

IBM Application Performance Analyzer for z/OS



Customization Guide

Version 1 Release 1

IBM Application Performance Analyzer for z/OS



Customization Guide

Version 1 Release 1

Second Edition (July 2005)

Note: Before using this information and the product it supports, be sure to read the general information under Appendix D, "Notices," on page 39.

This edition applies to IBM Application Performance Analyzer for z/OS Version 1 Release 1 (5697-N37) 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: <http://www.ibm.com/software/awdtools/apa/>

Copyright © 2005 BankNet. All rights reserved.

© Copyright International Business Machines Corporation 2005. All rights reserved.

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

About this document	v
Who might use this document	v
Other documentation that you might need	v
Document conventions.	vi
Some basic concepts and terminology.	vi
How to send your comments	vii

Chapter 1. Preparing to customize Application Performance Analyzer **1**

Checklist for installing and customizing Application Performance Analyzer	1
Library names after you finish installing	2
Library names created by Application Performance Analyzer	2
Planning for security	3
Establishing external security system access rules	3
Deciding on the Application Performance Analyzer configuration settings	4
Deciding on the Application Performance Analyzer access rules	4
Assigning priority	4
Using the Sysplex feature	5

Chapter 2. Customizing Application Performance Analyzer. **7**

Adding Application Performance Analyzer module to the LNKLST	7
Enabling the SMF step initiation exit (IEFUSI)	8
Authorizing Application Performance Analyzer modules	9
Modifying the TSO/E logon procedure to include Application Performance Analyzer user interface	9
Choosing an Application Performance Analyzer started task identifier	10
Making the CAZ0 started task available	10
Configuring Application Performance Analyzer	10
Creating the installation-level and started task configuration settings	10
Installation-level configuration settings	11
Started task configuration settings	14
Pre-allocating a checkpoint data set	20
Updating the descriptive program attribution tables	20
DPA table entry elements.	20
CAZDPAGN macro.	21
Example of CAZDPAGN Macros	21
Implementation	22
Modifying Application Performance Analyzer procedure for third party products	22
Starting the CAZ0 started task	23
Starting the IBM Application Performance Analyzer user interface.	23

Chapter 3. Verifying the installation **25**

Beginning the started task	25
Logged messages	25
Starting the ISPF dialog	26
Entering a measurement request	26
Displaying performance analysis reports.	28
Ending the Application Performance Analyzer started task	29

Chapter 4. Customizing Application Performance Analyzer for use with DB2 Universal Database **31**

Deciding which authorization ID to use	31
Binding a DB2 package and plan	31
Providing DB2 authorization	31
Updating the Application Performance Analyzer started task	32
Updating the started task configuration settings	32

Appendix A. Stopping the IBM Application Performance Analyzer started tasks **33**

Appendix B. Applying service APAR or PTF **35**

What you receive	35
Checklist for applying an APAR or PTF	35
Step 1. Prepare to install APAR or PTF	35
Step 2. Receive the APAR or PTF	35
Step 3. Review the HOLDDATA	35
Step 4. Accept a previously applied APAR or PTF (optional)	36
Step 5. Apply the APAR or PTF	36
Step 6. Test the APAR or PTF	36
Step 7. Accept the APAR or PTF	36

Appendix C. Accessibility **37**

Using assistive technologies	37
Keyboard navigation of the user interface	37
Accessibility of this document	37

Appendix D. Notices **39**

Trademarks	39
----------------------	----

Glossary **41**

Bibliography **43**

IBM Application Performance Analyzer for z/OS documents.	43
Documents for related products	43

Index **45**

About this document

This document describes IBM Application Performance Analyzer for z/OS, Version 1 Release 1. The product is hereafter referred to as Application Performance Analyzer.

Application Performance Analyzer is an application program performance measurement tool for use on z/OS systems. This document explains how to configure Application Performance Analyzer with parameters that offer the greatest advantage to your installation for monitoring job performance. Application Performance Analyzer shows total resource utilization within a target address space for items such as a job, started task, or TSO/E session. This document explains how to configure Application Performance Analyzer after it has been installed.

