SYSB.MEASURE.JOB.JOB123 FROM(CAZA.MEASURE.JOB.JOB123)-
      FCLASS(FACILITY)

 /* RDEL FACILITY CAZA.VIEW.USER123
 /* RDEL FACILITY CAZA.MEASURE.JOB.JOB123
/*
//
```

When the utility is run for CAZB with the following command,

```
CAZRACON STCID=CAZB
         SYSNAMES=(SYSA,SYSB)
         CLASS=CAZCLASS
         DSN=CAZB.COMMANDS
```

the following output is written to data set *tsoprefix*.CAZB.COMMANDS:

```
        (INSERT JOB CARD)
//*
//* Create RACF Profiles for Sysplex security enhancements.
//*
//TSO      EXEC PGM=IKJEFT01,DYNAMNBR=20
//SYSTSPRT DD  SYSOUT=*
//SYSPRINT DD  SYSOUT=*
//SYSUDUMP DD  SYSOUT=*
//SYSTSIN  DD  *
 RDEF CAZCLASS SYSA.VIEW.USER123 FROM(CAZB.VIEW.USER123) -
      FCLASS(FACILITY)
 RDEF CAZCLASS SYSB.VIEW.USER123 FROM(CAZB.VIEW.USER123) —
```

```
        FCLASS(FACILITY)

 RDEF CAZCLASS SYSA.MEASURE.JOB.JOB123 FROM(CAZB.MEASURE.JOB.JOB123) -
        FCLASS(FACILITY)
 RDEF CAZCLASS SYSB.MEASURE.JOB.JOB123 FROM(CAZB.MEASURE.JOB.JOB123)-
        FCLASS(FACILITY)

 /* RDEL FACILITY CAZB.VIEW.USER123
 /* RDEL FACILITY CAZB.MEASURE.JOB.JOB123
 /*
 //
```

Notice that there are RDEF commands in each output data set for the same profiles
from two different sources. If the source profiles have the same authorization lists,
then either can be used; otherwise, you need to determine which commands to use
and whether any changes to the authorization lists are required.

Appendix D. Application Performance Analyzer Data Extractor Module Resolution User Exit

Overview

The Application Performance Analyzer Data Extractor Module Resolution User Exit (DEXMR) is a programming interface to the Application Performance Analyzer Sampling Engine. Specifically, this exit enables users to attribute program execution to MVS modules or USS executable files that typically would be reported as unresolved (NOSYMB) by Application Performance Analyzer.

The exit is developed and maintained by the client. It requires an assembler developer who is experienced in programming in an SRB environment.

Module Resolution

For every observed PSW address, Application Performance Analyzer attempts to resolve it to either an MVS module or USS exec. Application Performance Analyzer is able to do this for all modules that are managed by Content Supervision Services. That is, modules that are brought into memory by MVS services such as: LOAD, LINK, XTCL, ATTACH, ATTACHX, IDENITFY or execs that are brought into memory by UNIX System Services such as exec. Application Performance Analyzer is able to resolve PSW addresses for modules that are in the JPA, LINKLIST, LPA, and Nucleus.

Occasionally, Application Performance Analyzer samples PSW addresses that are not managed by CSV and, therefore, are reported as unresolved. These unresolved addresses belong to modules that are managed by the sampled application itself or by third party products. The Data Extractor Module Resolution permits client sites with knowledge of the sampled application or third party product to resolve these addresses programmatically. If Data Extractor Module Resolution has been activated, the Application Performance Analyzer Sampling Engine calls this exit for all unresolved addresses as described above.

Data Extractor Module Resolution Program Design Considerations

The following describes information that a programmer needs to know for developing the Data Extractor Module Resolution.

1. The exit is called once to initialize itself at the start of the sample session and once again at the end of sampling to perform cleanup and termination.
2. For each sampling interval during the sample session, the Application Performance Analyzer Sampling Engine analyzes the Job Step TCB and all subtasks. Part of this analysis involves resolving the PSW addresses. Therefore, the Data Extractor Module Resolution should expect to be called one or more times per sample interval and once for each unresolved address.
3. The Application Performance Analyzer Sampling Engine executes as an SRB. Since the Data Extractor Module Resolution is called by the Sampling Engine, it must also comply with all the rules of an SRB as described in *z/OS MVS Programming: Authorized Assembler Services Guide; Chapter: Implications of Running in SRB Mode*.
4. It must not establish an FRR. Recovery is provided by the Sampling Engine.

5. A user can request that a Job Step be sampled from start of step to end of step. Therefore, it is very likely that the exit is called early in the Job Step's creation. Under this condition the sampled application does not have sufficient time to initialize itself. Therefore, the exit might not find the environment that it expects. This is especially true during the Initialization (First) function call to the exit. If the exit determines that the sampled application is not ready, the exit should remember this fact and not attempt to initialize itself at this time. Instead, it should reattempt initialization during Module Resolution function calls.

   The exit can also be called late in the Job Step termination and find that the sampled application already terminated. In this case, the exit might find it impossible to resolve an address or even perform its termination function.

   For both early and late calls, the exit can expect segment- or page-translation exceptions if it attempts to reference memory that has not been obtained yet. The exit should validate such addresses by checking for zeros and using the TPROT instruction to verify the address.

6. The exit must be reentrant and able to handle concurrent calls. It is very possible for the exit to be called by the Sampling Engine SRB, the DB2 intercept and the IMS intercept all at the same time.

7. If the exit requires its own memory, it should allocate this memory from subpool 235. Storage in this subpool is owned by the address space and is not freed at end of step. The exit can therefore be guaranteed that the memory persists until its Termination (Last) function call. The exit must free this memory to prevent memory leaks.

8. Economize on path length. The exit could be called multiple times per second and a long path length could adversely affect the overall performance of the Sampling Engine. Avoid calling MVS system services. Try to restrict these calls to Initialization or Termination functions as they are only called once per sample session.

9. Data Extractor Module Resolution is not activated by default. It must be activated by specifying DEXMR=YES in the CONFIG SAMPLE section of the configuration file found in the member CAZCNFG0 in *hlq*.SCAZSAMP.

