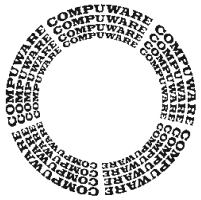
STROBE MVS

STROBE ADABAS/Natural Feature

Release 3.0



COMPUWARE®

Please direct questions about STROBE MVS
or comments on this document to:

STROBE MVS Technical Support
Compuware Corporation
124 Mount Auburn Street
Cambridge MA 02138-5758
1-800-585-2802 or
1-617-661-3020
1-617-498-4010 (fax)
strobe-sup@compuware.com

Outside the USA and Canada, please contact
your local Compuware office or agent.

Doc. CWSTUN3A
January 22, 2003.

Contents

Figures

Summary of Changes

This section lists the changes to the ADABAS/NATURAL Feature for STROBE MVS for Sysplex Release 3.0.

Changes to the STROBE ADABAS/NATURAL Feature

STROBE MVS for Sysplex Release 3.0 supports all NATURAL versions through 3.1.5 and all Adabas versions through 7.3. Also the CPU Usage by Natural Statement report now includes data that identifies the name of a global buffer pool if one exists, (if not the field reads *LOCAL*) and an IMS reference number column is provided if the NATURAL statement attempts to access an IMS database

Changes to this Manual

The STROBE ADABAS/NATURAL Feature provides an example of the new version of the CPU Usage by Natural Statement report and an explanation of the new data and an example of an IMS report that shows the cross-reference to the IMS database that was called.

Introduction

This manual describes measurement concepts applicable to and specific data made available by the STROBE ADABAS/NATURAL Feature of the STROBE MVS Application Performance Measurement System. The STROBE ADABAS/NATURAL Feature augments functions provided by the basic STROBE system.

The STROBE MVS Application Performance Measurement System and the STROBE ADABAS/NATURAL Feature are products designed for IBM MVS/ESA, IBM OS/390, and IBM z/OS systems. STROBE ADABAS/NATURAL Feature is designed for use with Software AG's ADABAS DBMS versions 5.2, 5.3, 6.1, 7.1, 7.2 and 7.3 and NATURAL 4GL versions 2.2.8, 2.3.1, 2.3.2, 2.3.4, 3.1.1, 3.1.2, 3.1.3, 3.1.4 and 3.1.5.

How This Manual Is Organized

Chapter 1, "Overview" presents an overview of the STROBE ADABAS/NATURAL Feature.

Chapter 2, "Using the STROBE ADABAS/NATURAL Feature" explains how to use the STROBE ADABAS/NATURAL Feature.

Chapter 3, "The STROBE Performance Profile for NATURAL Regions" describes and interprets a STROBE Performance Profile for an address space that processes NATURAL application programs.

Chapter 4, "The STROBE Performance Profile for ADABAS Regions" describes and interprets a STROBE Performance Profile for an ADABAS region.

Chapter 5, "STROBE Wait Time Attribution for 3GL Programs and ADABAS SQL Server Statements" describes a STROBE Performance Profile that shows attribution for a 3GL program that accesses ADABAS.

How to Use This Manual

You should read Chapter 1, "Overview" and Chapter 2, "Using the STROBE ADABAS/NATURAL Feature" before submitting a measurement request. If you have a STROBE Performance Profile for an address space executing NATURAL that you want to interpret, read Chapter 3, "The STROBE Performance Profile for NATURAL Regions". If you have a STROBE Performance Profile for an ADABAS region that you want to interpret, read Chapter 4, "The STROBE Performance Profile for ADABAS Regions". To learn how to interpret a STROBE Performance Profile that shows 3GL attribution, read Chapter 5, "STROBE Wait Time Attribution for 3GL Programs and ADABAS SQL Server Statements".

The STROBE Library

The STROBE base product manuals include:

- *STROBE MVS Concepts and Facilities*, document number CWSTGX3A

  STROBE MVS Concepts and Facilities explains how to decide which programs and online regions to measure, when to measure them, and how to interpret the reports in the STROBE Performance Profile.

- *STROBE MVS Messages*, document number CWSTXM3A

  STROBE MVS Messages lists all messages and abnormal termination (ABEND) codes, describes how to interpret them, and in many cases suggests a corrective action.

- *STROBE MVS System Programmer's Guide*, document number CWSTXI3A

  The *STROBE MVS System Programmer's Guide* explains how to install and maintain STROBE.

- *STROBE MVS User's Guide*, document number CWSTUX3A and the *STROBE MVS User's Guide with Advanced Session Management*, document number CWSTUA3A

  The *STROBE MVS User's Guide* explains how to use STROBE to measure application performance. The *STROBE MVS User's Guide with Advanced Session Management* explains how to use STROBE with the STROBE Advanced Session Management Feature to measure application performance. Users who have the STROBE Advanced Session Management Feature will use this manual rather than the *STROBE MVS User's Guide*.

- *STROBE MVS Application Performance Measurement System Quick Reference*, form number PGM001

  The *STROBE MVS Application Performance Measurement System Quick Reference* is a convenient reference for how to use STROBE and for interpreting the STROBE Performance Profile.

STROBE Feature Manuals

These manuals describe the optional features of the STROBE MVS Application Performance Measurement System. Each manual describes measurement concepts applicable to and specific data made available by the feature.

- *STROBE MVS User's Guide with Advanced Session Management*, document number CWSTUA3A
- *STROBE ADABAS/NATURAL Feature*, document number CWSTUN3A
- *STROBE CA-IDMS Feature*, document number CWSTUR3A
- *STROBE CICS Feature*, document number CWSTUC3A
- *STROBE COOL:Gen Feature*, document number CWSTUG3A
- *STROBE CSP Feature*, document number CWSTUP3A
- *STROBE DB2 Feature,* document number CWSTUD3A
- *STROBE IMS Feature,* document number CWSTUI3A
- *STROBE Interface Feature*, document number CWSTUF3A
- *STROBE Java Feature,* document number CWSTUJ3A
- *STROBE MQSeries Feature,* document number CWSTUM3A
- *STROBE UNIX System Services Feature,* document number CWSTUU3A

Online Documentation

STROBE manuals are available in HTML, Adobe Acrobat PDF format, and IBM BookManager format, on CD-ROM and at Compuware's technical support Web site at http://frontline.compuware.com.

Online Help

STROBE products provide the following online information:

- STROBE/ISPF Online Tutorials, Option T from the STROBE/ISPF STROBE OPTIONS menu
- STROBE/ISPF Online Message Facility, Option M from the STROBE/ISPF STROBE OPTIONS menu

Other Compuware Application Performance Management Products

The following products and features work in conjunction with STROBE MVS Application Performance Measurement System. These tools extend the benefits of application performance management (*APM*).

iSTROBE

iSTROBE enables you to view and analyze STROBE Performance Profile data on a workstation using a standard Web browser. Easy to install and easy to use, iSTROBE guides you through the performance analysis process and offers recommendations for improving performance. iSTROBE simplifies the performance analysis of applications that you measure with STROBE. For more information on iSTROBE, see the iSTROBE *Getting Started Guide*.

SQL Analysis Feature

The SQL Analysis Feature works in conjunction with STROBE and iSTROBE or APMpower to supply access path analyses and database and SQL coding recommendations for DB2 applications measured by STROBE. The SQL Analysis Feature pinpoints the most resource-consumptive static or dynamic SQL statements, explains why these statements might be inefficient, and provides recommendations to improve the performance of the DB2 application. For more information on the SQL Analysis Feature, see the *STROBE MVS User's Guide* or the *STROBE MVS User's Guide with Advanced Session Management*.

APMpower

The APMpower Application Performance Analysis System extends the benefits of STROBE to application developers who use workstations to develop, test, and maintain MVS applications. Developers employ the APMpower graphical user interface and advanced analytical aids to navigate the Performance Profile, analyze and improve application performance, and share performance knowledge across the IS organization. For more information about APMpower, see the APMpower documentation.

Compuware APM Technical Support

For North American customers, for technical support, please contact the Technical Support department by telephone at (800) 585-2802 or (617) 661-3020, by fax at (617) 498-4010, or by e-mail at strobe-sup@compuware.com.

To access online technical support, visit Compuware's FrontLine page on the World Wide Web at http://frontline.compuware.com and select the product "STROBE and APMpower."

For other international customers, please contact your local Compuware office or STROBE supplier.

Compuware APM Training

Compuware's Education Resources Group offers a range of training options for organizations that use STROBE, iSTROBE, and APMpower. To arrange Application Performance Management training, please contact Compuware at 1-800-835-3190 or visit Compuware's Education Resources Group at http://www.compuware.com/training

For other international customers, please contact your local Compuware office or STROBE supplier for a complete list of APM Training offerings.

Compuware APM Service Offerings

For North American customers, for information about current service offerings, please contact your local Compuware sales office or call Compuware Corporate Headquarters at 1-800-COMPUWARE (266-7892) or visit Compuware's APM Product page on the World Wide Web at http://www.compuware.com/products/strobe.

For other international customers, please contact your local Compuware office or STROBE supplier for a complete list of Services offerings.

APM Installation Assurance

The APM Installation Assurance service assists you in planning for, installing, customizing and using APM products. The service will help you maximize the value and benefits derived from the APM product family.

Consulting engineers work closely with your IT personnel to understand your operating environment and your organization's APM goals. The engineer will assist you in developing a customization and installation plan for STROBE, iSTROBE, and APMpower. The engineer will oversee the installation process and verify product readiness. The engineer will also help set up measurement request schedules, request groups, history records, AutoSTROBE measurement requests, and will verify the installation of the SQL Analysis Feature.

With APM Installation Assurance services, your organization can immediately maximize the value received from your investment in the APM product family. You will also benefit from a fully customized installation that will enhance the product functionality and increase the automation aspects of your APM initiatives.

Application Performance Management Consulting

The Application Performance Management (APM) Consulting services assist you in identifying and resolving specific performance problems in your OS/390 business-critical applications.

Using STROBE, iSTROBE, and APMpower, consulting engineers work closely with your IT personnel to measure an application's performance, identify performance improvement opportunities and make recommendations for implementing solutions.

With APM Consulting services, your organization cannot only resolve problems quickly and effectively, but also gain the skills necessary to prevent application performance degradation in the future.

Application Performance Assessment

The Application Performance Assessment (APA) service assists you in achieving a higher level of performance for your OS/390 business-critical applications.

Using STROBE, iSTROBE, and APMpower, consulting engineers work closely with your IT personnel to evaluate the efficiency of business-critical applications, identify opportunities for improving performance and document the potential savings that can result from implementing recommended solutions.

With APA services, you cannot only improve application performance quickly and effectively, but also gain the knowledge and skills necessary to implement and sustain a process-oriented application performance management (APM) program.

Chapter 1. Overview

STROBE is a product that determines where time is spent in online subsystems and batch processing programs. STROBE produces a collection of reports that help you determine where and how to revise applications to improve their performance.

The STROBE ADABAS/NATURAL Feature extends the functions of STROBE. With the Feature, you can measure and evaluate three types of performance data:

- the performance of applications developed and executing in a NATURAL environment
- the performance of the ADABAS database management system
- the amount of wait time for 3GL programs that access ADABAS

NATURAL Language Support

There are two types of NATURAL programs: *NATURAL nucleus programs* and *NATURAL 4GL programs*. NATURAL nucleus programs make up the NATURAL run-time system, and they are supplied by SOFTWARE AG. NATURAL 4GL programs are written in the NATURAL language by application programmers. Since nucleus programs cannot be modified by applications programmers, the STROBE ADABAS/NATURAL Feature focuses on identifying performance improvement opportunities in NATURAL 4GL programs.

NATURAL 4GL programs can execute in teleprocessing environments such as CICS and batch processing regions. They can access data managed by ADABAS, IBM's VSAM, and other database management systems. The STROBE ADABAS/NATURAL Feature does not support NATURAL 4GL programs executing in IMS or COM-PLETE environments.

The STROBE ADABAS/NATURAL Feature can measure the performance of NATURAL 4GL programs executing in CICS, IMS and batch regions. To measure NATURAL 4GL programs executing in a CICS or an IMS environment, you must use the STROBE CICS Feature or the STROBE IMS Feature in conjunction with the STROBE ADABAS/NATURAL Feature.

The STROBE ADABAS/NATURAL Feature

- identifies CPU usage in NATURAL 4GL programs by statement number and, optionally, shows the source code text associated with each statement number
- in batch environments, identifies wait by NATURAL statement number
- provides function descriptors for NATURAL nucleus routines
- provides the NATURAL Program CPU Usage Summary report, which shows a breakdown of activity by NATURAL 4GL program name, and the CPU Usage by NATURAL Statement reports, which show activity for each NATURAL 4GL program broken down by statement number.

ADABAS Support

When the STROBE ADABAS/NATURAL Feature measures an ADABAS region, the Performance Profile for that region shows:

- CPU activity in ADABAS modules and user-exit routines
- function descriptors for the ADABAS nucleus routines
- CPU usage attributed to callers within the NATURAL regions that caused the activity, identified:
  - for a batch job, by the job name
  - for a TSO region, by the TSO user ID
  - for a CICS region, by the logical terminal name
  - for an IMS region, by the IMS module name

3GL Wait Time Attribution

STROBE can measure a 3GL program written in assembler, COBOL, FORTRAN, or PL/I that makes direct calls or ADABAS SQL Server statement calls to an ADABAS region. It attributes the wait incurred when the direct calls or ADABAS SQL Server statement calls access the ADABAS database. When wait time is caused in the module which controls access to ADABAS, STROBE attributes the time to an offset in the calling module. The wait time is recorded in the Attribution of CPU Wait report along with wait time caused by NATURAL programs that access ADABAS.

Identification of Activity in Pseudo-Entities

The STROBE ADABAS/NATURAL Feature generates the following pseudo-entities under which it reports activity.

Pseudo-Entity	Type of Activity STROBE Reports
.ADABAS	ADABAS database activity. (Module names beginning with ADA.)
.ADASQL	Activity in ADABAS SQL Server statements. (Module names beginning with ESQ.)
.NATNUC	Activity in the NATURAL nucleus. (Module names beginning with ACM, NAT, NCI, and NDB.)
.NATURAL	Execution in the NATURAL nucleus that cannot be assigned to a NATURAL 4GL program.
.xxxxxxx	Activity in an ADABAS task other than a system service. The name begins with a "." followed by the first seven characters of the module name in which the execution of the task was initiated.

Benefits of the STROBE ADABAS/NATURAL Feature

The data provided by the STROBE ADABAS/NATURAL Feature will help you throughout the application life cycle:

- in design and build, to assess alternative programming techniques by measuring the performance of various test cases
- in test and quality assurance, to confirm design assumptions about new or modified application resource requirements
- in production, to ensure efficient performance of NATURAL 4GL programs and usage of ADABAS services
- in maintenance, to assess the impact of changes in business requirements and data volume on the performance of NATURAL 4GL applications and on the ADABAS database management system.

Chapter 2.
Using the STROBE ADABAS/NATURAL Feature

Before you can use the STROBE ADABAS/NATURAL Feature, your STROBE system programmer must customize and install the STROBE ADABAS/NATURAL Configurations Module (STRBANCM). For more information on this module, refer to "Customize the STROBE ADABAS/NATURAL Configurations Module" in Chapter 3 of the STROBE *MVS System Programmer's Guide*.