If you need to install Application Performance Analyzer, refer to the *IBM Application Performance Analyzer for z/OS 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 operating systems and subsystems are supported:

- z/OS Version 1 Release 1 (or later) and OS/390[®] Version 2 Release 10
 - CICS[®]
 - IMS[™]
 - DB2 Universal Database^{™™}
 - TSO/E

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 43.

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	<i>italics</i>	<i>xxx</i> OPERAND invalid <i>hlq</i> .SCAZSAMP
JCL code	monospace type	// <i>jobname</i> JOB
JCL lines that need to be altered for your system	Bold	DSN=xxxxxx
High-level qualifiers that you chose	<i>hlq</i>	<i>hlq</i> .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), a Application Performance Analyzer started task needs to be active in each MVS in which measurements are to be performed.

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.

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 LNKLIST.

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

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

Checklist for installing and customizing Application Performance Analyzer

To customize Application Performance Analyzer, you will need to ensure that the appropriate tasks in the following checklist are completed.

- ___ 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 SMF step initiation exit (IEFUSI).
- ___ 5. APF-authorize Application Performance Analyzer modules.
- ___ 6. Modify the TSO/E logon procedure.
- ___ 7. Choose a started task instance identifier.
- ___ 8. Make the CAZ0 started task available.
- ___ 9. Create installation-level settings and settings for the started task.
- ___ 10. Pre-allocate a checkpoint 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 CAZ0 started task.
- ___ 14. Start Application Performance Analyzer.
- ___ 15. Run an IVP to verify the customization of Application Performance Analyzer (optional).

You do not need to customize your DB2 Universal Database to use IBM Application Performance Analyzer. However, if you want Application Performance Analyzer to gather the necessary data to report plan and package BIND timestamps, do the following tasks:

- ___ 1. Create and bind a DB2 UDB access package and plan.
- ___ 2. Provide DB2 authorization.
- ___ 3. Update the started task JCL.
- ___ 4. Update the started task configuration settings.

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
<i>hlq</i> .SCAZLINK	LNKLST module	
<i>hlq</i> .SCAZAUTH	Load library modules	
<i>hlq</i> .SCAZEXEC	REXX exec library	
<i>hlq</i> .SCAZMENU	ISPF messages	
<i>hlq</i> .SCAZPENU	ISPF panels	
<i>hlq</i> .SCAZTENU	ISPF tables	
<i>hlq</i> .SCAZDBRM	Data	
<i>hlq</i> .SCAZSAMP	Sample JCL	

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
<i>yourhlq</i> .CheckpointDSN	Application Performance Analyzer checkpoint data set
<i>yourhlq</i> .LOG.Dyyyyddd.Thhmsst	Optional log data set
<i>yourhlq</i> .userid.Rnnnn.jobname.SF	Measurement data set or sample data set

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.

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.

Planning for security

IBM Application Performance Analyzer runs on a z/OS or OS/390 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 10.

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 20 to see 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.Dyyyyddd.Thhmsst

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.Rnnnn.jobname.SF

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.CURRENT.TEST	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

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 11.

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. For more information, see “Installation-level configuration settings” on page 11.

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. This must be configured 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. This is done 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 observe/sample and report on application programs that execute on an MVS system in a Sysplex, other than the one the user has started the Application Performance Analyzer user interface on. To accommodate this, the Application Performance Analyzer started task will communicate with other instances of the Application Performance Analyzer started task on different MVS systems in the Sysplex.

The communication between Application Performance Analyzer instances is facilitated by the implementation of an Application Performance Analyzer Sysplex group. It is assumed that all images in a Sysplex participate in DASD sharing. To take advantage of the Sysplex support in Application Performance Analyzer, a unique instance of Application Performance Analyzer must be installed on every MVS system in the Sysplex. Each of these instances must be defined to the same Application Performance Analyzer Sysplex group during the install of Application Performance Analyzer. Only one instance of Application Performance Analyzer on the same MVS system can be a member in the same Application Performance Analyzer Sysplex group, that is, you cannot have two or more instances of Application Performance Analyzer on the same MVS system as members of the same Application Performance Analyzer Sysplex group.