10. Data Extractor Module Resolution is a global exit. When it is properly activated, it is called during all sampling sessions. If this is undesirable, the programmer can add code to the Initialization function call to deactivate itself for the remainder of the sample session. This must be done on a sample session by sample session basis.

11. The exit is passed a pointer to a 4096-byte context area. This area is for the exclusive use of the exit. The Sampling Engine initializes this area to nulls once, prior to the First call to the exit. This exit can use this area for any purpose, which can include:
    - A portion for the exit's dynamic storage area.
    - A portion as a context area to keep information it might need across function calls.
    - A portion can optionally be used to communicate information back to the Sampling Engine e.g. USS pathname data.

Data Extractor Module Resolution Program Characteristics

The exit's load module name is CAZDEXMR. It is called with the following characteristics:

Authorization
   Supervisor state, key 0

Cross memory mode
> PASN = SASN = HASN

AMODE
> 31-bit

ASC mode
> Primary

Locks Local lock may be held. Refer to the **DEXMRI_Flags** parameter in "Input Parameter Fields" on page 110 for more information.

Reentrant
> Yes

Residency
> ECSA

Input Register contents:

R1 Address of a standard CALL parameter list.

> Word 1: Contains the address of the CAZDXMRP parameter block. The VL bit is on.

R13 Address of a standard register 72-byte save area

R14 Return address

R15 Entry point address of CAZDEXMR

Upon entry, all access registers contain zeros.

Output Register contents:

R0 – R14
> Same as upon entry

R15 The value set in: DEXMRO_RC

When control returns to the caller, the access registers must contain zeros.

The load module must be accessible from the Application Performance Analyzer started task. It is recommend that it reside in the Application Performance Analyzer *hlq*.SCAZAUTH load library.

The source for a sample CAZDEXMR exit program is provided in the *hlq*.SCAZSAMP dataset in member CAZDEXMR. The JCL that is used to assemble, link and apply the exit as an SMP USERMOD is also located in the *hlq*.SCAZSAMP dataset in member CAZDXMRA.

Data Extractor Module Resolution Parameter Block

All input and output parameters reside in a single parameter block. The parameter block is mapped by the macro: CAZDXMRP, which resides in *hlq*.SCAZSAMP. The DSECT name is CAZDXMRP. All input parameter field names are prefixed with DEXMRI_ and all output parameter field names are prefixed with DEXMRO_.

The following table describes the input fields:

Table 7. Data Extractor Module Resolution Parameter Block Input Fields

Offset	Name	Type	Length	Value	Description
0	DEXMRI_Input	Word	0		Start of input data
0	DEXMRI_Eye	Char	8	DEXMRPB_	Eye catcher
8	DEXMRI_Functn	Byte	1		Function code
	DEXMRI_FRSV	EQU	0	0	Reserved
	DEXMRI_FFirst	EQU	0	4	Initialization or First function call
	DEXMRI_FMR	EQU	0	8	Module Resolution Function call
	DEXMRI_FLast	EQU	0	12	Termination or Last function call
9	DEXMRI_Versn	Byte	1		Parameter block version
	DEXMRI_CVersn	EQU	0	1	Current version
A	DEXMRI_PLen	HWord	2	352	Length of the entire parameter block
C	DEXMRI_Flags	Byte	1		Flag byte
	DEXMRI_FLLock	EQU	0	x'80'	Local Lock is held
C		Byte	1	Nulls	Reserved
E	DEXMRI_ContxtL	HWord	2	4096	Length of exit's context area
10	DEXMRI_Contxt@	Word	4		Address of exit's context area
14	DEXMRI_StepPGM	Byte	8		Job Step program name
1C		Word	4	Nulls	Reserved
20	DEXMRI_Address	Word	4		Sampled address (from PSW)
24	DEXMRI_TCB@	Word	4		Address of sampled TCB
28	DEMRI_ASID	HWord	2		ASID of sampled address space
2A	DEMRI_SASID	HWord	2		ASID where address was sampled
2C		Byte	48	Nulls	Reserved
5C					

The following table will describe the output fields which follow immediately after the input fields:

Table 8. Data Extractor Module Resolution Parameter Block Output Fields

Offset	Name	Type	Length	Value	Description
5C	DEXMRO_Output	Word	0		Start of output data
5C	DEXMRO_RC	Byte	1		Return code
	DEXMRO_RCOK	EQU	0	0	Success
	DEXMRO_RCNF	EQU	0	4	Module not found
	DEXMRO_RCErr	EQU	0	8	Error

Offset	Name	Type	Length	Value	Description
	DEXMRO_RCSevr EQU	EQU	0	16	Severe error
	DEXMRO_RCNCall	EQU	0	20	Do not call this exit again
5D	DEXMRO_RSN	Byte	1		Reason code
	DEXMRO_RSOK	EQU	0	0	Success
	DEXMRO_RSNI	EQU	0	4	Exit not interested in Job Step
	DEXMRO_RSVersn	EQU	0	8	Parameter block version mismatch
	DEXMRO_RSEye	EQU	0	12	Parameter block eye catcher mismatch
	DEXMRO_RSCtxSz	EQU	0	16	Context area too small
	DEXMRO_RSBPSz	EQU	0	20	Parameter block size not equal
	DEXMRO_RSFunc	EQU	0	24	Unknown function code
	DEXMRO_RSExit	EQU	0	28	Error in exit (see DEXMRO_Msg)
5E		Byte	1	Nulls	Reserved
Resolved Module Information					
5F	DEXMRO_Flag1	Byte	1		Flag byte
	DEXMRO_F1_MOD	EQU	0	x'80'	Module data is an MVS module
	DEXMRO_F1_PTH	EQU	0	x'40'	Module data is a USS path name
60	DEXMRO_ModData	Byte	0		Start of module data
60	DEXMRO_ModName	Byte	8		Module name
68	DEXMRO_ModDDN	Byte	8		DDName of loadlib (or nulls)
70	DEXMRO_ModDSN	Byte	44		DSName of loadlib (or nulls)
60	DEXMRO_ModPTH@	Byte	4		Address of USS path name length prefix
9C		Word	4	Nulls	Reserved
A0	DEXMRO_MStart@	Word	4		Starting address of module
A4	DEXMRO_MLen	Word	4		Length of module
A8	DEXMRO_UTkn	Byte	8		Module uniqueness token
B0		Byte	48	Nulls	Reserved
E0	DEXMRO_MsgL	Byte	1		Length of error message
E1	DEXMRO_Msg	Byte	120		Error message
160					

Input Parameter Fields

The meaning and use of each of the input parameter fields are described below. The Sampling Engine places data into these fields used by the exit to complete the function call.

