

PROFILER for NATURAL

Note: All references to the Profiler version in this manual are indicated by *vrs* or *v.r.s*. The current release of Profiler is version 4.3.2

Comments pertaining to this document and the PROFILER for NATURAL package are encouraged. Please direct all comments in writing to:

Treehouse Software, Inc.
2605 Nicholson Road, Suite 1230
Sewickley, PA 15143
Phone: 724.759.7070
Fax: 724.759.7067
e-mail: support@treehouse.com
<http://www.treehouse.com>

Worldwide marketing of PROFILER for NATURAL and other Treehouse products is handled through the Sewickley office.

Any reproduction of any portion of this document without the written consent of Treehouse Software, Inc. is prohibited.

Copyright January 2013 by Treehouse Software, Inc., Sewickley, Pennsylvania.

Last Updated: 01/27/2014

This page intentionally left blank.

This PROFILER for NATURAL Manual explains the functions and capabilities of the Treehouse Software product PROFILER for NATURAL.

The first section of this manual introduces PROFILER and gives an overview of its functions.

Section 2, Getting Started, provides users a tutorial that takes users through the steps of How To Begin Using PROFILER. It is recommended that this section be read before using PROFILER.

The next seven sections describe the Session Maintenance, Session Execution, Session Reporting, Enhanced Reporting, Trace, Clarifications, and PROFILER Administration functions.

Section 10 describes how to use PROFILER in batch and the Section 11 presents PROFILER Education and Examples.

Section 12 explains the installation procedure for PROFILER. This section illustrates the procedure for OS and VM environments. Sample JCL is included for each environment.

Appendix A lists PROFILER messages and Appendix B lists common problems.

Throughout this manual, PROFILER refers to the Treehouse Software product PROFILER for NATURAL.

PROFILER for NATURAL is a product of Treehouse Software, Inc., and is copyright protected. ADABAS, NATURAL, NATURAL VSAM, NATURAL DB2, and COM-LETE are products of Software AG. CICS, Z/OS, TSO and DB2 are products of IBM. TRIM, N2O, N2O/3GL, AUDITRE, AUTOLOADER, tRelational, and SECURITRE are products of Treehouse Software, Inc.

This page intentionally left blank.

Table of Contents

I. INTRODUCTION	I-1
I.1 Introducing Application Profiling	I-1
I.2 Benefits of Application Profiling	I-2
I.2.1 Profiling for Quality Assurance	I-2
I.2.2 Profiling for Performance	I-3
I.2.3 Profiling for Debugging	I-3
I.2.4 Profiling for Testing	I-4
I.2.5 Profiling for Education and Evaluation	I-4
I.3 Overview of the PROFILER for NATURAL Environment	I-5
I.3.1 PROFILER Sessions	I-7
I.3.2 PROFILER Enhanced Reporting Facility	I-10
I.3.3 PROFILER Performance and Usage Considerations	I-11
I.4 The PROFILER User Interface	I-12
I.4.1 PROFILER Screen Standards	I-12
I.4.2 'About' Screen	I-13
I.4.3 Main "Session List" Menu	I-14
I.4.4 Data Entry Screens	I-16
I.4.5 Help Screens	I-17
I.4.6 Display Screens	I-18
I.4.7 Report Screens	I-19
I.4.8 Error Screens	I-21
II. GETTING STARTED	II-1
II.1 Introduction	II-1
II.2 How To Begin Using PROFILER	II-2
III. MAIN 'SESSION LIST' MENU	III-1
III.1 Menu Structure	III-1
III.2 Invoking PROFILER	III-2
III.3 Controlling the "Session List"	III-3
III.4 "Session Actions" on the "Session List"	III-3
III.5 PF Keys on the "Session List"	III-4
III.6 "Session List" Help	III-5
III.7 'About' Screen	III-5
IV. SESSION MAINTENANCE EXECUTION	IV-1
IV.1 Introduction to Session Maintenance and Execution	IV-1
IV.2 Define New Session	IV-3
IV.3 Display Session Definition	IV-7
IV.4 Modify Session Definition	IV-11
IV.5 Delete Existing Session	IV-15
IV.6 Activate Session	IV-19
IV.7 Deactivate Session	IV-24
IV.8 Reset Session Statistics	IV-26

IV.9	Show Session Active Users.....	IV-30
V.	SESSION REPORTING	V-1
V.1	Introduction to Session Reporting.....	V-1
V.2	Session Statistics Summary Reports.....	V-5
V.2.1	Statement Execution Count Summary Report	V-9
V.2.2	CPU Time Summary Report.....	V-11
V.2.3	Database Elapsed Time Summary Report.....	V-13
V.2.4	Number of Statements, CPU Time, Database Time Report.....	V-15
V.2.5	Percentage of Executions, CPU Time, and Database Time Report.....	V-17
V.2.6	Percent Graph of Executions, CPU and Database Report.....	V-19
V.2.7	Database/Work File Loops Report	V-21
V.2.8	Internal Subroutines/Non-Procedural Blocks Report.....	V-23
V.2.9	FOR/REPEAT Loops Report.....	V-25
V.2.10	IF/DECIDE Conditions Report.....	V-27
V.2.11	Percent Executed by Statement Type Report	V-29
V.3	Source Code Listing Report for Profiled Objects	V-31
V.4	Source Code Listing Report for Traced Objects	V-35
V.5	Application QA Report for Profiled Libraries	V-37
VI.	ENHANCED REPORTING	VI-1
VI.1	Introduction to Enhanced Reporting	VI-1
VI.2	Tag and Move Data Actions	VI-2
VI.2.1	Tag Data for Move.....	VI-3
VI.2.2	Untag Data for Move	VI-4
VI.2.3	Move Selected Tagged Data.....	VI-5
VI.2.4	List Tagged Data	VI-6
VI.3	Report Parameter Set Maintenance	VI-7
VI.3.1	Report Parameter Maintenance	VI-9
VI.3.1.1	Define Report Parameters	VI-10
	Field Description	VI-10
VI.3.1.2	Display Report Parameters	VI-13
	Field Description	VI-13
VI.3.1.3	Modify Report Parameters	VI-15
	Field Description	VI-15
VI.3.1.4	Delete Report Parameters	VI-17
	Field Description	VI-17
	Field Description	VI-18
VI.3.1.5	Copy Report Parameters	VI-20
	Field Description	VI-20
VI.3.2	User Group Maintenance	VI-22
VI.3.2.1	Define User Group	VI-23
	Field Description	VI-23
VI.3.2.2	Display User Group.....	VI-25
	Field Description	VI-25
VI.3.2.3	Modify User Group.....	VI-26

Field Description.....	VI-26
VI.3.2.4 Delete User Group.....	VI-27
Field Description.....	VI-27
VI.3.2.5 Copy User Groups.....	VI-28
Field Description.....	VI-28
VI.3.3 Session Group Maintenance.....	VI-29
VI.3.3.1 Define Session Group.....	VI-30
Field Description.....	VI-30
VI.3.3.2 Display Session Group.....	VI-32
Field Description.....	VI-32
VI.3.3.3 Modify Session Group.....	VI-33
Field Description.....	VI-33
VI.3.3.4 Delete Session Group.....	VI-34
Field Description.....	VI-34
VI.3.3.5 Copy Session Groups.....	VI-35
Field Description.....	VI-35
VI.4 Introduction to Enhanced Reporting.....	VI-37
VI.4.1 Enhanced Statistics Summary Reports.....	VI-40
VI.4.3.1 Enhanced Statement Execution Count Summary Report.....	VI-44
VI.4.1.2 Enhanced CPU Time Summary Report.....	VI-46
VI.4.1.3 Enhanced Database Elapsed Time Summary Report.....	VI-48
VI.4.1.4 Enhanced Number of Statements, CPU Time, Database Time Report..	VI-50
VI.4.1.5 Enhanced Percentage of Executions, CPU Time, and Database Time Report	VI-52
.....	
VI.4.1.6 Enhanced Percent Graph of Executions, CPU and Database Report....	VI-54
VI.4.1.7 Enhanced Database/Work File Loops Report.....	VI-56
VI.4.1.8 Enhanced Internal Subroutines/Non-Procedural Blocks Report.....	VI-58
VI.4.1.9 Enhanced FOR/REPEAT Loops Report.....	VI-60
VI.4.3.10 Enhanced IF/DECIDE Conditions Report.....	VI-62
VI.4.3.11 Enhanced Percent Executed by Statement Type Report.....	VI-64
VI.4.2 Enhanced Source Code Listing Report.....	VI-66
VI.4.3 Enhanced Application QA Report.....	VI-70
VII BACKGROUND MONITORING.....	VII-1
VII.1 Overview of Background Monitoring.....	VII-1
VII.2 Background Monitoring "Session List" Menu.....	VII-4
VII.3 Define Background Monitoring Record.....	VII-7
VII.4 Display Background Monitoring Record.....	VII-10
VII.5 Modify Background Monitoring Record.....	VII-12
VII.6 Delete Background Monitoring Record.....	VII-14
VIII. CLARIFICATION OF PROFILER STATISTICS DURING REPORTING.....	VIII-1
VIII.1 NATURAL Statements Coded on the Same Line.....	VIII-1
VIII.2 Statement Statistics Limits.....	VIII-1
VIII.3 Statistics for Objects Migrated with SYSMAIN.....	VIII-2
VIII.4 NATURAL STEPLIB Objects.....	VIII-2
VIII.5 Statistics for PROLIB, SYS, and SYSTEM Objects.....	VIII-2

VIII.6	Statistics for PREDICT "Free" and "Automatic" Verification Rules Included in NATURAL Maps	VIII-2
VIII.7	Statistics for Subroutine Statements	VIII-2
VIII.8	Statistics for IF/ELSE/END-IF	VIII-3
VIII.9	Statistics for Database/Work File Statements	VIII-4
VIII.10	Statistics for DECIDE Statements	VIII-5
VIII.11	Statistics for DEFINE DATA Statements	VIII-6
VIII.12	Statistics for FIND Statements	VIII-6
VIII.13	Statistics for CALL Statements	VIII-7
VIII.14	Statistics for AT END OF PAGE Statements	VIII-7
VIII.15	Statistics for END Statements	VIII-7
VIII.16	Statistics for TERMINATE Statements	VIII-7
VIII.17	Defining User Group Names	VIII-8
VIII.18	NATURAL OPTIMIZER COMPILER (NOC	VIII-8
VIII.19	How PROFILER Calculates CPU Time	VIII-8
VIII.20	Object CPU Time vs. Statement CPU Time	VIII-10
VIII.21	Considerations When Using the NATURAL Review Data Collector Interface f or Collection of PROFILER Statistics	VIII-10
VIII.22	Processing of Object Catalog Timestamps on the Enhanced Source Code Listing Report	VIII-10
VIII.23	PROFILER Statistics for NATURAL Object Types	VIII-12
VIII.24	PROFILER Sessions - Individual Versus Shared	VIII-12
	VIII.24.1 Individual Sessions	VIII-13
	VIII.24.2 Shared Session	VIII-13
VIII.25	How To Use Enhanced Reporting	VIII-14
IX.	PROFILER ADMINISTRATION	IX-1
IX.1	Introduction to PROFILER Administration	IX-1
IX.2	Reset Active Users	IX-3
IX.3	Purge All Profile Sessions	IX-6
IX.4	Purge All Trace Sessions for One User	IX-7
IX.5	Purge All Trace Sessions for All Users	IX-7
IX.6	Move Tagged Data for All Profile Sessions	IX-7
IX.7	Ask User about SYSRDC	IX-8
X.	USING PROFILER IN BATCH	X-1
X.1	Introduction to PROFILER in Batch	X-1
X.2	Activate/Deactivate Session in Batch	X-2
X.3	Reset Session Statistics in Batch	X-5
X.4	Tagged Data Batch Operations	X-7
	X.4.1 Tag Data For Move	X-7
	X.4.2 Move All/Selected Tagged Data in Batch	X-8
X.5	Batch Reports	X-10
X.6	Batch Report Programs and Parameters	X-11
	X.6.1 Session Statistics Summary Reports	X-11
	X.6.2 Source Code Listing Report for Profiled Objects	X-14

X.6.3	Source Code Listing Report for Traced Objects	X-15
X.6.4	Application QA Report	X-16
X.7	Batch Enhanced Report Programs and Parameters	X-17
X.7.1	Enhanced Session Statistics Summary Reports.....	X-17
X.7.2	Enhanced Source Code Listing Report.....	X-20
X.7.3	Enhanced Application QA Report	X-21
XI	EDUCATION AND EXAMPLES	XI-1
XI.1	Introduction	XI-1
XI.2	Quality Assurance	XI-2
XI.3	Performance Analysis	XI-4
XI.4	Debugging.....	XI-9
XI.5	Application Testing.....	XI-10
XI.6	Education and Evaluation	XI-11
XII.	INSTALLATION.....	XII-1
XII.1	Introduction to Installation	XII-1
XII.2	Z/OS Installation	XII-3
XII.2.1	Load Datasets From Either a Web/Email or Tape Distribution	XII-5
XII.2.1.1	Loading Datasets From a Web or Email Distribution	XII-5
XII.2.1.2	Load datasets from a tape distribution.....	XII-9
XII.2.1.3	Install zaps and fixes	XII-11
XII.2.2	Apply PROFILER Authorization Zap.....	XII-12
XII.2.3	Define your PROFILER configuration using the PROFCFG macro.....	XII-12
XII.2.4	NATLOAD the PROFILER NATURAL Modules.....	XII-14
XII.2.4.1	Copy the PROFILER NATURAL Library (Optional).....	XII-15
XII.2.7.2	SYSMAN Modules from SYSEXT to SYSTEM	XII-15
XII.2.4.3	Installing the LOGON Front-End.....	XII-16
XII.2.4.4	Installing the FIN Front-End	XII-17
XII.2.4.5	User-Exits.....	XII-17
XII.2.4.6	Step Library changes where NATURAL SECURITY is Installed	XII-18
XII.2.5	Load the PROFILER-REPORTING File DDE (optional)	XII-18
XII.2.6	Establish the ADABAS file Where PROFILER Statistics Will Be Stored ...	XII-19
XII.2.7	Establish the ADABAS file For PROFILER Enhanced Reporting	XII-20
XII.2.8	Increase Sizes of Partition or Region and Possibly the NATURAL Thread, and	
XII.2.9	Assemble NATPARMs and Relink the NATURAL Nucleus	XII-22
XII.2.10	Considerations of Using the NATURAL RDC Interface.....	XII-23
XII.2.11	Create Reentrant ADALINKs	XII-24
XII.2.12	Recycle CICS and COM-LETE/TPF (for CICS and COM-LETE/TPF Installation only).....	XII-24
XII.2.13	Verify the PROFILER Installation.....	XII-25
XII.3	VM (SP, XA, ESA) Installation	XII-26
XII.3.1	Allocate CMS Mini-disk Space	XII-28
XII.3.2	Load to Allocated Mini-disk	XII-28
XII.3.3	Apply PROFILER Authorization Zap.....	XII-28
XII.3.4	Apply GETMAIN Zap to PROFINIT (optional).....	XII-29
XII.3.5	Trace Subsystem Installation (optional).....	XII-30

XII.3.7	NATLOAD the PROFILER NATURAL Modules	XII-30
XII.3.7.1	SYSMAN Modules from SYSEXT to SYSTEM.....	XII-32
XII.3.7.2	Installing the LOGON Front-End	XII-32
XII.3.7.3	Installing the FIN Front-End.....	XII-34
XII.3.7.4	User-Exits	XII-34
XII.3.7.5	Step Library changes where NATURAL SECURITY is Installed.....	XII-34
XII.3.8	Load the PROFILER-REPORTING File DDE (optional)	XII-35
XII.3.9	Establish the ADABAS File Where PROFILER Statistics Will Be Stored..	XII-36
XII.3.10	Establish the ADABAS File For PROFILER Enhanced Reporting	XII-37
XII.3.11	Change and Assemble NATPARAMs, Apply Zap(s) for PROFILER Statistics Collection, and Relink NATURAL Executable Modules	XII-38
XII.3.12	Considerations of Using the RDC Interface	XII-39
XII.3.13	Increase Size of the Virtual Machine	XII-39
XII.3.14	Verify the PROFILER Installation	XII-40
APPENDIX A	PROFILER MESSAGES	A-1
APPENDIX B	PROFILER Common Problems	B-1

SECTION I

INTRODUCTION

I.1 Introducing Application Profiling

Programmers have always needed methods for monitoring the performance and testing the execution of their code. These needs have led to the development of a wide variety of monitoring tools. One of the most useful of these is the application profiler.

An application profiler monitors the execution of each line of code in an application. As part of the monitoring process, the profiler confirms the actual execution of the code and accumulates statistics about the performance of the code. These statistics may be reported by user, application, object, object statement, and other categories.

The development of sophisticated languages such as NATURAL has not affected the need for application profilers. In fact, the variety of application components in NATURAL (e.g., programs, subprograms, copycode, maps, help routines, etc.) makes applications more difficult to monitor and test. NATURAL applications may also be much easier to use, implying a greater level of execution activity and an increasing need to be concerned about performance.

Furthermore, NATURAL does not generate true object code to be executed directly by the CPU. NATURAL interpretively executes its own "object code", making it difficult to relate a particular NATURAL statement to the actual instructions executed by the CPU.

In addition, there might be differences between the amount of CPU time required to execute the same NATURAL statement under one version of NATURAL and another. Buffer pool activity, operating system differences, and teleprocessing system differences affect NATURAL performance. These make CPU usage by NATURAL somewhat unpredictable.

Programmers need to be able to test the relative performance of different NATURAL statements that accomplish the same function in order to determine which statement is the most efficient for a given function. Programmers may use the application profiler to make these determinations.

Treehouse Software's PROFILER for NATURAL is designed to meet the application profiling needs of NATURAL sites.

Note: All references to the Profiler version in this manual are indicated by *vrs* or **v.r.s. The current release of Profiler is version 4.3.1.**

I.2 Benefits of Application Profiling

Profiling NATURAL applications can provide a site with benefits in these and other areas:

- Quality Assurance - Enables a site to more thoroughly test code, resulting in lower maintenance and support requirements.
- Performance Analysis - Identifies problem objects and statements, inefficient code, poor application structure/design, and expensive database access methods.
- Debugging - Reveals object and statement execution statistics and displays object flow through tracing.
- Application Testing - Identifies untested objects, code not executed, weaknesses in test data and procedures, and assesses the impact of new functions and database changes.
- Education and Evaluation - Gives insight into NATURAL internals and NATURAL statement efficiencies for performance optimization.

I.2.1 Profiling for Quality Assurance

Before any application is placed into production, it is given a series of formal tests. These tests use sample data to demonstrate that the application functions properly. Any errors or problems encountered during testing are corrected by the programmer. However, it is virtually impossible to manually determine that an entire object has been tested without using an application profiling tool.