See parameter "SPXGroupName" on the CONFIG BASIC statement in the member CAZCNFG0 in *hlq*.SCAZSAMP. This parameter is coded as "SPXGroupName=Sysplexname" and identifies the Application Performance Analyzer Sysplex group that the Application Performance Analyzer instance will be a member of. It can be a maximum of eight alphanumeric characters. All Application Performance Analyzer instances that will communicate with each other must code an identical "SPXGroupName=Sysplexname". The Application Performance Analyzer instances in a Application Performance Analyzer Sysplex group must be from different MVS systems.

All Application Performance Analyzer instances defined in the same Application Performance Analyzer Sysplex group must share a common checkpoint file; therefore, it is important that the HLQ, UNIT, and CheckpointDSN parameters are identical. Failure to do so will cause the started task to fail upon startup with an appropriate error message. It is important that you choose an HLQ which 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. For more information, see "CONFIG BASIC statement" on page 15.

In summary, the CONFIG BASIC statement for every Application Performance Analyzer instance defined in the same Application Performance Analyzer Sysplex group must be coded identically, with the exception of the STCid, which might be unique.

Chapter 2. Customizing Application Performance Analyzer

You need to perform the following tasks to set up the basic product after installing it:

- “Adding Application Performance Analyzer module to the LNKLST” on page 7
- “Enabling the SMF step initiation exit (IEFUSI)” on page 8
- “Authorizing Application Performance Analyzer modules” on page 9
- “Modifying the TSO/E logon procedure to include Application Performance Analyzer user interface” on page 9
- “Choosing an Application Performance Analyzer started task identifier” on page 10
- “Making the CAZ0 started task available” on page 10
- “Creating the installation-level and started task configuration settings” on page 10
- “Pre-allocating a checkpoint data set” on page 20 (optional)
- “Updating the descriptive program attribution tables” on page 20 (optional)
- “Modifying Application Performance Analyzer procedure for third party products” on page 22 (optional)
- “Starting the CAZ0 started task” on page 23
- “Starting the IBM Application Performance Analyzer user interface” on page 23

Adding Application Performance Analyzer module to the LNKLST

The library SCAZLINK contains Application Performance Analyzer’s step initiation exit (IEFUSI) routine, which receives control at job/step initiation. SCAZLINK must be added to the link list for the product to execute successfully. Add SCAZLINK to the LNKLST 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 library is managed by SMS, use:
LNKLST ADD NAME(*lnklst.name*) DSNAME (*hlq*.SCAZLINK) SMS
- If the library is not managed by SMS, use:
LNKLST ADD NAME(*lnklst.name*) DSNAME(*hlq*.SCAZLINK) VOLUME(*parm*)

The variable *parm* is one of the following:

- A six-character identifier for the volume on which *hlq*.SCAZLINK resides
 - ***** , which indicates that *hlq*.SCAZLINK is located on the current SYSRES volume
 - *MCAT* , which indicates that *hlq*.SCAZLINK is located on the volume that contains the master catalog
2. To make the change effective, issue one of the following commands:
 - 'SET PROG=xx' where xx is the suffix of the PROG member
 - 'SETPROG LNKLST,ACTIVATE,NAME=*lnklst.name*'
 3. If your link list data set is managed by LLA, issue a 'F LLA,REFRESH' MVS command to refresh the Library Lookaside Directories.

Enabling the SMF step initiation exit (IEFUSI)

IEFUSI is a System Management Facilities (SMF) exit, and it must be enabled for the types of workloads that Application Performance Analyzer is going to report on. The IEFUSI exit point can be defined to SMF such 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 an SMFPRM xx 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 SMFPRM xx correctly, the CAZ0 started task will inform you of each exit point that Application Performance Analyzer is using. For example, if you set up SMFPRM xx to take SMF exit IEFUSI for TSO users, then you would see a message similar to the following for SYSTSO:

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

Note: Before applying any maintenance or installing a new release of Application Performance Analyzer, make sure that you stop the started task CAZ0.