You can find detailed instructions for submitting measurement requests in the STROBE/ISPF Online Tutorials and in Chapter 2 and Chapter 3 of the STROBE *MVS User's Guide* or the STROBE *STROBE MVS User's Guide with Advanced Session Management*.

Invoking the STROBE ADABAS/NATURAL Feature

As described in Chapter 1, "Overview", you can obtain three categories of measurement data with the STROBE ADABAS/NATURAL Feature. One of three data collectors must be invoked depending on whether you want NATURAL, ADABAS, or 3GL measurement data.

Invoking the NATURAL Data Collector

If you want to include NATURAL 4GL program measurement data in your STROBE Performance Profile, the NATURAL data collector must be operating. Use the NATURAL data collector when you want to measure

- an address space that is running a NATURAL batch program
- a CICS region in which NATURAL transactions are running
- an IMS region in which NATURAL transactions are running.
- a TSO region that is running NATURAL

STROBE automatically invokes the STROBE NATURAL data collector when:

- You measure a target program whose name (as given in the PGM= operand of the EXEC statement that invokes the job step) begins with "NATL".
- You measure a CICS region.

Otherwise, you must specify the STROBE NATURAL data collector when you add a measurement request.

- With STROBE/ISPF, you must enter "Y" in the DATA COLLECTORS field of the ADD ACTIVE or ADD QUEUED panel, then enter "Y" in the NATURAL field on the STROBE - DATA COLLECTORS panel (Figure 2-1 on page 2-2).
- With STROBE command language, you must include the NATURAL operand when you submit an ADD operation.

Invoking the ADABAS Data Collector

If you want to include ADABAS program measurement data in your Performance Profile, the ADABAS data collector must be operating. Use the ADABAS data collector when you want to measure the region that is running the ADABAS database. STROBE automatically invokes the STROBE ADABAS data collector when you measure a target program whose

name (as given in the PGM= operand of the EXEC statement that invokes the job step) begins with “ADAB”, or “ADAR”.

Otherwise, you must specify the STROBE ADABAS data collector when you add a measurement request:

- With STROBE/ISPF, you must enter “Y” in the DATA COLLECTORS field of the STROBE ADD ACTIVE or STROBE ADD QUEUED panel, then enter “Y” in the ADABAS field on the STROBE - DATA COLLECTORS panel.
- With STROBE command language, you must include the ADABAS operand when you submit an ADD operation.

Figure 2-1. STROBE - DATA COLLECTORS Panel

```
--------------------------- STROBE - DATA COLLECTORS --------------------------
COMMAND ===>

OVERRIDE DATA COLLECTOR DEFAULTS FOR JOBNAME:  WPAFXC

DATA COLLECTORS: (Y or N; Y adds to and N removes from your system defaults)
  ADABAS           ===>     ADA3GL           ===>     C                ===>
  CICS             ===>     COBOL            ===>     CSP              ===>
  DB2              ===>     IDMS             ===>     IDMS BATCH DML   ===>
  IEF              ===>     IMS              ===>     NATURAL          ===>  Y
  PL/I             ===>     SVC              ===>


CICS Options: (Y or N; default is Y)
 Produce Performance Supplement ===>   Collect terminal activity ===>


CAPTURE Options: (Y or N; default is Y)
          DB2    ===>    IMS    ===>


OTHER DATA COLLECTORS:
  PROGRAM NAME   ===>           ===>           ===>           ===>
  PROGRAM NAME   ===>           ===>           ===>           ===>
```

Invoking the ADA3GL Direct Call and ADABAS SQL Server Statement Call Data Collector

If you have an ADABAS database that is accessed by direct calls or ADABAS SQL Server statement calls from a 3GL program running in a batch environment, you can invoke the ADA3GL data collector to measure the wait the 3GL program incurred. The data collector for direct calls and ADABAS SQL Server statement calls from an ADA3GL program cannot be automatically invoked; instead you must either:

- specify “Y” in the ADA3GL field of the STROBE - DATA COLLECTORS panel
- with STROBE command language, include the ADA3GL operand with the ADD operation

The wait time measurement data is placed in the Attribution of CPU Wait Time Report (Figure 3-14 on page 3-17.)

Note: If you specify ADA3GL attribution, then the STROBE ADABAS data collector is automatically invoked. You do not need to specify “Y” in both the ADA3GL and the ADABAS fields on the STROBE - DATA COLLECTORS panel. STROBE generates an error message if you specify both options. To collect both 3GL and NATURAL information, you have to specify both the ADA3GL and the NATURAL data collectors.

If you are not interested in the measurement data collected by the STROBE ADABAS/NATURAL Feature, you can disable it:

- with STROBE/ISPF,
  - enter "Y" in the DATA COLLECTORS field of the STROBE - ADD ACTIVE REQUEST or STROBE - ADD QUEUED REQUEST panel,
  - then enter "N" in the NATURAL or ADABAS field on the STROBE - DATA COLLECTORS panel
- with STROBE command language,
  - include the operand NOADABAS or NONATURAL when you submit an ADD operation

Indexing NATURAL Programs

For NATURAL 4GL programs, the ADABAS/NATURAL Feature can match CPU activity to statement number. You can create a Performance Profile that will display this information in the CPU Usage by NATURAL Program Statement report with DB2 Information (Figure 3-4 on page 3-8). To associate the reported statement number with the source code text is a two-step process. You must

- create the NATURAL 4GL program listing data set
- create a map data set from the listing data set

STROBE provides a procedure named STROXNAT that executes both processes. Then you just need to specify the name of the map data set when you create the Performance Profile. STROBE also provides a procedure named STROXN that only creates the NATURAL 4GL program listing data set. Once you have created the listing data set, you have two options to create a map data set:

- in batch, with the STROX procedure
- in STROBE/ISPF, with Option 5

These procedures are described in the sections that follow. If you do not have access to the STROBE procedures, see your STROBE system programmer. The installation of these procedures is described in Chapter 4 of the STROBE *MVS System Programmer's Guide*.

Using the STROXNAT Procedure to Create a Map Data Set in Batch

The STROXNAT procedure (Figure 2-2 on page 2-4) creates a listing data set that includes the NATURAL 4GL program source listing, the database identification number, and the file number. The procedure uses the listing data set to create the map data set.

Setting Default Parameters in the STROXNAT Procedure

Before you execute STROXNAT for the first time, specify default values for the parameters highlighted in Figure 2-2 on page 2-4. Once you have specified values, execute STROXNAT, supplying its run-time parameters as described in "Executing the STROXNAT Procedure" on page 2-4.

You can override any of the STROXNAT parameters at run time by specifying a new value for that parameter when you execute the procedure.

Figure 2-2. STROXNAT Procedure

```
//STROXNAT PROC NATBNUC=,              NATURAL BATCH NUCLEUS NAME
//         NATREG=2048K,               REGION SIZE
//         NPARM=,                     NATURAL RUN PARAMETER
//         NINDX=,                     NATURAL HIGHER LEVEL INDEX NAME
//         AINDX=,                     ADABAS HIGHER LEVEL INDEX NAME
//         ADAPARM=,                   ADABAS PARM FILE NAME
//         DAUNIT=SYSDA,               UNIT ASSIGNMENTS FOR DA DEVICES
//         PRTATYP=TRK,                CMPRINT ALLOCATION TYPE (NATURAL utility listing data set)
//         PRTPRIM=3,                  CMPRINT PRIMARY ALLOCATION
//         PRTSEC=1,                   CMPRINT SECONDARY ALLOCATION
//         MAPDSN=MAPDSN,              MAP FILE DSN
//         MAPVOL=                     MAP FILE VOLSER
//         MAPDISP=CATLG,              MAP FILE DISPOSITION
//         INDXPRM=,                   INDEXER STEP PARAMETER
//         PRTCLAS='*',                SYSOUT PRINT CLASS
//         STRLIB=STROBLIB,            STROBE LOAD MODULE LIBRARY
//         SYSUDMP=DUMMY               SYSUDUMP DATA SET
//*                                    .
                                       .
```

Executing the STROXNAT Procedure

The following description and the example at the end of this section explain how to execute the STROXNAT procedure.

- Invoke the STROXNAT procedure and supply the following run-time parameters:

Parameter	Value
NATBNUC	The name of your NATURAL batch nucleus
NPARM	Additional NATURAL utility parameters (use the FUSER parameter to specify the ADABAS database ID and file number of the NATURAL program library)
NINDX	High-level qualifier of the NATURAL load library
AINDX	High-level qualifier of your ADABAS load library
ADAPARM	The name of your ADABAS parameters library
MAPDSN	The name of the output map data set to be created by the STROXNAT procedure

- In the UNLOAD.SYSIN data set:
  - Specify the NATUNLD utility to include the database identification number and the file number in the listing data set.
  - Specify the LIST system command to include the NATURAL 4GL program source listing in the listing data set. STROXNAT uses the listing data set to create the map data set.
  - Replace *natprog* with the name of the NATURAL 4GL program to be included in the listing data set.
  - Replace *libname* with the name of the NATURAL library in which the NATURAL 4GL program resides.

- Execute the STROXNAT procedure as shown below:

```
//NATURAL  EXEC STROXNAT,
//         NATBNUC=NAT222BA,
//         NPARM='FUSER=(1,9)',
//         NINDX='NATURAL.NAT222',
//         AINDX='ADABAS.V519',
//         ADAPARM='NATURAL.NAT222.JCL(ADAPARM)',
//         MAPDSN='NATURAL.NATPROG.MAPDSN'
//UNLOAD.SYSIN   DD *

NATUNLD SAVED natprog FM LIB libname WITH TYPE P
.
LOGON libname
LIST natprog EXP C *
```

For more information on NATUNLD, see the *NATURAL 2.2 Utilities Manual*. For more information on using the LIST system command, see the *NATURAL Reference Manual* specific to the release you are using.

Using the STROXN Procedure to Create a Listing Data Set in Batch

The STROXN procedure (Figure 2-3) creates a listing data set that includes the NATURAL 4GL program source listing, the database identification number, and the file number. After executing the STROXN procedure, you can either submit the listing data set to STROBE/ISPF or run the STROX procedure to create a map data set.

Setting Default Parameters in the STROXN Procedure

Before you execute STROXN for the first time, specify values for the default parameters highlighted in Figure 2-3. Once you have specified values, execute STROXN, supplying its run-time parameters as described in "Executing the STROXNAT Procedure" on page 2-4.

You can override any of the STROXN parameters at run time by specifying a new value for that parameter when you execute the procedure.

Figure 2-3. STROXN Procedure

```
//STROXN   PROC NATBNUC=,              NATURAL BATCH NUCLEUS NAME
//         NATREG=2048K,               REGION SIZE
//         NPARM=,                     NATURAL RUN PARAMETER
//         NINDX=,                     NATURAL HIGHER LEVEL INDEX NAME
//         AINDX=,                     ADABAS HIGHER LEVEL INDEX NAME
//         ADAPARM=,                   ADABAS PARM FILE NAME
//         DAUNIT=SYSDA,               UNIT ASSIGNMENTS FOR DA DEVICES
//         LISTDSN=,                   LISTING DSN
//         LSTATYP=TRK,                LISTING ALLOCATION TYPE
//         LSTPRIM=3,                  LISTING MERGE PRIMARY ALLOCATION
//         LSTSEC=1,                   LISTING SECONDARY ALLOCATION
//         PRTCLAS='*',                SYSOUT PRINT CLASS
//         SYSUDMP=DUMMY               SYSUDUMP DATA SET
//*                                    .
                                       .
```

Executing the STROXN Procedure

The following description and the example at the end of this section explain how to execute the STROXN procedure.

- Invoke the STROXN procedure and supply the following run-time parameters:

Parameter	Value
NATBNUC	The name of your NATURAL batch nucleus
NPARM	Additional NATURAL utility parameters (use the FUSER parameter to specify the ADABAS database ID and file number of the NATURAL program library)
NINDX	High-level qualifier of the NATURAL load library
AINDX	High-level qualifier of your ADABAS load library
ADAPARM	The name of your ADABAS parameters library
LISTDSN	The name of the output data set to be created by the STROXN procedure

- In the UNLOAD.SYSIN data set:
  - Specify the NATUNLD utility to include the database identification number and the file number in the listing data set.
  - Specify the LIST system command to include the NATURAL 4GL program source listing in the listing data set.
  - Replace *natprog* with the name of the NATURAL 4GL program to be included in the listing data set.
  - Replace *libname* with the name of the NATURAL library in which the NATURAL 4GL program resides.
- Execute the STROXN procedure as shown below:

```
//NATURAL EXEC STROXN,
//          NATBNUC=NAT222BA,
//          NPARM='FUSER=(1,9)',
//          NINDX='NATURAL.NAT222',
//          AINDX='ADABAS.V519',
//          ADAPARM='NATURAL.NAT222.JCL(ADAPARM)',
//          LISTDSN='NATURAL.NATPROG.LISTING'
//UNLOAD.SYSIN   DD *

NATUNLD SAVED natprog FM LIB libname WITH TYPE P
.
LOGON libname
LIST natprog EXP C *
```

For more information on NATUNLD, see the *NATURAL 2.2 Utilities Manual*. For more information on using the LIST system command, see the *NATURAL Reference Manual* for the release you are using.

Using STROBE/ISPF to Create a Map Data Set

To create a map data set with STROBE/ISPF, enter the following on the STROBE - INDEX TO CREATE A MAP DATA SET panel (Figure 2-4 on page 2-7):

- "Y" in the ADABAS/NATURAL PROGRAM LANGUAGE field
- the name of the map data set in the OUTPUT: MAP DATA SET field
- the name of the output listing data set, created by the STROXN procedure, in one of the INPUT: COMPILER SYSPRINT DATA SETS fields

You can concatenate several listing data sets to produce one map data set by entering more than one listing data set name in the INPUT fields.

Figure 2-4. STROBE - INDEX TO CREATE A MAP DATA SET Panel

```
------------------- STROBE - INDEX TO CREATE A MAP DATA SET -------------------
OPTION ===>
            B  - Background processing   F  - Foreground processing

PROGRAM LANGUAGE: (Specify Y -- select one language)
 ADABAS/NATURAL  ===> Y     ADS/O           ===>       ASSEMBLER       ===>
 C (IBM OR SAS)  ===>       CA OPTIMIZER    ===>       COBOL           ===>
 CSP             ===>       DB2 DBRM        ===>       FORTRAN G       ===>
 FORTRAN VS OR H ===>       IEF             ===>       PL/I            ===>

OUTPUT: MAP DATA SET
        ===> 'wpa.mapfile'
  UNIT ===> WPAANY     VOLUME ===>

INPUT: COMPILER SYSPRINT DATA SETS
        ===> 'natural.natprog.listing'
        ===>
        ===>
        ===>
        ===>
        ===>
```

Using the STROX Procedure to Create a Map Data Set

Execute the STROX procedure to index the program and create a map data set, supplying these parameters:

Parameter	Value
MAPPGM	STRXNAT (the NATURAL Indexer program)
LISTDSN	The name of the listing data set created by the STROXN procedure
MAPDSN	The name of the output map data set

```
//INDEX    EXEC  STROX,
//               MAPPGM=STRXNAT,
//               LISTDSN='NATURAL.NATPROG.LISTING',
//               MAPDSN='MY.MAP.FILE'
```

To index several source listings and produce only one map data set, override the LISTING DD statement in the STROX procedure with a concatenation of the sequential data set names.

```
//INDEX    EXEC STROX,
//                  MAPPGM=STRXNAT,
//                  MAPDSN='MY.MAP.FILE'
//MAP.LISTING DD    DSN=NATURAL.NATPROG.LISTDSN1,DISP=SHR
//                  DSN=NATURAL.NATPROG.LISTDSN2,DISP=SHR
```

Indexing Performance Profiles

A NATURAL map data set is identical in format to those produced by the other STROBE Indexers. When you produce a Performance Profile, you can concatenate NATURAL map data sets with each other and with map data sets produced by the other STROBE Indexers. Chapter 2 and Chapter 4 in the STROBE *MVS User's Guide* explain how to produce STROBE reports.