DEXMRI_Eye

This field contains an eye catcher and identifies this entire parameter block. It contains 'DEXMRPB _'. The exit can use this field to verify the parameter list. It can also be used to find the parameter block in a dump.

DEXMRI_Functn

This field contains the code of the function to be performed by the exit. Refer to item 5 in "Data Extractor Module Resolution Program Design Considerations" on page 105 for additional information. There are three function codes:

1. DEXMRI_FFirst - indicates that this is a First time or initialization call. The exit is called with this function code once at the start of the sampling session.
2. DEXMRI_FMR - indicates that this is a Module Resolution call. The exit is called once per unresolved address.
3. DEXMRI_FLast- indicates that this is Last or termination call. The exit is called with this function code once at the end of the sampling session. The exit should use this opportunity to clean up its environment.

DEXMRI_Versn

This field contains the version of the entire parameter block. The exit can use this value to ensure that parameter block it expects is at the correct version. This protects the exit from changes to the parameter block.

DEXMRI_PLen

This field contains the length of the entire parameter block. The exit can use this value to ensure that parameter block it expects is at the correct version. This protects the exit from changes to the parameter block.

DEXMRI_Flags

This field contains some flag bits.

1. DEXMRI_FLLock – Indicates that the exit is called with the local lock held.

DEXMRI_ContxtL

This field contains the length of the context area provided by Application Performance Analyzer for the exclusive use of the exit.

DEXMRI_Contxt@

This field contains the address of the context area.

DEXMRI_StepPGM

This field contains the Job Step program name. The exit can use this field to determine if it is interested in monitoring this Job Step. If not, the exit can return with a return code of DEXMRO_RCNCall, indicating to the Sampling Engine to not call the exit again until the Last (or termination) call.

DEXMRI_Address

This field contains the sampled PSW (instruction) address. It is the address that the Sampling Engine requires the exit to resolve to a load module or USS exec.

DEXMRI_TCB@
This field contains the address of the TCB under which the sampled PSW address was found.

DEXMRI_ASID
This field contains the address space ID of the sampled address space.

DEXMRI_SASID
This field contains the address space ID of the address space where the PSW address was sampled. Typically, this is the same as DEXMRI_ASID. The exit can decide to compare the two ASIDs to determine if it is interested in resolving the address.

Output Parameter Fields

The meaning and use of each of the output parameter fields are described below. The exit places data into these fields in response to the function call.

DEXMRO_RC
This field contains the return code for this function call. The Sampling Engine reports it and the reason code in a message if this value is eight or higher. This message is displayed with a message ID of: CAZ4880E in the REQ#nnnn SYSOUT dataset of the Application Performance Analyzer Started Task. *nnnn* is the observation request number associated with the error.

A return code of DEXMRO_RCSevr or higher causes the Sampling Engine to stop calling the exit for Module Resolution functions. The Sampling Engine only calls the exit again at the end of sampling to perform the Last function call.

DEXMRO_RSN
This field contains the reason code for this function call. The reason code provides details on the cause of the error.

DEXMRO_Flag1
This field contains some flag bits.
- DEXMRO_F1_MOD indicates that the module information returned is an MVS module.
- DEXMRO_F1_PTH indicate that the module information returned is a USS exec.

DEXMRO_ModData
This label denotes the start of an overlay point. The data at this location either contains MVS module information or USS exec information. Refer to the **DEXMRO_Flag1** in "Output Parameter Fields" for more information.

If bit DEXMRO_F1_MOD is on, then DEXMRO_ModName, DEXMRO_ModDDN, and DEXMRO_ModDSN contain MVS module information. If bit DEXMRO_F1_PTH is on, then DEXMRO_ModPTH@ points to the address of a pathname length prefix.

DEXMRO_ModName
This field contains the resolved MVS module name in which the instruction at PSW address in DEXMRI_Address was executing.

DEXMRO_ModDDN
This field can contain nulls or the DDName of the load library from which the MVS module was loaded.

DEXMRO_ModDSN

This field can contain nulls or the DSName of the load library from which the MVS module was loaded. This field is required if the module's CSECT information is to be resolved by Application Performance Analyzer.

DEXMRO_ModPTH@

This field can contain nulls or a pointer to the length prefix of the full pathname of the path from which the USS exec was executed. The length is that of the name only.

DEXMRO_MStart@

This field contains the starting address (load point address) of either the MVS module or USS exec described by the data in DEXMRO_ModData.

DEXMRO_MLen

This field contains the length of either the MVS module or USS exec described by the data in DEXMRO_ModData.

DEXMRO_UTkn

This field contains nulls or an 8-byte token to uniquely identify this instance of an MVS module or USS exec. This token provides an indication of whether or not the module or exec has been loaded and deleted multiple times such that the starting and ending addresses have changed.

DEXMRO_MsgL

This field contains nulls or the length of a message provided in DEXMRO_Msg. This field is ignored for all return codes less than or equal to four.

DEXMRO_Msg

This field contains nulls or the message text. This field is ignored for all return codes less than or equal to four. The message is displayed with a message ID of: CAZ4881I in the REQ#nnnn SYSOUT dataset of the Application Performance Analyzer Started Task. *nnnn* is the observation request number associated with the error.