It is widely acknowledged that a large percentage of object execution time is spent on a small amount of the object's code. Test procedures often perform an excellent job of testing the more frequently used portion of the code, but sometimes ignore the less frequently used portion. At some point in time, however, it is likely that data will be introduced that will cause the untested portion of the code to be executed. Any errors encountered could corrupt important production data or bring the entire application down. This is not an acceptable situation for any site with critical applications and data. Therefore, a better methodology for testing applications is required.

Profiling tools help identify an object's "hot spots", or areas where significant amounts of computing resources are being used. Profiling tools may also play an important role in the testing and quality assurance processes.

When testing NATURAL applications, sites need an effective tool, such as PROFILER, for profiling the execution of NATURAL objects. With PROFILER, sites can monitor the execution of a NATURAL application in Testing or Quality Assurance environments, identifying which objects and statements have and have not been executed. Using this information, sites can verify the effectiveness of their test data and procedures. If the statistics show that a particular object or section of an object has not been tested, the site should modify its test data and procedures so that the untested code is executed during further testing.

I.2.2 Profiling for Performance

In order to get the most from its hardware and software investment, a site needs applications which deliver peak performance. Studies have shown that less than 4 percent of the code for an application generally accounts for more than half of its running time. Therefore, it is important for a site to determine which parts of applications consume the most resources and to tune those components for maximum efficiency.

When analyzing application performance, PROFILER may help by identifying:

- CPU intensive objects and statements
- Inefficient code
- Poor application structure and design
- Expensive database access methods

With this information, a site may quickly focus its tuning efforts on the areas which have the greatest impact on an application's performance.

Once problem objects and statements are identified, PROFILER continues to assist the site. Using PROFILER, a site may determine the performance impact of using different NATURAL statements to perform a function, using different algorithms to accomplish a function, or the effect of changes on the object's structure.

I.2.3 Profiling for Debugging

When debugging an application, it is helpful to know exactly what the object is doing by determining which parts of it are executing. PROFILER provides this sort of information.

Seeing the number of times a statement has executed can sometimes provide a clue to finding the bug in an object. For example, if the statements within a loop seem to have executed an excessive number of times, this could indicate that the loop has been incorrectly coded. Similarly, if a statement which should have executed during testing did not execute, this could indicate incomplete test data.

More detailed information on objects can be obtained through Trace sessions. The Trace Source Code Report lists all statements of a NATURAL object in the order they were executed.

I.2.4 Profiling for Testing

During application testing efforts, it is imperative to ensure that the entire application is tested. PROFILER can easily pinpoint unexecuted objects, helping to ensure better test coverage.

During testing, it is important to identify any weaknesses in test data or procedures used to test the application. By identifying unexecuted objects and statements, PROFILER can help to identify missing or incorrect test data and procedures. The test data can be adjusted, and the application profiled again to ensure that the revised test data and procedures exercise all of the application code.

Changes to parts of an application can impact the function or performance of other parts of the application. It is important to ensure that the entire application has been tested after any significant changes have been made to the application functionality or structure. PROFILER can help the site to ensure that proper testing is done.

Database modifications can have profound effects on an application. Changing the layout or physical structure of a database file can increase or decrease the performance of applications which access that file. Changing database parameters can improve or hinder performance. By comparing the elapsed time of database accessing statements before and after the database change, sites can determine the impact of those database changes.

I.2.5 Profiling for Education and Evaluation

The education of programmers is an ongoing process. As NATURAL grows and evolves, its internals change. Statements which may have performed poorly in a previous release of NATURAL may now perform very well. By profiling a given statement under different releases of NATURAL, programmers can gain insight into NATURAL internals and see the improvements which are being made behind the scenes. This insight can be invaluable to programming efforts.

Like any language, NATURAL provides programmers with more than one way to solve a given programming problem. There may be several different approaches, using different NATURAL statements, algorithms, etc. Each potential solution may have a different performance impact. Programmers need a way to determine, for any given situation, the most efficient solution to a programming problem. PROFILER allows programmers to evaluate the performance impact of different options.

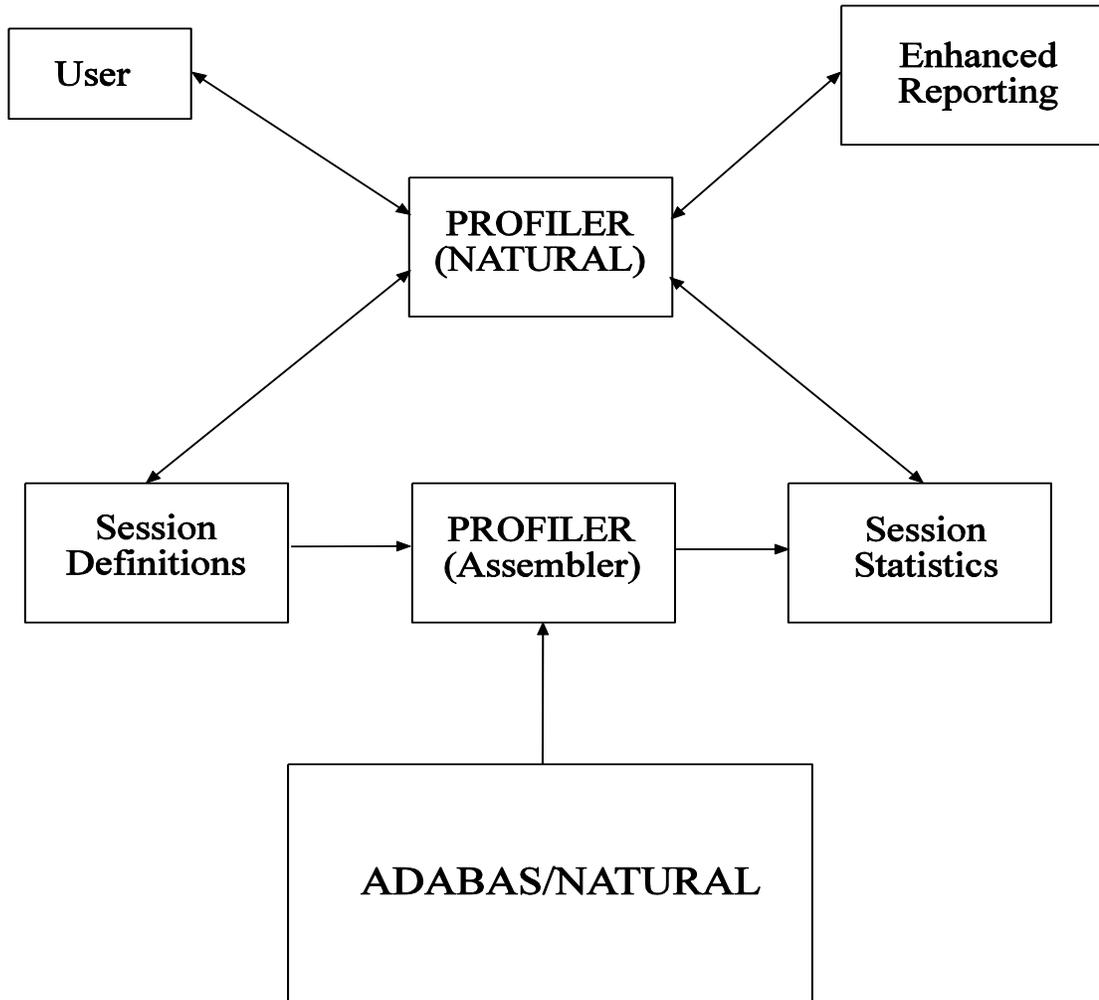
Database access is also a very important part of most NATURAL applications. Based on the structure and content of a given database, the type of database access used can make a significant difference in the performance of the object. In some cases, a READ statement might be the most appropriate way to access the database. In others, a FIND statement might be more efficient. By profiling the results of both statements with the actual data, it is easy to see which solution is better in a given situation.

Application profiling benefits like these are all available in PROFILER for NATURAL.

I.3 Overview of the PROFILER for NATURAL Environment

PROFILER for NATURAL is a powerful tool for monitoring and testing the execution of NATURAL objects and applications. As users test their applications, PROFILER can be instructed to monitor their testing activity and collect statistics for the objects they are executing.

The diagram below provides an overview of the PROFILER for NATURAL operating environment:



The user begins by creating a profile session on the PROFILER "session list" menu. A profile session is a repository for storing application execution statistics in an ADABAS file and is described in the next subsection. The user defines the statistics to be collected and then activates this session.

When PROFILER is activated for a user's NATURAL session, PROFILER opens an independent session with ADABAS, and establishes PROFILER control information. As the user invokes NATURAL objects, the NATURAL Dispatcher interpretively executes each line of object code. As each statement in a NATURAL object is executed, PROFILER identifies the object and statement being executed. PROFILER increments the run count, records the CPU time for the current NATURAL statement, and accumulates the CPU time for the previous NATURAL statement. For NATURAL statements resulting in database calls, the elapsed time is also accumulated. Control is then returned to NATURAL, which executes the command.

This interaction with NATURAL ensures that PROFILER provides continuous monitoring of NATURAL object and statement executions. This ensures that PROFILER collects complete, accurate statistics. Periodically, these statistics are stored in the PROFILER statistics repository, which is an ADABAS file.

The PROFILER Reporting system uses the statistics stored in the ADABAS file and the NATURAL source code (if displayed in the report) to produce a variety of on-line and batch reports of NATURAL application execution activity.

The PROFILER Enhanced Reporting system allows statistics to be moved to the PROFILER Reporting file where the statistics may be merged and manipulated to produce PROFILER Enhanced Reports and customized user-written reports.

The Trace Subsystem allows the activation and maintenance of Trace sessions, which display object statements in the order they were executed. The PROFILER Trace assembler modules maintain a buffer which may contain up to 900 traced statements. When this buffer is full, PROFILER will write the data for the 900 statements to the PROFILER repository. Trace data is also written to the PROFILER repository when the Trace is deactivated. No reporting may take place on an active Trace session.

I.3.1 PROFILER Sessions

In order to provide reports containing only the desired statistics, PROFILER must be able to identify related and unrelated statistics and distinguish between statistics collected by one user and those collected by another. PROFILER must also be able to relate statistics from different users, libraries, objects, etc. One way that PROFILER accomplishes these tasks is by relating statistics to a particular session and user.

Sessions can also be used to relate the testing activity of multiple users. For example, a group of users assigned to test a specific application may all use the same session. PROFILER will allow users to view their own statistics for the session independently of other users' statistics or allow them to view all users' statistics for the session on one report.

Users activate a session at the start of their testing process. PROFILER begins collecting the desired statistics and storing them in an ADABAS file. Statistics collection stops when the user deactivates the session or exits NATURAL.

The same PROFILER session may be activated many times by any number of users, each profiling the same or different applications/objects.

Much of the day-to-day use of PROFILER centers around the use of sessions. For this reason, PROFILER includes facilities that ensure the easy management of sessions.

Sessions may be defined (using PF9), 'DI'splayed, 'MO'dified, 'CO'pied, and 'PU'rged. When defining a session, the user may specify certain libraries, objects, and NATURAL object types (i.e., programs, subprograms, subroutines, maps, and help routines) for which statistics should be collected by PROFILER.

To collect statistics about a user's testing activity, the user must 'AC'tivate a previously defined PROFILER session. Sessions may be manually activated by the user or automatically activated by the PROFILER Background Monitoring facility (accessed with PF6 from the "session list" menu).

When a user manually activates PROFILER to collect statistics for application testing activity, the user selects a session to be activated and specifies the libraries/objects for which statistics should be collected. PROFILER is then instructed to activate the session and begin collecting statistics.

Although PROFILER activation and deactivation are very simple, some sites prefer a more automated approach. The Background Monitoring facility allows authorized users, such as Project Leaders, to cause a PROFILER session to be automatically activated for themselves or others based on library/object masks, NATURAL object types, and date/time. When users LOGON to the desired NATURAL library, the specified PROFILER session is automatically activated. As the user tests the application with a PROFILER session activated, PROFILER collects the desired statistics and places them in an ADABAS file.

The statistics stored in a session may become outdated or inappropriate (e.g., objects may have been modified after they have been tested). In this case, session statistics may be reset ('RS' action) by the user for the entire session or for a specific user, library, or object.

When testing is complete, the PROFILER session is deactivated ('DA' action). At this point, PROFILER ceases collection of statistics for this user's testing activity. Reports may then be produced from the statistics using the 'S.' (Summary Statistics), 'QA' (Application Quality Assurance), or 'CU' (Customized Report) actions.

Many ADABAS/NATURAL applications have on-line and batch components. On-line components provide quick and easy access to information. Batch components provide efficiency in processing a large volume of data, enable processing to occur when on-line activities are minimal, and offer users the option of obtaining hardcopy output.

For these same reasons, PROFILER provides a number of useful reports on-line and in batch, including:

Summary statistics for a PROFILER session which display the following object information:

- Libraries/objects executed
- Statement Execution Count reports ('SS' action)
- CPU time reports ('SC' action)
- Database Elapsed Time reports ('SD' action)
- Number of Statements, CPU Time, Database ('SN' action)
- Percentage of Executions, CPU and Database ('SP' action)
- Percent Graph of Executions, CPU and Database ('SG' action)

Each of these reports may be sorted in one of four ways:

- Object (within library) (OBJ)
- Statement Execution Count (STA)
- CPU time (CPU)
- Database Elapsed Time (DAT)

Source listing of a NATURAL Object showing execution statistics, including copycode execution statistics and the identification of NATURAL Optimized Code.

NATURAL Statement Type Statistics:

- Database/Work File Loops ('SF' action)
- Internal Subroutines/Non-Procedural Blocks ('SB' action)
- FOR/REPEAT Loops ('SR' action)
- IF/DECIDE Conditions ('SI' action)
- Percent Executed by Statement Type ('ST' action)
- Threshold Statistics, showing objects which equaled or exceeded a specific threshold(s) for object executions, statement executions, CPU time, Database Elapsed Time, etc.
- Summary for Application Library - Quality Assurance ('QA' action), showing which objects in a given application library have not been tested.
- Trace Source Code Report ('S=' action), which displays object statements in the order they were executed.

The PROFILER Reports present statistics which show the following information:

For each object:

- Execution counts
- CPU time usage
- Database Elapsed Time usage
- Percent of statements tested

Within each object:

- Statement Execution counts
- CPU time used per statement
- Database Elapsed Time used per statement
- Statements not tested

For each application:

- Percent of object tested
- Percent of application tested

Note: Database Elapsed Time represents elapsed time for calls to databases such as ADABAS, DB2, and VSAM.
--

I.3.2 PROFILER Enhanced Reporting Facility

The PROFILER Enhanced Reporting facility allows statistics to be moved to a separate ADABAS file so that PROFILER Enhanced Reports and customized user-written reports can access this data. Statistics may be combined in a variety of ways for flexible reporting.

These are some of the combinations a user may specify for the Enhanced Reports:

- One session with one User-ID
- One session with multiple User-IDs
- Group of users
- Group of sessions
- All User-IDs
- All sessions
- Library/object masks
- Multiple versions (Catalog Timestamps) of an object

In addition to these combinations, PROFILER allows a user to specify exceptions which are to be included or excluded from the report.

Consider a site where a team of users tests the same application. Each user tests a specific function or component of the application. Using the Enhanced Reporting facility, the project leader can merge individual team members' statistics on one report to verify that the entire application has been tested.

In addition to combining statistics from multiple users, sessions, etc. the Enhanced Reporting facility also provides additional options for viewing PROFILER statistics. For example, the Enhanced Source Code Listing Report automatically expands copycode in the listing.

One of the most significant aspects of the Enhanced Reporting facility is that it allows users to code their own reports to display PROFILER statistics in any desired format. This makes the reporting capabilities of PROFILER virtually unlimited.

I.3.3 PROFILER Performance and Usage Considerations

When using a profiling tool such as PROFILER, it is important to consider the performance impact of the tool, the monitoring technique used by the tool, and the accuracy of the tool. These factors may affect the output produced.

PROFILER has been designed to have minimal impact on the systems being profiled. Continuous storage of profiling statistics on the database would dramatically affect the performance of the application being profiled. Therefore, PROFILER statistics are collected in memory and stored on the database periodically. This decreases the database access requirements and minimizes the overhead created by PROFILER. PROFILER statistics collection is performed by ASSEMBLER routines to ensure their speed and efficiency.

When using PROFILER for performance tuning, it is necessary to ensure that the profiling session allows for several executions of each object and statement in the application. NATURAL buffer activity, operating system events, and teleprocessing system events may occasionally cause some statement and object execution figures to appear abnormally large. By monitoring several executions of the object or statement, the effect of these non-object activities on the performance statistics is reduced or eliminated.

Although PROFILER is very efficient, monitoring the execution of each NATURAL statement in a production environment can generate substantial overhead. Therefore, it is recommended that sites use PROFILER in production environments on a very limited basis.

I.4.2 'About' Screen

Pressing PF2 on the PROFILER "session list" menu displays the PROFILER 'About' screen.

PPPPP	RRRRR	OOOOO	FFFFF	IIIII	LL	EEEEEE	RRRRR
PP PP	RR RR	OO OO	FF	II	LL	EE	RR RR
PP PP	RR RR	OO OO	FFFF	II	LL	EEEE	RR RR
PP PP	RR RR	OO OO	FF	II	LL	EE	RR RR
PP PP	RR RR	OO OO	FF	II	LL	EE	RR RR
PPP	RR RR	OOOOO	FF	IIIII	LLLLL	EEEEEE	RR RR
PP	RR	RR					
PP	RR.	RR	.	f	o	r	.
PP	TP Mon TSO
PP	.	.	.	N	A	T	U
				R	A	L	.
				.	.	.	Version 4.3.2
				.	.	.	Released 01/03/13
				.	.	.	Zap Level 0000
COPYRIGHT							ADALNK
TREEHOUSE SOFTWARE							Only Authorized for Use by
2605 Nicholson Road Suite 230			*	*			Licensee until 2013-12-31
Sewickley PA 15143 USA							
phone: (724) 759 7070						DBID	FNR
fax: (724) 759 7067						Zap	0 0
http\www.treehouse.com						LF=120	4 120
Statistics can be collected for up to 1080 statements.						LF=122	4 122

Pressing Enter will return you to the PROFILER "session list" menu.

I.4.3 Main "Session List" Menu

Type the command "PRF" at the 'NEXT' prompt or in the NATURAL 'Command:' line, press Enter, and PROFILER displays the Main "session list" menu. Sessions matching the 'From', 'Owner', and 'Profile/Trace/All' settings will be displayed. With the cursor in the 'Com' field, press PF1 and a list of valid session "actions" will be displayed. These "actions" allows the user to administer profile and/or trace sessions.

```

*** PROFILER 4.3.2 for Natural ***

2011-11-20 11:38   Active Profile Session: PAYROLL TEST           USER24
C A P             Active Trace Session: NO                PAYTEST
o c or
m t T Session Name      Date      Time   Owner   LastUser  Description
--- P PAYROLL EXAMPLE   2011-06-05 16:52 USER24   USER23   Example for
--- * P PAYROLL TEST    2011-06-12 15:33 USER24   USER27   Payroll Test

From PAYROLL_____ Owner _____ Profile/Trace/All P

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
Help About End Admin Backg Up Down New-S Left Right Exit

```

If the current user has a PROFILER session active, the session name is displayed to the right of "Active Profile Session". For example, the screen above displays "PAYROLL TEST" in the session name field, indicating that USER24 has a profile session named PAYROLL TEST active. "NO" in this field indicates that no profile session is active for the current user.