Using the STROBE Advanced Session Management Feature

You may also be interested in measuring multiple jobs at the same time, providing you with a broad picture of what is happening on a system at a given time. The STROBE Advanced Session Management Feature enables you to measure jobs that are executing in different, but related, address spaces at the same time.

For example, you can measure a batch NATURAL program and the related ADABAS region. By reviewing the Performance Profiles for the NATURAL program as well as the ADABAS region, you can then analyze the complete performance impact of the application.

Chapter 3.
The STROBE Performance Profile for NATURAL Regions

This chapter describes the STROBE Performance Profile reports produced when you measure CICS, TSO, or batch address space running NATURAL. A NATURAL environment is an MVS address space that processes NATURAL 4GL programs. NATURAL 4GL programs can execute in teleprocessing environments, such as CICS, TSO, and in batch processing regions. It also discusses DB2-related information in the CPU Usage by NATURAL Program Statement report and the CPU Usage by SQL Statement report, which is the DB2 report produced when a NATURAL statement accesses DB2 using NATURAL for DB2 as the interface. For STROBE to collect CICS information, the STROBE CICS Feature must be invoked during measurement. For STROBE to collect DB2 information, the STROBE DB2 Feature must be invoked. To detect NATURAL for DB2 activity in a CICS address space, both the STROBE DB2 Feature and the STROBE CICS Feature must be invoked. In most cases, STROBE automatically invokes these features.

You can find a complete description of the Performance Profile in Chapter 3 of STROBE *STROBE MVS Concepts and Facilities*.

NATURAL Program Activity

An unoptimized NATURAL 4GL program is interpreted by the NATURAL nucleus programs. Without the STROBE ADABAS/NATURAL Feature, STROBE shows execution and wait time as occurring in these nucleus programs. The STROBE ADABAS/NATURAL Feature determines, when the nucleus is active, the identity of any NATURAL 4GL program and statement being interpreted.

An optimized NATURAL 4GL program contains machine instructions that are directly executable by the CPU. Optimization permits the conversion of many NATURAL language statements directly to machine code, thus requiring much less support from the NATURAL nucleus. The STROBE ADABAS/NATURAL Feature identifies activity by statement number in optimized NATURAL 4GL programs.

STROBE presents the following NATURAL performance data:

- Program name—NATURAL 4GL program name or .NATURAL.
- Library name, database ID, and file number—the library name, ADABAS database ID, and ADABAS file number of the NATURAL program library.
- CICS Transaction ID—if CICS, the name of the CICS transaction under which the NATURAL nucleus executed (The STROBE CICS Feature is required to see CICS data.)
- For NATURAL statements using NATURAL for DB2 to access data, the database request module (DBRM) name associated with the program, its creation date, and any related SQL statement numbers. (The STROBE DB2 Feature is required to see DB2 data.)
- Release compiled—release of the NATURAL nucleus under which the NATURAL 4GL program was compiled.
- Statement number—NATURAL 4GL program statement number.
- Statement text—NATURAL 4GL program statement text (if the program was indexed).

STROBE reports any execution in the NATURAL nucleus that cannot be assigned to a NATURAL 4GL program under the pseudo-4GL name .NATURAL.

STROBE assigns initialization activity in a NATURAL 4GL program that cannot be attributed to a NATURAL statement to the pseudo-activity .INIT, which appears in the STATEMENT NUMBER field.

STROBE assigns activity in an optimized NATURAL 4GL program that cannot be assigned to a specific NATURAL statement number to the pseudo-activity .OPTZ, which appears in the STATEMENT NUMBER field.

Measurement Session Data Report

The Measurement Session Data report (Figure 3-1 on page 3-3) describes the environment during a measurement session. When STROBE measures a job step that is executing NATURAL, the SUBSYSTEM field of the report shows the name and version number of the NATURAL nucleus program.

The format of the SUBSYSTEM field varies, however, depending on the number of different NATURAL releases running in the address space that is being measured. The formats are as follows:

Number of different NATURAL releases	Format
One	NATURAL X.X.X
Two	NATURAL XXX, XXX
Three	NAT XXX,XXX,XXX
More than three	NAT XXX,XXX,XXX+ (In this case, the three highest releases are shown, with the + indicating the presence of additional releases.)

In Figure 3-1 on page 3-3, the SUBSYSTEM field shows that there are two different releases of NATURAL running in the measured address space. The report additionally indicates that the STROBE CICS Feature was invoked in conjunction with the STROBE ADABAS/NATURAL Feature by also showing the name and release level of CICS in the SUBSYSTEM field. If the program measured is using NATURAL for DB2 to access DB2 data and the STROBE DB2 Feature is installed, the name and release level of DB2 would appear in the SUBSYSTEM field.

Choosing Between the Execution and Wait Reports

Wait time for a typical NATURAL 4GL application program is relatively high when there is intensive database I/O activity. Therefore, the value of CPS TIME PERCENT may be much smaller than the WAIT TIME PERCENT value. The STROBE ADABAS/NATURAL Feature attributes wait time for NATURAL 4GL programs running under batch or TSO. For all other programs, you should use the Program Usage by Procedure reports to identify performance improvement opportunities. If you are:

- a NATURAL application programmer, refer to the section titled “Reports That Show NATURAL 4GL Program CPU Usage” on page 3-3
- a NATURAL system administrator, refer to the section titled “Reports That Show NATURAL System CPU Usage” on page 3-10
- interested in examining the wait time for 4GL programs running under batch or TSO, refer to the section titled“Reports That Show Wait Time” on page 3-15.

Figure 3-1. Measurement Session Data Report

```
                                    ** MEASUREMENT SESSION DATA **

------- JOB ENVIRONMENT --------       ------- MEASUREMENT PARAMETERS ------       ----- MEASUREMENT STATISTICS ----

PROGRAM MEASURED   -      DFHSIP       ESTIMATED SESSION TIME  -      10 MIN       CPS TIME PERCENT  -          4.15
JOB NAME           -     CICSRGN       TARGET SAMPLE SIZE   -         20,000       WAIT TIME PERCENT -         95.85
JOB NUMBER         -    JOB05210       REQUEST NUMBER   (A)    -          60       RUN MARGIN OF ERROR PCT -     .61
STEP NAME          -     CICSRGN       FINAL SESSION ACTION    -        QUIT       CPU MARGIN OF ERROR PCT -     .13

                                                                                  TOTAL SAMPLES TAKEN -      20,000
DATE OF SESSION    -    02/23/01       OPTIONS -                          CICS      TOTAL SAMPLES PROCESSED -  20,000
TIME OF SESSION    -    13.49.23                                                  INITIAL SAMPLING RATE - 33.33/SEC
CONDITION CODE     -      C-0000                                                  FINAL SAMPLING RATE   - 33.33/SEC
                                       --------- REPORT PARAMETERS ---------
SYSTEM -             ESA SP4.2.2
DFSMS          -           1.1.0       REPORT RESOLUTION    -       64 BYTES     CPU TIME     -    0 MIN 32.14 SEC
SUBSYSTEM          -    CICS 2.1       SORTSIZE             -        999,999     WAIT TIME    -   10 MIN  7.20 SEC
                NATURAL 224,225        LINE/PAGE                         60      STRETCH TIME -    0 MIN 16.43 SEC
CPU MODEL          -         9021
SMF/SYSTEM ID      -         S4VS       DASD=2.0%  DASDGAP=5                    SRB TIME -        0 MIN  0.24 SEC
                                       ATTR= 0.0%                              SERVICE UNITS                 213
REGION SIZE BELOW 16M  -  6,764K                                               PAGES IN-      0     OUT-       0
REGION SIZE ABOVE  -     65,536K       DATE FORMAT                MM/DD/YYYY   PAGING RATE    -         0.00/SEC
                                       TIME FORMAT                  HH:MM:SS   EXCPS -       2,055       3.13/SEC
PTF LVL- 2.5.0.FS000000/FS000000
STROBE TAPE NUMBER -  000-S92ZZZ

SAMPLE DATA SET  - STROBE.CICSRGN.S001D001
```

Reports That Show NATURAL 4GL Program CPU Usage

Two STROBE reports are available that show the CPU usage for NATURAL 4GL programs and for statements contained in the programs.

NATURAL Program CPU Usage Summary Report

The NATURAL Program CPU Usage Summary report (Figure 3-2 on page 3-4) indicates which NATURAL 4GL programs are responsible for CPU activity. The columns in this report show:

Report Header	Contents
PROGRAM NAME	NATURAL 4GL program name or .NATURAL
LIBRARY	NATURAL program library name.
DBID	ADABAS database ID for the NATURAL program library
FNR	ADABAS file number for the NATURAL program library
TRANID	If CICS or IMS, the name of the transaction under which the NATURAL nucleus executed
CPU TIME PERCENT	
SOLO	CPU activity without any concurrent I/O activity
TOTAL	CPU activity with or without any concurrent I/O activity

In this example, NATURAL 4GL program NATTESTR from program library PRACTICE (identified by ADABAS database ID 006, file number 052), running under CICS transaction DIS1, used 92.90% of the total CPU activity. To determine which statements within program NATTESTR were responsible for the CPU usage, refer to the CPU Usage by NATURAL Program Statement report for program NATTESTR (Figure 3-3 on page 3-5).

Figure 3-2. NATURAL Program CPU Usage Summary Report

```
                                   ** NATURAL PROGRAM CPU USAGE SUMMARY **

PROGRAM NAME    LIBRARY    DBID    FNR    TRANID              % CPU TIME    CPU TIME HISTOGRAM  MARGIN OF ERROR:  2.29%
                                                             SOLO   TOTAL   .00     23.50     47.00     70.50     94.00

.NATURAL                                                      .11     .11   .
LOGON           SYSLIB     006     051                       .05     .05   .
NATTESTR        PRACTICE   006     052                     92.90   92.90   .****************************************
                                                           -----   -----
PROGRAM TOTALS                                             93.07   93.07
```

CPU Usage by NATURAL Program Statement Report

The CPU Usage by NATURAL Program Statement report (Figure 3-3 on page 3-5) shows the distribution of the CPU activity among the statements that make up the NATURAL 4GL programs. The header information for each report shows:

- the NATURAL 4GL program name (PROGRAM)
- the NATURAL program library in which it resides (LIBRARY)
- the ADABAS database ID(DBID) and file number for the program library (FNR)
- if CICS or IMS, the transaction ID under which NATURAL was executed (TRANID)
- the release of NATURAL under which the 4GL program was compiled (RELEASE COMPILED
- if the NATURAL program accessed DB2 data using NATURAL for DB2, the name of the database request module (DBRM) and its creation date
- The global buffer pool name (BPNAME), if the name exists. If a local buffer pool is being used, the field value is the literal *LOCAL*.

Note: If the NATURAL program is using NATURAL for DB2 and is associated with more than one DBRM, the DBRM field will read “MULTIPLE“.

- IF IMS, the module name (MODULE), CSECT name (CSECT) and name of the program status block (PSB) that initiated the request for IMS DL/I services.

Each detail line shows:

- The NATURAL 4GL statement number.
- If present, the SQL statement number. The SQL statement number may be suffixed with a STROBE-generated number “-nn“.

Note: When activity in DB2 SQL statements causes multiple instances of the same NATURAL statement number to appear on the CPU Usage by NATURAL Program Statement report, STROBE indicates the total CPU time used by the NATURAL statement in a line labeled TOTAL. The spike in the histogram (a row of asterisks and pluses) noting a high proportion of activity, appears next to this line.

- The IMS reference number if the Natural statement accesses am IMS database.
- If the program was indexed, the statement text.
- The CPU solo and total time used by each statement.

Reporting INCLUDEd Code

For INCLUDEd code (often referred to as copycode), an asterisk and a number to the right of the statement number indicates the level of nesting. For indexed copycode in the first level of nesting, STROBE reports statement text along with statement number. For copycode in the second level of nesting and higher, STROBE does not report statement text.

STROBE reports the statement number and statement text of the INCLUDE statement that included the copycode. STROBE then prints the copycode object in-line at the point where it was included. This enables you to identify the copycode object and its statement number that are consuming CPU resources.

For example, statement number 2040 in Figure 3-3 includes the copycode statement that follows it. The statement that follows 2040 is at the first level of nesting, as indicated by the *1 to the right of its statement number. Located at statement number 0400 in its own program, this statement has the text INCLUDE NATTESTW. It consumes no CPU time but is important because it INCLUDEs the two copycode statements that follow it and enables the user to identify the nested statements that consume CPU time. The statements that follow statement number 0400 have statement numbers 0100 and 0110 in their own programs. Because they are at the second level of nesting, as indicated by a *2 to the right of their statement numbers, STROBE does not report their statement text. The statements at the second level of nesting consume 1.32% and .06% of CPU.

Note: STROBE prints the INCLUDE statement text even if the program is not using indexing and reports CPU activity for the INCLUDE statement even if it is 0%.

Note: STROBE does not index copycode that is nested more than one level because of a limitation of the NATURAL LIST command, which does not expand copycode beyond the first level.