Appendix E. Implement Secure Socket Layer (SSL) communication in Application Performance Analyzer

This appendix describes the basic tasks required to implement SSL communication between Application Performance Analyzer for z/OS and the Application Performance Analyzer GUI remote clients, through the Application Performance Analyzer Listener. Simplified **examples** are provided as a guide and are not meant to be explicit instructions, unless otherwise noted. The examples use RACF® to define authorization and manage digital certificates and the key ring. If you are using a system other than RACF®, refer to that system's documentation for instructions. For detailed information on advanced AT-TLS configuration options refer to the documents listed in the "Bibliography" on page 131. It is strongly recommended that you discuss your configuration requirements with the system programmers responsible for TCP/IP and security. The following tasks are described:

- Enable AT-TLS for the TCP/IP stack
- Configure the Policy Agent (PAGENT) started task
- Define the AT-TLS policy rules
- Define the PAGENT authorization
- Define the Application Performance Analyzer Listener Authorization
- Generate the Digital Certificates
- Create the Key Ring
- Export the CA Certificate
- Install the CA Certificate on the Remote Client
- Enforce SSL communication in the Application Performance Analyzer GUI

All information in this document has been derived from the documents listed under AT-TLS in the "Bibliography" on page 131.

Enable AT-TLS for the TCP/IP Stack

Note: If AT-TLS is already running and active on the TCP/IP stack this task is bypassed.

TCP/IP AT-TLS support is controlled by the TTLS or NOTTLS parameter of the TCPCONFIG statement. AT-TLS is enabled by specifying TTLS. By default, the TCPCONFIG statement is found in the PROFILE.TCPIP data set. AT-TLS can also be enabled without modifying the PROFILE.TCPIP and stopping and starting the TCP/IP stack. To do this, define a separate file that contains the TCPCONFIG TTLS statement and issue the VARY TCPIP OBEYFILE command, as illustrated in the following example:

- Create an OBEYFILE member called: TTLSON in SYS1.TCPPARMS. The member will contain the following line: TCPCONFIG TTLS
- On the MVS console, issue the following command:

```
VARY TCPIP,,O,DSN=SYS1.TCPPARMS(TTLSON)
```

Configure the Policy Agent (PAGENT) Started Task

Note: If a Policy Agent started task is already configured on your system this task is bypassed.

Policy Agent started task

The Policy Agent (PAGENT) started task communicates with the TCP/IP started task. It provides TCP/IP with the AT-TLS rules. You can find a sample JCL procedure for the Policy Agent in: TCPIP.SEZAINST(PAGENT).

An example PAGENT JCL procedure is also provided here:

```
//PAGENT    PROC
//PAGENT    EXEC PGM=PAGENT,REGION=0K,TIME=NOLIMIT,
//        PARM='ENVAR("_CEE_ENVFILE=DD:STDENV")/'
//*
//STDENV   DD DSN=SYS1.TCPENV(PAGENV),DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSOUT   DD SYSOUT=*
//CEEDUMP  DD SYSOUT=*,DCB=(RECFM=FB,LRECL=132,BLKSIZE=132)
```

The Policy Agent started task runs as a USS process and requires input environment variables to define its environment. These environment variables may be passed via the EXEC PARM operand, an HFS file, or an MVS data set. Refer to the comments in the sample procedure TCPIP.SEZAINST(PAGENT) for more information. In the above example an MVS data set is used for the environment variables. The MVS data set must be defined as a variable blocked PDS, and called SYS1.TCPENV. Member PAGENV contains the environment variables.

Policy Agent Configuration Files

PAGENT Environment Variables

The example PAGENT JCL procedure above allocated SYS1.TCPENV(PAGENV) to DDName: STDENV. The member contains the following assignments:

1. TZ=*EST5EDT4*
2. PAGENT_CONFIG_FILE=//'*SYS1.TCPPARMS(PAGENT)*'
3. PAGENT_LOG_FILE=*SYSLOGD*

Where:

- Statement #1: *TZ* is the time zone variable
- Statement #2: *PAGENT_CONFIG_FILE* specifies the HFS file or MVS data set containing the Policy Agent's configuration parameters.
- Statement #3: *PAGENT_LOG_FILE* specifies the destination where the Policy Agent will log diagnostic information.

PAGENT Configuration Statements

The contents of the configuration data set:

'SYS1.TCPPARMS(PAGENT)' are itemized below. Comment lines are denoted by a '#' in column 1.

1. # TCPIP Policy Agent Configuration
2. # LogLevel statement
3. # ... SYSERR, OBJERR, PROTERR, and WARNING messages are logged.
4. LogLevel 15
5. TcpImage *TCPIP FLUSH PURGE*
6. TTLSConfig //'*SYS1.TCPPARMS(TCPPOLS)' FLUSH PURGE*

Where:
- Statement #4: *Loglevel* specifies the level of tracing for the Policy Agent.
- Statement #5: *TcpImage* specifies the job name of the TCP/IP stack that this Policy Agent will control.
- Statement #6: *TTLSConfig* specifies the location of the HFS file or MVS data set containing the policy rules.

Define the AT-TLS Policy Rules

This task defines the policy rules required to implement SSL communication in Application Performance Analyzer.

The policy rules required in: 'SYS1.TCPPARMS(TCPOLS)' are itemized below.

1. TTLSRule *APAListener*
2. {
3.   LocalPortRange *8100 8199*
4.   JobName *LI**
5.   Direction **Inbound**
6.   TTLSGroupActionRef *APAListASecureGrpAct*
7.   TTLSEnvironmentActionRef *APAListASecureEnvAct*
8. }
9. TTLSGroupAction *APAListASecureGrpAct*
10. {
11.   GroupUserInstance 1
12.   TTLSEnabled **On**
13.   Trace 3
14. }
15. TTLSEnvironmentAction *APAListASecureEnvAct*
16. {
17.   EnvironmentUserInstance 1
18.   HandShakeRole **Server**
19.   TTLSKeyRingParms
20.   {
21.   Keyring *APAListAKeyRing*
22.   }
23. }

Where
- Statement #1: TTLSRule defines the AT-TLS rule paragraph. In this example, the rule paragraph is named *APAListener*. Select a paragraph name appropriate for your installation.