If the user has an active Trace session, the name of the session will appear to the right of "Active Trace Session". If no Trace session is active, this field will display "NO".

Profile sessions and/or Trace sessions can be included in the "session list" by entering 'P', 'T', or 'A' in the "Profile/Trace/All" field. Listed sessions can be limited to those "owned" (i.e., defined) by a particular user by entering a value in the "Owner" field. The "session list" can be started from a chosen position by entering a value in the "From" field. The required page of the "session list" can be entered directly in the "Page" field.

"Session actions" are two-letter acronyms and can be entered directly beside the required session. 'AU', 'S?', 'QA' and 'CU' actions are only valid for Profile sessions:

- DI Display Session Definition
- MO Modify Session Definition
- CO Copy Session Definition
- PU Purge Session Definition and Statistics
- AC Activate Session
- DA De-Activate Session
- RS Reset Statistics for Session
- AU List Active Users for Session
- S= Statistics Report for Session (most recent report format)
- S? Statistics Help (list of available report formats)
- QA Application Quality Assurance Report for Session
- CU Customized Report for Session

The following are the standard PF-keys for the PROFILER Main menu:

Key	Function	Description
PF1	Help	Provides cursor-sensitive help information
PF2	About	Displays installation, license and environment details
PF3	End	Returns to the previous screen (NATURAL Main menu)
PF4	Admin	Gives access to the Administrator options
PF6	Backg	Gives access to the Background Monitoring facility
PF7	Up	Scrolls up the "session list"
PF8	Down	Scrolls down the "session list"
PF9	New-S	Allows a new session
PF10	Left	Scrolls left on the "session list"
PF11	Right	Scrolls right on the "session list"
PF12	Exit	Exits PROFILER and returns to the NATURAL Main menu

I.4.4 Data Entry Screens

PROFILER Data Entry screens allow a user to interact with PROFILER by entering data necessary to perform PROFILER functions.

```

PRO0013: Use PF5 to access Included/Excluded Objects.
      ***
      Modify Profile Session
      SessioN PAYROLL EXAMPLE  Status NOT ACTIVE
      Desc Example for Payroll System Testers
2011-11-20 11:38 Act
C A P Act
o c o r
m t T Session Name
mo P PAYROLL EXAMPLE
__ * P PAYROLL TEST

      User      Date      Time      Version
Defined USER24 2011-06-12 15:33 431
Last USER23 2011-06-12 15:36
      Lib.Mask  Obj.Mask & Types  Source
Defined *      *      PNSMH DBID 1
Last *      *      PNSMH FUSER 29
      ADABAS Password
      Statements Executed 76
      CPU Time (secs) 0.13
From PAYROLL Database Elapsed (secs) 0.00

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help      End      Objts      Exit

```

For example, in the screen above, a user has chosen to 'MO'dify the session named "PAYROLL EXAMPLE".

I.4.5 Help Screens

PROFILER provides screen-level help on all screens and field-level help when appropriate. To invoke screen-level help, press PF1, and to invoke field-level help, enter a question mark or press PF1 while the cursor is on the field where help is desired.

Note: When invoking screen-level help, the cursor must not be on a field that has field-level help.

Pressing PF1 with the cursor away from an 'input' field on the "session list" menu displays screen-level help about the PROFILER "session list" menu.

```

2011-11-20 11:38   Help for Session Menu   USER24   PAYTEST
  ■
Active ... - Name   of any currently active 'Profile' session
Session(s) - Number of any currently active 'Trace' session.
1 Com (-mand) - Command (if any) to be actioned for that session.
C Act (-ive) - '*' in this column indicates 'active' session.
o Sessions - Names of defined Sessions.
m Date/Time - When the session was defined.
- Owner - User who defined the session.
- LastUser - User who most recently used the session.
- Description - ... of the session at definition time.
From - Starting point (if any) for listed sessions.
Owner - Only sessions owned by this user will be listed.
      If blank, sessions for all users will be listed.

PF1 - Display this Help           PF7 - Scroll Up
PF2 - PROFILER Installation       PF8 - Scroll Down
PF3 - Exit PROFILER              PF9 - Define New Session
PF4 - Administration Options     PF10 - Scroll Left
                                PF11 - Scroll Right
                                PF12 - Exit PROFILER
                                P
Ente                                12---
Help About End Admin Backg Up Down New-S Left Right Exit

```

Entering a "?" on the 'Com' field displays the field-level help screen below for "session actions" (i.e., "session commands").

```

*** PROFILER 4.3.2 for Natural ***
2011-11-20 11:38   Acti ■   Mark Command ...   USER24
C A P             Acti   _ DI Display           PAYTEST
o c or           _ MO Modify                       Page 1_ of 1
m t T Session Name _ CO Copy                       LastUser Description
_ P PAYROLL EXAMPLE _ PU Purge                     USER23 Example for
?_ P PAYROLL TEST                                USER27 Payroll Test

_ AC Activate
_ DA De-Activate
_ RS Reset Stats
_ AU Active User *

_ S= Stats Report
_ S? Stats Help *
_ QA Applic QA *
_ CU Customized *

From PAYROLL_____ Profile/Trace/All P
Only for Profile: *
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
Help About End Admin Backg Up Down New-S Left Right Exit

```

After using help screens, a user may press Enter to return to the screen from which a user invoked help.

I.4.6 Display Screens

PROFILER Display screens return PROFILER data to the user. This information may be data the user previously entered, PROFILER statistics, or PROFILER environment information.

```

PRO0013: Use PF5 to access Included/Excluded Objects.

***
          Display Profile Session
          Session PAYROLL EXAMPLE Status NOT ACTIVE
          Desc Example for Payroll System Testers
2011-11-20 11:38 Act
C A P Act
o c o r
m t T Session Name
di P PAYROLL EXAMPLE
__ * P PAYROLL TEST

          User      Date      Time      Version
Defined USER24    2011-06-12 15:33    431
Last USER23     2011-06-12 15:36
          Lib.Mask  Obj.Mask & Types  Source
Defined *          *          PNSMH DBID 1
Last *            *          PNSMH FUSER 29

          Statements Executed 76
          CPU Time (secs) 0.13
From PAYROLL Database Elapsed (secs) 0.00

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help      End      Objts      Exit

```

For example, in the screen above, information about the session PAYROLL EXAMPLE is displayed. It is currently not active, and USER23 was the last user to activate the session.

I.4.7 Report Screens

PROFILER Report screens, such as the Statement Execution Summary Report, display statistics that have been collected during an active PROFILER session.

```

PR00096: 'S'elect Object to see its Source Code Listing Report.

                Summary Report for Session
2011-11-20 11:38      Session PAYROLL EXAMPLE      USER24  PAYTEST
Report Format S  Sort Order OBJ Types _____  QA? N  Page 1__ of 1
Start Library _____ Start Object _____  View/Amend Thresholds N
                Total Stmt Execs                2263

S                T                Total  Exec  Exec  %Exec  % Graph of
e                y                Stmt  utbl  utbl  -utbl  Executable
l Library      Object p      Count  Execs Stmt  Exec  Exec  Executed
- PAYTEST     PAYROLL P      1      1    2    1    50.00  *****
- PAYTEST     PAY0100M M      4      8    2    2   100.00  *****
- PAYTEST     PAY0100P P      1     59   26   20   76.92  *****
- PAYTEST     PAY0110M M      1      4    2    2   100.00  *****
- PAYTEST     PAY0125M M      1      4    2    2   100.00  *****
- PAYTEST     PAY0130M M      1      4    2    2   100.00  *****

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help User End  Stmts CPU  Dbase Up   Down  Sourc Left  Right Exit

```

For example, in the screen above, statistics are reported for objects executed from library PAYTEST. The statistics presented in this report were stored under the session "PAYROLL EXAMPLE".

The current 'Report Format', 'Sort Order', and (shown object) 'Types' settings are shown on the next line. These values may be changed directly or by help selection windows (with PF1 or a '?'). The 'QA?' field allows direct access to the Application Quality Assurance report for the current session. The current and maximum page number of the report is located on the right of this line.

The 'Start Library' and 'Start Object' fields allow the summary report to be started from a chosen point. The 'View/Amend Thresholds?' field allows selection of various values to limit the objects included in the summary report.

Key	Function	Description
PF1	Help	Provides cursor-sensitive help information
PF2	User	Toggles between the display of the profiled library and the user who performed the profiling
PF3	End	Returns to the "session list" menu
PF4	Stmts	Changes the summary report format directly to 'S' (Statement Execution)
PF5	CPU	Changes the summary report format directly to 'C' (CPU Time Used)
PF6	Dbase	Changes the summary report format directly to 'D' (Database Elapsed Time Used)
PF7	Up	Scrolls up through the report
PF8	Down	Scrolls down through the report
PF9	Sourc	Allows the location of source code (i.e., libraries) to be changed
PF10	Left	Scrolls left through the other summary report formats
PF11	Right	Scrolls right through the other summary report formats
PF12	Exit	Exits PROFILER

I.4.8 Error Screens

PROFILER Error screens display information about an error which may occur while using PROFILER. PROFILER has a standard error-trapping program. When a NATURAL error occurs, a pop-up error window, similar to the example below, is displayed.

```
+-----+
| PRO0020: Error NAT0082 in Object PRF0010P on line (0310) Contact System
Administrator. |
+-----+
```

Pressing Enter after receiving a NATURAL error message returns the user to the PROFILER "session list" menu.

Occasionally, a NATURAL error message that is not trapped by the PROFILER error program is displayed. For example, this may occur if non-numeric data is entered in a numeric field. The error message may be too long to completely display, but the error message number is visible.

PROFILER also displays internal error messages which initiate from its assembler modules. For example, if PROFILER has expired, the message below displays in the top left corner of the screen.

```
PRO0017: PROFILER for NATURAL Trial EXPIRED..., Contact System Administrator.
```

Depending on the severity of the internal error message received, pressing Enter after receiving a message either exits PROFILER or allows processing to continue.

For an explanation of PROFILER error messages, refer to **Appendix A**.

This page intentionally left blank.

SECTION II

GETTING STARTED

II.1 **Introduction**

This section is intended for users with little or no experience using PROFILER. It guides users through a step-by-step illustration of the following basic PROFILER functions:

- Listing defined sessions
- Defining a profiling session
- Displaying a session definition
- Activating a session
- Showing the active users of a session
- Deactivating a session
- Displaying several of the PROFILER reports
- Resetting session statistics
- Purging a session

After completing this section, the user should be familiar with how PROFILER is used. This familiarity will help the user to better understand the reference information in this manual.

Users who are already familiar with the usage of PROFILER may wish to skip this section of the manual.

Note that this section purposely does not demonstrate the more advanced features of PROFILER for NATURAL, such as Background Monitoring and Enhanced Reporting. Users are encouraged to learn more about these functions by referring to the appropriate sections in the manual.

II.2 How To Begin Using PROFILER

Enter PROFILER by typing "PRF" from any library at the NATURAL Command Line, "Next" prompt, or "More" prompt.

Press Enter to display the PROFILER Main "session list" menu.

```

*** PROFILER 4.3.2 for Natural ***

2011-11-20 15:04   Active Profile Session: NO           USER24
C A P             Active Trace Session: NO           PAYTEST
o c or
m t T Session Name      Date   Time   Owner   LastUser Description
___ P PAYROLL EXAMPLE    2011-06-05 16:52 USER24   USER23   Example for
___ P PAYROLL TEST       2011-06-02 15:33 USER24   USER27   Payroll Test

From PAYROLL_____ Owner _____ Profile/Trace/All P

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10---PF11---PF12---
Help About End Admin Backg Up Down New-S Left Right Exit
    
```

Press PF9 on the "session list" menu to invoke 'Define New Profile Session', and create a profile session environment to be used for the collection of PROFILER statistics.

Note: Assign a unique, meaningful name and description to the profiling session. In this scenario, the session name is PAYROLL and the library is PAYTEST.

```

PRO0013: Use PF5 to access Included/Excluded Objects.
*** PROFILER 4.3.2 for Natural ***

2011-11-20 15:06   Active Profile Session: NO           USER24
Act ■
C A P             Define New Profile Session
o c or           Session payroll_____
m t T Session Name   Desc Getting Started with PROFILER 4.3.1_____
___ P PAYROLL EXAMPLE (Test 20 new Payroll objects. Give_____
___ P PAYROLL TEST   efficiency results to supervisor. Give_
                    QA results to Quality Assurance team.____
                    Test will be conducted by Mary B.____
                    User      Date      Time      Version
                    Defined USER24  2011-11-20 15:06  431
                    Lib.Mask  Obj.Mask & Types  Source
                    Defined paytest_ pay*_____ PNSMH DBID 1_____
                    FUSER 29_____
From PAYROLL_____ ADABAS Password _____

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10---PF11---PF12---
Help End Objts Exit
    
```

In the screen above, objects of all types in library PAYTEST whose names begin with "PAY" will be profiled. All object types are automatically selected. Remove unwanted object types from the list or press PF1 for a help selection window.

Press PF5 to access the Included/Excluded Objects window. Include Objects and Exclude Objects allow a user to specify five libraries/objects to be included or excluded from the profiling session.

The NATURAL source code to be reported resides on DBID 1 and FUSER 29. NATURAL source code is not needed to profile an application. NATURAL source code is needed for the Source Code Listing Report and the Enhanced Source Code Listing Report, which each display source code.

NATURAL source code may reside on a different DBID and FUSER than the DBID and FUSER where NATURAL object code is being profiled and where PROFILER is installed. The DBID and FUSER fields should not be changed unless PROFILER source code resides on a different DBID and FUSER than the ones where PROFILER is installed.

Press Enter to define the new session.

The 'Define New Profile Session' window is re-displayed with all input fields protected and a message confirming that the session is defined.

```

PRO0061: New Session PAYROLL Defined successfully for USER24
*** PROFILER 4.3.2 for Natural ***

2011-11-20 15:06 Active Profile Session: NO USER24
Act ■

C A P Define New Profile Session
o c or Session PAYROLL
m t T Session Name Desc Getting Started with PROFILER 4.3.1
___ P PAYROLL EXAMPLE (Test 20 new Payroll objects. Give
___ P PAYROLL TEST efficiency results to supervisor. Give
QA results to Quality Assurance team.
Test will be conducted by Mary B.
User Date Time Version
Defined USER24 2011-11-20 15:06 431

Lib.Mask Obj.Mask & Types Source
Defined PAYTEST_ PAY* PNSMH DBID 1
FUSER 29

From PAYROLL ADABAS Password

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10---PF11---PF12---
Help End Objts Exit

```

Press Enter to continue.

```

PRO0061: New Session PAYROLL Defined successfully for USER24
*** PROFILER 4.3.2 for Natural ***

2011-11-20 15:07 Active Profile Session: NO USER24
C A P Active Trace Session: NO PAYTEST
o c or Page 1_ of 1
m t T Session Name Date Time Owner LastUser Description
___ P PAYROLL 2011-11-20 15:06 USER24 Getting Star
___ P PAYROLL EXAMPLE 2011-06-05 16:52 USER24 USER23 Example for
___ P PAYROLL TEST 2011-06-02 15:33 USER24 USER27 Payroll Test

From PAYROLL Owner Profile/Trace/All P

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10---PF11---PF12---
Help About End Admin Backg Up Down New-S Left Right Exit

```

Enter action code "DI" next to session PAYROLL on the "session list" menu to invoke 'Display Profile Session Definition'.

```

*** PROFILER 4.3.2 for Natural ***

2011-11-20 15:07 Active Profile Session: NO USER24
C A P Active Trace Session: NO PAYTEST
o c or Page 1_ of 1
m t T Session Name Date Time Owner LastUser Description
di P PAYROLL 2011-11-20 15:06 USER24 Getting Star
___ P PAYROLL EXAMPLE 2011-06-05 16:52 USER24 USER23 Example for
___ P PAYROLL TEST 2011-06-02 15:33 USER24 USER27 Payroll Test

From PAYROLL Owner Profile/Trace/All P

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10---PF11---PF12---
Help About End Admin Backg Up Down New-S Left Right Exit

```

Press Enter to continue.

```

PRO0013: Use PF5 to access Included/Excluded Objects.
*** PROFILER 4.3.2 for Natural ***

2011-11-20 15:06 Active Profile Session: NO USER24
Act
C A P          Display Profile Session
o c or        Session PAYROLL          Status NEVER ACTIVE
m t T Session Name Desc Getting Started with PROFILER 4.3.1
di P PAYROLL   (Test 20 new Payroll objects. Give
P PAYROLL EXAMPLE efficiency results to supervisor. Give
P PAYROLL TEST QA results to Quality Assurance team.
Test will be conducted by Mary B.
User Date Time Version
Defined USER24 2011-11-20 15:06 431

Lib.Mask Obj.Mask & Types Source
Defined PAYTEST PAY* PNSMH DBID 1
FUSER 29

Statements Executed 0
CPU Time (secs) 0.00
Database Elapsed (secs) 0.00

From PAYROLL_____

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
Help End Objts Exit
    
```

This screen displays an existing session definition. Because this is a newly defined session, there are no statistics and the session status is "NEVER ACTIVE".

Press Enter or PF3 to return to the "session list" menu.

Enter action code "MO" next to session PAYROLL on the "session list" menu to invoke 'Modify Profile Session Definition'.

```

*** PROFILER 4.3.2 for Natural ***

2011-11-20 15:08 Active Profile Session: NO USER24
C A P          Active Trace Session: NO PAYTEST
o c or
m t T Session Name Date Time Owner LastUser Description
mo P PAYROLL 2011-06-10 15:06 USER24 USER24 Getting Star
P PAYROLL EXAMPLE 2011-06-05 16:52 USER24 USER23 Example for
P PAYROLL TEST 2011-06-02 15:33 USER24 USER27 Payroll Test

From PAYROLL_____ Owner _____ Profile/Trace/All P

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
Help About End Admin Backg Up Down New-S Left Right Exit
    
```

Press Enter to continue.

```

PRO0013: Use PF5 to access Included/Excluded Objects.
*** PROFILER 4.3.2 for Natural ***

2011-11-20 15:08 Active Profile Session: NO USER24
Act
C A P          Modify Profile Session
o c or        Session PAYROLL_____ Status NEVER ACTIVE
m t T Session Name Desc Getting Started with PROFILER 4.3.1_____
mo P PAYROLL   (Test 20 new Payroll objects. Give_____
P PAYROLL EXAMPLE efficiency results to supervisor. Give_
P PAYROLL TEST QA results to Quality Assurance team.____
Test will be conducted by Mary B._____
User Date Time Version
Defined USER24 2011-06-10 15:06 431

Lib.Mask Obj.Mask & Types Source
Defined PAYTEST_ PAY*_____ PNSMH DBID 1_____
FUSER 29_____
ADABAS Password _____

Statements Executed 0
CPU Time (secs) 0.00
Database Elapsed (secs) 0.00

From PAYROLL_____

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
Help End Objts Exit
    
```

Modify the fifth comment line by changing "Mary B" to "Susan M".