Figure 3-3. CPU Usage by NATURAL Program Statement Report with INCLUDEd Code

```
                                   ** CPU USAGE BY NATURAL PROGRAM STATEMENT **

PROGRAM - NATTESTR       LIBRARY - PRACTICE   DBID - 00712   FNR - 00309
                         RELEASE COMPILED - 3.1.4  BPNAME - NATGBP31

 STMT     SQL    IMS       STATEMENT                              % CPU TIME   CPU TIME HISTOGRAM MARGIN OF ERROR: 2.40%
NUMBER    STMT  REFNO        TEXT                                SOLO   TOTAL   .00   16.00   32.00   48.00   64.00

0810           00001  CALL FILE 'NATREAD' #CONTROL #IN-REC          .12     .12   .
1980           00002  WRITE (M) #MESSLINE                           .12     .12   .
2010                  INCLUDE NATTESTM                              .00     .00   .
1640*1                REPEAT UNTIL #LOOPCNT GT #LOOPMAX             .78     .78   .
1650*1                COMPUTE #WORKNUM = SQRT(#LOOPCNT)           62.61   62.61   .**************************************
1660*1                ADD 1 TO #LOOPCNT                             .96     .96   .
1670*1                END-REPEAT                                    .06     .06   .
2030                  INCLUDE NATTESTS                              .00     .00   .
0680*1                REPEAT UNTIL #LOOPCNT GT #LOOPMAX             .36     .36   .
0690*1                STACK TOP 10 'TESTIT' 123.4                  2.52    2.52   .*
0700*1                INPUT #WORKNUM1 #WORKCHAR #WORKNUM2         18.97   18.97   .***********
0710*1                ADD 1 TO #LOOPCNT                            1.26    1.26   .
0720*1                END-REPEAT                                    .18     .18   .
2040                  INCLUDE NATTESTO                              .00     .00   .
0400*1                INCLUDE NATTESTW                              .00     .00   .
0100*2                                                             1.32    1.32   .
0110*2                                                              .06     .06   .
2050                  INCLUDE NATTESTA                              .00     .00   .
0390*1                READ SAL-VIEW BY ISN STARTING FROM 10000      .06     .06   .
0410*1                READ SAL-VIEW BY NAME.. SALARY(1) > 100000   2.28    2.28   .*
0420*1                END-READ                                      .12     .12   .
0530*1                EMP. FIND EMP-VIEW WITH NAME = 'SMITH'        .30     .30   .
0540*1                VEH.   FIND SAL-VIEW .. = EMP.PERSONNEL-ID    .66     .66   .
0550*1                END-FIND                                      .06     .06   .
0570*1                ADD 1 TO #LOOPCNT                             .06     .06   .
0710*1                ADD 1 TO #LOOPCNT                             .06     .06   .
1530*1                END TRANSACTION                               .06     .06   .
1650*1                FIND EMPUPD-VIEW WITH CITY='BOSTON'           .06     .06   .
                                                                  -----   -----
PROGRAM  NATTESTR  TOTALS                                        93.04   93.04
```

Reporting DB2 Information

The ADABAS/NATURAL Feature can now report DB2 information obtained by a NATURAL for DB2 program.

Programs that use NATURAL for DB2

When NATURAL programs use NATURAL for DB2 to access data in a DB2 database, STROBE will produce reports that contain both NATURAL and DB2 information. The STROBE ADABAS/NATURAL Feature will produce CPU Usage by NATURAL Program Statement reports, and the STROBE DB2 Feature will produce CPU Usage by SQL Statement reports. To obtain in-depth information on SQL statements called by NATURAL program statements, you can cross-reference the reports: the CPU Usage by NATURAL Program Statement report contains the DBRM name, its creation date, and the SQL statement numbers associated with the program.

The CPU Usage by SQL Statement Report

The CPU Usage by SQL Statement report, Figure 3-5 on page 3-9, shows the distribution of CPU activity among SQL statements within a Procedure, Query, or DBRM and identifies the target SQL that caused the activity. The description section, which is the top half of Figure 3-5 on page 3-9, displays SQL statement text for all target statements. The CPU usage section, which is the bottom half of Figure 3-5 on page 3-9, shows the amount of CPU activity consumed by each executing statement.

In the description section, STROBE lists SQL statements under one of these headings: STATIC, NON-CURSOR SQL; STATIC, CURSOR SQL; DYNAMIC, NON-CURSOR SQL; and DYNAMIC, CURSOR SQL. For STATIC, NON-CURSOR SQL statements, STROBE begins each detail line with the precompiler listing statement number followed by statement type. For all other types of SQL statement numbers, STROBE begins each line with a uniquely generated statement number, which is based on the sequence in which STROBE detects activity for the statement.

In the CPU activity section of the report, STROBE begins each detail line with the precompiler statement number followed by the verb under the heading STATEMENT TEXT. For SQL verbs with target SQL statements, STROBE then displays the generated statement number and the SQL verb of the target statement.

Cross-Referencing the Reports

To get detailed information about SQL statement activity associated with NATURAL program statements, cross-reference the CPU Usage by NATURAL Program Statement report with the CPU Usage by SQL Statement report. To cross-reference the CPU Usage by NATURAL Program Statement report with the CPU Usage by SQL Statement report, find the CPU Usage by SQL Statement report that corresponds to the DBRM named in the CPU Usage by NATURAL Program Statement report. Then locate the SQL text identified in the NATURAL report in the description section of the DB2 report. If STROBE appended the SQL statement number with a dash number, then locate the statement corresponding to this number. Otherwise, locate the statement corresponding to the SQL statement number itself.

You can cross-reference the reports in this way:

- Locate the NATURAL program statement with high CPU activity. On Figure 3-4 on page 3-8, this is NATURAL program statement number 0400.
- Locate the SQL statement number associated with the NATURAL program statement number. On Figure 3-4 on page 3-8, this is SQL statement number 1150. Note that the NATURAL program statement number text is printed the first time the number is printed; for statement 400, this is `STORE WPACI410-DEMO`. Activity that is not associated with an SQL number, such as the 1.15% for NATURAL program statement number 0400, is attributable to initialization or other overhead routines.

- Match the DBRM name on the CPU Usage by NATURAL Program Statement report with the DBRM name on the CPU Usage by SQL Statement report (Figure 3-5 on page 3-9).
- Go to the CPU Usage by SQL Statement report, Figure 3-5 on page 3-9, and in the description section of the report locate the text of the DB2 statement number. You know to look in the STATIC, NON-CURSOR section because the SQL statement number does not have an associated dash number. For statement 1150, the text is
  ```
  INSERT INTO WPACI410.DEMO(NAME,ADDRESS,DATEOFBIRTH,SALARY)
  VALUES(:H,:H,:H,:H)
  ```
- Look in the CPU usage section of the CPU Usage by SQL Statement, Figure 3-5 on page 3-9. The report shows that statement 1150 is an INSERT statement, which uses 3.45% CPU time.

NATURAL program statement number 0480 in Figure 3-4 on page 3-8 also uses a significant amount of CPU time. Because this is a different type of SQL statement, the cross-referencing procedure is somewhat different.

- Locate the SQL statement number associated with this NATURAL program statement number. The SQL statement number is 1187 with an -04 appended to it. The dash number is a uniquely generated STROBE sequence number.
- Go to the CPU Usage by SQL Statement report and in the description section of the report, Figure 3-5 on page 3-9, look up the dash number, which tells you the type of SQL DECLARE statement being used. The report shows that -04 is

  ```
  DECLARE C0050480 CURSOR FOR SELECT COUNT(*),NAME FROM WPACI410.DEMO WHERE
  NAME>=???? GROUP BY NAME
  ORDER BY NAME FOR FETCH ONLY
  ```

- To find the SQL verb associated with this statement go to the CPU usage section of the report and look up the base number 1187 and in the second column match the dash number. The statement text tells you that 1187-04 is an OPEN statement.
- Look up the other SQL statement associated with NATURAL program statement number 480. The CPU Usage by NATURAL Program Statement report, Figure 3-4 on page 3-8, indicates that the SQL statement is 1191-04.
- Go to the CPU Usage by by SQL Statement report, Figure 3-5 on page 3-9, and look up 1191-04. The dash number indicates, as seen previously, a DECLARE statement.
- Go to the CPU Usage section of the report and look up 1191 matching the -04 in the second column. This section of the report tells you that 1191 is a FETCH statement.

For more information about the CPU Usage by SQL Statement report and other DB2 reports collected by the DB2 Feature, such as the Transaction Summary, the SQL CPU Usage Summary, the SQL Wait Time Summary, and the Attribution of CPU Execution Time, see the STROBE *DB2 Feature*.

Figure 3-4. CPU Usage by NATURAL Program Statement Report with DB2 Information

```
                                  ** CPU USAGE BY NATURAL PROGRAM STATEMENT **

 PROGRAM - TDB2YONI       LIBRARY - PRACTICE   DBID - 00712   FNR - 003092
 TRANID  - N313D710       RELEASE COMPILED- 3.1.3   DBRM - TDB2DBR<


 STMT     SQL     IMS       STATEMENT                              % CPU TIME  CPU TIME HISTOGRAM MARGIN OF ERROR:  7.43%
NUMBER    STMT   REFNO        TEXT                                SOLO   TOTAL    .00    1.50    3.00    4.50       6.00

 0290                     FIND WPACI410-DEMO WITH NAME > ' '          .57     .57   .***
 0350                     DELETE                                     1.15    1.15
 0350     1141                                                       1.15    1.15
                                                                    -----   -----
 0350                     TOTALS                                     2.30    2.30   .***************

 0400                     STORE WPACI410-DEMO                        1.15    1.15
 0400     1150                                                       3.45    3.45
                                                                    -----   -----
 0400                     TOTALS                                     4.60    4.60   .******************************

 0440     1169-09         READ WPACI410-DEMO BY NAME                  .57     .57   .***
 0450                     WRITE NAME *COUNTER                        1.15    1.15   .*******
 0480     1187-04         HISTOGRAM WPACI410-DE..E STARTING FROM ' '  1.15    1.15
 0480     1191-04                                                    1.15    1.15
                                                                    -----   -----
 0480                     TOTALS                                     2.30    2.30   .***************

 0490                     DISPLAY  NAME 'NUMBER..S' *NUMBER *COUNTER  1.15    1.15   .*******
 0520                     SELECT ADDRESS                             1.15    1.15   .*******
 0570                     FIND WPACI410-DEMO WITH NAME > ' '           .57     .57
 0570     1233-19                                                      .57     .57
                                                                    -----   -----
 0570                     TOTALS                                     1.15    1.15   .*******
```

Figure 3-5. CPU Usage by SQL Statement Report

```
                                    ** CPU USAGE BY SQL STATEMENT **

DBRM - TDB2DBRM                                CREATED - 23 JUL 02 23:43:28

  STATIC, NON-CURSOR SQL

         797 DELETE FROM WPACI410.DEMO WHERE CURRENT OF C0030270
        1108 UPDATE WPACI410.DEMO SET NAME=:H,ADDRESS=:H,DATEOFBIRTH=:H,SALARY=:H WHERE CURRENT OF C0040620
        1141 DELETE FROM WPACI410.DEMO WHERE CURRENT OF C0050290
        1150 INSERT INTO WPACI410.DEMO(NAME,ADDRESS,DATEOFBIRTH,SALARY) VALUES(:H,:H,:H,:H)
        1247 UPDATE WPACI410.DEMO SET NAME=:H,ADDRESS=:H,DATEOFBIRTH=:H,SALARY=:H WHERE CURRENT OF C0050570
        1280 UPDATE WPACI410.DEMO SET NAME=:H,ADDRESS=:H,DATEOFBIRTH=:H,SALARY=:H WHERE CURRENT OF C0050620
        1313 DELETE FROM WPACI410.DEMO WHERE CURRENT OF C0060360
        1322 INSERT INTO WPACI410.DEMO(NAME,ADDRESS,DATEOFBIRTH,SALARY) VALUES(:H,:H,:H,:H)
        1419 UPDATE WPACI410.DEMO SET NAME=:H,ADDRESS=:H,DATEOFBIRTH=:H,SALARY=:H WHERE CURRENT OF C0060570
  STATIC, CURSOR SQL

          01 DECLARE C0030600 CURSOR FOR SELECT NAME,ADDRESS,DATEOFBIRTH,SALARY FROM WPACI410.DEMO WHERE NAME>???
                ? FOR UPDATE OF NAME,ADDRESS,DATEOFBIRTH,SALARY
          02 DECLARE C0040360 CURSOR FOR SELECT NAME,ADDRESS,DATEOFBIRTH,SALARY FROM WPACI410.DEMO WHERE NAME>???
                ?
          03 DECLARE C0040620 CURSOR FOR SELECT NAME,ADDRESS,DATEOFBIRTH,SALARY FROM WPACI410.DEMO WHERE NAME>???
                ? FOR UPDATE OF NAME,ADDRESS,DATEOFBIRTH,SALARY
          04 DECLARE C0050480 CURSOR FOR SELECT COUNT(*),NAME FROM WPACI410.DEMO WHERE NAME>=???? GROUP BY NAME
                ORDER BY NAME FOR FETCH ONLY
          05 DECLARE C0060040 CURSOR FOR SELECT COUNT(*),NAME FROM WPACI410.DEMO WHERE NAME>=???? GROUP BY NAME
                ORDER BY NAME FOR FETCH ONLY
                                    .
                                    .
                                    .

 STMT   STATEMENT                                          % CPU TIME    CPU TIME HISTOGRAM  MARGIN OF ERROR:  7.43%
NUMBER    TEXT                                           SOLO   TOTAL   .00      1.50      3.00      4.50      6.00

   806 INSERT                                            3.45    3.45   .************************
   978 INSERT                                            2.30    2.30   .***************
  1150 INSERT                                            3.45    3.45   .************************
  1322 INSERT                                            2.30    2.30   .***************
                                    .
                                    .
                                    .
  1191 FETCH                  04 DECLARE                 1.15    1.15   .*******
  1233 FETCH                  19 DECLARE                  .57     .57   .***
  1266 FETCH                  10 DECLARE                  .57     .57   .***
  1299 FETCH                  11 DECLARE                 1.72    1.72   .***********
  1363 FETCH                  05 DECLARE                 1.15    1.15   .*******
  1383 FETCH                  12 DECLARE                 1.15    1.15   .*******
  1405 FETCH                  17 DECLARE                  .57     .57   .***
  1438 FETCH                  18 DECLARE                  .57     .57   .***
                                                        -----   -----
DBRM - TDB2DBRM                          TOTALS        57.47   57.47
```

Reporting IMS Information

When NATURAL programs access data in an IMS database, STROBE will produce reports that contain both NATURAL and IMS information. The STROBE ADABAS/NATURAL Feature will produce CPU Usage by NATURAL Program Statement reports, and the STROBE IMS Feature will produce various types of CPU usage reports. To obtain in-depth information on IMS databases called by NATURAL program statements, you can cross-reference the reports by using the IMS REFNO column values that appear to the right of the SQL STMT column shown in Figure 3-6. The report also provides the IMS module, CSECT and Program Status Block (PSB) names. You can take this data and then examine IMS Feature reports to determine what IMS activity related to the Natural activity.

Note: You must have the IMS Feature installed at your site to obtain IMS data in your CPU Usage by NATURAL Program Statement report.

Figure 3-6. CPU Usage by Natural Program Statement Report Showing IMS Information

```
                                  ** CPU USAGE BY NATURAL PROGRAM STATEMENT **

 PROGRAM - TDB2YONI       LIBRARY - PRACTICE   DBID - 008   FNR - 052
 DBRM    - ILM00001       CREATED - 23 JUL 02 23:43:28
 IMS: MODULE - STRIL140 CSECT - ***LI401  PSB - STRIL140

 STMT     SQL     IMS       STATEMENT                          % CPU TIME  CPU TIME HISTOGRAM MARGIN OF ERROR:  7.43%
NUMBER    STMT   REFNO        TEXT                            SOLO   TOTAL    .00    1.50    3.00    4.50       6.00

 0810             000001   FIND EMP-VIEW WITH NAME =  'SMITH'        10.12   10.12   .*****
 1980             000002   FIND SAL-VIEW WITH PERSONNEL-ID = EMP       .12     .12   .
 2115    1187-04           WRITE NAME *COUNTER                        .02     .02   .
 2120             000003   FIND EMP-VIEW WITH NAME =  'SMITH'        10.12   10.12   .
 2340             000004   FIND SAL-VIEW WITH PERSONNEL-ID = EMP       .12     .12   .
 3124    1187-04           WRITE NAME *COUNTER                        .02     .02   .
                                                                     -----   -----
 PROGRAM NATTESTR  TOTALS                                           20.52   20.52
```

Figure 3-7 shows an example of an IMS report that you can cross reference to see what IMS databases were called by NATURAL program statements.