- Statement #3: LocalPortRange specifies a local port or range of ports, for the inbound connections that this rule will act on. In this example all clients connecting using ports *8100* to *8199* inclusive will be affected by this rule and will communicate using SSL. Select a port range appropriate for your installation.
- Statement #4: JobName specifies the name or name prefix of the Application Performance Analyzer Listener started task or tasks that this rule will act on.

They are the rule owners. In this example, *LI** is the prefix for the Application Performance Analyzer Listener started task. Select a job name appropriate for your installation.

- Statement #5: Direction specifies the direction the connection must be initiated from. The direction of the Application Performance Analyzer connection must be 'Inbound', indicating it is initiated by the client.
- Statement #6: TTLSGroupActionRef specifies the name of paragraph that will define the TTLS group actions for this rule. In this example, the paragraph is named *APAListASecureGrpAct*. Select a paragraph name appropriate for your installation.
- Statement #7: TTLSEnvironmentActionRef specifies the name of the paragraph that will define the TTLS environment actions for this rule. In this example, the paragraph is named *APAListASecureEnvAct*. Select a paragraph name appropriate for your installation.
- Statement #9: TTLSGroupAction defines the group action paragraph for the rule. This must specify the paragraph name used in Statement #6.
- Statement #11: GroupUserInstance specifies an instance identifier for this GroupAction paragraph. The value must be incremented each time you change this paragraph and wish to refresh the rule using the REFRESH MODIFY command for the PAGENT started task.
- Statement #12: TTLSEnabled specifies whether AT-TLS is active. To enable SSL communication in Application Performance Analyzer this value must be 'On'.
- Statement #13: Trace specifies the level of AT-TLS tracing to be used for the generated AT-TLS policy. Select a trace level appropriate for your installation.
- Statement #15: TTLSEnvironmentAction defines the environment action paragraph for this rule. This must specify the paragraph name used in Statement #7.
- Statement #17: EnvironmentUserInstance specifies an instance identifier for this EnvironmentAction paragraph. The value must be incremented each time you change this paragraph and wish to refresh the rule using the REFRESH MODIFY command for the PAGENT started task.
- Statement #18: HandShakeRole specifies whether the client or server performs the SSL handshake. In Application Performance Analyzer the value must be 'Server', indicating the server always performs the SSL handshake.
- Statement #19: TTLSKeyRingParms specifies an inline paragraph for key ring parameters. This is a required parameter.
- Statement #21: KeyRing specifies the name of the key ring used to store the digital certificates. This must match the name of the key ring created for Application Performance Analyzer. Refer to the later section "Create the Key Ring" on page 119 for more information.

Define the PAGENT authorization

Note: If the Policy Agent started task is already defined to RACF and a RACF user ID with z/SO UNIX superuser authority (UID=0) is associated with it, this task is bypassed.

The following example JCL defines the PAGENT started task to RACF and associates a RACF user ID to it. The RACF user ID must have z/OS UNIX superuser authority (UID-0).

```
//* RACF commands to define the PAGENT started task
//*
//DAEMONS EXEC PGM=IKJEFT01
//SYSTSPRT DD SYSOUT=*
//SYSTSIN DD *
SETROPTS CLASSACT(STARTED)
SETROPTS RACLIST(STARTED)
SETROPTS GENERIC(STARTED)
RDEFINE STARTED PAGENT.*
SETROPTS RACLIST(STARTED) REFRESH
SETROPTS GENERIC(STARTED) REFRESH
/*
//* RACF command to define a user ID for the PAGENT started tasks
//*
/PAUSER EXEC PGM=IKJEFT01
//SYSTSPRT DD SYSOUT=*
//SYSTSIN DD *
ADDUSER PAGENTU DFLTGRP(OMVSGRP) OMVS(UID(0) HOME('/'))
/*
//* RACF command to associate user ID with PAGENT started task
//*
//ASCPAUSR EXEC PGM=IKJEFT01
//SYSTSPRT DD SYSOUT=*
//SYSTSIN DD *
RALTER STARTED PAGENT.*  STDATA(USER(PAGENTU))
```

Define the Application Performance Analyzer Listener authorization

This task defines the authorization required by the Application Performance
Analyzer Listener to invoke RACF digital certificate commands (RACDCERT) to
manage digital certificates.

In the previous section 'Define the AT-TLS Policy Rules' an AT-TLS rule called
APAListener was defined. It specified the owner of the rule to be any started task
whose job name begins with "*LI*" (that is, the Application Performance Analyzer
Listener started task). The user ID associated with the Application Performance
Analyzer Listener started task is used by AT-TLS to invoke RACDCERT commands
therefore it must be granted authority to the IRR.DIGTCERT.function resource in
the FACILITY class.

In the example below, *START2* is the user ID that has been associated with the
Application Performance Analyzer Listener started task during the customization
of the Application Performance Analyzer Listener. You must replace *START2* in the
following example with the Application Performance Analyzer Listener user ID
defined at your installation.

```
//RACDCERT EXEC PGM=IKJEFT01
//SYSTSPRT DD SYSOUT=*
//SYSUADS  DD DSN=SYS1.UADS,DISP=SHR
//SYSLBC   DD DSN=SYS1.BRODCAST,DISP=SHR
//SYSTSIN  DD *
 SETROPTS CLASSACT(DIGTCERT DIGTRING DIGTNMAP)
 RDEFINE FACILITY IRR.DIGTCERT.LIST     UACC(NONE)
 RDEFINE FACILITY IRR.DIGTCERT.LISTRING UACC(NONE)
 RDEFINE FACILITY IRR.DIGTCERT.GENCERT  UACC(NONE)
 PERMIT IRR.DIGTCERT.LIST     CLASS(FACILITY) ID(START2) ACCESS(CONTROL)
 PERMIT IRR.DIGTCERT.LISTRING CLASS(FACILITY) ID(START2) ACCESS(CONTROL)
 PERMIT IRR.DIGTCERT.GENCERT  CLASS(FACILITY) ID(START2) ACCESS(CONTROL)
 SETROPTS RACLIST (DIGTCERT) REFRESH
 SETROPTS RACLIST (DIGTRING) REFRESH
 SETROPTS RACLIST (DIGTNMAP) REFRESH
 SETROPTS RACLIST (FACILITY) REFRESH
//
```

Generate the Digital Certificates

Note: If an appropriate CA certificate and site certificate already exist at your installation, this task is bypassed.