Press Enter to modify the session definition.

The 'Modify Profile Session' screen is re-displayed with a message confirming that PROFILER has successfully modified the session.

```

PRO0027: Session PAYROLL Modified Successfully.
*** PROFILER 4.3.2 for Natural ***

2011-11-20 15:08   Active Profile Session: NO           USER24
                   Act■
C A P                               Modify Profile Session
o c or                               Session PAYROLL           Status NEVER ACTIVE
m t T Session Name       Desc Getting Started with PROFILER 4.3.1
mo P PAYROLL             (Test 20 new Payroll objects. Give
  P PAYROLL EXAMPLE      efficiency results to supervisor. Give
  P PAYROLL TEST         QA results to Quality Assurance team.
                           Test will be conducted by Susan M.
                           User           Date           Time           Version
Defined USER24         2011-06-10    15:06          431
                           Lib.Mask      Obj.Mask & Types  Source
Defined PAYTEST_       PAY*          PNSMH          DBID 1
                           ADABAS Password
                           Statements Executed 0
                           CPU Time (secs) 0.00
From PAYROLL           Database Elapsed (secs) 0.00

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10---PF11---PF12---
Help           End           Objts                               Exit

```

Press Enter to return to the "session list" menu.

```

*** PROFILER 4.3.2 for Natural ***

2011-11-20 15:09   Active Profile Session: NO           USER24
C A P               Active Trace Session: NO           PAYTEST
o c or                                                     Page 1_ of 1
m t T Session Name   Date      Time      Owner      LastUser   Description
  P PAYROLL           2011-06-10 15:06 USER24     USER24     Getting Star
  P PAYROLL EXAMPLE   2011-06-05 16:52 USER24     USER23     Example for
  P PAYROLL TEST      2011-06-02 15:33 USER24     USER27     Payroll Test

From PAYROLL           Owner           Profile/Trace/All P

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10---PF11---PF12---
Help About End Admin Backg Up Down New-S Left Right Exit

```

Enter action code "AC" next to session PAYOLL on the "session list" menu to invoke 'Activate Profile Session'.

```

*** PROFILER 4.3.2 for Natural ***

2011-11-20 15:09   Active Profile Session: NO           USER24
C A P               Active Trace Session: NO           PAYTEST
o c or                                                     Page 1_ of 1
m t T Session Name   Date      Time      Owner      LastUser   Description
ac P PAYROLL           2011-06-10 15:06 USER24     USER24     Getting Star
  P PAYROLL EXAMPLE   2011-06-05 16:52 USER24     USER23     Example for
  P PAYROLL TEST      2011-06-02 15:33 USER24     USER27     Payroll Test

From PAYROLL           Owner           Profile/Trace/All P

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10---PF11---PF12---
Help About End Admin Backg Up Down New-S Left Right Exit

```

Press Enter to display the Activate 'Profile Session' screen.

```

PRO0013: Use PF5 to access Included/Excluded Objects.
*** PROFILER 4.3.2 for Natural ***

2011-11-20 15:09 Active Profile Session: NO USER24
Ac

C A P Activate Profile Session
o c or Session PAYROLL Status NEVER ACTIVE
m t T Session Name Desc Getting Started with PROFILER 4.3.1
ac P PAYROLL (Test 20 new Payroll objects. Give
P PAYROLL EXAMPLE efficiency results to supervisor. Give
P PAYROLL TEST QA results to Quality Assurance team.
Test will be conducted by Susan M.

User Date Time Version
Defined USER24 2011-06-10 15:06 431

Lib.Mask Obj.Mask & Types Source
Defined PAYTEST_ PAY* PNSMH DBID 1
FUSER 29
ADABAS Password

Statements Executed 0
CPU Time (secs) 0.00
Database Elapsed (secs) 0.00

From PAYROLL

Enter-PF1---PF2---PF3---PF
Help End Objts Exit
    
```

PROFILER displays a message 'PRO0035: Warning: Other Users already active on Session. Use "AU" to list them.' if the session is currently being used and activating it may interfere with another user's reporting process.

Assign libraries, objects, and object types to be profiled. Values in the "Library Mask", "Object Mask", and "Object Types" fields are defaults from the session definition. In the example above, all object types beginning with PAY in library PAYTEST will be profiled when they are executed by USER24.

Use PF5 to access the 'Included/Excluded Objects' window. Include Objects and Exclude Objects allow a user to specify five libraries/programs to be included or excluded from the profiling session.

Statistics will be stored in DBID 1 and File Number 29.

Press Enter to activate the session and a message is displayed confirming that the session is activated.

```

PRO0037: Session PAYROLL Activated successfully for USER24.
*** PROFILER 4.3.2 for Natural ***

2011-11-20 15:10 Active Profile Session: PAYROLL USER24
C A P Active Trace Session: NO PAYTEST
o c or Page 1_ of 1
m t T Session Name Date Time Owner LastUser Description
* P PAYROLL 2011-06-10 15:06 USER24 USER24 Getting Star
P PAYROLL EXAMPLE 2011-06-05 16:52 USER24 USER23 Example for
P PAYROLL TEST 2011-06-02 15:33 USER24 USER27 Payroll Test

From PAYROLL Owner Profile/Trace/All P

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10---PF11---PF12---
Help About End Admin Backg Up Down New-S Left Right Exit
    
```

Note: A user may have only one profiling session active at a time.

The screen above shows that session PAYROLL has been activated by USER24.

Enter action code "AU" on the "session list" menu to invoke 'Show Active Users for Session' screen.

```

*** PROFILER 4.3.2 for Natural ***

2011-11-20 15:11   Active Profile Session: PAYROLL           USER24
C A P             Active Trace Session: NO              PAYTEST
o c or
m t T Session Name      Date      Time  Owner  LastUser  Description
au * P PAYROLL          2011-11-20 15:06 USER24  USER24  Getting Star
  _ P PAYROLL EXAMPLE    2011-06-05 16:52 USER24  USER23  Example for
  _ P PAYROLL TEST       2011-06-02 15:33 USER24  USER27  Payroll Test

From PAYROLL_____ Owner _____ Profile/Trace/All P

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
Help About End Admin Backg Up Down New-S Left Right Exit

```

Press Enter to continue.

```

*** PROFILER 4.3.2 for Natural ***

2011-11-20 15:11   Active Profile Session: PAYROLL           USER24
C A P             Active Trace Session: NO              PAYTEST
o c or
m t T Session Name      Date      Time  Owner  LastUser  Description
au * P PAYROLL          2011-11-20 15:12 PAYROLL  USER24  PAYTEST
  _ P PAYROLL EXAMPLE    User      Date      Time  Mask  Mask  Object  Object
  _ P PAYROLL TEST       USER24   2011-06-10 15:09 *    *    PNSMH

From PAYROLL_____

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
Help End Up Down Exit

```

The screen above shows USER24 is the only user who has activated session PAYROLL.

Press PF3 to return to the "session list" menu.

A user must exit PROFILER to execute programs to be profiled. Press PF3 or PF12 to exit PROFILER.

Now a user is ready to profile the library and object(s) specified in the activated profiling session.

If not already on the required library, LOGON to the library specified on the Activate Session screen. This is the library where objects to be profiled reside.

Execute each of the specified objects at least once.

Note: Do not reSTOW programs during profiling.

When session profiling is complete (i.e., when all objects being profiled have been executed), a user may reenter PROFILER to access the statistics. Invoke PROFILER by entering "PRF" at the NATURAL Command Line, "Next" prompt, or "More" prompt.

Press Enter to continue.

Enter action code "DA" next to session PAYROLL on the "session list" menu to invoke 'Deactivate Profile Session'. Deactivating the session stops the collection of statistics for this user. This does not stop the collection of statistics for any other user who has the session active. When a user selects the Deactivate Session function, a user may only deactivate the session that user has active.

```

*** PROFILER 4.3.2 for Natural ***

2011-11-20 15:21   Active Profile Session: PAYROLL           USER24
C A P             Active Trace Session: NO                PAYTEST
o c or                                                  Page 1_ of 1
m t T Session Name      Date   Time   Owner   LastUser Description
da * P PAYROLL          2011-06-10 15:06 USER24   USER24   Getting Star
___ P PAYROLL EXAMPLE   2011-06-05 16:52 USER24   USER23   Example for
___ P PAYROLL TEST      2011-06-02 15:33 USER24   USER27   Payroll Test

From PAYROLL_____ Owner _____ Profile/Trace/All P

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help About End Admin Backg Up Down New-S Left Right Exit
    
```

Press Enter to continue.

```

*** PROFILER 4.3.2 for Natural ***

2011-11-20 15:21   Active Profile Session: PAYROLL           USER24
                  Ac
C A P             De-Activate Profile Session
o c or           Session PAYROLL           Status NEVER ACTIVE
m t T Session Name      Desc Getting Started with PROFILER 4.3.1
da * P PAYROLL          (Test 20 new Payroll objects. Give
___ P PAYROLL EXAMPLE   efficiency results to supervisor. Give
___ P PAYROLL TEST      QA results to Quality Assurance team.
                        Test will be conducted by Susan M.
                        User      Date      Time   Version
                        Defined USER24  2011-06-10 15:06  431
                        Lib.Mask  Obj.Mask & Types  Source
                        Defined PAYTEST  PAY*             PNSMH DBID 1
                                                FUSER 29

From PAYROLL_____ Owner _____ Profile/Trace/All P

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help      End                               Exit
    
```

Press Enter to deactivate the session and display a message which confirms that PROFILER successfully deactivated the session.

```

PRO0040: Session PAYROLL De-Activated successfully for USER24
*** PROFILER 4.3.2 for Natural ***

2011-11-20 15:22 Active Profile Session: NO USER24
C A P Active Trace Session: NO PAYTEST
o c or Page 1_ of 1
m t T Session Name Date Time Owner LastUser Description
--- P PAYROLL 2011-06-10 15:06 USER24 USER24 Getting Star
--- P PAYROLL EXAMPLE 2011-06-05 16:52 USER24 USER23 Example for
--- P PAYROLL TEST 2011-06-02 15:33 USER24 USER27 Payroll Test

From PAYROLL_____ Owner _____ Profile/Trace/All P

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
Help About End Admin Backg Up Down New-S Left Right Exit

```

Statistics are no longer collected for the objects specified for Session PAYROLL for USER24 only.

Note: A user may only deactivate the one session that the user has active.

Enter action code "S?" on the PROFILER Main "session list" menu to invoke the 'Summary Statistics Help Window'. The 'Summary Statistics' action codes allow a user to retrieve the statistics that have been calculated during an active profiling session.

```

*** PROFILER 4.3.2 for Natural ***

2011-11-20 15:22 Active Profile Session: NO USER24
C A P Active Trace Session: NO PAYTEST
o c or Page 1_ of 1
m t T Session Name Date Time Owner LastUser Description
s? P PAYROLL 2011-06 ■ Mark Report Format..
--- P PAYROLL EXAMPLE 2011-06 _ C CPU Time Used Summary
--- P PAYROLL TEST 2011-06 _ D Database Elapsed Time Summary
_ N Num,Times: Stmts, CPU, Dbase
_ P Percents: Stmts, CPU, Dbase
_ G % Graphs: Stmts, CPU, Dbase

_ F Database/Work File Loops
_ B Inline Sub/Non-proc'al Blocks
_ R FOR and/or REPEAT Loops
_ I IF and/or DECIDE Conditions
_ T Percent Executed by Stmt Type

From PAYROLL_____

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
Help About End Admin Backg Up Down New-S Left Right Exit

```

Mark report format 'S' and press Enter to display the 'Statement Execution Count' Summary Report.

```

PRO0096: 'S'elect Object to see its Source Code Listing Report.

                Summary Report for Session
2011-11-20 15:23      Session PAYROLL      USER24  PAYTEST
Report Format S  Sort Order OBJ Types _____ QA? N  Page 1__ of 1
Start Library _____ Start Object _____ View/Amend Thresholds N
      Total Stmt Execs                2263

S          T          Run          Total Exec          Exec          %Exec          % Graph of
e          y          Count         Execs Stmt         utbl -utbl      Executable
l Library  Object p          Execs Stmt         Exec  Stmt      Stmts  Statements
- PAYTEST  PAYROLL P          1          1    2          1  50.00  *****
- PAYTEST  PAY0100M M         4          8    2          2 100.00  *****
- PAYTEST  PAY0100P P          1         59   26         20  76.92  *****
- PAYTEST  PAY0110M M          1          4    2          2 100.00  *****
- PAYTEST  PAY0125M M          1          4    2          2 100.00  *****
- PAYTEST  PAY0130M M          1          4    2          2 100.00  *****

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help User End  Stmts CPU   Dbase Up   Down  Sourc Left  Right Exit
    
```

The Statement Execution Count Summary Report shows statement execution information about session PAYROLL. The report is sorted in library/object order (OBJ).

Sort order may be changed to Statement Execution Count order (STA), CPU time order (CPU), or Database Elapsed Time order (DAT).

The current 'Report Format', 'Sort Order', and (shown object) 'Types' settings are shown on the line below the session name. These values may be changed directly or by help selection windows (with PF1 or a '?').

The 'QA?' field allows direct access to the Application Quality Assurance report for the current session. The current and maximum page number of the report is located on the top right of this screen.

The 'Start Library' and 'Start Object' fields allow the summary report to be started from a chosen point. The 'View/Amend Thresholds?' field allows selection of various values to limit the objects included in the summary report.

Key	Function	Description
PF1	Help	Provides cursor-sensitive help information
PF2	User	Toggles between the display of the profiled library and the user who performed the profiling
PF3	End	Returns to the "session list" menu
PF4,	Stmts	Changes the summary report format directly to 'S' (Statement Execution)
PF5	CPU	Change the summary report format directly to 'C' (CPU Time Used)
PF6	Dbase	Change the summary report format directly to 'D' (Database Elapsed Time Used)
PF7	Up	Scrolls up through the summary report
PF8	Down	Scrolls down through the summary report
PF9	Sourc	Allows the location of source code (i.e., libraries) to be changed
PF10	Left	Scrolls left through the other summary report formats
PF11	Right	Scroll right through the other summary report formats
PF12	Exit	Exits PROFILER

All eleven Summary Report formats allow all the features and options shown on this example, including toggling/scrolling to the other report formats.

Select an object on the 'Summary Report' to invoke the Source Code Listing Report.

```

PRO0096: 'S'elect Object to see its Source Code Listing Report.

                Summary Report for Session
2011-11-20 15:23      Session PAYROLL      USER24  PAYTEST
Report Format S Sort Order OBJ Types _____ QA? N Page 1__ of 1
Start Library _____ Start Object _____ View/Amend Thresholds N
                Total Stmt Execs                2263

s                T                Total Exec  utbl  %Exec  % Graph of
e                y                Stmt  utbl  -utbl  -utbl  Executable
l Library Object p                Excs  Stmt  Stmt  Stmts  Statements
s PAYTEST PAY0100P P                1          2    1  50.00  *****
_ PAYTEST PAY0100M M                4          2    2 100.00  *****
s PAYTEST PAY0100P P                1          2    2  76.92  *****
_ PAYTEST PAY0110M M                1          2    2 100.00  *****
_ PAYTEST PAY0125M M                1          2    2 100.00  *****
_ PAYTEST PAY0130M M                1          2    2 100.00  *****

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help User End  Stmts CPU  Dbase Up   Down Sourc Left Right Exit

```

Press Enter to invoke the Source Code Listing Report.

2011-11-20 15:25 Profile Session PAYROLL			Object Profiled by USER24	
Execs	Total CPU	Avg CPU	> ____+....1....+....2...
				PAY0100P Lib PAYTEST
			0010	* Program: PAY0100P
			0020	DEFINE DATA LOCAL USING PAY0100L
			0030	END-DEFINE
1	0.003	0.003	0040	INCLUDE PAY0100C
			0050	*
1	0.005	0.005	0060	SET KEY PF8 = PGM NAMED ' +
			0070	PF20 = PGM NAMED ' +
			0080	*
10	0.326	0.003	0090	READ EMPLOYEES BY NAME STARTING FROM PERSON
<ADABAS	1005.584	100.558>		
10	0.030	0.003	0100	ADD 1 TO #I
10	0.036	0.004	0110	MOVE PERSONNEL-ID TO #ID(#I)
10	0.050	0.005	0120	COMPRESS FIRST-NAME MIDDLE-NAME INTO #NAM
10	0.040	0.004	0130	MOVE DEPT TO #DEPT(#I)
10	0.036	0.004	0140	MOVE JOB-TITLE TO #TITLE(#I)
PF1 ?	PF2 COPY	PF3 QUIT	PF4 SCAN	PF5 SC=
	PF6 SHOW			Page 1 of 5

The NATURAL source code for program PAY0100P in library PAYTEST is listed with statistics for each executable statement. If an executable statement has not executed, it is marked with a ">" immediately to the left of the source code line number. Non-executable statements such as comments and continuation lines appear in the report listing, but have no statistics and are not marked with a ">". If the first statement in a block of NATURAL Optimized Code has executed, it is marked with "O" immediately to the left of the source code line number.

The > ____ field allows a specific line number or one of the following commands to be entered. Most of these commands can be invoked by use of a PF key.

PF2	COPY	Show Copycode (at cursor)
PF3	Q{UIT}	Quit from Object Listing
PF4	SC{AN}..	SCAN for text string..
PF5	SCAN=, SC=	Repeat previous SCAN
PF6	SHOW..	Show statement options.
	T{OP}, --	Go to Top of Object
PF7	-P, -	Up one Page of Object
PF8	+P, +	Down one Page of Object
PF9	B{OT}, ++	Go to Bottom of Object
	____	Go to line ... of Object
PF10	L{EFT}, <	Scroll to Left of Page
PF11	R{IGHT}, >	Scroll to Right of Page
PF12	EXIT	Exit from PROFILER

Press PF3 to return to the 'Summary Report' format "S" (Statement Execution Summary) screen.

Press PF10 or PF11 to scroll to 'Summary Report' format "F" (Database / Work File Loops).

```

PRO0096: 'S'elect Object to see its Source Code Listing Report.

                          Summary Report for Session
2011-11-20 15:26         Session PAYROLL          USER24  PAYTEST
Report Format F  Sort Order OBJ Types _____ QA? N  Page 1__ of 1
Start Library _____ Start Object _____ View/Amend Thresholds N
                          Total Stmt Execs          2263

S          T          Total  DB/WF  %DB/WF  DB/WF  %DB/WF
e          y          Run    DB/WF  Loops   Loops  Bodies
l Library  Object  p    Count  Loops  Exec   Exec   Exec   Exec
- PAYTEST  PAYROLL P      1
- PAYTEST  PAY0100M M      4
- PAYTEST  PAY0100P P      1      1      1  100.00      1  100.00
- PAYTEST  PAY0110M M      1
- PAYTEST  PAY0125M M      1
- PAYTEST  PAY0130M P      1      1      1  100.00

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
Help User End  Stmts CPU  Dbase Up   Down  Sourc Left  Right Exit

```

This report shows that the one Database/Work File Loop in PAY0100P was executed and that at least one statement within the body of the loop was executed. The Database/Work File Loop in PAY0130M was executed but no statement within the body of the loop was executed.