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 start task CAZ0, 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:

```
SETPROG EXIT,DELETE,EXITNAME=SYS.IEFUSI,MODNAME=CAZ00990      /* for the entire system */  
SETPROG EXIT,DELETE,EXITNAME=SYSJES2.IEFUSI,MODNAME=CAZ00990 /* for JES2 environment */  
SETPROG EXIT,DELETE,EXITNAME=SYSJES3.IEFUSI,MODNAME=CAZ00990 /* for JES3 environment */  
SETPROG EXIT,DELETE,EXITNAME=SYSTSO.IEFUSI,MODNAME=CAZ00990 /* for TSO logons */  
SETPROG EXIT,DELETE,EXITNAME=SYSSTC.IEFUSI,MODNAME=CAZ00990 /* for Started Tasks */
```

Then restart the started task CAZ0, which will load the new Application Performance Analyzer exits.

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

Authorizing Application Performance Analyzer modules

The library *hlq.SCAZAUTH* contains all of the Application Performance Analyzer modules that must be authorized for the product to execute successfully. Add *hlq.SCAZAUTH* 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 library is managed by SMS, use:
APF ADD DSN(hlq.SCAZAUTH) SMS
 - If the library is not managed by SMS, use:
APF ADD DSN(hlq.SCAZAUTH) VOLUME(*parm*)
The variable *parm* is one of the following
 - A six-character identifier for the volume on which *hlq.SCAZAUTH* resides
 - *********, which indicates that *hlq.SCAZAUTH* is located on the current SYSRES volume
 - ***MCAT***, which indicates that *hlq.SCAZAUTH* is 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. 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(CAZ1)' /*Application Performance Analyzer*/ ( NEW )
...

```

For more information about configuring your ISPF Primary Option menu, see *z/OS ISPF Planning and Customizing* or *OS/390 ISPF Planning and Customizing*.

Choosing an Application Performance Analyzer started task identifier

An Application Performance Analyzer started task is identified by a four-character identifier. The IBM-default value shipped with the product is CAZ0. You can choose a different name to meet your site's requirement.

Making the CAZ0 started task available

To make the CAZ0 started task available, do 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.
3. Define a user ID for the CAZ0 started task.
4. Grant the user ID you created in step READ access to Application Performance Analyzer libraries.
5. Define a profile in the RACF STARTED class for the CAZ0 started task so that it can access protected resources. To do this, update your security definitions by using the RDEFINE command to associate CAZ0 with user IDs as follows:

```
RDEFINE STARTED CAZ0.* STDATA(USER(userid) GROUP(group))
```

Where *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* or *OS/390 Security Server (RACF) Security Administrator's Guide*.

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

```
COM='START CAZ0'
```

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 14 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,*

              RULE END
* ----- *
```

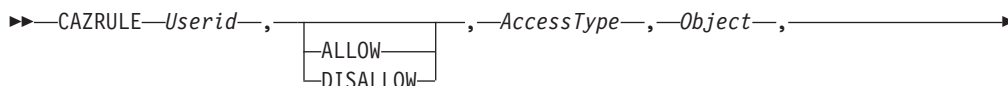
Specifying access rules

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 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 13 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 13 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,=====
```

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 the access is granted to FRANK01. The third positional parameter specifies the type of access. In this case, ViewRequestsOwnedBy; it specifies which owners' measurement requests FRANK01 will be allowed to display. The fourth positional parameter specifies *. FRANK01 is allowed to display 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 fourth 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.

Access rule descriptions:

Table 4. Coding for access types for object types

Code this	For this type of access	Object type
ViewRequestsOwnedBy	Controls which measurement requests will be visible to the user.	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).
UseDB2P1us	Controls whether the user is allowed to use the DB2+ data extractor.	User ID of user creating the request.

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 only their own requests. The second statement, however, specifies that User ID FRANK01 is allowed to view 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 STC measurement.

The rules supplied in CAZCNFG1 leave access fairly open, they are shown here:

```
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,*
```

Rule number	Explanation
01	Users can view only their own measurement requests. Other users' requests are hidden.
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.
07	Any user is allowed to turn on the DB2+ data extractor.

Started task configuration settings

The Application Performance Analyzer started task CAZ0 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

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 might be specified:

STCid Required.

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

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 17, and LogHLQ

in “CONFIG LOGFILE statement” on page 19 for descriptions of how to set a unique high-level qualifier for the sample files and log files.

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

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.

SPXGroupName

Optional.

Specifies the Application Performance Analyzer Sysplex group that this Application Performance Analyzer instance will be a member of. It can be a maximum of eight alphanumeric characters. All Application Performance Analyzer instances that will communicate with each other must code an identical SPXGroupName=sysplexname.

The Application Performance Analyzer instances in a Application Performance Analyzer Sysplex group must be from different MVS systems. All Application Performance Analyzer instances defined in the same

Application Performance Analyzer Sysplex group must share a common checkpoint file; therefore, it is important that the HLQ, UNIT, and CheckpointDSN are identical.

In summary, the CONFIG BASIC statement for every Application Performance Analyzer instance defined in the same Application Performance Analyzer Sysplex group must be coded identically, with the exception of the STCid, which might be unique.

Default: None.

ExpiryDays

Optional.

Specifies the default number of days a sample file should be retained. The user can override this when entering an observation request by over-typing the Retain File for (days) field in the Schedule New Measurement panel.

CONFIG SAMPLE statement

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

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.

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.

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.

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.

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. 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.

SampleHLQ

Optional.

Specifies the high-level qualifier to be used when creating sample data sets. SampleHLQ must comply with the JCL rule for specifying a data set name. It can be no more than 17 characters long. Optionally, the keyword &USERID might be specified as part of the SampleHLQ. At sample data set allocation time, the TSO userid of the user requesting the sample will be substituted for this variable. The length of SampleHLQ is still restricted to 17 characters, including the keyword &USERID.

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

Default: none.

DB2I

Optional.

This is used to specify that the DB2 Intercept should be turned on. Specify DB2I=YES to turn on this feature. Turning this feature on makes the DB2+ data extractor available, which allows additional DB2 data to be collected.

Default: NO.

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 as 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.

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.

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:

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.

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.

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.

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 will be used.

Default: none.

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 15.

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 and control section names. This feature allows Application Performance Analyzer to report descriptions of customer programs.

DPA table entry elements

Each DPA table entry contains the following elements:

Program 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

A unique eight-character name defines a program group to which the program belongs. The supplied table uses a group name of CUSTOMER.

Sub Group

A unique (within the sub groups) eight-character name defines a program sub groups to which the program belongs. The supplied table uses a sub group name of CUSTOMER.

Description

This field contains a description of the load module or control section function.

CAZDPAGN macro

The CAZDPAGN macro is used to generate assembly statements which, when assembled, produce an object module defining a DPA table. One positional parameter and several keyword parameters are specified in each CAZDPAGN statement. The positional parameter specifies the function code and is one of the following values:

PROLOG

Generates prologue logic at the beginning of the CAZDPAGN module. This must be the first statement in the assembly source. The name field specifies the DPA table CSECT and load module name. This must be USRDPA01. 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 name field. One keyword parameter must be specified: DESCRP. The supplied sample table uses CUSTOMER for the group name.

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 name field. One keyword parameter must be specified: DESCRP. The supplied sample table uses CUSTOMER for the sub group name.

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 format NAME=C-*tran*, where *tran* is the CICS Transaction name (or wild card pattern). CICS Transactions can also be specified as a range. The keyword parameter DESCRP specifies the functional description. The SUBGRP keyword specifies the associated Program Group name.

END Used to mark the end of the block of GROUP statements and the end of the block of PROG statements. No keyword parameters are required.

Example of CAZDPAGN Macros

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

```
1 USRDPA01 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
5 CAZDPAGN NAME=CICSDB2,SUBGRP=CUSTOMER, X
  DESCRP='DB2 Test Program'
  CAZDPAGN NAME=MQSAMP1,SUBGRP=CUSTOMER, X
  DESCRP='MQ Test Program'
```



```

6      CAZDPAGN NAME=C-TDB2,SUBGRP=CUSTOMER,          X
      DESCRP='DB2 Test Trans'
      CAZDPAGN NAME=C-MQS1,SUBGRP=CUSTOMER,          X
      DESCRP='MQ Test Trans'
7      CAZDPAGN END
8      END USRDPA01

```

Notes:

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 CAZDPA01. 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 USRDPA01. 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 CAZR XOEM 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. CAZR XOEM is invoked whenever a user specifies a repository of T for "Third party" on the Source Program Mapping panel, A01.

Edit the member CAZR XOEM in the *hlq.SCAZEXEC* and make the following changes:

1. Comment out the line:

```
QUEUE "CAZR XOEM"
```

2. Replace "SYS1.LINKLIB" in the following statement:

```
QUEUE "ISPEXEC LIBDEF ISPLLIB DATASET",
      "ID('SYS1.LINKLIB') STKADD"
```

with the data set name of the library that contains the utility to be run.

3. Replace "OEMUTIL" in the statement:

```
QUEUE "ISPEXEC SELECT CMD(OEMUTIL)"
```

with the name of the utility to be run.

4. Increase the space to more than the default allocation of one track, if needed, in the statement:

```
QUEUE "INPUT SPACE(1) TRACKS"
```


- 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 CAZO started task

The CAZO 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 10 of the task for making CAZO available, you can start CAZO by entering the command 'S CAZO' on the system console. When you re-IPL your system, the started task CAZO is automatically started.

Starting the IBM Application Performance Analyzer user interface

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

- Enter 'EX 'hlq.SCAZEXEC(CAZ@SPF)'' at the TSO/E command 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 IBM Application Performance Analyzer Primary Options menu.

2. To begin an analysis session, choose one of the options from the menu and press Enter.

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

Reqnum  Owned By  Description      Job Name  Date/Time      Samples  Status
-----
0082    USER1 id    MACHIND         MACHIND   Jan-3  18:12        50,000  Ended
0080 +  USER1 ims   DONDRVRN        DONDRVRN  Jan-3  18:11        50,000  STEPS
0073 +  USER2      MQGET           MQGET     Jan-3  18:11        60,000  STEPS
0072 +  USER2      MQPUT           MQPUT     Jan-3  18:10        60,000  STEPS
0071 +  USER2      CICSC22F        CICSC22F  Jan-3  18:07        60,000  REPEAT
0068 +  USER2      VSAMJOB         VSAMJOB   Jan-3  18:08        60,000  STEPS
0067    USER2      CICSC22F        CICSC22F  Jan-3  17:55        60,000  Ended
0066    USER2      DBJOB8          DBJOB8    Jan-3  17:56       100,001  Cancel
0065    USER2      AMENT1          AMENT1    Jan-3  17:54         4,567  Ended
0064    USER2      AMENT1          AMENT1    Jan-3  17:54           1  Failed
0063    USER2      AMENTPL1        AMENTPL1  Jan-3  17:52           1  Failed

Welcome to IBM APA for z/OS ISPF Version 1.1000. 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 1.100C starting.
```

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

```
CAZ0100I CAZ0-MAIN (PLXF067) 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
	JESMSG LG	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 JESMSG LG. If the started task fails to start successfully, look in JESMSG LG 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
ReqNum  Owned By  Description  Job Name  Date/Time  Samples  Status
-----
+-----+
| Welcome to Application Performance Analyzer/ISPF Version 1.1. You are |
| currently connected to measurement task id CAZ0.                    |
| Enter CONNECT for an alternate connection, VERSION                 |
| for additional version information.                                  |
+-----+
```

The screen should have displayed a message indicating that you are connected to the measurement task. If this has not appeared, enter a CONNECT command specifying your Application Performance Analyzer instance identifier. For example:

```
==> CONNECT CAZ0
```

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. CICS Txns      7. Schedule
  2. Options              4. Active Jobs      6. Sysplex        8. Sched Options