For SSL communications, Application Performance Analyzer requires a Certificate Authority (CA) certificate and a Site certificate signed by the CA certificate. The CA certificate is later exported and delivered to every Application Performance Analyzer GUI user for installation on their personal computer.

Generate a Certificate Authority (CA) certificate

The example below will generate a Certificate Authority (CA) certificate. Replace the values shown in italics with values appropriate for your installation.

```
//RACDCERT EXEC PGM=IKJEFT01
//SYSTSPRT DD SYSOUT=*
//SYSUADS  DD DSN=SYS1.UADS,DISP=SHR
//SYSLBC   DD DSN=SYS1.BRODCAST,DISP=SHR
//SYSTSIN  DD *
RACDCERT GENCERT -
   CERTAUTH -
   SUBJECTSDN(OU('YourOrganizationalUnit') -
   O('YourOrganization') -
   L('YourLocality') -
   SP('YourState/Province') -
   C('YourCountry')) -
   NOTAFTER(DATE(yyyy-mm-dd)) -
   WITHLABEL('CA-label-name') -
   KEYUSAGE(CERTSIGN)
```

Generate a Site certificate

The example below will generate a site certificate that is signed by the CA certificate created above.

In this example, *START2* is the user ID that has been associated with the Application Performance Listener started task during the customization of the Application Performance Analyzer Listener. Replace *START2* in the following example with the Application Performance Analyzer Listener user ID defined at your installation.

Replace CA-*label-name* in the following example with the label assigned to the CA certificate created above.

Replace other values shown in italics with values appropriate for your installation.

```
//RACDCERT EXEC PGM=IKJEFT01
//SYSTSPRT DD SYSOUT=*
//SYSUADS  DD DSN=SYS1.UADS,DISP=SHR
//SYSLBC   DD DSN=SYS1.BRODCAST,DISP=SHR
//SYSTSIN  DD *
RACDCERT GENCERT -
   ID(START2) -
   SUBJECTSDN(CN('Your Common Name') -
   OU('Your Organizational Unit'') -
   O('YourOrganization') -
   C('YourCountry')) -
   NOTAFTER(DATE(yyyy-mm-dd)) -
   WITHLABEL('Site-label-name') -
   SIGNWITH(CERTAUTH LABEL('CA-label-name'))
```

Create the Key Ring

This task creates a key ring to store the digital certificates generated above.

The name of the key ring must match the name specified in the KeyRing parameter of the TTLSEnvironmentAction statement within the AT-TLS policy rule defined for the Application Performance Analyzer Listener. Refer back to the section 'Define the AT-TLS Policy Rules' for the name assigned to KeyRing. The example below creates the key ring named APAListAKeyRing and stores the digital certificates in this key ring. Replace APAListAKeyRing with the name specified in the KeyRing parameter of the TTLSEnvironmentAction statement.

The user ID associated with the Application Performance Analyzer Listener started task must be assigned ownership of the key ring. In this example, *START2* is the user ID that has been associated with the Application Performance Listener started task during the customization of the Application Performance Analyzer Listener. Replace *START2* in the following example with the Application Performance Analyzer Listener user ID defined at your installation.

Replace other values shown in *italics* with values appropriate for your installation.

```
//RACDCERT EXEC PGM=IKJEFT01
//SYSTSPRT DD SYSOUT=*
//SYSUADS  DD DSN=SYS1.UADS,DISP=SHR
//SYSLBC   DD DSN=SYS1.BRODCAST,DISP=SHR
//SYSTSIN  DD *
RACDCERT ADDRING(APAListAKeyRing) ID(START2)
RACDCERT CONNECT(CERTAUTH LABEL('CA-label-name') -
   RING(APAListAKeyRing)) ID(START2)
RACDCERT CONNECT(ID(START2) LABEL('Site-label-name') -
   RING(APAListAKeyRing) DEFAULT) ID(START2)
```

Export the CA Certificate

Note: If an appropriate exported CA certificate already exists at your installation, this task is bypassed.

This task exports the CA certificate created earlier from RACF and saves it to an MVS data set. Remote clients (users of the Application Performance Analyzer GUI) require this certificate to establish a secure connection with the Application Performance Analyzer for z/OS through the Application Performance Analyzer Listener.

The following example will export the CA certificate created earlier, and save it to an MVS data set. Replace CA-label-name with the label assigned to the CA certificate created earlier. Replace other values shown in italics with values appropriate for your installation.

```
//RACDCERT EXEC PGM=IKJEFT01
//SYSTSPRT DD SYSOUT=*
//SYSUADS  DD DSN=SYS1.UADS,DISP=SHR
//SYSLBC   DD DSN=SYS1.BRODCAST,DISP=SHR
//SYSTSIN  DD *
RACDCERT EXPORT(LABEL('CA-label-name')) -
   CERTAUTH -
   DSN('output-data-set-name') FORMAT(CERTB64)
/*
//
```

Install the CA Certificate on the Remote Client

This task installs the CA certificate exported above into the keystore file on the Application Performance Analyzer GUI user's personal computer. Every organization will have their own standards, therefore detailed instructions are not provided here. Follow your installation's key management policies and procedures to transfer and import the CA certificate into your local keystore file.

Note: When transferring the exported CA certificate from the MVS data set to the remote platform, do not use Binary transfer mode. If using FTP use the GET command with the ASCII option.

Enforce SSL communication in the Application Performance Analyzer GUI

This task enforces SSL communication between the Application Performance Analyzer GUI and Application Performance Analyzer for z/OS through the Application Performance Analyzer Listener.

The Application Performance Analyzer Listener must be configured with a unique secure port in the ListenTo and/or PORT settings of the CONFIG BASIC statement described in Chapter 7. See "CONFIG BASIC statement" on page 82 for more details.