For example:

```

FIND PAY-FILE WITH SSN = 999999999
WRITE 'RECORD IS ON FILE'
END-FIND

```

If the SSN equal to 999999999 does not exist, the entry condition of the FIND loop will be executed but the body of the FIND loop will never be entered. Therefore, no statements within the body of the loop are executed.

Press PF10 or PF11 until the 'Statement Execution Summary' format is displayed and then enter "Y" in the 'View/Amend Thresholds?' field to invoke the Session Thresholds window.

```

PRO0096: 'S'elect Object to see its Source Code Listing Report.

                          Summary Report for Session
2011-11-20 15:27         Session PAYROLL          USER24  PAYTEST
Report Format S  Sort Order OBJ Types _____ QA? N  Page 1__ of 1
Start Library _____ Start Object _____ View/Amend Thresholds y
                          Total Stmt Execs          2263

S          T          Total  Exec  %Exec  % Graph of
e          y          Stmt  Stmt  utbl  -utbl Executable
l Library  Object  p    Count  Execs Stmt  Exec   Exec   Executed
- PAYTEST  PAYR  p      1
- PAYTEST  PAY0
- PAYTEST  PAY0          View/Amend Thresholds
- PAYTEST  PAY0  Exclude Objects where.. ..is less than
- PAYTEST  PAY0          .. Run Count for Object _____ 0 runs
- PAYTEST  PAY0          .. Stmts Executed for Object _____ 0 statements
- PAYTEST  PAY0          .. CPU Time used by Object _____ 0.0000 msecs
- PAYTEST  PAY0          .. D'base Elapsed Time for Object _____ 0.0000 msecs

or Object's Percentage of.. ..is less than
.. Executable Stmts Executed _____ 0.00 %
.. Total Session CPU Time used _____ 0.00 %
.. Total Session D'base Elapsed _____ 0.00 %

Enter-PF1---PF2---
Help User

```

Enter '10' in the 'Stmts Executed for Object' field.

Press Enter to continue.

```

PRO0103: 5 objects EXCLUDED due to Threshold settings.

                Summary Report for Session
2011-11-20 15:29      Session PAYROLL      USER24  PAYTEST
Report Format S Sort Order OBJ Types _____ QA? N Page 1__ of 1
Start Library _____ Start Object _____ View/Amend Thresholds N
                Total Stmt Execs                2263

S                T                Total Exec      Exec  %Exec  % Graph of
e                y                Stmt utbl      utbl  -utbl Executable
l Library Object p      Run      Execs Stmt      Stmt  Stmts  Statements
_ PAYTEST PAY0100P P      1          59  26      20  76.92  *****

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help User End  Stmts CPU   Dbase Up    Down Sourc Left Right Exit
    
```

In this scenario, only program PAY0100P in library PAYTEST satisfied the threshold entered.

Message 'PRO0103: 5 objects EXCLUDED due to Threshold settings.' is displayed at the top of the screen.

Enter "Y" in the 'QA?' field on the 'Summary Report' screen to invoke the Application QA Report. Mark the library required for the QA report (i.e., PAYTEST).

```

PRO0103: 5 objects EXCLUDED due to Threshold settings.

                Summary Report for Session
2011-11-20 15:29      Session PAYROLL      USER24  PAYTEST
Report Format S Sort Order OBJ Types _____ QA? y Page 1__ of 1
Start Library _____ Start Object _____ View/Amend Thresholds N
                Total Stmt Execs                2263

S                T                Total Exec      Exec  %Exec  % Graph of
e                y                Stmt utbl      utbl  -utbl Executable
l Library Object p      Run      Execs Stmt      Stmt  Stmts  Statements
_ PAYTEST PAY0100P P      1          59  26      20  76.92  *****

      ■
                Mark Profiled Library ...
                ... for Session PAYROLL
      x PAYTEST

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help User End  Stmts CPU   Dbase Up    Down Sourc Left Right Exit
    
```

Press Enter to continue.

```

PRO0096: 'S'elect Object to see its Source Code Listing Report.

                          Summary Report for Session
2011-11-20 15:30          Session PAYROLL          USER24  PAYTEST
QA Report:              52 Objects in PAYTEST_    View Executed
of which                6 ( 11.54% ) were Executed.  Objects? N
Objects NOT Executed starting.. _____ types.. _____ Page 1_ of 2

S          T Exec S          T Exec S          T Exec S          T Exec
e          y utbl e          y utbl e          y utbl e          y utbl
l Object p Stmt l Object p Stmt l Object p Stmt l Object p Stmt
- CITYTAXL L 1 - FICAM M 9 - LIFEINSS S 3 - PAY0120M M 3
- CITYTAXM M 9 - FICAP P 14 - PAYBATCH P 10 - PAY0120P P 30
s CITYTAXP P 13 - FICAS S 3 - PAYEMPL L 0 - PAY0120T M 2
- CITYTAXS S 3 - KAH0100M M 2 - PAYKH P 30 - PAY0130P P 28
- CITYTX2L L 0 - KAH0100P P 26 - PAYL L 0 - PAY0130T M 2
- FEDTAXL L 1 - KAH1080 P 874 - PAYLOCL L 0 - PAY0140M M 2
- FEDTAXM M 9 - KHBIBM M 2 - PAYROLLG C 1 - PAY0140P P 25
- FEDTAXP P 15 - LIFEINSL L 1 - PAY0100T M 2 - PAY0140T M 2
- FEDTAXS S 3 - LIFEINSM M 6 - PAY0110P P 27 - PENSIONL L 0
- FICAL L 1 - LIFEINSP P 13 - PAY0110T M 2 - PENSIONM M 6

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help      End      Up      Down      Sourc      Exit
    
```

For extended reporting capabilities, refer to the **Enhanced Reporting** section of the PROFILER manual.

Press PF3 to return to the PROFILER "session list" menu.

Enter action code "RS" next to session PAYROLL on the "session list" menu to invoke 'Reset Session Statistics'. Once on-line and batch reporting is complete, session statistics may be reset so the session may be used to collect new statistics and begin profiling again.

```

*** PROFILER 4.3.2 for Natural ***

2011-11-20 15:32      Active Profile Session: NO      USER24
C A P                Active Trace Session: NO      PAYTEST
o c o r
m t T Session Name      Date      Time      Owner      LastUser      Description
rs P PAYROLL            2011-06-10 15:06 USER24      USER24      Getting Star
___ P PAYROLL EXAMPLE    2011-06-05 16:52 USER24      USER23      Example for
___ P PAYROLL TEST       2011-06-02 15:33 USER24      USER27      Payroll Test

From PAYROLL_____ Owner _____ Profile/Trace/All P

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help About End Admin      Backg Up      Down New-S Left Right Exit
    
```

Press Enter to continue.

```

PRO0116: You may limit Reset of Stats by User, Library, Object, and/or Type
*** PROFILER 4.3.2 for Natural ***

2011-11-20 15:32 Active Profile Session: NO USER24
Ac

C A P Reset Statistics for Profile Session
o c or Session PAYROLL Status NOT ACTIVE
m t T Session Name Desc Getting Started with PROFILER 4.3.1
rs P PAYROLL (Test 20 new Payroll objects. Give
_ P PAYROLL EXAMPLE efficiency results to supervisor. Give
_ P PAYROLL TEST QA results to Quality Assurance team.
Test will be conducted by Susan M.

Reset User Date Time Version
only for USER24_ 2011-06-10 15:06 431
Last USER24 2011-06-10 15:10
Reset Lib.Mask Obj.Mask & Types Source
only for PAYTEST_ PAY*_ PNSMH DBID 1
Last PAYTEST PAY*_ PNSMH FUSER 29

Statements Executed 2263
CPU Time (secs) 2.03
From PAYROLL_ Database Elapsed (secs) 1906.78

Enter-PF1---PF2---PF3---P--
Help End
    
```

Confirm session reset by pressing Enter. Press PF3 to avoid reset of statistics.

```

PRO0115: 12 'Detail' Records DELETED for Session PAYROLL
*** PROFILER 4.3.2 for Natural ***

2011-11-20 15:33 Active Profile Session: NO USER24
C A P Active Trace Session: NO PAYTEST
o c or Page 1_ of 1
m t T Session Name Date Time Owner LastUser Description
_ P PAYROLL 2011-11-20 15:06 USER24 USER24 Getting Star
_ P PAYROLL EXAMPLE 2011-06-05 16:52 USER24 USER23 Example for
_ P PAYROLL TEST 2011-06-02 15:33 USER24 USER27 Payroll Test

From PAYROLL_ Owner _ Profile/Trace/All P

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10---PF11---PF12---
Help About End Admin Backg Up Down New-S Left Right Exit
    
```

A message is displayed to confirm that PROFILER successfully reset the session statistics.

The statistics have been deleted. A user may activate the session at any time to collect new statistics.

Enter action code "PU" next to session PAYROLL on the "session list" menu to invoke 'Purge Profile Session'. If the session is no longer needed, purge (i.e., delete) the session to free PROFILER internal allocations. There is a limit to the number of profile sessions that can be defined in PROFILER at any one time.

```

*** PROFILER 4.3.2 for Natural ***

2011-11-20 15:34 Active Profile Session: NO USER24
C A P Active Trace Session: NO PAYTEST
o c or Page 1_ of 1
m t T Session Name Date Time Owner LastUser Description
pu P PAYROLL 2011-06-10 15:06 USER24 USER24 Getting Star
___ P PAYROLL EXAMPLE 2011-06-05 16:52 USER24 USER23 Example for
___ P PAYROLL TEST 2011-06-02 15:33 USER24 USER27 Payroll Test

From PAYROLL_____ Owner _____ Profile/Trace/All P

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
Help About End Admin Backg Up Down New-S Left Right Exit

```

Press Enter to continue.

```

PRO0102: Answer MUST be 'Y' or 'N'.
*** PROFILER 4.3.2 for Natural ***

2011-11-20 15:34 Active Profile Session: NO USER24
Ac
C A P Purge Profile Session
o c or Session PAYROLL Status NOT ACTIVE
m t T Session Name Desc Getting Started with PROFILER 4.3.1
pu P PAYROLL (Test 20 new Payroll objects. Give
___ P PAYROLL EXAMPLE efficiency results to supervisor. Give
___ P PAYROLL TEST QA results to Quality Assurance team.
Test will be conducted by Susan M.
User Date Time Version
Defined USER24 2011-06-10 15:06 431
Last USER24 2011-06-10 15:10
Lib.Mask Obj.Mask & Types Source
Defined PAYTEST PAY* PNSMH DBID 1
Last PAYTEST PAY* PNSMH FUSER 29
Proceed with PURGE of this Session? y
Statements Executed 2263
CPU Time (secs) 2.03
From PAYROLL_____ Database Elapsed (secs) 1906.78

Enter-PF1---PF2---PF3---P-
Help End Exit

```

To purge the session, enter "Y" next to the 'Proceed with PURGE of this Session?' text and press Enter.

```

PRO0118: Session PAYROLL Purged. 10 PROFILER records deleted in total.
*** PROFILER 4.3.2 for Natural ***

2011-11-20 15:35 Active Profile Session: NO USER24
C A P Active Trace Session: NO PAYTEST
o c or Page 1_ of 1
m t T Session Name Date Time Owner LastUser Description
___ P PAYROLL EXAMPLE 2011-06-05 16:52 USER24 USER23 Example for
___ P PAYROLL TEST 2011-06-02 15:33 USER24 USER27 Payroll Test

From PAYROLL_____ Owner _____ Profile/Trace/All P

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
Help About End Admin Backg Up Down New-S Left Right Exit

```

A message is displayed confirming that PROFILER deleted the session. PROFILER deleted the session and all PROFILER records associated with the session.

Press PF3 or PF12 to exit PROFILER.

This page intentionally left blank.

SECTION III

MAIN "SESSION LIST" MENU

III.1 Menu Structure

The PROFILER menu structure consists of a "session list" with a number of "session actions" available for each session.

```
*** PROFILER 4.3.2 for Natural ***
■
2011-11-20 16:42 Acti Mark Command ... USER24
C A P Acti _ DI Display PAYTEST
o c or _ MO Modify Page 1_ of 1
m t T Session Name X CO Copy LastUser Description
_ P PAYROLL EXAMPLE _ PU Purge USER23 Example for
co P PAYROLL TEST USER27 Payroll Test

_ AC Activate
_ DA De-Activate
_ RS Reset Stats
_ AU Active User *

_ S= Stats Report
_ S? Stats Help *
_ QA Applic QA *
_ CU Customized *

From PAYROLL_____ Profile/Trace/All P
Only for Profile: *
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
Help About End Admin Backg Up Down New-S Left Right Exit
```

Profile sessions and/or Trace sessions can be included in the "session ". Listed sessions can be limited to those "owned" (i.e., defined) by a particular user. The "session list" can be started from a chosen position. PF keys are provided for scrolling through the "session list".

"Session actions" are two-letter acronyms and can be entered directly beside the required session. Some actions are only valid for Profile sessions. PF keys are available to access help, environment, administrator, and background monitoring functions.

III.2 Invoking PROFILER

Enter PROFILER by typing "PRF" from any library at the NATURAL Command Line, "Next" prompt, or "More" prompt. Press Enter to display the PROFILER Main "session list" menu.

```

*** PROFILER 4.3.2 for Natural ***

2011-11-20 16:44   Active Profile Session: PAYROLL TEST           USER24
C A P             Active Trace Session: NO                   PAYTEST
o c or
m t T Session Name      Date      Time   Owner   LastUser  Description
--- P PAYROLL EXAMPLE   2011-06-05 16:52 USER24   USER23   Example for
--- * P PAYROLL TEST    2011-06-12 15:33 USER24   USER27   Payroll Test

From PAYROLL_____ Owner _____ Profile/Trace/All P

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help About End Admin      Backg Up   Down New-S Left Right Exit

```

This is the main menu for PROFILER. All functions can be accessed from this screen via "session actions" or PF keys. After performing any function, the user is returned by default to this menu. Basic information about the user's current PROFILER environment is shown at the head of this menu.

If the current user has a PROFILER session active, the session name is displayed to the right of "Active Profile Session". For example, the screen above displays "PAYROLL TEST" in the session name field, indicating that USER24 has a profile session named PAYROLL TEST active. "NO" in this field indicates that no profile session is active for the current user.

If the user has an active Trace session, the name of the session will appear to the right of "Active Trace Session". If no Trace session is active, this field will display "NO".

PROFILER can be invoked directly from any library (on the database where it has been installed). When PROFILER is invoked, it immediately establishes PROLIB as a step library (using Software AG subprogram USR1025N). In this way, the PROFILER software can be used without moving the user from their original library.

On first invoking PROFILER, only sessions owned by that user-id will be displayed. However, PROFILER records each user's most recent PROFILER settings and options. Upon re-entering PROFILER, these settings and options are restored so that the user returns to the same "place".

III.3 Controlling the "Session List"

Sessions matching the 'From', 'Owner', and 'Profile/Trace/All' settings will be displayed on the Main "session list" menu.

Listed sessions can be limited to those "owned" (i.e., defined) by a particular user by entering a value in the "Owner" field. Leave the "Owner" field blank to list sessions "owned" by all users.

The "session list" can be started from a chosen position by entering a value in the "From" field. Leave the "From" field blank to list all sessions matching the other settings.

Profile sessions and/or Trace sessions can be included in the "session list" by entering 'P', 'T', or 'A' in the "Profile/Trace/All" field.

The required page of the "session list" can be entered directly in the "Page" field. The entered value must not be less than one (1) or greater than the last page of the listing.

III.4 "Session Actions" on the "Session List"

"Session actions" are two-letter acronyms and can be entered in the 'Com' field beside the session that is to be processed. (Please note that actions 'AU', 'S?', 'QA', and 'CU' actions are only valid for Profile sessions).

Entering a "?" on the 'Com' field displays the field-level help screen below for "session actions" (i.e., "session commands").

```

*** PROFILER 4.3.2 for Natural ***
  2011-11-20 16:48  Acti  Mark Command ...                USER24
C A P              Acti  _ DI Display                    PAYTEST
o c or            _ MO Modify                    Page 1_ of 1
m t T Session Name _ CO Copy                    LastUser Description
_ P PAYROLL EXAMPLE _ PU Purge                    USER23 Example for
?_ P PAYROLL TEST   _ AC Activate                    USER27 Payroll Test
                  _ DA De-Activate
                  _ RS Reset Stats
                  _ AU Active User *
                  _ S= Stats Report
                  _ S? Stats Help *
                  _ QA Applic QA *
                  _ CU Customized *
From PAYROLL_____ Profile/Trace/All P
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
Help About End Admin Backg Up Down New-S Left Right Exit

```

Mark the command that is required and press enter. The marked command will be copied into the 'Com' field ready for processing.

Refer to the **Session Maintenance and Execution** section for more information about the following "session actions":

- DI Display Session Definition
- MO Modify Session Definition
- CO Copy Session Definition
- PU Purge Session Definition and Statistics
- AC Activate Session
- DA De-Activate Session
- RS Reset Statistics for Session
- AU List Active Users for Session

Refer to the **Session Reporting** section for more information about the following "session actions":

- S= Statistics Report for Session (most recent report format)
- S? Statistics Help (list of available report formats)
- SS Statement Execution Count Summary Report
- SC CPU Time Used Summary Report
- SD Database Elapsed Time Summary Report
- SN Number of Statements, CPU Time, Database Summary
- SP Percentage of Executions, CPU and Database Summary
- SG Percent Graph of Executions, CPU and Database Summary
- SF Database/Work File Loops Summary Report
- SB Internal Subroutines/Non-Procedural Blocks Summary
- SR FOR/REPEAT Loops Summary Report
- SI IF/DECIDE Conditions Summary Report
- ST Percent Executed by Statement Type Summary Report
- QA Application Quality Assurance Report for Session
- CU Customized Report for Session

III.5 **PF Keys on the "Session List"**

The following are the standard PF-keys for the PROFILER Main "session list" menu:

Key	Function	Description
PF1	Help	Provides cursor-sensitive help information (refer to the "Session List" Help section for more information)
PF2	About	Displays installation, license, and environment details (refer to the About Screen section for more information)
PF3	End	Returns to the previous screen (NATURAL Main menu)
PF4	Admin	Gives access to the Administrator options (refer to the PROFILER Administration section for more information)
PF6	Backg	Gives access to the Background Monitoring facility (refer to the Background Monitoring section for more information)
PF7	Up	Scrolls up the "session list"
PF8	Down	Scrolls down the "session list"
PF9	New-S	Allows a new session to be defined (refer to the Session Maintenance and Execution section for more information)
PF10	Left	Scrolls left on the "session list"
PF11	Right	Scrolls right on the "session list"
PF12	Exit	Exits PROFILER and returns to the NATURAL Main menu

III.6 "Session List" Help

PROFILER provides screen-level help on all screens and field-level help when appropriate. To invoke screen-level help, press PF1, and to invoke field-level help, enter a question mark or press PF1 while the cursor is on the field where help is desired.