Figure 3-7. CPU Usage by DL/I Request Report for Batch Regions

```
                                        ** CPU USAGE BY DL/I REQUEST **

 MODULE      - STRIL140
 SECTION     - ***L1401
 PCB         - STRIL140

                                                CALL PARAMETERS

     REF      PCB   PCB NAME   RESOURCE
     NUMBER   TYPE  OR LABEL   NAME          SSA DATA

     000001   DB    LOANPCB    STRIDLON
     000002   DB    LOANPCB    STRIDLON

                                              FUNCTION ACTIVITY

             PROCEDURE   REQUEST     DL/I    REF.      CPU TIME PERCENT       CPU TIME HISTOGRAM  MARGIN OF ERROR:  3.33%
 LINE NO.    NAME        LOCATION    FUNC    NUMBER     SOLO     TOTAL       .00      7.00     14.00     21.00     28.00

     17      CALL        000001CC    GHU     000001    26.04     27.07       .************************************+
     19      CALL        00000256    GHN     000002     2.65      2.65       .***
                                                       -----     -----
 TOTALS                                                28.69     29.72
```

Reports That Show NATURAL System CPU Usage

Several STROBE reports can be generated that show CPU usage and activity within a NATURAL environment.

Program Section Usage Summary Report

The Program Section Usage Summary report (Figure 3-8) shows the distribution of CPU time used by each active control section of each active module in the NATURAL environment. NATURAL nucleus modules and IBM system modules are compressed and shown as pseudo-sections under the pseudo-module .SYSTEM.

This report can help you focus your investigation by showing whether NATURAL nucleus programs or user programs are responsible for the most CPU activity. If NATURAL user programs are the heaviest CPU users, refer to the NATURAL 4GL Program reports to identify performance improvement opportunities. Otherwise, refer to the Program Usage by Procedure reports.

In this example, the NATURAL nucleus programs shown under module name .SYSTEM and section name .NATNUC account for 36.94% of the execution time. Therefore, you

should look at the Program Usage by Procedure report (Figure 3-9 on page 3-12) for .NATNUC to determine what caused this activity.

Figure 3-8. Program Section Usage Summary Report

```
                                    ** PROGRAM SECTION USAGE SUMMARY **

 MODULE  SECTION  16M SECT  FUNCTION                 CPU TIME PERCENT       CPU TIME HISTOGRAM  MARGIN OF ERROR:  3.40%
   NAME     NAME  <,> SIZE                            SOLO     TOTAL        .00     11.00     22.00     33.00     44.00

.SYSTEM  .ADABAS            ADABAS SYSTEM SERVICES     2.17      2.53        .**
.SYSTEM  .CICS              CICS SYSTEM SERVICES      38.26     42.59        .************************************+++++
.SYSTEM  .COBLIB            COBOL LIBRARY SUBROUTINE    .48       .48        .
.SYSTEM  .IOCS              DATA MANAGEMENT SERVICES    .12       .12        .
.SYSTEM  .NATNUC            NATURAL NUCLEUS SERVICES  34.78     36.94        .***********************************++
.SYSTEM  .NUCLEUS           MVS NUCLEUS                6.38      6.86        .***+
.SYSTEM  .SVC               SUPERVISOR CONTROL          .36       .36        .
.SYSTEM  .VSAM              VIRTUAL STORAGE ACC METH    .12       .12        .
.SYSTEM  .VTAM              VIRTUAL TELECOM ACC METH    .24       .24        .
                                                       -----     -----
.SYSTEM  TOTALS             SYSTEM SERVICES           82.91     90.24

ACFAEDFH          <  24576                             1.56      1.68        .*
ACFAEICX          <  46384                              .36       .48        .
CCATCVT1          <  31104                             4.69      5.05        .**+
CJTBLACC          <   3504                              .12       .12        .
DMNLPIS1          < 113792                              .60       .72        .
IGC0023D          <  16672  SUPERVISOR SERVICES        1.56      1.56        .*
PXSCRTRD          <  12088                              .12       .12        .
                                                       -----     -----
PROGRAM DFHSIP    TOTALS                              91.94    100.00
```

Program Usage by Procedure Report

The Program Usage by Procedure reports present a detailed accounting of CPU time spent by each procedure within each program that was active during a measurement session. They include both application programs and system modules, and the two types of reports look slightly different. For application programs, STROBE displays the module name, control section name, and starting location. If the control section was indexed, STROBE shows the line number and procedure name as well.

For system modules, STROBE displays the module and control section name and a function descriptor. Both types of reports show the solo and total CPU time used by each control section within the module. (For a complete description of this report, see Chapter 3 in STROBE *STROBE MVS Concepts and Facilities*.)

Activity in System Modules

The second type of Program Usage by Procedure report details activity in system modules. The example in Figure 3-9 on page 3-12 shows the NATURAL nucleus programs that support the NATURAL 4GL programs in this CICS environment. STROBE identifies activity in NATURAL nucleus programs with the pseudo-module name .NATNUC.

Of particular interest in this example are control section NATADRI of module NAT224RE, which accounts for 6% of CPU activity, and section NATARI2 of module NAT224RE, which accounts for 4.8%. To determine which NATURAL 4GL programs and statements caused the activity, examine the Attribution of CPU Execution Time report (Figure 3-10 on page 3-13).

Figure 3-9. Program Usage by Procedure Report

```
                                    ** PROGRAM USAGE BY PROCEDURE **

MODULE - NAT224RE                              SECTION -.NATNUC   NATURAL NUCLEUS SERVICES

MODULE   SECTION        FUNCTION            INTERVAL  CPU TIME PERCENT       CPU TIME HISTOGRAM  MARGIN OF ERROR:  3.40%
  NAME     NAME                           LENGTH   SOLO     TOTAL        .00      2.00      4.00      6.00      8.00

         NATADRI   NATURAL ADABAS SUPPORT       3940   5.78      6.02        .*****************************++
         NATADX2   NATURAL ADABAS SUPPORT       7144    .48       .48        .**
         NATARI2   NATURAL ARITHMETIC FUNC     15728   4.69      4.81        .************************+
         NATBPMG   NATURAL BUFFER POOL MNGR     4456   1.68      1.81        .********+
         NATCCR    NATURAL NUCLEUS               750    .12       .12        .
         NATDB     NATURAL NUCLEUS               970    .48       .84        .**++
         NATDISP   NATURAL NUCLEUS              9872   1.81      1.81        .*********
         NATDYPR   NATURAL PARM EVALUATION      9274    .12       .12        .
         NATEVP    NATURAL NUCLEUS              1798    .24       .24        .*
         NATINO2   NATURAL I/O PROCESSING      24840    .96      1.32        .****++
         NATIOC    NATURAL I/O PROCESSING       3164    .48       .48        .**
         NATMASH   NATURAL NUCLEUS               120    .24       .24        .*
         NATMISC   NATURAL NUCLEUS              5526    .72       .72        .***
         NATMUP2   NATURAL WORKFILE SUPPORT     2856    .36       .36        .*
         NATNSR1   NATURAL/ADABAS CALL          4012   1.08      1.08        .*****
         NATPRX2   NATURAL CPU                  6840   3.97      4.21        .*******************++
         NATPTS    NATURAL I/O PROCESSING      10740   2.17      2.29        .**********+
         NATRUN1   NATURAL NUCLEUS              1332   4.21      4.21        .**********************
         NATRUN4   NATURAL NUCLEUS              6852   2.05      2.05        .**********
         NATSCP    NATURAL STORAGE CNTL PGM     5724    .24       .36        .*
         NATUB     NATURAL NUCLEUS              7032    .12       .12        .
         NAT327O   NATURAL 3270 TERM CNVRTR     1912    .12       .12        .
         NAT3279   NATURAL 3279 TERM CNVRTR     2276    .12       .24        .+
         CMSTART   NATURAL INITIALIZATION      20112    .84      1.08        .****+
         NATCODEC                              1048   1.32      1.44        .******+
         NSPMWRIT                               846    .24       .24        .*
         SPOOLET                               1576    .12       .12        .
                                                      -----     -----
         .NATNUC   TOTALS                             34.76     36.93
```

Attribution of CPU Execution Time Report

The Attribution of CPU Execution Time report (Figure 3-10 on page 3-13) identifies the sites of invocation of attributed modules. Use this report to identify which NATURAL 4GL statements caused activity in a NATURAL nucleus module or system service routine.

Note: Activity in DB2 system modules will not show attribution to a NATURAL 4GL statement.

Header Lines

The report header identifies the invoked routine, showing:

- its pseudo-module, module, and control section name (when available)
- a function descriptor for either the control section or the module

Detail Lines

Each report detail line for a NATURAL nucleus program identifies an invoker of the module and displays:

- a transaction name, if the NATURAL nucleus is running under a CICS transaction
- the name of the NATURAL 4GL program
- for copycode, the name of the object that included the code
- the NATURAL 4GL program statement number
- the NATURAL program library name
- the ADABAS database ID and file number of the NATURAL program library

- the solo and total CPU time spent on behalf of the invoker

Note: The Attribution of CPU Execution report combines CPU activity for individual copycode objects that the program includes more than once. The CPU activity shown is the sum of the activity for the copycode object. In addition, the NATINCLU column identifies the most immediate includer of the copycode. This column lists only the most immediate includer of the copycode.

Total Line

The total line shows the total time attributed to the invokers of the module. It may be less than the time shown in the Program Usage by Procedure or Most Intensively Executed Procedures reports because STROBE cannot always identify an invoker of a service routine.

Figure 3-10. Attribution of CPU Execution Time Report

```
                                        ** ATTRIBUTION OF CPU EXECUTION TIME **

.NATNUC  NAT227BA NATARI2   NATURAL ARITHMETIC FUNC
-------------------------WAS INVOKED BY--------------------   -----------------VIA----------------------  CPU TIME %
XACTION  NAT4GLNM  NATINCLU    STMT #  LIBRARY   DBID     FNR                                            SOLO  TOTAL

         NATTESTR  NATTESTA 0410       PRACTICE   006     052                                              .33   .33
         NATTESTR  NATTESTM 1640       PRACTICE   006     052                                              .60   .60
         NATTESTR  NATTESTM 1650       PRACTICE   006     052                                            60.53 60.53
         NATTESTR  NATTESTM 1660       PRACTICE   006     052                                              .44   .44
         NATTESTR  NATTESTO 0390       PRACTICE   006     052                                              .11   .11
         NATTESTR  NATTESTS 0680       PRACTICE   006     052                                              .27   .27
         NATTESTR  NATTESTS 0710       PRACTICE   006     052                                              .55   .55
                                                                                                          ----- -----
                                                                                                          62.83 62.83
```

Transaction Reports for CICS Address Spaces

When you measure an online subsystem, the transaction reports show you a breakdown of CPU activity by transaction name. These reports are helpful when:

- You want to see the transaction counts and service time statistics.
- You are interested in the program activity within transactions.
- You want to improve the performance of the CICS system in which your NATURAL programs execute.

Transaction reports in the STROBE Performance Profile vary according to the type of address space measured and the STROBE Features that were installed and active during the measurement session.

STROBE presents transaction reports for CICS transactions executing NATURAL in the STROBE CICS Performance Supplement.

Pseudo-Transactions

When the STROBE CICS Feature detects activity in the main CICS task and neither a user transaction nor a CICS transaction is in control, the STROBE CICS Feature, assuming that CICS is performing activities on its own behalf, assigns that time to a pseudo-transaction named .CICS.

When STROBE identifies activity in a task other than the main CICS task, STROBE generates a pseudo-transaction name beginning with a "." and followed by the first seven characters of the name of the load module that initiated execution in the task.

Transaction Summary Report

When STROBE produces the CICS Performance Supplement, the Transaction Summary report (for a CICS address space) (Figure 3-11) shows the distribution of service time and CPU usage among transactions that were active during a measurement session. In this report, the transaction DIS0 shows a transaction count of 499 and 35.74% of CPU time is attributed to it.

When STROBE identifies activity in a CICS transaction, the transaction count, message rates, service time statistics, and the name of the initial program appear in the detail line. When STROBE uses a pseudo-transaction name, it leaves these values blank.

The Transaction Summary report aggregates all CPU execution for a NATURAL nucleus executing under a CICS transaction and shows the name of that transaction in the NAME column. If the NATURAL nucleus is:

- loaded from CICS-managed storage, the INITIAL PROGRAM column shows the nucleus' program name
- loaded from the operating system using the NATURAL NCILOAD module (OS LOADED), the INITIAL PROGRAM column shows the NCILOAD module name

Figure 3-11. Transaction Summary Report

** TRANSACTION ACTIVITY **

TRANSACTION SUMMARY

NAME	CLASS	FUNCTION	TRANS-ACTION COUNT	MESSAGES PER SECOND	TOTAL CPU TIME PERCENT	SERVICE TIME (SEC) MEAN	85 PC	MEDIAN	15 PC	INITIAL PROGRAM
.DFHSTSK					.12	.00	--	--	--	
ACFT-X			11	0.00	.12	.00	--	--	--	ACFAETSP
ACF1			6	0.01	.00	.07	--	--	--	ACFAEUSP
CATD-X			11	0.00	.00	.00	--	--	--	DFHZATD
CLS1-X		ISC/MRO LU SERVICE MODEL	4	0.00	.00	.00	--	--	--	DFHLUP
CQRY-X			3	0.00	.00	.00	--	--	--	DFHQRY
CRSQ-X		INTERSYSTEM COMM.	1	0.00	.00	.00	--	--	--	DFHCRQ
CSGM-X		GOOD MORNING	5	0.00	.00	.00	--	--	--	ACFAEUSP
CSKC		TRANSACTION DISPATCHER			.48					
CSKP-X		KEYPOINT FOR RESTART	10	0.00	.00	.00	--	--	--	DFHAKP
CSM3		ISC/MRO QUEUE MODEL	36	0.05	.12	.06	.10	.07	.03	DFHMIR
CSSY		JOURNAL KICK-OFF PRGM			.60					
CSTP		TERMINAL CONTROL TASK			6.14					
DIS0			499	0.76	35.74	1.07	1.46	.71	.33	NCX224PC
LINK			2	0.00	.12	.02	--	--	--	FEPLINK
LOGO-X			2	0.00	.00	.00	--	--	--	ACFAEUSP
NATP-X			24	0.00	4.81	.00	--	--	--	NCX224PC
NAT2			8	0.01	.12	.00	--	--	--	NCI224FP
ORD0			279	0.42	20.82	.68	1.25	.62	.20	NCX224PC
ORD1-X	6		3	0.00	28.16	.00	--	--	--	PXPCGNAT
ORD1-X	7		5	0.00	2.65	.00	--	--	--	PXPCGNAT
TOTAL--		INTERACTIVE TRANSACTIONS	830	1.26	64.14	.85	1.30	.65	.36	
TOTAL--		OTHER TRANSACTIONS	89	0.14	35.86					

Transaction Activity: CPU Usage by Control Section Report

The CPU Usage by Control Section report shows, for each transaction that was active during the measurement session, the CPU time spent in each control section within each active module that was invoked to process that transaction.