Every Application Performance Analyzer GUI user must set up their Network Connection preferences to enforce SSL communication. The host port must be the unique secure port that is configured in the Application Performance Analyzer Listener. The SSL Connection checkbox must be checked, the name of the local keystore file must be selected and if applicable, the keystore passphrase must be entered. The Get Keystore button can be used to locate the local keystore file where the CA certificate has been saved. Once these values are entered, the Application Performance Analyzer GUI must be restarted for the changes to take effect. Refer to Chapter 14 in the Application Performance Analyzer *User's Guide* for additional details on the Network Preferences window.

Appendix F. Applying service APAR or PTF

This section describes how to apply updates to IBM Application Performance
Analyzer. To use the maintenance procedures effectively, you must install IBM
Application Performance Analyzer with SMP/E before doing the maintenance
procedures below.

What you receive

If you report a problem with IBM Application Performance Analyzer to your IBM
Support Center, you might receive a tape containing one or more Authorized
Program Analysis Reports (APARs) or Program Temporary Fixes (PTFs) that were
created to solve your problem.

You might also receive a list of prerequisite APARs or PTFs, which you must apply
to your system before applying the current APAR. These prerequisite APARs or
PTFs might relate to IBM Application Performance Analyzer or any other licensed
product you have installed, including z/OS.

Checklist for applying an APAR or PTF

The following checklist describes the steps and associated SMP/E commands to
install the APAR or PTF:

__ 1. Prepare to install the APAR or PTF

__ 2. Receive the APAR or PTF (SMP/E RECEIVE)

__ 3. Review the HOLDDATA

__ 4. Accept previously applied APARs or PTFs (optional) (SMP/E ACCEPT)

__ 5. Apply APAR or PTF (SMP/E APPLY)

__ 6. Test APAR or PTF

__ 7. Accept APAR or PTF (SMP/E ACCEPT)

Step 1. Prepare to install APAR or PTF

Before you start to install an APAR or PTF, do the following steps:

1. Create a backup copy of the current IBM Application Performance Analyzer
   libraries. Save this copy of IBM Application Performance Analyzer until you
   have completed installing the APAR or PTF and you are confident that the
   service runs correctly.

2. Research each service tape through the IBM Support Center for any errors or
   additional information. Note all errors on the tape that were reported by
   APARs or PTFs and apply the relevant fixes. You should also review the
   current Preventive Service Planning (PSP) information.

Step 2. Receive the APAR or PTF

Receive the service using the SMP/E RECEIVE command from either the SMP/E
dialogs in ISPF or by using a batch job similar to CAZWRECV in *hlq*.SCAZSAMP.

Step 3. Review the HOLDDATA

Review the HOLDDATA summary reports for the APAR or PTF. Follow any
instructions described in the summary reports.

Step 4. Accept a previously applied APAR or PTF (optional)

If there is any APAR or PTF that you applied earlier but did not accept, and the earlier APAR or PTF is not causing problems in your installation, accept the applied service from either the SMP/E dialogs in ISPF or by using a batch job.

Accepting the earlier service allows you to use the SMP/E RESTORE command to return to your current level if you encounter a problem with the service that you are currently applying. You can do this either from the SMP/E dialogs in ISPF or by using a batch job similar to CAZWACPT in *hlq*.SCAZSAMP.

Step 5. Apply the APAR or PTF

We recommend that you first use the SMP/E APPLY command with the CHECK operand. Check the output; if it shows no conflict, rerun the APPLY command without the CHECK operand. This can be done from the SMP/E dialogs in ISPF or by using a batch job similar to CAZWAPLY in *hlq*.SCAZSAMP.

Step 6. Test the APAR or PTF

Thoroughly test your updated IBM Application Performance Analyzer. Do not accept an APAR or PTF until you are confident that it runs correctly.

Step 7. Accept the APAR or PTF

We recommend that you first use the SMP/E ACCEPT command with the CHECK operand. Check the output; if it shows no conflict, rerun the ACCEPT command without the CHECK operand. You can do this either from the SMP/E dialogs in ISPF or by using a batch job similar to CAZWACPT in *hlq*.SCAZSAMP.

Appendix G. Accessibility

Accessibility features help a user who has a physical disability, such as restricted mobility or limited vision, to use software products successfully. The accessibility features in z/OS provide accessibility for IBM Application Performance Analyzer.

The major accessibility features in z/OS enable users to:

- Use assistive technology products such as screen readers and screen magnifier software
- Operate specific or equivalent features by using only the keyboard
- Customize display attributes such as color, contrast, and font size

Using assistive technologies

Assistive technology products work with the user interfaces that are found in z/OS. For specific guidance information, consult the documentation for the assistive technology product that you use to access z/OS interfaces.

Keyboard navigation of the user interface

Users can access z/OS user interfaces by using TSO/E or ISPF. Refer to the following publications:

- *z/OS TSO/E Primer*
- *z/OS TSO/E User's Guide*
- *z/OS ISPF User's Guide Volume 1*

These guides describe how to use TSO/E and ISPF, including the use of keyboard shortcuts or function keys (PF keys). Each guide includes the default settings for the PF keys and explains how to modify their functions.

Accessibility of this document

The XHTML format of this document that will be provided in the IBM Problem Determination Tools information center at http://publib.boulder.ibm.com/infocenter/pdthelp/index.jsp is accessible to visually impaired individuals who use a screen reader.

To enable your screen reader to accurately read syntax diagrams, source code examples, and text that contains the period or comma picture symbols, you must set the screen reader to speak all punctuation.

When you use JAWS for Windows, the links to accessible syntax diagrams might not work. Use IBM Home Page Reader to read the accessible syntax diagrams.

Appendix H. Notices

References in this publication to IBM products, programs, or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Subject to IBM's valid intellectual property or other legally protectable rights, any functionally equivalent product, program, or service may be used instead of the IBM product, program, or service. The evaluation and verification of operation in conjunction with other products, except those expressly designated by IBM, are the responsibility of the user.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

> IBM Director of Licensing
> IBM Corporation
> 500 Columbus Avenue
> Thornwood, NY 10594
> U.S.A.