Note: When invoking screen-level help, the cursor must not be on a field that has field-level help.

Pressing PF1 with the cursor away from an 'input' field on the "session list" menu displays screen-level help about the PROFILER "session list" menu.

```

2011-11-20 16:49      Help for Session Menu      USER24      PAYTEST

  ■ Active ... - Name of any currently active 'Profile' session
  Session(s) - Number of any currently active 'Trace' session.
1 Com (-mand) - Command (if any) to be actioned for that session.
C Act (-ive) - '*' in this column indicates 'active' session.
o Sessions - Names of defined Sessions.
m Date/Time - When the session was defined.
- Owner - User who defined the session.
- LastUser - User who most recently used the session.
- Description - ... of the session at definition time.
From - Starting point (if any) for listed sessions.
Owner - Only sessions owned by this user will be listed.
      If blank, sessions for all users will be listed.

PF1 - Display this Help          PF7 - Scroll Up
PF2 - PROFILER Installation      PF8 - Scroll Down
PF3 - Exit PROFILER             PF9 - Define New Session
PF4 - Administration Options    PF10 - Scroll Left
                                PF11 - Scroll Right
                                PF12 - Exit PROFILER
PF6 - Background Monitoring

```

III.7 'About' Screen

Pressing PF2 on the PROFILER "session list" menu displays the PROFILER 'About' screen.

```

PPPPP  RRRRR  OOOOOO  FFFFFF  IIIIII  LL  EEEEE  RRRRR
PP PP  RR RR  OO OO  FF      II  LL  EE  RR RR
PP PP  RR RR  OO OO  FFFF  II  LL  EEEE  RR RR
PP PP  RR RR  OO OO  FF      II  LL  EE  RR RR
PP PP  RR RR  OO OO  FF      II  LL  EE  RR RR
PPP   RR RR  OOOOOO  FF      IIIIII  LLLLLL  EEEEE  RR  RR
PP    RR  RR
PP    RR.  RR
PP
PP
PP .

          f o r
          N A T U R A L
          * * *

COPYRIGHT
TREEHOUSE SOFTWARE
2605 Nicholson Road Suite 230
Sewickley PA 15143 USA
phone: (724) 759 7070
fax: (724) 759 7067
http\www.treehouse.com
Statistics can be collected for up to 1080 statements.

          TP Mon TSO
          Version 4.3.1
          Released 04/02/12
          Zap Level 0000
          ADALNK
          Only Authorized for Use by
          Licensee until 2012-12-31

          DBID  FNR
          Zap 0  0
          LF=120 4  120
          LF=122 4  122

```

4.3.1 Pressing Enter will return the user to the PROFILER "session list" menu.

This page intentionally left blank.

SECTION IV

SESSION MAINTENANCE AND EXECUTION

IV.1 **Introduction to Session Maintenance and Execution**

This section describes the PROFILER functions used to define sessions, to manage sessions, and to manage the collection of PROFILER statistics. A session defines an environment where users may establish parameters that PROFILER uses to collect statistics about NATURAL applications and objects. These statistics are saved in a PROFILER repository by session.

Profiling and Tracing are based on a library, a range of objects, and either object types (to be profiled) or statement types (to be traced) specified at session activation. Statistics are only calculated when the objects specified for the session are executed by a user for whom that session is active.

Profile sessions report which object statements were executed, how many times they were executed, how much CPU and Database time they used, which statements were not executed, and which statements were not executable. A range of summary reports are available for Profile sessions, as well as the detailed 'Source Code Listing' report for each object executed.

Trace sessions only report object statements in the order they were executed. No CPU or Database time usage is collected during Tracing.

When a session is manually activated by a user, PROFILER collects session statistics when the user activates the session and stops collecting statistics when the user deactivates the session. PROFILER also stops collecting statistics when the user's NATURAL session is terminated.

The Background Monitoring function (refer to the **Background Monitoring** section for more information) allows a Profile session to be automatically activated for a user by the user or by a third party, based on library/object masks, NATURAL object types, and date/time. Use of the Background Monitoring function is optional and does not prevent users from manually activating Profile sessions.

When Background Monitoring is in effect for a particular user or library, the specified session is activated when a user logs on to a library specified in the Background Monitoring parameters. The session remains active until the user logs off that library or the NATURAL session is terminated.

<p>Note: Use of the NATURAL Debugging Facility during profiling or tracing can interfere with the collection of statistics by PROFILER. For more information, refer to the Considerations When Using the NATURAL Statement Interface for Collection of PROFILER Statistics sub-section of the Clarification of PROFILER Statistics section.</p>
--

PROFILER provides the following functions that allow users to control session definition and execution:

(PF9) Define New Session

Adds and defines a new session.

(DI) Display Session Definition

Displays information about a session that a user has previously defined.

(MO) Modify Session Definition

Changes the definition of a previously defined session.

(CO) Copy Session Definition

Copies the definition of a previously defined session to a new session.

(PU) Delete Existing Session

Deletes a session and all PROFILER statistics associated with the session.

(AC) Activate Session

Activates a session to begin collecting statistics.

(DA) Deactivate Session

Deactivates a session to stop collecting statistics.

(RS) Reset Session Statistics

Resets the statistics for a session. Resetting statistics for a session deletes the statistics but not the session definition from the PROFILER repository. The session may be re-activated to collect new statistics.

(AU) Show Session Active Users

Displays all users who are active on a particular session.

Note: A user may have only one Profile session AND one Trace session active at a time.

"Session actions" are two-letter acronyms and can be entered directly beside the required session. (The 'AU' action is only valid for Profile sessions.) Apart from 'Define New Session', the session maintenance and execution functions are available on the Main "session list" menu either by direct entry of the "session action" or by use of the help selection window shown below.

```

*** PROFILER 4.3.2 for Natural ***
2011-11-20 16:48 Acti Mark Command ... USER24
C A P Acti _ DI Display PAYTEST
o c or _ MO Modify Page 1_ of 1
m t T Session Name X CO Copy LastUser Description
_ P PAYROLL EXAMPLE _ PU Purge USER23 Example for
co P PAYROLL TEST _ AC Activate USER27 Payroll Test
_ DA De-Activate
_ RS Reset Stats
_ AU Active User *
_ S= Stats Report
_ S? Stats Help *
_ QA Applic QA *
_ CU Customized *
From PAYROLL Profile/Trace/All P
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
Help About End Admin Backg Up Down New-S Left Right Exit
    
```

Refer to the **Main "Session List" Menu** section for more details about this screen.

The 'Define New Profile Session' screen is displayed below.

```

PRO0013: Use PF5 to access Included/Excluded Objects.
*** PROFILER 4.3.2 for Natural ***

2011-11-2015:06 Active Profile Session: NO USER24
Act █

C A P Define New Profile Session
o c o r Session payroll profile_
m t T Session Name Desc Getting Started with PROFILER 4.3.1-
P PAYROLL (Test 20 new Payroll objects. Give
P PAYROLL TEST efficiency results to supervisor. Give
QA results to Quality Assurance team.

User Date Time Version
Defined USER24 2011-06-10 15:06 431

Lib.Mask Obj.Mask & Types Source
Defined paytest_ pay* PNSMH DBID 1
FUSER 29
ADABAS Password

From PAYROLL

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
Help End Objts Exit
    
```

The 'Define New Trace Session' screen is displayed below.

```

*** PROFILER 4.3.2 for Natural ***

2011-11-2015:06 Active Profile Session: NO USER24
Act █

C A P Define New Trace Session
o c o r Session payroll trace_
m t T Session Name Desc Trace the execution of three new
P PAYROLL Payroll objects. Examine results.
P PAYROLL TEST

User Date Time Version
Defined USER24 2011-06-10 15:06 431

Lib.Mask Obj.Mask Types Source
Defined paytest_ pay* DIXCPFRO DBID 1
Statements to FUSER 29

From PAYROLL

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
Help End Exit
    
```

Field	Description
Session	The name the user assigns to the session. The name must begin with an alphanumeric character and contain 1 to 16 characters.
Desc(ription)	A description of the session. Upper and lower case may be used.
(Defined) User, Date, Time	The user, date, and time at session definition.
(Defined) Library Mask	The NATURAL library or libraries to be profiled or traced. Mask options include the following: * Matches any character in remainder of string ? Matches any single character . Matches any numeric character @ Matches any non-numeric character
(Defined) Object Mask	The NATURAL object or objects to be profiled or traced.

(continued from the previous page)

Field	Description
(Defined) Object Types (Profile Sessions Only)	<p>The NATURAL object types to be profiled.</p> <p>P Program N Subprogram S Subroutine M Map H Helproutine</p> <p>Press PF1 on this field to access a help selection window.</p>
(Defined) Statement Types (Trace Sessions Only)	<p>The NATURAL statement types to be traced.</p> <p>D Database I I/O X Calls C CallNat P Perform F Fetch R For/Repeat O Other</p> <p>Press PF1 to access a help selection window.</p>
Source DBID / FUSER	<p>The number of the database and FUSER where PROFILER accesses NATURAL source code. These fields default to the DBID and FUSER where PROFILER is installed.</p> <p>These fields should not be changed unless source code resides on a different DBID/FUSER. These fields are only used by PROFILER for the Source Code Listing Report the Enhanced Source Code Listing Report, and the Trace Source Code Report.</p>
Statements ____ to ____ (Trace Sessions Only)	<p>If only one library/object is to be traced, the 'from' and 'to' statement numbers may be specified in order to reduce the amount of Trace output.</p>
ADABAS PASSWORD (Profile Sessions Only)	<p>A non-display field that is required if the PROFILER File (FNR) is protected by ADABAS security.</p>

(continued from the previous page)

Field	Description
Included Objects Press PF5 to view/amend these settings (Profile Sessions Only)	Allows a user to specify five libraries/objects to be included in the Profile session. Mask options include the following: * Matches any character in remainder of string ? Matches any single character . Matches any numeric character @ Matches any non-numeric character
Excluded Objects (Profile Sessions Only)	Allows a user to specify five libraries/ objects to be excluded from the Profile session.

The following PF-keys are provided for the Define New Session screens.

Key	Function	Description
PF1	Help	Provides help information about Define New Session
PF3	End	Returns to the "session list" menu (and the new session is NOT defined)
PF5	Objts	Access to the 'Included / Excluded Objects' window
PF12	Exit	Exits PROFILER

Pressing Enter adds a new session definition and displays a message "PRO0061: New Session {name} Defined successfully for (user)" confirming that PROFILER successfully defined the session. The user is returned to the "session list" menu and is positioned at the entry for the newly defined session.

If users attempt to define more than the maximum number of sessions, PROFILER displays the message "PRO0058: Maximum {current-maximum} sessions used. Delete one before adding another." at the top of the screen.

IV.3 Display Session Definition

The Display Session Definition function allows a user to display information about a previously defined session. To Display a session's definition, enter action code "DI" next to the session that is to be displayed.

```

PRO0061: New Session PAYROLL TRACE Defined successfully for USER24
*** PROFILER 4.3.2 for Natural ***

2011-11-2015:07 Active Profile Session: NO USER24
C A P Active Trace Session: NO PAYTEST
o c or Page 1_ of 1
m t T Session Name Date Time Owner LastUser Description
di P PAYROLL 2011-06-05 16:52 USER24 USER23 Example for
___ P PAYROLL TEST 2011-06-02 15:33 USER24 USER27 Payroll Test
___ T PAYROLL TRACE 2011-06-10 15:06 USER24 Payroll Trace

From PAYROLL_____ Owner _____ Profile/Trace/All A
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---

```

Pressing Enter displays the 'Display Profile Session' screen or the 'Display Trace Session' screen depending on the type of session selected. The 'Display ... Session' fields are not modifiable.

```

PRO0013: Use PF5 to access Included/Excluded Objects.
*** PROFILER 4.3.2 for Natural ***

2011-11-2015:07 Active Profile Session: NO USER24
Act
C A P Display Profile Session
o c or Session PAYROLL Status NEVER ACTIVE
m t T Session Name Desc Getting Started with PROFILER 4.3.1
di P PAYROLL (Test 20 new Payroll objects. Give
___ P PAYROLL TEST efficiency results to supervisor. Give
___ T PAYROLL TRACE QA results to Quality Assurance team.
Test will be conducted by Mary B.
User Date Time Version
Defined USER24 2011-06-10 15:06 431
Last
Lib.Mask Obj.Mask & Types Source
Defined PAYTEST PAY* PNSMH DBID 1
Last FUSER 29

Statements Executed 0
CPU Time (secs) 0.00
From PAYROLL_____ Database Elapsed (secs) 0.00

Enter-PF1---PF2---PF3---PF
Help End Objts Exit

```

```

*** PROFILER 4.3.2 for Natural ***

2011-11-2015:07   Active Profile Session: NO           USER24
                  Act
C A P
o c or           Display Trace Session
m t T Session Name   Desc Trace the execution of three new
                  P PAYROLL           Payroll objects.  Examine results.
                  P PAYROLL TEST
di T PAYROLL TRACE

                  User      Date      Time      Version
Defined USER24   2011-06-10  15:06    431
                  Statement
                  Lib.Mask  Obj.Mask  Types     Source
Defined PAYTEST  PAY*     DIXCPFRO DBID 1
                  Statements to      FUSER 29

From PAYROLL_____ 0 Statements have been TRACEd so far.

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
Help      End                               Exit

```

Field	Description
Session	The name the user assigns to the session. The name must begin with an alphanumeric character and contain 1 to 16 characters.
Session Status (Profile Sessions Only)	The status of this session. Valid values include the following: ACTIVE, NOT ACTIVE, or NEVER ACTIVE.
Desc(ription)	A description of the session. Upper and lower case may be used.
(Defined) User, Date, Time	The user, date, and time at session definition.
(Last) User, Date, Time (Profile Sessions Only)	The user, date, and time at the last session activation. (These fields are blank if the session has never been activated.)
(Defined) Library Mask	The NATURAL library or libraries to be profiled or traced. Mask options include the following: <ul style="list-style-type: none"> * Matches any character in remainder of string ? Matches any single character . Matches any numeric character @ Matches any non-numeric character
(Defined) Object Mask	The NATURAL object or objects to be profiled or traced.

(continued from the previous page)

Field	Description
(Defined) Object Types (Profile Sessions Only)	<p>The NATURAL object types to be profiled.</p> <p>P Program N Subprogram S Subroutine M Map H Helproutine</p> <p>Press PF1 on this field to access a help selection window.</p>
(Defined) Statement Types (Trace Sessions Only)	<p>The NATURAL statement types to be traced.</p> <p>D Database I I/O X Calls C CallNat P Perform F Fetch R For/Repeat O Other</p> <p>Press PF1 to access a help selection window.</p>
(PROFILER) Version	The PROFILER version installed when the session was defined.
(Last) Library Mask (Profile Sessions Only)	The library mask used when the session was last activated.
(Last) Object Mask (Profile Sessions Only)	The object mask used when the session was last activated.
(Last) Object Types (Profile Sessions Only)	The object types used when the session was last activated.
Source DBID / FUSER	The number of the database and FUSER where PROFILER accesses NATURAL source code. These fields default to the DBID and FUSER where PROFILER is installed.
Statements _____ to _____ (Trace Sessions Only)	If only one library/object is to be traced, the 'from' and 'to' statement numbers may be specified in order to reduce the amount of Trace output.

(continued from the previous page)

Field	Description
Included Objects Press PF5 to view/amend these settings (Profile Sessions Only)	Allows a user to specify five libraries/objects to be included in the Profile session. Mask options include the following: * Matches any character in remainder of string ? Matches any single character . Matches any numeric character @ Matches any non-numeric character
Excluded Objects (Profile Sessions Only)	Allows a user to specify five libraries/objects to be excluded from the Profile session.
Statements Exec (Profile Sessions Only)	The number of statements executed for this session by all session users. This field is blank if the session has never been activated.
CPU Time (sec) (Profile Sessions Only)	The total CPU time in seconds that this session has used. This field is blank if the session has never been activated.
Database Elapsed (sec) (Profile Sessions Only)	The total Database Elapsed Time in seconds that this session has used. This field is blank if the session has never been activated.
Statements TRACEd so far (Trace Sessions Only)	The number of statements traced for this session. This field is zero if the session has never been activated.

The following PF-keys are provided for the 'Display ... Session' screens.

Key	Function	Description
PF1	Help	Provides help information about 'Display ... Session'
PF3	End	Returns to the "session list" menu
PF5	Objts	Access to the 'Included / Excluded Objects' window
PF12	Exit	Exits PROFILER

Pressing Enter returns to the "session list" menu.

IV.4 Modify Session Definition

The Modify Session Definition function allows a user to change information about a previously defined session.

This function also allows a user to modify a library or range of libraries, an object or range of objects, and object types to be profiled. If the PROFILER file where PROFILER stores statistics is protected by ADABAS security, the ADABAS Password may also be modified. These values will become the default parameters for the Activate Session function.

A user may change the Source Code DBID and FUSER where PROFILER accesses NATURAL source code. NATURAL source code is only used by the Source Code Listing Report, the Enhanced Source Code Listing Report, and the Trace Source Code Report.

To Modify a session's definition, enter action code "MO" next to the session that is to be modified.

```

*** PROFILER 4.3.2 for Natural ***

2011-11-2015:08   Active Profile Session: NO           USER24
C A P             Active Trace Session: NO       PAYTEST
o c or
m t T Session Name      Date      Time   Owner   LastUser Description
mo P PAYROLL            2011-06-05 16:52 USER24  USER23  Example for
  P PAYROLL TEST       2011-06-02 15:33 USER24  USER27  Payroll Test
  T PAYROLL TRACE      2011-06-10 15:06 USER24           Payroll Trace

From PAYROLL_____ Owner _____ Profile/Trace/All A
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
```

If the session is already active (for any user) the message "PRO0011: Warning - Session is ACTIVE. Press <enter> to execute your command." is displayed.