The CICS example (Figure 3-12), for transaction DIS0, shows that most of the CPU activity was caused by the NATURAL nucleus programs. In this case, the NATURAL nucleus was OS LOADED, so the activity appears under the OS LOADED module name (NAT224RE) rather than the CICS-defined initial program name (NCX224PC) for transaction DIS0, shown in the Transaction Summary report (Figure 3-11 on page 3-14).

Figure 3-12. CPU Usage by Control Section Report

```
                                CPU USAGE BY CONTROL SECTION FOR TRANSACTION DISO
-------------------------------------------------------------------------------------------------------------------
                                  SECTION                                CPU TIME USED, PERCENT
MODULE    ------------------------------------------- ------------------------------------------------------------
NAME      NAME     COMPRESSED FUNCTION                  SOLO  TOTAL    .00      4.50      9.00      13.50     18.00
                                                                         '         '         '         '         '
.ADABAS   ADABAS              ADABAS SYSTEM SERVICES    1.32   1.56    .**+
.CICS     DFHCMP   DFHCMPDY   CICS MONIT COMPAT INT     1.08   1.08    .**
.CICS     DFHEIP               EXEC(CMD-LEVEL)INT PGM    .36    .36    .
.CICS     DFHEPC               EXEC INTERFACE-PGM CNTL   .12    .12    .
.CICS     DFHERM               RESOURCE MGR INT(RMI)MOD  .12    .12    .
.CICS     DFHETS               EXEC INTERFACE-TEMP STOR  1.20   1.56    .**+
.CICS     DFHKCP   DFHKCP      TRANS MNGR BOOTSTRP ROUT   .60    .60    .*
.CICS     DFHPCP               PROGRAM CONTROL PROGRAM   .36    .36    .
.CICS     DFHSCP   DFHSCP      STORAGE CONTROL PGM       5.67   6.59    .***********++
.CICS     DFHTMP               TABLE MANAGER PGM         2.43   2.75    .*****+
.CICS     DFHTSP               TEMPORARY STOR CNTL PGM   3.03   3.47    .******+
.NATNUC   NAT224RE NATADRI     NATURAL ADABAS SUPPORT    8.42   9.51    .******************+++
.NUCLEUS  IEAVEPST             POST SERVICE              2.17   2.29    .****+
.NUCLEUS  IEAVESCO             SCHEDULE SERVICE           .24    .36    .
.NUCLEUS  IEAVTSFR             SETFRR SERVICE             .24    .24    .
.SVC      SVC 011              TIME                       .12    .12    .
ACFAEDFH                                                  .60    .72    .*
ACFAEICX                                                  .24    .36    .
CCATCVT1                                                  .72    .72    .*
DMNLPIS1                                                  .60    .72    .*
IGC0023D                                                 2.05   2.05    .****
                                                        ------ ------
  TOTAL                                                 31.77  35.74
```

Reports That Show Wait Time

The Performance Profile shows wait time in NATURAL environments in two reports, the Wait Time by Module report and the Attribution of CPU Wait Time report. The performance data in these reports can help you improve the performance of the NATURAL batch regions that show wait.

The ADABAS/NATURAL Feature attributes wait time only for NATURAL 4GL programs running under TSO or batch, not for those running under CICS.

Wait Time by Module Report

The Wait Time by Module report (Figure 3-13 on page 3-16) shows the modules that are sites of wait. In this example of a NATURAL batch region, a majority of the wait time occurs in SVC 001, the Wait SVC. This is normal for a NATURAL environment, which often waits for I/O in the database region.

Figure 3-13. Wait Time by Module Report

```
                                        ** WAIT TIME BY MODULE **

 MODULE   SECTION COMPRESSED   FUNCTION                        % RUN TIME    RUN TIME HISTOGRAM  MARGIN OF ERROR:  1.42%
  NAME      NAME    SECTION                                   PAGE   TOTAL   .00     14.00     28.00     42.00     56.00

 .ADABAS  ADALNK   ADABAS   ADABAS SYSTEM SERVICES             .00    5.03   .+++
 .IOCS    IGG019AQ          QSAM GET  NEXT BUFFER              .00     .23   .

 .SVC     SVC 001           WAIT                              .00   55.44   .++++++++++++++++++++++++++++++++++++++++++++++++
 .SVC     SVC 008           PROGRAM MANAGER/LOAD              .00     .40   .
 .SVC     SVC 009           PROGRAM MANAGER/DELETE            .00     .02   .
 .SVC     SVC 013           TERMINATION                       .00     .02   .
 .SVC     SVC 018           BLDL/FIND                         .00     .02   .
 .SVC     SVC 019           OPEN                              .00     .02   .
 .SVC     SVC 020           CLOSE                             .00     .02   .
 .SVC     SVC 056           RESOURCE MANAGER/ENQUEUE          .00     .02   .
 .SVC     SVC 111           JES2/JES3                         .00     .08   .
                                                            -----   -----
 .SVC     TOTALS            SUPERVISOR CONTROL                .00   56.04
                                                            -----   -----
 PROGRAM NAT227BA TOTALS                                     .00   61.30
-
```

Attribution of CPU Wait Time Report

The Attribution of CPU Wait Time report (Figure 3-14 on page 3-17) identifies the sites of invocation of system routines. When wait time is attributed to a service routine, the report header displays the invoked routine, showing:

- its pseudo-module, module, and control section name (when available)
- a function descriptor for either the control section or the module

Each report detail line for a service routine identifies an invoker and displays:

- A transaction name, if the NATURAL nucleus is running under a CICS transaction.
- The NATURAL program name (or .NATURAL).
- For copycode, the name of the object that included the code.
- Its NATURAL statement number. For DB2 activity, STROBE shows the DB2 SQL statement number.
- The NATURAL program library name.
- The ADABAS database ID and file number of the NATURAL program library.
- Page wait and total wait time caused by the invoker.

In addition, the Attribution of CPU Wait Time Report:

- Combines wait time for individual copycode that the program includes more than once.
- In the NATINCLU column, displays the most immediate includer of the copycode. If the statement is nested, this column lists the name of the innermost copycode object.

Figure 3-14. Attribution of CPU Wait Time Report

```
                                        ** ATTRIBUTION OF CPU WAIT TIME **

.SVC     SVC 001            WAIT
--------------------------WAS INVOKED BY--------------------   -----------------VIA---------------------- WAIT TIME %
XACTION  MODULE   SECTION    RETURN   LINE   PROCEDURE NAME       MODULE   SECTION  FUNCTION                 PAGE  TOTAL

         .NATNUC  NAT227BA NATOS      NATURAL MVS INTERFACE       SVC 020           CLOSE                     .00   .04

XACTION  NAT4GLNM  NATINCLU    STMT #  LIBRARY   DBID     FNR

         .NATURAL                                                                                             .00   .76
         LOGON              3110       SYSLIB     006     051                                               .00   .06
         LOGON              4520       SYSLIB     006     051                                               .00   .02
         NATTESTR           0810       PRACTICE   006     052                                               .00   .27
         NATTESTR  NATTESTA 0270       PRACTICE   006     052                                               .00   .91
         NATTESTR  NATTESTA 0390       PRACTICE   006     052                                               .00   .99
         NATTESTR  NATTESTA 0410       PRACTICE   006     052                                               .00 39.98
         NATTESTR  NATTESTA 0530       PRACTICE   006     052                                               .00  3.38
         NATTESTR  NATTESTA 0540       PRACTICE   006     052                                               .00  5.70
         NATTESTR  NATTESTA 0700       PRACTICE   006     052                                               .00   .04
         NATTESTR  NATTESTA 1480       PRACTICE   006     052                                               .00   .42
         NATTESTR  NATTESTA 1520       PRACTICE   006     052                                               .00   .08
         NATTESTR  NATTESTA 1530       PRACTICE   006     052                                               .00   .36
         NATTESTR  NATTESTA 1650       PRACTICE   006     052                                               .00   .13
         NATTESTR  NATTESTA 1670       PRACTICE   006     052                                               .00   .25
         NATTESTR  NATTESTA 1680       PRACTICE   006     052                                               .00   .53
         NATTESTR  NATTESTA 1690       PRACTICE   006     052                                               .00   .82
                                                                                                           ----- -----
                                                                                                            .00 55.44
```

Chapter 4.
The STROBE Performance Profile for ADABAS Regions

An ADABAS region is an MVS address space that processes requests to access ADABAS databases from teleprocessing regions such as CICS and TSO, and batch processing programs. Requests to move data from and to the caller's address space are made through the ADABAS SVC.

When STROBE measures an ADABAS region, the STROBE Performance Profile for that region shows:

- CPU activity in ADABAS modules and in user exit routines
- function descriptors for the ADABAS routines
- CPU usage attributed to callers within the user regions that caused the activity, identified:
  - for a batch job, by the job name
  - for a TSO region, by the TSO user ID
  - for a CICS region, by the logical terminal name

When STROBE measures a 3GL program, the STROBE Performance Profile reports CPU wait time for the ADALNK module back to the 3GL (Assembler, PL/I, FORTRAN or COBOL) program that called it.

This chapter describes a Performance Profile of an ADABAS region and provides examples of the key reports.

Measurement Session Data Report

The Measurement Session Data report describes the environment during a measurement session. This example (Figure 4-1 on page 4-2) shows a busy ADABAS region. The SUBSYSTEM field of the report shows the version number of ADABAS in the measured region. One or more CPUs were active during 8% of the 5-minute measurement session. The ADABAS databases were also relatively busy, averaging more than 32 accesses per second.

Figure 4-1. Measurement Session Data Report for ADABAS Regions

```
                                        ** MEASUREMENT SESSION DATA **

-------- JOB ENVIRONMENT --------      ------- MEASUREMENT PARAMETERS ------      ----- MEASUREMENT STATISTICS ----
PROGRAM MEASURED   -      ADARUN       ESTIMATED SESSION TIME  -      20 MIN      CPS TIME PERCENT-             8.00
JOB NAME           -     ADABAS12      TARGET SAMPLE SIZE      -     100,000      WAIT TIME PERCENT -          92.00
JOB NUMBER         -     STC04859      REQUEST NUMBER (Q) -                72      RUN MARGIN OF ERROR PCT  -     .61
STEP NAME   -            ADABAS12      FINAL SESSION ACTION     -       QUIT      CPU MARGIN OF ERROR PCT -      .13

DATE OF SESSION    -    02/13/97                                                  TOTAL SAMPLES TAKEN -       25,979
TIME OF SESSION    -    17.09.14      --------- REPORT PARAMETERS ---------      TOTAL SAMPLES PROCESSED -   25,979
CONDITION CODE     -      C-0000                                                 INITIAL SAMPLING RATE- 100.00/SEC
                                        REPORT RESOLUTION     -       64 BYTES   FINAL SAMPLING RATE -  100.00/SEC
SYSTEM -             ESA SP4.3.0       SORTSIZE              -       999,999
DFSMS            -         1.1.0       LINES/PAGE            -            60
SUBSYSTEM        -     ADABAS 5.2
CPU MODEL        -      3090-600S      DASD=2.0%  DASDGAP=5                   SESSION TIME -     5 MIN 22.78 SEC
SMF/SYSTEM ID             -   SESA                                             CPU TIME  -        0 MIN 27.80 SEC
                                        DATE FORMAT                MM/DD/YYY   WAIT TIME    -     4 MIN 24.74 SEC
REGION SIZE BELOW 16M -    1,088K      TIME FORMAT                 HH:MM:SS    STRETCH TIME -     0 MIN 30.59 SEC
REGION SIZE ABOVE -       32,768K
                                                                               SRB TIME  -        0 MIN  0.24 SEC
                                                                               SERVICE UNITS                  125
PTF LVL- 2.2.0.FS000000/FS000000                                               PAGES IN-         418      OUT-   1
STROBE TAPE NUMBER -  000-$$$$$$                                               PAGING RATE  -             1.30/SEC
                                                                               EXCPS -          10,423   32.29/SEC

SAMPLE DATA SET  - STROBE.ADABAS12.S001D001
```

Time Distribution and Resource Demand Distribution Reports

The Time Distribution of Activity Level report (Figure 4-2) and the Resource Demand Distribution report (Figure 4-3 on page 4-3) show the level of all resource utilization in the ADABAS region during the measurement session. The reports show the execution of two ADABAS tasks and the I/O activity of the ADABAS data sets. These data sets include the databases and the associator, work, and log data sets. In this example, the associator data set, DDASSOR1, which contains access information for the databases, and DDDATAR1, the database data set, are the most heavily accessed.

Figure 4-2. Time Distribution of Activity Level Report for ADABAS Regions

```
                                        ** TIME DISTRIBUTION OF ACTIVITY LEVEL **

TASK OR   RESOURCE   N X 10 PLUS OR MINUS 5 IS PERCENT OF FULL UTILIZATION  * IS GREATER THAN 95%   - IS LESS THAN 5%
 DDNAME          .-------------------------------------------------------------------------------------------------.
ADARUN    CPU    .   21 21-12--12- 121 12--1-1112-1211-111131--1-1--11 131--1111111--11111 1111111112----1 - --      .
ADRIVSUB  CPU    .    -- ----1- ---  --  -- ---- -- -1- -1- ------- --  ---------------- --- -- - --------- -         .
DDDATAR1  3380   .    17 171153 373  25  52 4444283-253-284-43335441-47  43333744455-265657 435544262531324-2 1-   .
DDASSOR1  3380   .   -6 16  72-571  73  9- 63-2671 64 -55 -91 -- 1- 74  91 275 15655983434  -455666371 -              .
DDWORKR1  3380   .    1  1 - 1  -1-  2   2  1 ---1  2-  1 - 1-  - -  1   2  -1-  - --1-111- -  -- 1- 1    -    -   .
DDPLOGR1  3380   .                                                                                  -    -   .
              0----0----1----1----2----2----3----3----4----4----5----5----6----6----7----7----8----8----9----9----*.
              0----5----0----5----0----5----0----5----0----5----0----5----0----5----0----5----0----5----0----5----*.
              START RUN                PERCENT OF ALLOCATED RUN TIME                                    END RUN
```

The I/O activity for each of the data sets in the ADABAS region seldom reaches levels as high as 80 or 90 percent. This lack of high I/O activity indicates that you probably can improve performance by increasing the overlap of data set access. The Resource Demand Distribution report indicates that the ADABAS data sets DDDATAR1 and DDASSOR1 incur wait time of around 25% of total run time and show high solo execution time percentages. These values also hint that the lack of data set overlap could be a problem. The I/O Facility Utilization Summary report for ADABAS Regions (Figure 4-8 on page 4-5) indicates where these data sets are located and displays their I/O access. This report can show you if performance can be improved by increasing data set overlap.