Licensees of this program who want to have information about it for the purpose of enabling (1) the exchange of information between independently created programs and other programs (including this one) and (2) the mutual use of the information that has been exchanged, should contact:

> IBM Corporation, Department HHX/H3
> 555 Bailey Avenue
> San Jose, CA 95141-1099
> U.S.A.

Such information might be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

Trademarks

IBM, the IBM logo, and ibm.com® are trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at www.ibm.com/legal/copytrade.shtml.

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

Java and all Java-based trademarks and logos are trademarks of Oracle and/or its affiliates.

LINUX is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Glossary

A

abend Abnormal end of a task. The termination of a task before its completion because of an error condition that cannot be resolved by recovery facilities while the task is executing.

access plan

The set of access paths that are selected by the optimizer to evaluate a particular SQL statement.

APF Authorized program facility. A facility that permits the identification of programs that are authorized to use restricted functions.

authorized program

A system program or user program that is allowed to use restricted functions.

B

batch A processing mode in which a predefined series of actions are performed with little or no interaction between the user and the system.

bind To convert the output from the SQL compiler to a usable control structure.

C

checkpoint

A place in a program at which a check is made, or at which data is recorded, to provide real-time monitoring.

checkpoint data set

A data set that contains checkpoint records.

connection authorization exit

An exit that approves or disapproves requests for a connection to another program.

D

data set

The major unit of data storage and retrieval, consisting of a collection of data in one of several prescribed arrangements and described by control information to which the system has access.

dynamic

Pertaining to events that occur at run time or during processing.

E

explain

To capture detailed information about the access plan that was chosen by the SQL compiler to resolve an SQL statement.

I

ISPF Interactive system productivity facility. An IBM licensed program that serves as a full-screen editor and dialog manager.

J

JCL Job control language. A control language that is used to identify a job to an operating system and to describe the job's requirements.

M

monitor

To collect data about an application from the running agents that are associated with the application.

P

panel An area of the screen that displays formatted information and that can include entry fields.

Bibliography

IBM Application Performance Analyzer for z/OS documents

Customization Guide, SC14-7546-02

Messages Guide, SC14-7544-02

Program Directory, GI11-9490

User's Guide, SC14-7545-02

Documents for related products

z/OS

ISPF Planning and Customizing, GC34-4814

MVS JCL Reference, SA22-7597

MVS JCL User's Guide, SA22-7598

MVS System Commands, SA22-7627

Security Server RACF Command Language Reference, SA22-7687

Security Server RACF Security Administrator's Guide, SA22-7683

TSO/E Command Reference, SA22-7782

TSO/E Programming Guide, SA22-7788

TSO/E System Programming Command Reference, SA22-7793

TSO/E User's Guide, SA22-7794

z/OS Initialization and Tuning Reference, SA22-7533

OS/390®

ISPF Planning and Customizing, SC28-1298

MVS JCL Reference, GC28-1757

MVS JCL User's Guide, GC28-1758

MVS System Commands, GC28-1781

Security Server (RACF) Command Language Reference, SC28-1919

Security Server (RACF) Security Administrator's Guide, SC28-1915

TSO/E Command Reference, SC28-1969

AT-TLS

z/OS Communications Server: IP Configuration Guide, SC31-8775-15

z/OS Communications Server: IP Configuration Reference, SC31-8776-16

z/OS Security Server RACF Command Language Reference, SA22-7687-13

DB2 9 for z/OS: Configuring SSL for Secure Client-Server Communications, REDP-4630-00

CICS

Application Programming Guide, SC34-5993

Application Programming Primer, SC33-0674

Application Programming Reference, SC34-5994

CICS Supplied Transactions, SC34-5992

CICS Transaction Server for z/OS Release Guide, GC34-6218

DB2 for OS/390 and z/OS

Administration Guide, SC26-9931

Application Programming and SQL Guide, SC26-9933

Command Reference, SC26-9934

Data Sharing: Planning and Administration, SC26-9935

Installation Guide, GC26-9936

Messages and Codes, GC26-9940

Reference for Remote DRDA* Requesters and Servers, SC26-9942

Release Planning Guide, SC26-9943

SQL Reference, SC26-9944

Utility Guide and Reference, SC26-9945

IMS

IMS Version 8 Installation Volume 1: Installation Verification, GC27-1297

IMS Version 7 Installation Volume 1: Installation Verification, GC26-9429

WebSphere MQ

WebSphere MQ for z/OS Concepts and Planning Guide, GC34-6926

WebSphere MQ for z/OS System Administration Guide, GC34-6929

WebSphere MQ for z/OS System Setup Guide, GC34–6927

Index

Readers' Comments — We'd Like to Hear from You

IBM Application Performance Analyzer for z/OS
Customization Guide
Version 12 Release 1

Publication No. SC14-7546-02

We appreciate your comments about this publication. Please comment on specific errors or omissions, accuracy, organization, subject matter, or completeness of this book. The comments you send should pertain to only the information in this manual or product and the way in which the information is presented.

For technical questions and information about products and prices, please contact your IBM branch office, your IBM business partner, or your authorized remarketer.

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you. IBM or any other organizations will only use the personal information that you supply to contact you about the issues that you state on this form.

Comments:

Thank you for your support.

Send your comments to the address on the reverse side of this form.

If you would like a response from IBM, please fill in the following information:

Name _____   Address _____

Company or Organization _____
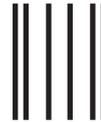
Phone No. _____   Email address _____

IBM®

Fold and Tape | **Please do not staple** | Fold and Tape
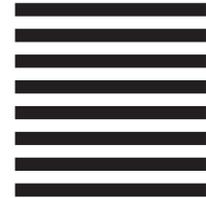
NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST-CLASS MAIL   PERMIT NO. 40   ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

IBM Corporation
H150/090
555 Bailey Avenue
San Jose, CA
USA 95141-9989

Fold and Tape | **Please do not staple** | Fold and Tape

IBM®

Product Number: 5697-Q03

Printed in USA