Pressing Enter displays the 'Modify Profile Session' screen or the 'Modify Trace Session' screen depending on the type of session selected.

```

PRO0013: Use PF5 to access Included/Excluded Objects.
*** PROFILER 4.3.2 for Natural ***

2011-11-2015:09   Active Profile Session: NO           USER24
                  Ac
C A P             Modify Profile Session
o c or            Session PAYROLL_____ Status NEVER ACTIVE
m t T Session Name Desc Getting Started with PROFILER 4.3.1_____
mo P PAYROLL      (Test 20 new Payroll objects. Give_____
  P PAYROLL TEST  efficiency results to supervisor. Give_
  T PAYROLL TRACE Test will be conducted by Mary B._____
                  User      Date      Time   Version
                  Defined USER24  2011-06-10 15:06  431
                  Last
                  Lib.Mask  Obj.Mask & Types Source
                  Defined PAYTEST_ PAY*_____ PNSMH DBID 1_____
                  Last
                  ADABAS Password _____
                  Statements Executed 0
                  CPU Time (secs) 0.00
                  Database Elapsed (secs) 0.00

From PAYROLL_____

Enter-PF1---PF2---PF3---P-
Help      End      Objts      Exit
```

```

*** PROFILER 4.3.2 for Natural ***

2011-11-2015:09   Active Profile Session: NO           USER24
                  Ac
C A P              Modify Trace Session
o c or            Session PAYROLL TRACE
m t T Session Name Desc Trace the execution of three new
_ P PAYROLL       Payroll objects. Examine results.
_ P PAYROLL TEST
mo T PAYROLL TRACE

                  User      Date      Time      Version
Defined USER24   2011-06-10  15:06    431
                  Statement
                  Lib.Mask  Obj.Mask  Types     Source
Defined PAYTEST_ PAY*_   DIXCPFRO  DBID 1
                  Statements ___ to ___ FUSER 29

From PAYROLL_____ 0 Statements have been TRACed so far.

Enter-PF1---PF2---PF3---PF      Help      End
Exit

```

Field	Description
Session	The name the user assigns to the session. The name must begin with an alphanumeric character and contain 1 to 16 characters.
Session Status (Profile Sessions Only)	The status of this session. Valid values include the following: ACTIVE, NOT ACTIVE, or NEVER ACTIVE.
Desc(ription)	A description of the session. Upper and lower case may be used.
(Defined) User, Date, Time	The user, date, and time at session definition.
(Last) User, Date, Time (Profile Sessions Only)	The user, date, and time at the last session activation. (These fields are blank if the session has never been activated.)
(Defined) Library Mask	The NATURAL library or libraries to be profiled or traced. Mask options include the following: <ul style="list-style-type: none"> * Matches any character in remainder of string ? Matches any single character . Matches any numeric character @ Matches any non-numeric character
(Defined) Object Mask	The NATURAL object or objects to be profiled or traced.

(continued from the previous page)

Field	Description
(Defined) Object Types (Profile Sessions Only)	<p>The NATURAL object types to be profiled.</p> <p>P Program N Subprogram S Subroutine M Map H Helproutine</p> <p>Press PF1 on this field to access a help selection window.</p>
(Defined) Statement Types (Trace Sessions Only)	<p>The NATURAL statement types to be traced.</p> <p>D Database I I/O X Calls C CallNat P Perform F Fetch R For/Repeat O Other</p> <p>Press PF1 to access a help selection window.</p>
(PROFILER) Version	The PROFILER version installed when the session was defined.
(Last) Library Mask (Profile Sessions Only)	The library mask used when the session was last activated.
(Last) Object Mask (Profile Sessions Only)	The object mask used when the session was last activated.
(Last) Object Types (Profile Sessions Only)	The object types used when the session was last activated.
Source DBID / FUSER	The number of the database and FUSER where PROFILER accesses NATURAL source code. These fields default to the DBID and FUSER where PROFILER is installed.
Statements ____ to ____ (Trace Sessions Only)	If only one library/object is to be traced, the 'from' and 'to' statement numbers may be specified in order to reduce the amount of Trace output.
ADABAS PASSWORD (Profile Sessions Only)	A non-display field that is required if the PROFILER File (FNR) is protected by ADABAS security.

(continued from the previous page)

Field	Description
Included Objects Press PF5 to view/amend these settings (Profile Sessions Only)	Allows a user to specify five libraries/objects to be included in the Profile session. Mask options include the following: * Matches any character in remainder of string ? Matches any single character . Matches any numeric character @ Matches any non-numeric character
Excluded Objects (Profile Sessions Only)	Allows a user to specify five libraries/objects to be excluded from the Profile session.
Statements Exec (Profile Sessions Only)	The number of statements executed for this session by all session users. This field is blank if the session has never been activated.
CPU Time (sec) (Profile Sessions Only)	The total CPU time in seconds that this session has used. This field is blank if the session has never been activated.
Database Elapsed (sec) (Profile Sessions Only)	The total Database Elapsed Time in seconds that this session has used. This field is blank if the session has never been activated.
Statements TRACEd so far (Trace Sessions Only)	The number of statements traced for this session. This field is zero if the session has never been activated.

The following PF-keys are provided for the 'Modify ... Session' screens.

Key	Function	Description
PF1	Help	Provides help information about 'Modify ... Session'
PF3	End	Returns to the "session list" menu (and the session is NOT modified)
PF5	Objts	Access to the 'Included / Excluded Objects' window
PF12	Exit	Exits PROFILER

Pressing Enter modifies the session definition and displays a message "PRO0027: Session {name} Modified Successfully. " confirming that PROFILER successfully modified. Pressing Enter again returns to the "session list" menu.

IV.5 Delete Existing Session

The Delete Existing Session function allows a user to delete a defined Profile or Trace session and all PROFILER statistics and data associated with the session. To delete a session's definition and statistics, enter action code "PU" next to the session that is to be deleted.

When a session is active, PROFILER displays the message "PRO0084: Can NOT Purge an Active Session. De-Activate Session and then Purge." to indicate that the session is currently being used and may not be deleted while any user is active on the session.

If a user other than the owner tries to delete a session, the message "PRO0083: Command ONLY available to Session Owner (i.e., ...)." will be displayed.

```

*** PROFILER 4.3.2 for Natural ***

2011-11-2015:10  Active Profile Session: NO          USER24
C A P           Active Trace Session: NO          PAYTEST
o c o r
m t T Session Name      Date      Time      Owner      LastUser    Description
pu P PAYROLL            2011-06-05 16:52 USER24     USER23     Example for
  P PAYROLL TEST       2011-06-02 15:33 USER24     USER27     Payroll Test
  T PAYROLL TRACE      2011-06-10 15:06 USER24                      Payroll Trace

From PAYROLL_____ Owner _____ Profile/Trace/All A

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10---PF11---PF12---

```

Pressing Enter displays the 'Purge Profile Session' screen or the 'Purge Trace Session' screen depending on the type of session selected. The 'Purge ... Session' fields are not modifiable, apart from the 'Proceed with PURGE' field.

```

PRO0013: Use PF5 to access Included/Excluded Objects.
*** PROFILER 4.3.2 for Natural ***

2011-11-2015:10  Active Profile Session: NO          USER24
                  Ac
C A P
o c o r
m t T Session Name      Desc
pu P PAYROLL            Getting Started with PROFILER 4.3.1
  P PAYROLL TEST       (Test 20 new Payroll objects. Give
  T PAYROLL TRACE      efficiency results to supervisor. Give
                        QA results to Quality Assurance team.
                        Test will be conducted by Mary B.
                        User      Date      Time      Version
Defined USER24      2011-06-10 15:06 431
Last
Lib.Mask  Obj.Mask & Types  Source
Defined PAYTEST    PAY*      PNSMH  DBID 1
Last
Proceed with PURGE of this Session? _
Statements Executed 0
CPU Time (secs) 0.00
Database Elapsed (secs) 0.00

From PAYROLL_____

Enter-PF1---PF2---PF3---P--
Help      End      Objts      Exit

```

```

*** PROFILER 4.3.2 for Natural ***

2011-11-2015:10   Active Profile Session: NO           USER24
                  Act
C A P
o c or           Purge Trace Session
m t T Session Name Desc Trace the execution of three new
                  Payroll objects.  Examine results.
  _ P PAYROLL
  _ P PAYROLL TEST
pu T PAYROLL TRACE

Proceed with PURGE of this Session? _
      User      Date      Time      Version
Defined USER24  2011-06-10  15:06  431
                  Statement
      Lib.Mask  Obj.Mask  Types      Source
Defined PAYTEST PAY*      DIXCPFRO  DBID 1
Statements to FUSER 29

From PAYROLL_____ 0 Statements have been TRACEd so far.

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help      End                               Exit

```

Field	Description
Session	The name the user assigns to the session. The name must begin with an alphanumeric character and contain 1 to 16 characters.
Session Status (Profile Sessions Only)	The status of this session. Valid values include the following: ACTIVE, NOT ACTIVE, or NEVER ACTIVE.
Desc(ription)	A description of the session. Upper and lower case may be used.
Proceed with PURGE ?	Answer 'Y' or 'N'.
(Defined) User, Date, Time	The user, date, and time at session definition.
(Last) User, Date, Time (Profile Sessions Only)	The user, date, and time at the last session activation. (These fields are blank if the session has never been activated.)
(Defined) Library Mask	The NATURAL library or libraries to be profiled or traced. Mask options include the following: <ul style="list-style-type: none"> * Matches any character in remainder of string ? Matches any single character . Matches any numeric character @ Matches any non-numeric character
(Defined) Object Mask	The NATURAL object or objects to be profiled or traced.

(continued from the previous page)

Field	Description
(Defined) Object Types (Profile Sessions Only)	<p>The NATURAL object types to be profiled.</p> <p>P Program N Subprogram S Subroutine M Map H Helproutine</p> <p>Press PF1 on this field to access a help selection window.</p>
(Defined) Statement Types (Trace Sessions Only)	<p>The NATURAL statement types to be traced.</p> <p>D Database I I/O X Calls C CallNat P Perform F Fetch R For/Repeat O Other</p> <p>Press PF1 to access a help selection window.</p>
(PROFILER) Version	The PROFILER version installed when the session was defined.
(Last) Library Mask (Profile Sessions Only)	The library mask used when the session was last activated.
(Last) Object Mask (Profile Sessions Only)	The object mask used when the session was last activated.
(Last) Object Types (Profile Sessions Only)	The object types used when the session was last activated.
Source DBID / FUSER	The number of the database and FUSER where PROFILER accesses NATURAL source code. These fields default to the DBID and FUSER where PROFILER is installed.
Statements ____ to ____ (Trace Sessions Only)	If only one library/object is to be traced, the 'from' and 'to' statement numbers may be specified in order to reduce the amount of Trace output.

(continued from the previous page)

Field	Description
Included Objects Press PF5 to view/amend these settings (Profile Sessions Only)	Allows a user to specify five libraries/objects to be included in the Profile session. Mask options include the following: * Matches any character in remainder of string ? Matches any single character . Matches any numeric character @ Matches any non-numeric character
Excluded Objects (Profile Sessions Only)	Allows a user to specify five libraries/objects to be excluded from the Profile session.
Statements Exec (Profile Sessions Only)	The number of statements executed for this session by all session users. This field is blank if the session has never been activated.
CPU Time (sec) (Profile Sessions Only)	The total CPU time in seconds that this session has used. This field is blank if the session has never been activated.
Database Elapsed (sec) (Profile Sessions Only)	The total Database Elapsed Time in seconds that this session has used. This field is blank if the session has never been activated.
Statements TRACED so far (Trace Sessions Only)	The number of statements traced for this session. This field is zero if the session has never been activated.

The following PF-keys are provided for the 'Purge ... Session' screens.

Key	Function	Description
PF1	Help	Provides help information about 'Purge ... Session'
PF3	End	Returns to the "session list" menu (and session is not purged)
PF5	Objts	Access to the 'Included / Excluded Objects' window
PF12	Exit	Exits PROFILER

Entering "N" in the 'Proceed with Purge' field or pressing PF3 cancels the delete function and returns to the "session list" menu.

Entering "Y" in the 'Proceed with Purge' field continues the process of deleting a session. The user is returned to the "session list" menu with the message " " at the top of the screen. (The deleted session will have been removed from the "session list".)

IV.6 **Activate Session**

The Activate Session function allows a user to define activation parameters and start a Profile or Trace session to collect statistics during the execution of a set of NATURAL objects in a NATURAL application.

One user or many users may activate a Profile session, each specifying a different set of activation parameters. PROFILER allows 84 users to be active on each of 64 (FUSER used) or 255 (alternate FUSER used) Profile sessions. A user may have only one Profile session active at a time, and this session is active during the user's current NATURAL session.

Only the owner of a Trace session may activate it. Each user may have 65,535 Trace sessions. A user may have only one Trace session active at a time, and this session is active during the user's current NATURAL session.

When a user exits NATURAL, statistics are no longer collected. The user may reactivate a specific session after logging on to NATURAL and reentering PROFILER.

When a user activates a session, the user specifies a library or range of libraries, an object or range of objects, and either object types (to be profiled) or statement types (to be traced). PROFILER collects statistics for an object in the specified library/object range when the object is executed during the active session. In addition to the library/object masks, a user may specify five libraries/objects to be included in a Profile session and five libraries/objects to be excluded from a Profile session.

PROFILER collects statistics for objects executed by the user who activated the session. PROFILER also collects statistics for objects executed by the user who has an active Profile session as a result of Background Monitoring. Statistics are not collected for objects executed by users who do not have an active session.

Profile sessions report which object statements were executed, how many times they were executed, how much CPU and Database time they used, which statements were not executed, and which statements were not executable. A range of summary reports are available for Profile sessions, as well as the detailed 'Source Code Listing' report for each object executed.

Trace sessions only report object statements in the order they were executed. No CPU or Database time usage is collected during Tracing.

<p>Note: It is recommended that a user enter a library and a range of objects to avoid collecting statistics on objects that are executed outside the scope of the application. For example, if a user does not specify a library or objects, PROFILER collects statistics on all activity performed by the user, except for objects which execute from library PROLIB and from libraries beginning with SYS.</p>
--

To activate a session, enter action code "AC" next to the session that is to be activated.

If the session is already active for that user, PROFILER displays the message "PRO0038: Can NOT Activate. Session is ALREADY active.". If the user already has another active session, PROFILER displays the message "PRO0031: You have {session-name} active already. You can NOT activate another session." Note that a user can have an active Profile session and an active Trace session at the one time.

If other user(s) already have the session active, PROFILER displays the message "PRO0035: Warning: Other Users already active on Session. Use "AU" to list them. If you want to know which users are active, use the 'AU' action code on the session. Otherwise, press enter to continue with the activation. If more than one user activates the same session, statistics are collected based on the library/object mask and object types defined by each user. Statistics for each user are stored in a PROFILER repository based on the User-ID and session.

```

*** PROFILER 4.3.2 for Natural ***

2011-11-2015:11  Active Profile Session: NO          USER24
C A P           Active Trace Session: NO          PAYTEST
o c or
m t T Session Name      Date      Time  Owner  LastUser Description
ac  P PAYROLL           2011-06-05 16:52 USER24  USER23  Example for
   P PAYROLL TEST      2011-06-02 15:33 USER24  USER27  Payroll Test
   T PAYROLL TRACE     2011-06-10 15:06 USER24           Payroll Trace

From PAYROLL_____ Owner _____ Profile/Trace/All A
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10---PF11---PF12---
    
```

Pressing Enter displays the 'Activate Profile Session' screen or the 'Activate Trace Session' screen depending on the type of session selected.

```

PRO0013: Use PF5 to access Included/Excluded Objects.
*** PROFILER 4.3.2 for Natural ***

2011-11-2015:11  Active Profile Session: NO          USER24
                  Ac
C A P           Activate Profile Session
o c or          Session PAYROLL           Status NEVER ACTIVE
m t T Session Name      Desc Getting Started with PROFILER 4.3.1
ac  P PAYROLL           (Test 20 new Payroll objects. Give
   P PAYROLL TEST      efficiency results to supervisor. Give
   T PAYROLL TRACE     QA results to Quality Assurance team.
                       Test will be conducted by Mary B.
                       User      Date      Time  Version
Defined USER24      2011-06-10 15:06  431
Last
Lib.Mask Obj.Mask & Types Source
Defined PAYTEST_   PAY*_____ PNSMH DBID 1
Last
ADABAS Password _____
Statements Executed 0
CPU Time (secs) 0.00
Database Elapsed (secs) 0.00

From PAYROLL_____
Enter-PF1---PF2---PF3---P
Help      End      Objts      Exit
    
```

```

*** PROFILER 4.3.2 for Natural ***

2011-11-2015:11  Active Profile Session: NO          USER24
                  Ac
C A P           Activate Trace Session
o c or          Session PAYROLL TRACE
m t T Session Name      Desc Trace the execution of three new
ac  P PAYROLL           Payroll objects. Examine results.
   P PAYROLL TEST
   T PAYROLL TRACE

User      Date      Time  Version
Defined USER24  2011-06-10 15:06  431
Statement
Lib.Mask Obj.Mask Types Source
Defined PAYTEST PAY*_____ DIXCPFR DBID 1
Statements to FUSER 29
ADABAS Password _____

From PAYROLL_____
Enter-PF1---PF2---PF3---P
Help      End      Exit
    
```

If the Profile session already has statistics collected, the message "PRO0032: Statistics exist: <pf3> & reset stats (RS) OR <enter> & add to them." will be displayed. Press enter to continue with the activation, or press PF3 and use the 'RS' action code on the session to reset the statistics prior to activation.

Field	Description
Session	The name the user assigns to the session. The name must begin with an alphanumeric character and contain 1 to 16 characters.
Session Status (Profile Sessions Only)	The status of this session. Valid values include the following: ACTIVE, NOT ACTIVE, or NEVER ACTIVE.
Desc(ription)	A description of the session. Upper and lower case may be used.
(Defined) User, Date, Time	The user, date, and time at session definition.
(Last) User, Date, Time (Profile Sessions Only)	The user, date, and time at the last session activation. (These fields are blank if the session has never been activated.)
(Defined) Library Mask	The NATURAL library or libraries to be profiled or traced. Mask options include the following: <ul style="list-style-type: none"> * Matches any character in remainder of string ? Matches any single character . Matches any numeric character @ Matches any non-numeric character
(Defined) Object Mask	The NATURAL object or objects to be profiled or traced.
(Defined) Object Types (Profile Sessions Only)	The NATURAL object types to be profiled. <ul style="list-style-type: none"> P Program N Subprogram S Subroutine M Map H Helproutine <p>Press PF1 on this field to access a help selection window.</p>

(continued from the previous page)

Field	Description
(Defined) Statement Types (Trace Sessions Only)	<p>The NATURAL statement types to be traced.</p> <p>D Database I I/O X Calls C CallNat P Perform F Fetch R For/Repeat O Other</p> <p>Press PF1 to access a help selection window.</p>
(PROFILER) Version	The PROFILER version installed when the session was defined.
(Last) Library Mask (Profile Sessions Only)	The library mask used when the session was last activated.
(Last) Object Mask (Profile Sessions Only)	The object mask used when the session was last activated.
(Last) Object Types (Profile Sessions Only)	The object types used when the session was last activated.
Source DBID / FUSER	The number of the database and FUSER where PROFILER accesses NATURAL source code. These fields default to the DBID and FUSER where PROFILER is installed.
Statements ____ to ____ (Trace Sessions Only)	If only one library/object is to be traced, the 'from' and 'to' statement numbers may be specified in order to reduce the amount of Trace output.
ADABAS PASSWORD	A non-display field that is required if the PROFILER File (FNR) is protected by ADABAS security.
Included Objects Press PF5 to view/amend these settings (Profile Sessions Only)	<p>Allows a user to specify five libraries/objects to be included in the Profile session. Mask options include the following:</p> <ul style="list-style-type: none"> * Matches any character in remainder of string ? Matches any single character . Matches any numeric character @ Matches any non-numeric character

(continued from the previous page)

Field	Description
Excluded Objects (Profile Sessions Only)	Allows a user to specify five libraries/objects to be excluded from the Profile session.
Statements Exec (Profile Sessions Only)	The number of statements executed for this session by all session users. This field is blank if the session has never been activated.
CPU Time (sec) (Profile Sessions Only)	The total CPU time in seconds that this session has used. This field is blank if the session has never been activated.
Database Elapsed (sec) (Profile Sessions Only)	The total Database Elapsed Time in seconds that this session has used. This field is blank if the session has never been activated.