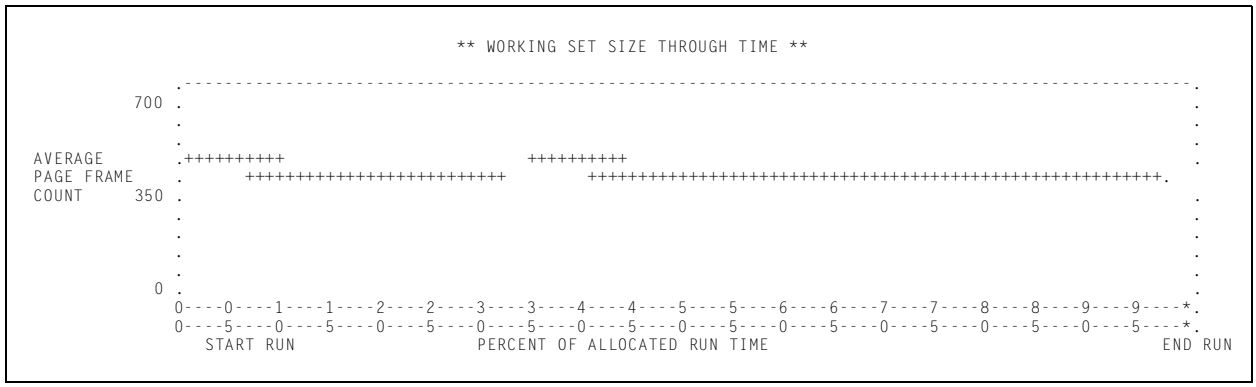
Figure 4-3. Resource Demand Distribution Report for ADABAS Regions

TASK OR DDNAME	RESOURCE	---- PERCENT OF RUN TIME ----- SERVICED BY CPU	SERVICED BY I/O	SERVICED BY EITHER	----- PERCENT OF RUN TIME SPENT ------- SOLO IN CPU	SOLO IN I/O	SOLO IN EITHER	CAUSING CPU WAIT	CUMULATIVE PERCENTAGES SOLO TIME	CAUSING CPU WAIT
ADARUN	CPU	6.82	.00	6.82	5.06	.00	5.06	.00	5.06	.00
ADRIVSUB	CPU	1.25	.00	1.25	.92	.00	.92	41.80	5.98	41.80
DDDATAR1	3380	.00	26.84	26.84	.00	25.61	25.61	25.54	31.59	67.34
DDASSOR1	3380	.00	24.35	24.35	.00	23.47	23.47	23.18	55.06	90.52
DDWORKR1	3380	.00	2.06	2.06	.00	1.51	1.51	1.47	56.57	91.99
DDPLOGR1	3380	.00	.01	.01	.00	.01	.01	.01	56.58	92.00

Working Set Size Through Time Report

The Working Set Size Through Time report (Figure 4-4) shows a fairly uniform working set size throughout the measurement session. The Measurement Session Data report (Figure 4-1 on page 4-2) confirms a relatively low paging rate of 1.30 pages per second.

Figure 4-4. Working Set Size Through Time Report



Wait Time by Module Report

The Wait Time by Module report (Figure 4-5 on page 4-3) shows the majority of wait time in the ADABAS nucleus module that waits for the completion of all events. This wait time includes I/O activity and time spent waiting to work. To see wait time caused by the ADABAS data set access, refer to the Resource Demand Distribution report for ADABAS Regions (Figure 4-3).

Figure 4-5. Wait Time by Module Report for ADABAS Regions

** WAIT TIME BY MODULE **

MODULE NAME	SECTION NAME	COMPRESSED SECTION	FUNCTION	RUN TIME PERCENT PAGE	TOTAL	RUN TIME HISTOGRAM .00	23.00	46.00	69.00	92.00	MARGIN OF ERROR: .61%
.SVC	SVC 111		JES 2	.00	.02	.					
DBGNUC	ASUPRMVS			.10	91.98	.++					
PROGRAM ADARUN	TOTALS			.10	92.00						

I/O Reports

Three different types of reports are available that provide information about I/O and data set access.

Data Set Characteristics Report

The Data Set Characteristics report (Figure 4-6) shows, for each data set accessed during the measurement session, the access method (ADABAS data sets are BDAM-based), block size, and data set name. The number of I/O operations performed (EXCPs) during the measurement session appears as well. In the example shown, the EXCP counts confirm the heavy use of the associator data set, DDASSOR1, and the database DDDATAR1.

Figure 4-6. Data Set Characteristics Report

```
                                        ** DATA SET CHARACTERISTICS **

DDNAME     ACCESS    POOL  REC  BLK/CI    HBUF  BUF  RPL  - SPLITS-     EXCP DATA SET NAME
           METHOD      NO  SIZE  SIZE       NO   NO STRNO CI    CA   COUNTS

DDASSOR1 BDAM                    2004                                 4,261 ADABAS.V411.DPC12.ASSO
DDDATAR1 BDAM                    4820                                 5,328 ADABAS.V411.DPC12.DATA
DDPLOGR1 BDAM                    5492                                     2 ADABAS.V411.DPC12.PROTECT1
DDWORKR1 BDAM                    5492                                   832 ADABAS.V411.DPC12.WORK
```

DASD Usage by Cylinder Report

A DASD Usage by Cylinder report (Figure 4-7 on page 4-5) shows data set access by unit, volume, ddname, and cylinder.

In this example, a small portion of the database accounts for most of the DASD usage. Identifying the frequently used data may suggest opportunities for improvement.

For the Performance Profile, this report was compressed by the specification of the parameter DASD=.1 as a report option. As a result, STROBE combined activity into a single line for contiguous cylinders that show less than 0.1 percent activity.

Figure 4-7. DASD Usage by Cylinder Report

```
                                        ** DASD USAGE BY CYLINDER **


DEVICE ADDRESS  - 31A    TYPE - 3380

  VOLUME          DDNAME            CYLINDER          RUN TIME PERCENT       RUN TIME HISTOGRAM  MARGIN OF ERROR:.61%
    ID                                NUMBER           SOLO     TOTAL        .00      3.50      7.00     10.50  14.00

  ADAF03          DDASSOR1            123              .01       .01        .
                  DDASSOR1            133-134          .10       .10        .
                  DDASSOR1            136              .03       .03        .
                  DDASSOR1            139              .64       .67        .*
                  DDASSOR1            141            10.57     10.99        .*******************************+
                  DDASSOR1            142            12.13     12.57        .***********************************+

                  DDDATAR1            241              .01       .01        .
                  DDDATAR1            288              .02       .03        .
                  DDDATAR1            298              .00       .00        .
                  DDDATAR1            306              .01       .01        .
                  DDDATAR1            316-317          .01       .01        .
                  DDDATAR1            325-326          .02       .03        .
                  DDDATAR1            363             7.74      8.18        .***********************+
                  DDDATAR1            364             7.41      7.76        .*********************+
                  DDDATAR1            365             9.68     10.13        .****************************+
                  DDDATAR1            366              .70       .72        .**

                  DDPLOGR1            442              .01       .01        .
                                                      -----     -----
DEVICE ADDRESS  - 31A TOTALS                         49.10     51.24


DEVICE ADDRESS  - 847    TYPE - 3380

  VOLUME          DDNAME            CYLINDER          RUN TIME PERCENT       RUN TIME HISTOGRAM  MARGIN OF ERROR:.61%
    ID                                NUMBER           SOLO     TOTAL        .00       .50      1.00      1.50   2.00

  ADAD01          DDWORKR1            749              .49       .55        .*********++
                  DDWORKR1            750              .64       .87        .************+++++
                  DDWORKR1            751              .38       .64        .*******++++++
                                                      -----     -----
DEVICE ADDRESS  - 847 TOTALS                          1.51      2.06
```

I/O Facility Utilization Summary Report

The I/O Facility Utilization Summary report for ADABAS Regions (Figure 4-8) summarizes I/O access by device, volume, and ddname. In this example, the database, the associator data set, and the log data set are on the same unit. I/O overlap should increase here if the database and associator data set were assigned to separate units. This reassignment should result in better response time in the online regions and faster batch processing jobs, as suggested by the Resource Demand Distribution report for ADABAS Regions (Figure 4-3 on page 4-3).

This example also shows that unit 31A is 51% utilized, implying that there must be significant queuing on this device.

Figure 4-8. I/O Facility Utilization Summary Report for ADABAS Regions

```
                                        ** I/O FACILITY UTILIZATION SUMMARY **

  UNIT      DEVICE      CACHE  VOLUME DDNAME              RUN TIME PERCENT       RUN TIME HISTOGRAM  MARGIN OF ERROR:  3.61%
   NO        TYPE       ELIG    ID                 I/O    SOLO     TOTAL        .00      7.00     14.00     21.00     28.00

   31A  DA 3380K              ADAF03 DDASSOR1       I    23.47     24.35        .*********************************+
   31A  DA 3380K              ADAF03 DDDATAR1       O    25.61     26.87        .************************************++
   31A  DA 3380K              ADAF03 DDPLOGR1 INDEX O      .01       .01        .
                                                           -----     -----
 UNIT   31A TOTALS                                       49.09     51.23
```

Reports Showing CPU Activity

Several STROBE reports indicate CPU usage by program and within a program.

Most Intensively Executed Procedures Report

The Most Intensively Executed Procedures report (Figure 4-9) shows the ten most frequently used programs in the ADABAS region. The ADABAS nucleus modules, ADANC6, ADANC2, and ADANC8 are the most heavily executed.

Figure 4-9. Most Intensively Executed Procedures Report

** MOST INTENSIVELY EXECUTED PROCEDURES **

MODULE NAME	SECTION NAME	LINE NUMBER	PROCEDURE/FUNCTION NAME	STARTING LOCATION	PROCEDURE LENGTH	CPU TIME PERCENT SOLO	CPU TIME PERCENT TOTAL	CUMULATIVE PERCENT SOLO	CUMULATIVE PERCENT TOTAL
.ADABAS	ADANC6		ADANC6 ADABAS NUCLEUS			18.13	25.95	18.13	25.9
.ADABAS	ADANC2		ADANC2 ADABAS NUCLEUS			12.74	15.70	30.87	41.65
.ADABAS	ADANC8		ADANC8 ADABAS NUCLEUS			7.87	10.16	38.74	51.81
.ADABAS	ADANC0		ADANC0 ADABAS NUCLEUS			5.44	7.49	44.18	59.30
.ADABAS	ADANC1		ADANC1 ADABAS NUCLEUS			5.15	7.01	49.33	66.31
.SVC	SVC 249		USER SVC			2.34	3.05	51.67	69.36
ADRIVERM	PROCESSM			0009C0	64	2.05	2.81	53.72	72.17
DBGNUC	HEXCONVM			000200	64	2.15	2.77	55.87	74.94
.ADABAS	ADANC7		ADANC7 ADABAS NUCLEUS			2.34	2.72	58.21	77.66
ATTACHEM	ATTACHEM			000200	64	1.67	2.43	59.88	80.09

Program Section Usage Summary Report

The Program Section Usage Summary report for ADABAS Regions (Figure 4-10 on page 4-7) shows the distribution of CPU time used by each active control section of each active module in the ADABAS region. ADABAS and IBM system modules are compressed and shown as pseudo-sections under the pseudo-module .SYSTEM.

Most of the CPU time is spent in ADABAS modules. In this example however, some of the time is spent in a user exit module, UE4MAINM. Review the Program Usage by Procedure reports for these control sections to locate the active areas of code and identify any performance improvement opportunities.

Figure 4-10. Program Section Usage Summary Report for ADABAS Regions

```
                                    ** PROGRAM SECTION USAGE SUMMARY **

 MODULE  SECTION  16M SECT  FUNCTION                CPU TIME PERCENT       CPU TIME HISTOGRAM MARGIN OF ERROR:2.14%
   NAME     NAME  <,> SIZE                           SOLO     TOTAL        .00     19.00     38.00     57.00  76.00

.SYSTEM  .ADABAS            ADABAS SYSTEM SERVICES   56.01     75.24        .**********************************++++++++++
.SYSTEM  .SVC               SUPERVISOR CONTROL        2.34      3.05        .*
                                                     -----     -----
.SYSTEM  TOTALS             SYSTEM SERVICES          58.35     78.29

ADRIVERM ADRIVERM     1600                             .05       .10        .
ADRIVERM DETAILM      4992                             .00       .05        .
ADRIVERM INTWRITM    10566                             .05       .05        .
ADRIVERM PROCESSM     3664                            6.68      9.40        .***+
                                                     -----     -----
ADRIVERM TOTALS   <  20822                           6.78      9.60

APASUEX4          <   1328                            .19       .19        .
ATTACHEM ATTACHEM <   1496                           2.19      2.96        .*

DBGNUC   ACLOCKM      5160                            .52       .76        .
DBGNUC   ASUPRMVS     5216                            .00       .05        .
DBGNUC   HEXCONVM     1368                           3.91      5.01        .**
DBGNUC   TABLES       2600                            .52       .91        .
                                                     -----     -----
DBGNUC   TOTALS   <  14344                           4.95      6.73

UE4MAINM DERIVEDM <   5256                           1.24      1.72        .
UE4MAINM UE4MAINM <   1664                            .29       .29        .
                                                     -----     -----
UE4MAINM TOTALS   <   6920                           1.53      2.01

UE4PROCM          <   1720                            .14       .24        .
                                                     -----     -----
PROGRAM ADARUN   TOTALS                             74.14    100.00
```

Program Usage by Procedure Report

The Program Usage by Procedure report for ADABAS Regions (Figure 4-11 on page 4-8) shows, for IBM and ADABAS system routines, the modules and control sections responsible for the CPU activity in the ADABAS region. Function descriptors appear for each control section or, if STROBE has not gathered control section information, for each module. (STROBE reports control section data depending on the BASELINE value specified when the measurement request is issued. For more information, see the STROBE *STROBE MVS User's Guide*.)

Figure 4-11. Program Usage by Procedure Report for ADABAS Regions

```
                                    ** PROGRAM USAGE BY PROCEDURE **

         .SYSTEM   SYSTEM SERVICES                      .ADABAS   ADABAS SYSTEM SERVICES

 MODULE   SECTION        FUNCTION            INTERVAL  CPU TIME PERCENT       CPU TIME HISTOGRAM MARGIN OF ERROR 2.14%
   NAME      NAME                              LENGTH   SOLO     TOTAL        .00      6.50     13.00     19.50  26.00

 ADAIOI   ADAIOI    ADABAS I/O PROCESSING       29064    .67      1.00        .*
 ADAIOR             ADABAS I/O PROCESSING       14464    .48       .86        .+
 ADALOG             ADABAS COMMAND LOGGING       6912    .72       .95        .*
 ADAMPM             ADABAS NUCLEUS-USER COMM     6880    .67      1.00        .*
 ADANC0   ADANC0    ADABAS NUCLEUS              12128   5.58      7.68        .********+++
 ADANC1   ADANC1    ADABAS NUCLEUS              17604   5.15      7.01        .*********++
 ADANC2   ADANC2    ADABAS NUCLEUS              45594  12.74     15.70        .**********************+++++
 ADANC5   ADANC5    ADABAS NUCLEUS              25080    .84      1.10        .+
 ADANC6   ADANC6    ADABAS NUCLEUS              30556  18.13     25.95        .*****************************++++++++++++
 ADANC7   ADANC7    ADABAS NUCLEUS              49925   2.34      2.72        .***+
 ADANC8   ADANC8    ADABAS NUCLEUS              41118   7.87     10.16        .************+++
 ADANC9             ADABAS NUCLEUS              17408    .81      1.10        .*
                                                       -----     -----
          .ADABAS   TOTALS                              56.00     74.23

          .SYSTEM   SYSTEM SERVICES                         .SVC      SUPERVISOR CONTROL

 MODULE   SECTION        FUNCTION          INTERVAL  CPU TIME PERCENT       CPU TIME HISTOGRAM  MARGIN OF ERROR:  2.14%
   NAME      NAME                            LENGTH   SOLO     TOTAL        .00      1.00      2.00      3.00      4.00

 SVC 249            USER SVC                          2.34      3.05        .*************************+++++++
                                                      -----     -----
          .SVC      TOTALS                            2.34      3.05
```

Reports Showing CPU Usage Attributed to User Regions

STROBE provides Performance Profile reports that can trace CPU usage to user regions.