Panel 1. Job Information

Job Name/Pattern . . _____

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

```

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

Regnum  Owned By  Description      Job Name  Date/Time      Samples  Status
-----
0001    USER1    my userid       MACHIND   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

Regnum  Owned By  Description      Job Name  Date/Time      Samples  Status
-----
0001    USER1 id  MACHIND         Jun-10  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)      Row 00001 of 00007
Command ==>>> _____ Scroll ==>>> CSR

Select a category from  __A Admin/Miscellaneous      __I IMS Measurement
the list to the right  __S Statistics/Storage      __E CICS Measurement
to view the available  __C CPU Usage Analysis      __F DB2 Measurement
reports in the selection __D DASD I/O Analysis      __Q MQ Measurement
list below.            __W CPU WAIT Analysis      __G Coupling Facility

-----

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
__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:

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

This chapter describes the tasks for configuring the optional DB2 Universal Database component. These tasks are only required if you want Application Performance Analyzer to gather the necessary data to report plan and package BIND timestamps.

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

1. "Deciding which authorization ID to use"
2. "Binding a DB2 package and plan"
3. "Providing DB2 authorization"
4. "Updating the Application Performance Analyzer started task" on page 32
5. "Updating the started task configuration settings" on page 32

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").

Binding a DB2 package and plan

To bind a DB2 UDB 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 UDB 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.

Providing DB2 authorization

Grant EXECUTE authority on the plan CAZPLAN1 to the authorization ID of the Application Performance Analyzer started task. You can use the following SQL statement to grant this privilege:

```
GRANT EXECUTE ON PLAN CAZPLAN1 TO authid
```

Note: CAZPLAN1 is the default plan name. If you have chosen to use a different plan name, then replace CAZPLAN1 with the plan name that was used to BIND the Application Performance Analyzer DB2 Plan.

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 must contain the following load modules: DSNALI, DSNTIAR, and alias DSNHLI2.

Updating the started task configuration settings

1. Edit member CAZCNFG0 in library *hlq.SCAZSAMP*.
2. Find the CONFIG SAMPLE statement, then specify the DB2 plan name in the DB2PlanName parameter.
3. Restart your Application Performance Analyzer started task CAZ0 to make the change effective.

Appendix A. Stopping the IBM Application Performance Analyzer started tasks

To stop the Application Performance Analyzer started tasks, do the following steps:

1. Enter the command 'P CAZ0' from the system console.
2. If 'P CAZ0' fails to stop the started task, retry the command.

Appendix B. 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:

- Step 1. Prepare to install the APAR or PTF
- Step 2. Receive the APAR or PTF (SMP/E RECEIVE)
- Step 3. Review the HOLDDATA
- Step 4. Accept previously applied APARs or PTFs (optional) (SMP/E ACCEPT)
- Step 5. Apply APAR or PTF (SMP/E APPLY)
- Step 6. Test APAR or PTF
- Step 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 C. 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 D. 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 might be used. Subject to IBM's valid intellectual property or other legally protectable rights, any functionally equivalent product, program, or service might 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 might 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

The following terms are trademarks and/or registered trademarks of the IBM Corporation in the United States or other countries or both:

BookManager	MVS
CICS	OS/390
DB2	RACF
DB2 Universal Database	VTAM
IBM	z/OS
IMS	

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, SC18-9599
Messages Guide, GC18-9600
Program Directory, GI10-8693
User's Guide, SC18-9601

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

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 UDB 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

Index

A

APAR, applying 35
Application Performance Analyzer
 authorizing modules 7
 customizing 7
 starting 23
 stopping tasks 33
authorizing modules 7

C

CAZ0
 making available 10
 starting 23
CAZAM
 starting 23
CAZPLINS member 31
checklist 1
CICS component 29
customizing
 tasks 7
 with DB2 UDB 31

D

DB2 UDB
 access plan 31
 customizing 31
DB2 UDB access plan, creating 31
DB2 UDB component, configuring 31

I

ISPF Primary Option menu 9

L

library names 2

M

modifying TSO/E logon 9

P

PTE, applying 35

S

starting Application Performance
 Analyzer 23

T

tasks, stopping 33
TSO/E logon, modifying 9

Readers' Comments — We'd Like to Hear from You

IBM Application Performance Analyzer for z/OS
Customization Guide
Version 1 Release 1

Publication No. SC18-9599-02

Overall, how satisfied are you with the information in this book?

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Overall satisfaction	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

How satisfied are you that the information in this book is:

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Accurate	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Complete	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Easy to find	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Easy to understand	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Well organized	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Applicable to your tasks	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Please tell us how we can improve this book:

Thank you for your responses. May we contact you? Yes No

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.

Name

Address

Company or Organization

Phone No.



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 95141-9989



Fold and Tape

Please do not staple

Fold and Tape



Printed in USA

SC18-9599-02