Callers' DB Usage Summary Report

The Callers' DB Usage Summary report (Figure 4-12 on page 4-9) shows the percentage of total CPU time consumed by each caller of ADABAS services.

Callers are identified:

- for a batch job, by the job name
  - for a TSO region, by the TSO user ID
- for a CICS region, by the logical terminal name

Overhead activity in an ADABAS region is attributed to the pseudo-section .ADABAS.

When a task other than an ADABAS system service is executing, the STROBE ADABAS/NATURAL Feature attributes the CPU time to a pseudo-section whose name begins with a "." followed by the first seven characters of the module name in which execution of the task was initiated.

During the measurement session, this ADABAS region was accessed by one batch job, WDC9582, and four CICS terminals. The batch job was the heaviest consumer of CPU time. If the CICS response time suffers because of contention for database access, the batch program might be scheduled to run when the CICS region is less busy or has been shut down.

Figure 4-12. Callers' DB Usage Summary Report

```
                                    ** CALLERS' DB USAGE SUMMARY **

  CALLER           FUNCTION                      CPU TIME PERCENT       CPU TIME HISTOGRAM  MARGIN OF ERROR:  2.14%
   NAME                                          SOLO     TOTAL        .00     11.50     23.00     34.50     46.00

  .ADABAS          ADABAS SYSTEM SERVICES          9.45     14.50       .*********++++
  CICSC321                                         7.22      9.62       .*******++
  CICSC348                                        12.03     14.44       .**********++
  CICSC407                                         4.81      7.22       .*****++
  CICSC448                                         5.53      9.09       .*******+++
  WDC9582                                         35.11     45.13       .************************************+++++++++
                                                  -----     -----
PROGRAM ADARUN   TOTALS                           74.15    100.00
```

Callers' DB Usage by Control Section Report

A Callers' DB Usage by Control Section report (Figure 4-13) is produced for each of the database users identified in the Callers' DB Usage Summary report. The report header line identifies the calling region. Each detail line shows solo and total CPU time spent on behalf of the caller in either an ADABAS or IBM system module or in a user-written control section.

Figure 4-13. Callers' DB Usage by Control Section Report

```
                                  ** CALLERS' DB USAGE BY CONTROL SECTION **

  CALLER    WDC9582

  MODULE        SECTION                              CPU TIME PERCENT       CPU TIME HISTOGRAM  MARGIN OF ERROR:  2.14%

  NAME           NAME  COMPRESSED   FUNCTI           SOLO     TOTAL        .00      3.00      6.00      9.00     12.00

 .ADABAS       ADAIOI   ADAIOI     ADABAS I/O PROCESSING          .38       .52        .*
 .ADABAS       ADAIOR              ADABAS I/O PROCESSING          .10       .24        .
 .ADABAS       ADALOG              ADABAS COMMAND LOGGING         .29       .38        .+
 .ADABAS       ADANC0   ADANC4     ADABAS NUCLEUS                2.67      3.34        .**********+++
 .ADABAS       ADANC1   ADANC1     ADABAS NUCLEUS                2.39      3.20        .*********+++
 .ADABAS       ADANC2   ADANC2     ADABAS NUCLEUS                8.40     10.40        .************************++++++
 .ADABAS       ADANC5   ADANC5     ADABAS NUCLEUS                 .10       .14        .
 .ADABAS       ADANC6   ADANC6     ADABAS NUCLEUS                8.97     11.55        .**********************+++++++++
 .ADABAS       ADANC7   ADANC7     ADABAS NUCLEUS                1.67      1.86        .******+
 .ADABAS       ADANC8   ADANC8     ADABAS NUCLEUS                2.72      3.34        .**********++
 .ADABAS       ADANC9   ADANC9     ADABAS NUCLEUS                 .43       .57        .*
 .PRIVATE                          PRIVATE AREA                   .19       .24        .
 .SVC          SVC 249             USER SVC                      1.00      1.29        .***+
 ADRIVERM      DETAILM                                            .00       .05        .
 ADRIVERM      PROCESSM                                          2.34      3.34        .*********++++
 APASUEX4                                                         .10       .10        .
 ATTACHEM      ATTACHEM                                           .86      1.05        .**+
 DBGNUC        ACLOCKM                                            .33       .38        .*
 DBGNUC        HEXCONVM                                          1.53      2.10        .******++
 DBGNUC        TABLES                                             .14       .19        .
 UE4MAINM      DERIVEDM                                           .33       .62        .*+
 UE4MAINM      UE4MAINM                                           .14       .14        .
 UE4PROCM                                                        .05       .10        .
                                                                -----     -----
  CALLER    WDC9582   TOTALS                                   35.13     45.14
```

Chapter 5.
STROBE Wait Time Attribution for 3GL Programs and ADABAS SQL Server Statements

STROBE can measure 3GL programs that make direct calls or ADABAS SQL Server statement calls to ADABAS. The 3GL programs can be written in FORTRA, PL/I, COBOL, or assembler. STROBE does not attribute wait time if it is measuring a 3GL program running in a CICS region.

This chapter describes a measurement of a COBOL program (CB2ADA1), which makes direct calls to read ADABAS database records. A series of STROBE Performance Profile reports are included that show how the program interacts with ADABAS.

Measurement Session Data Report

After you run a STROBE session to measure how the 3GL program makes direct calls or ADABAS SQL Server statement calls to ADABAS, you can create a Performance Profile to see what STROBE reported. The first report to examine is the Measurement Session Data Report shown in Figure 5-1.

Figure 5-1. Measurement Session Data Report

```
                                    ** MEASUREMENT SESSION DATA **

-------- JOB ENVIRONMENT --------      ------- MEASUREMENT PARAMETERS ------      ----- MEASUREMENT STATISTICS ----

 PROGRAM MEASURED   -     CB2ADA1      ESTIMATED SESSION TIME  -       1 MIN      CPS TIME PERCENT -           3.77
 JOB NAME           -    WPACLHNT      TARGET SAMPLE SIZE      -      10,000      WAIT TIME PERCENT -         96.23
 JOB NUMBER         -    JOB31851      REQUEST NUMBER   (Q)    -         271      RUN MARGIN OF ERROR PCT  -   1.30
 STEP NAME   -             ADACOB                                               CPU MARGIN OF ERROR PCT  -   6.67

 DATE OF SESSION    -  08/27/1997                                               TOTAL SAMPLES TAKEN -       5,711
 TIME OF SESSION    -    12:25:07      SYS REQ -      SCS1                     TOTAL SAMPLES PROCESSED -   5,711
 CONDITION CODE     -      C-0000                                              INITIAL SAMPLING RATE- 166.67/SEC
                                        OPTIONS -                   ADA3GL     FINAL SAMPLING RATE  - 166.67/SEC
 SYSTEM -          OS/390 01.02.00
 DFSMS              -       1.3.0                                              SESSION TIME -    1 MIN 22.94 SEC
 CPU MODEL          -    9672-R42      --------- REPORT PARAMETERS ---------    CPU TIME -        0 MIN  3.59 SEC
 SMF/SYSTEM ID -         SCS1/SCS1                                             WAIT TIME    -    0 MIN 59.89 SEC
 LPAR               -        SCS1      REPORT RESOLUTION    -       64 BYTES   STRETCH TIME -    0 MIN 19.46 SEC
                                        SORTSIZE             -      999,999
 REGION SIZE BELOW 16M  -  1,088K       LINES/PAGE           -           60     SRB TIME     -    0 MIN  2.10 SEC
 REGION SIZE ABOVE  -     32,768K                                              SERVICE UNITS-              3177
                                        DASD= 2.0%  DASDGAP= 5
 PTF LVL- 2.2.0.         FS000000                                              PAGES IN-      0     OUT-       0
 STROBE TAPE NUMBER -  000-S00DSK       DATE FORMAT             -  MM/DD/YYYY  PAGING RATE  -           0.00/SEC
                                        TIME FORMAT             -    HH:MM:SS  EXCPS -          4       0.05/SEC

 SAMPLE DATA SET  - ZZ.WPACLHNT.S001D001.VERX
```

This report shows that the wait time accounted for 96.23 percent of the total time that elapsed during the STROBE measurement session. High wait time is often due to frequent I/O operations that occur when a program or ADABAS SQL Server statement makes repetitive direct calls to the database. You can probably improve program performance by making the direct calls to ADABAS more efficient.

Wait Time by Module Report

The Wait Time by Module report (Figure 5-2) shows which modules cause the application wait time.

Figure 5-2. Wait Time by Module Report Showing 3GL Attribution

```
                                        ** WAIT TIME BY MODULE **

MODULE   SECTION COMPRESSED   FUNCTION                      % RUN TIME    RUN TIME HISTOGRAM  MARGIN OF ERROR:  1.30%
 NAME      NAME    SECTION                                 PAGE   TOTAL   .00     20.50     41.00     61.50     97.00

.SVC     SVC 001           WAIT                             .00   95.26   .+++++++++++++++++++++++++++++++++++++++
.SVC     SVC 008           PROGRAM MANAGER/LOAD             .00     .93   .
.SVC     SVC 211           USER SVC                         .00     .04   .
                                                           -----   -----
.SVC     TOTALS            SUPERVISOR CONTROL               .00   96.23
                                                           -----   -----
PROGRAM CB2ADA1  TOTALS                                     .00   96.23
```

The report shows that 95.26 percent of application wait time occurred in SVC 001, which is a supervisor call routine that causes a task to cease until some event, such as I/O completion, occurs.

After identifying the module that is causing the most wait time, the final step is to determine which 3GL program statement incurred the wait time or whether the wait is attributable to ADABAS SQL Server statements. If direct calls from a 3GL program are causing the wait time, you can determine the places within the program code that initiate the database access. If the 3GL program makes ADABAS SQL Server database calls that are incurring wait time, you need to examine the ADABAS SQL Server statements. Improving these areas of code or making the ADABAS SQL Server statements more efficient will likely produce the largest decreases in wait time and therefore provide the best opportunities to decrease the total time that elapses as the application runs.

Attribution of Wait for ADABAS 3GL Programs

To attribute SVC 001 module wait time to the originating program, turn to the Attribution of CPU Wait Time report (Figure 5-3 on page 5-3). The report shows which COBOL statements are causing the wait time total shown in the Wait Time by Module report. The module showing significant wait time was one of the supervisor call routines SVC 001. An Attribution of CPU Wait Time report is included in the STROBE Performance Profile for each of the modules that incurred wait time. For this measurement session, you should focus your attention on SVC 001.

Figure 5-3. Attribution of CPU Wait Time Report

```
                                   ** ATTRIBUTION OF CPU WAIT TIME **

 .SVC     SVC 001            WAIT
--------------------------WAS INVOKED BY---------------------  -----------------VIA---------------------- WAIT TIME %
XACTION  MODULE   SECTION    RETURN   LINE   PROCEDURE NAME       MODULE   SECTION  FUNCTION                 PAGE  TOTAL

        .ADABAS  ADAIOR             ADABAS I/O PROCESSING      SVC  008  STEPLIB  PROGRAM MANAGER/LOAD      .00   .04
        .ADABAS  ADARUN    RUNMVS   ADABAS NUCLEUS STARTUP     SVC  008           PROGRAM  MANAGER/LOAD      .00   .02
        .ADABAS  ADASSF             ADABAS SYSTEM SERVICES     SVC  008           PROGRAM  MANAGER/LOAD      .00   .07
        .ADABAS  ADASSF             ADABAS SYSTEM SERVICES     SVC  008  STEPLIB  PROGRAM  MANAGER/LOAD      .00   .04
        .IOCS    IGG0199G           SAM INTERFACE OPEN         SVC  022           OPEN (TYPE = J)           .00   .07
        .IOCS    IGG0201W           DATA MANAGEMENT SERVICES   SVC  020           CLOSE                     .00   .09
        CB2ADA1  .ADABAS  ADABAS    ADABAS SYSTEM SERVICES     SVC  008           PROGRAM MANAGER/LOAD      .00   .02
        CB2ADA1  COBADA1    000FD8    209      CALL                                                        .00   .02
        CB2ADA1  COBADA1    001120    242      CALL             ADALNK   ADABAS   ADABAS SYSTEM SERVICES   .00   .28
        CB2ADA1  COBADA1    0011CA    255      CALL             ADALNK   ADABAS   ADABAS SYSTEM SERVICES   .00 39.96
        CB2ADA1  COBADA1    0012DE    292      CALL             ADALNK   ADABAS   ADABAS SYSTEM SERVICES   .00 51.65
        CB2ADA1  COBADA1    0013BE    311      CALL             ADALNK   ADABAS   ADABAS SYSTEM SERVICES   .00  1.91
        CB2ADA1  COBADA1    0015B0    358      CALL             ADALNK   ADABAS   ADABAS SYSTEM SERVICES   .00   .02
        IGC003              000A80                               SVC 013           TERMINATION              .00  1.07
                                                                                                            ----- -----
                                                                                                            .00 95.26
```

STROBE attributes most of the wait time incurred by SVC 001 to module CB2ADA1, including high wait times of 51.65 percent and 39.96 percent. In each case, CB2ADA1 calls the ADALNK module, as shown under the VIA section of the report. The ADALNK module is significant because it serves as an application interface to ADABAS. All types of database access from 3GL program direct calls directed at ADABAS use the ADALNK module.

All 3GL program ADABAS SQL Server statement calls to the ADABAS database use a module named ESQOBT as the application interface to ADABAS. For 3GL program SQL statement calls to ADABAS, this module will appear in the VIA section of the report.

Applications making I/O requests always have to wait some period of time while the request is being processed. The wait time usually occurs while the data is being retrieved from or written to DASD. You should focus your tuning efforts on improving the efficiency of the program statements that invoke the ADALNK or ESQOBT modules.

For this example, decreasing the frequency of the direct calls to ADALNK offers the best opportunity to improve the performance of this program. By examining the COBOL code at the statement numbers indicated in the Attribution of CPU Wait Time report, you may be able to reduce the number of ADABAS access direct calls. You could also try adjusting parameters for ADABAS features such as PREFETCH or changing the buffer space size to cut wait time.

By indexing this program, you obtain the COBOL statement number and verb associated with the ADABAS call. This will help you determine where to revise the program code. For further information about indexing, refer to Chapter 2, “Using the STROBE ADABAS/NATURAL Feature” of this manual or Chapter 2 of the STROBE *STROBE MVS User’s Guide*.

Attribution of Wait in ADABAS SQL Server Statements

If the wait time was incurred on behalf of ADABAS SQL Server statements, STROBE provides attribution under the pseudo-section “.ADASQL”. STROBE reports all wait in modules beginning with “ESQ“ under “.ADASQL.”.

If execution time is incurred by modules beginning with “ESQ”, STROBE also reports this under the pseudoname “.ADASQL”.

Index

Special Characters

Numerics

A

C

D

E

F

I

M

N

P

R

S

T

W