The following PF-keys are provided for the 'Activate ... Session' screens.

Key	Function	Description
PF1	Help	Provides help information about 'Activate ... Session'
PF3	End	Returns to the "session list" menu (and session is not activated)
PF5	Objts	Access to the 'Included / Excluded Objects' window
PF12	Exit	Exits PROFILER

Pressing enter will activate the session and the message " PRO0037: Session ... Activated successfully for ..." will be displayed upon return to the "session list" menu.

IV.7 Deactivate Session

The Deactivate Session function allows a user to stop a Profile session from collecting statistics during the execution of a set of NATURAL objects in a NATURAL application. PROFILER also stops collecting a user's statistics when that user's NATURAL session is terminated. To begin collecting statistics again, a user must reactivate the session.

Users may only deactivate sessions they have activated. A user may not deactivate another user's active session.

To Deactivate a session, enter action code "DA" next to the session that is to be deactivated. If the session is not active for the user, PROFILER will display the message "PRO0039: Can NOT De-Activate. Session is NOT active for you."

```

*** PROFILER 4.3.2 for Natural ***

2011-11-2015:12  Active Profile Session: PAYROLL          USER24
C A P           Active Trace Session: PAYROLL TRACE      PAYTEST
o c or
m t T Session Name      Date      Time      Owner      LastUser  Description
da * P PAYROLL          2011-06-05 16:52 USER24     USER24     Example for
   P PAYROLL TEST       2011-06-02 15:33 USER24     USER27     Payroll Test
   * T PAYROLL TRACE    2011-11-2015:06 USER24     USER24     Payroll Trace

From PAYROLL_____ Owner _____ Profile/Trace/All A

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
    
```

Pressing Enter displays the 'De-Activate Profile Session' screen or the 'De-Activate Trace Session' screen depending on the type of session selected. The 'De-Activate ... Session' fields are not modifiable.

```

PRO0013: Use PF5 to access Included/Excluded Objects.
*** PROFILER 4.3.2 for Natural ***

2011-11-2015:12  Active Profile Session: PAYROLL          USER24
                  Ac
C A P           De-Activate Profile Session
o c or          Session PAYROLL          Status ACTIVE
m t T Session Name      Desc Getting Started with PROFILER 4.3.1
da * P PAYROLL          (Test 20 new Payroll objects. Give
   P PAYROLL TEST       efficiency results to supervisor. Give
   * T PAYROLL TRACE    QA results to Quality Assurance team.
                        Test will be conducted by Mary B.
                        User      Date      Time      Version
                        Defined USER24 2011-06-10 15:06 431
                        Last   USER24 2011-06-10 15:11
                        Lib.Mask Obj.Mask & Types Source
                        Defined PAYTEST PAY*      PNSMH DBID 1
                        Last   PAYTEST PAY*      PNSMH FUSER 29

From PAYROLL_____

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
Help          End          Objts          Exit
    
```

```

*** PROFILER 4.3.2 for Natural ***

2011-11-2015:12   Active Profile Session: NO           USER24
                  Act
C A P             De-Activate Trace Session
o c or           Session PAYROLL TRACE
m t T Session Name Desc Trace the execution of three new
___ P PAYROLL      Payroll objects. Examine results.
___ P PAYROLL TEST
da * T PAYROLL TRACE

                  User      Date      Time      Version
Defined USER24   2011-06-10  15:06    431
                  Statement
                  Lib.Mask  Obj.Mask  Types      Source
Defined PAYTEST  PAY*      DIXCPFRO  DBID 1
                  Statements to      FUSER 29

From PAYROLL_____ 154 Statements have been TRACEd so far.

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help      End                               Exit

```

Key	Function	Description
PF1	Help	Provides help information about 'De-Activate ... Session'
PF3	END	Cancels the deactivation of a session and returns to the "session list" menu
PF5	Objts	Access to the 'Included / Excluded Objects' window
PF12	Exit	Exits PROFILER

Pressing Enter deactivates the session and returns to the "session list" menu with the message "PRO0040: Session ... De-Activated successfully for ...".

IV.8 Reset Session Statistics

The Reset Session function allows a user to reset session statistics without affecting the session definition. A user may then activate the session and collect new statistics.

To Reset Statistics for a session, enter action code "RS" next to the session that is to have its statistics reset. If the session is active (for any user), the message "PRO0011: Warning - Session is ACTIVE. Press <enter> to execute your command." is displayed. Resetting statistics while users are collecting statistics will give unpredictable report results.

```

*** PROFILER 4.3.2 for Natural ***

2011-11-2015:13   Active Profile Session: NO           USER24
C A P             Active Trace Session: NO       PAYTEST
o c or
m t T Session Name   Date      Time   Owner   LastUser  Description
rs  P PAYROLL         2011-06-05 16:52 USER24  USER24   Example for
   P PAYROLL TEST    2011-06-02 15:33 USER24  USER27   Payroll Test
   T PAYROLL TRACE   2011-06-10 15:06 USER24  USER24   Payroll Trace

From PAYROLL_____ Owner _____ Profile/Trace/All A

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
    
```

Pressing Enter displays the 'Reset Statistics for Profile Session' screen or the 'Reset Statistics for Trace Session' screen depending on the type of session selected.

```

PRO00116: You may limit Reset of Stats by User, Library, Object, and/or Type
*** PROFILER 4.3.2 for Natural ***

2011-11-2015:13   Active Profile Session: NO           USER24
C A P             Active Trace Session: NO       PAYTEST
o c or
m t T Session Name   Date      Time   Owner   LastUser  Description
rs  P PAYROLL         2011-06-05 16:52 USER24  USER24   Example for
   P PAYROLL TEST    2011-06-02 15:33 USER24  USER27   Payroll Test
   T PAYROLL TRACE   2011-06-10 15:06 USER24  USER24   Payroll Trace

Reset Statistics for Profile Session
Session PAYROLL         Status NOT ACTIVE
Desc Getting Started with PROFILER 4.3.1
(Test 20 new Payroll objects. Give
efficiency results to supervisor. Give
QA results to Quality Assurance team.
Test will be conducted by Susan M.

Reset   User      Date      Time   Version
only for USER24_ 2011-06-10 15:06   431
Last   USER24   2011-06-10 15:11
Reset   Lib.Mask  Obj.Mask & Types  Source
only for PAYTEST_ PAY*_____ PNSMH  DBID 1
Last   PAYTEST  PAY*_____ PNSMH  FUSER 29

Statements Executed 2263
CPU Time (secs) 2.03
Database Elapsed (secs) 1906.78

From PAYROLL_____

Enter-PF1---PF2---PF3---P-
Help           End                               Exit
    
```

```

*** PROFILER 4.3.2 for Natural ***

2011-11-2015:13   Active Profile Session: NO           USER24
C A P             Active Trace Session: NO       PAYTEST
o c or
m t T Session Name   Date      Time   Owner   LastUser  Description
rs  P PAYROLL         2011-06-05 16:52 USER24  USER24   Example for
   P PAYROLL TEST    2011-06-02 15:33 USER24  USER27   Payroll Test
   T PAYROLL TRACE   2011-06-10 15:06 USER24  USER24   Payroll Trace

Reset Statistics for Trace Session
Session PAYROLL TRACE
Desc Trace the execution of three new
Payroll objects. Examine results.

User      Date      Time   Version
Defined USER24 2011-06-10 15:06   431
Statement
Lib.Mask  Obj.Mask  Types  Source
Defined PAYTEST PAY*_____ DIXCPRO DBID 1
Statements to FUSER 29

From PAYROLL_____ 154 Statements have been TRACed so far.

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
Help           End                               Exit
    
```

Field	Description
Session	The name the user assigns to the session. The name must begin with an alphanumeric character and contain 1 to 16 characters.
Session Status (Profile Sessions Only)	The status of this session. Valid values include the following: ACTIVE, NOT ACTIVE, or NEVER ACTIVE.
Desc(ription)	A description of the session. Upper and lower case may be used.
Reset only for User ... (Profile Sessions Only)	The user for which Profile statistics are to be reset. If blank, statistics for all users for this session will be reset.
(Last) User, Date, Time (Profile Sessions Only)	The user, date, and time at the last session activation. (These fields are blank if the session has never been activated.)
Reset only for Lib. Mask... (Profile Sessions Only)	The NATURAL library or libraries for which statistics are to be reset. Mask options include the following: <ul style="list-style-type: none"> * Matches any character in remainder of string ? Matches any single character . Matches any numeric character @ Matches any non-numeric character
Reset only for Object Mask... (Profile Sessions Only)	The NATURAL object or objects for which statistics are to be reset.
Reset only for Obj. Types... (Profile Sessions Only)	The NATURAL object types for which statistics are to be reset. <ul style="list-style-type: none"> P Program N Subprogram S Subroutine M Map H Helproutine <p>Press PF1 on this field to access a help selection window.</p>

(continued from the previous page)

Field	Description
(Defined) Statement Types (Trace Sessions Only)	The NATURAL statement types to be traced. D Database I I/O X Calls C CallNat P Perform F Fetch R For/Repeat O Other Press PF1 to access a help selection window.
(PROFILER) Version	The PROFILER version installed when the session was defined.
(Last) Library Mask (Profile Sessions Only)	The library mask used when the session was last activated.
(Last) Object Mask (Profile Sessions Only)	The object mask used when the session was last activated.
(Last) Object Types (Profile Sessions Only)	The object types used when the session was last activated.
Source DBID / FUSER	The number of the database and FUSER where PROFILER accesses NATURAL source code. These fields default to the DBID and FUSER where PROFILER is installed.
Statements ____ to ____ (Trace Sessions Only)	If only one library/object is to be traced, the 'from' and 'to' statement numbers may be specified in order to reduce the amount of Trace output.
Statements Exec (Profile Sessions Only)	The number of statements executed for this session by all session users. This field is blank if the session has never been activated.
CPU Time (sec) (Profile Sessions Only)	The total CPU time in seconds that this session has used. This field is blank if the session has never been activated.
Database Elapsed (sec) (Profile Sessions Only)	The total Database Elapsed Time in seconds that this session has used. This field is blank if the session has never been activated.
Statements TRACEd so far (Trace Sessions Only)	The number of statements traced for this session. This field is zero if the session has never been activated.

The following PF-keys are provided for the 'Reset Statistics for ... Session' screens.

Key	Function	Description
PF1	Help	Provides help information about 'Reset Statistics for ...'
PF3	End	Returns to the "session list menu (and session statistics are not reset)
PF12	Exit	Exits PROFILER

Pressing enter will reset the statistics nominated and the message "PRO0113: Statistics for ... Reset successfully by ..." will be displayed upon return to the "session list menu.

If statistics in the session have been tagged for move to the Enhanced Reporting facility, the message "PRO0114: Warning - "Reset" may stop "tagged" objects being moved to 'Enhanced Reporting'." is displayed, indicating that resetting this session may prevent some of these objects from being moved.

IV.9 Show Session Active Users

The Show Session Active Users function displays all active users for a Profile session, the date and time a user activated a session, the library and objects masks, and the object types that are being used to collect PROFILER statistics for the Profile session. Note that this function is not available for Trace sessions because they can only be active for one user (i.e., the owner of the Trace Session).

To Show the Active Users for a Profile session, enter action code "AU" on the "session list" menu beside the required Profile session.

```

*** PROFILER 4.3.2 for Natural ***

2011-11-2015:15  Active Profile Session: PAYROLL          USER24
C A P           Active Trace Session: NO                PAYTEST
o c or
m t T Session Name      Date      Time  Owner  LastUser  Description
au * P PAYROLL          2011-06-05 16:52 USER24  USER24  Example for
   P PAYROLL TEST      2011-06-02 15:33 USER24  USER27  Payroll Test
   T PAYROLL TRACE     2011-06-10 15:06 USER24  USER24  Payroll Trace

From PAYROLL_____ Owner _____ Profile/Trace/All A

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10---PF11---PF12---
Help About End Admin Backg Up Down New-S Left Right Exit
    
```

Press Enter to continue.

```

*** PROFILER 4.3.2 for Natural ***

2011-11-2015:15  Active Profile Session: PAYROLL          USER24
Ac
C A P           Show Active Users for...
o c or          2011-11-2015:15 PAYROLL      USER24  PAYTEST
m t T Session Name      Active   Active   Active Library  Object  Object
au * P PAYROLL          User     Date     Time  Mask      Mask    Types
   P PAYROLL TEST      USER24  2011-06-10 15:14 *      *      PNSMH
   T PAYROLL TRACE

From PAYROLL_____

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
Help          End          Up      Down          Exit
    
```

The screen above shows USER24 is the only user who has activated session PAYROLL. Press PF3 to return to the "session list" menu.

Field	Description
Active User	The User-ID of the active user for the session.
Active Date	The date the user activated the session.
Active Time	The time the user activated the session
Library Mask	The NATURAL library or libraries to be profiled. Mask options include the following: <ul style="list-style-type: none"> * Matches any character in remainder of string ? Matches any single character . Matches any numeric character @ Matches any non-numeric character
Object Mask	The NATURAL object or objects to be profiled.
Object Types	The NATURAL object types to be profiled. Valid values are as follows: <ul style="list-style-type: none"> P Program N Subprogram S Subroutine M Map H Helproutine

The following PF-keys are provided for the Show Session Active Users function.

Key	Function	Description
PF1	Help	Provides help information about Show Session Active Users
PF3	End	Returns to the "session list" menu
PF7	Up	Pages backward
PF8	Down	Pages forward
PF12	Exit	Exits PROFILER

Pressing Enter or PF8 on the last page of the report returns to the first page of the report.

This page intentionally left blank.

SECTION V

SESSION REPORTING

V.1 Introduction to Session Reporting

The PROFILER Reporting facility allows a user to display the statistics that have been calculated during an active profile or trace session. PROFILER retrieves these statistics from the PROFILER repository. Reports that display these statistics may be obtained on-line or in batch. There are four types of PROFILER Session reports: Profile Session Statistics Summary, Source Code Listing for Profiled Objects, Source Code Listing for Traced Objects, and Application QA (Quality Assurance) for Profiled Libraries.

Profile Session Statistics Summary Reports

Provide information about statement executions, CPU time, Database Elapsed Time, and type of NATURAL statement executed for NATURAL objects executed during an active profile session. There are eleven different summary report formats available directly from the "session list" menu. While viewing a particular report format, another report format can be selected using a pop-up menu or by use of PF keys:

- Statement Execution Count ('SS' action)
- CPU time ('SC' action)
- Database Elapsed Time ('SD' action)
- Number of Statements, CPU Time, Database ('SN' action)
- Percentage of Executions, CPU and Database ('SP' action)
- Percent Graph of Executions, CPU and Database ('SG' action)
- Database/Work File Loops ('SF' action)
- Internal Subroutines/Non-Procedural Blocks ('SB' action)
- FOR/REPEAT Loops ('SR' action)
- IF/DECIDE Conditions ('SI' action)
- Percent Executed by Statement Type ('ST' action)

The scope of any of these report formats can be limited by starting library/object, object type(s), and several threshold settings. Any objects that have statistics greater than the relevant threshold setting will be included in the summary report. Thresholds may be entered for the following object statistics:

- Run Count
- Statement Executions
- CPU time
- Database Elapsed Time
- Percent of Statement Executions
- Percent of CPU time
- Percent of Database Elapsed Time

The summary reports may also be sorted in one of four ways (without leaving the report):

- Object (within Library) order (ascending)
- Statement Execution Count order (descending)
- CPU time order (descending)
- Database Elapsed Time order (descending)

Source Code Listing Report (for Profiled Objects)

Displays the source code, including copycode, for an individual NATURAL object, and gives statistics for run count. Also displays CPU and database usage for the object, as well as for each individual executed, un-executed, and non-executable statement within the object. The source code listing can be limited to any combination of executed, un-executed, and non-executable statements required. The source code can also be scanned for particular text.

Source Code Listing Report (for Traced Objects)

Lists each statement that was executed during the Trace statistics collection. Statements are listed in the exact order in which they were executed, showing the Library name, Object name, Statement number, and source code for the statement. The source code can also be scanned for particular text. A Trace session can report up to 13,500 executed statements relating to at most 200 objects.

Application QA Report for Profiled Libraries

Provides the percentage of a NATURAL library, or for a range of objects in a library, that has been tested. It also shows a listing of which objects have not been tested at all. The source code of these untested objects can be viewed directly from this listing. Direct access is provided to the 'Statement Execution Count' Summary report which shows the percentage of each object tested.

Notes about Session Reporting

When reporting on PROFILER statistics for numerous objects, it is recommended that reports be run in batch. The eleven Profile Session Statistics Summary Report formats show up to 500 profiled objects on-line.

The Application QA Report shows up to 4,000 untested objects for one library on-line. The Source Code Listing Report (for Profiled Objects) allows 2,500 object statements and 2,500 copycode statements to be displayed on-line. The Source Code Listing Report (for Traced Objects) allows 13,500 object statements to be displayed on-line. (This is the maximum number of traced statements that can be collected in a Trace session.)

In batch, an unlimited number of lines and pages may be displayed.

Database Elapsed Time represents elapsed time for calls to databases, such as ADABAS, DB2, and VSAM.

All statistics on the PROFILER reports are rounded.

Object code must exist on the NATURAL FUSER where PROFILER is installed. To create the object code, NATURAL objects must be STOWed before profiling.

It is recommended that the object code be cataloged in the library from which it is to be profiled. For more information, refer to the **PROFILER Statistics for Objects Migrated with SYSMAIN** section.

Source code is only necessary for the Source Code Listing Reports.

PROFILER does not collect statistics on objects executed from library PROLIB or from a library that begins with SYS.

It is recommended that all users of a profile session deactivate the profile session before reporting begins on that session. This will prevent statistics from changing and reduce overhead while reporting. When a user deactivates a session, only that user's current session is deactivated. If another user has the same session active, this user's session is not deactivated. This may cause statistics to change during the reporting process.

There is overhead for a user who is reporting on an active session. When a user has a session active, PROFILER is invoked every time any NATURAL module is executed by this active user. This causes overhead which may slow down the reporting process for the active user.

When a session is active, the message "PRO0011: Warning - Session is ACTIVE. Press <enter> to execute your command." is displayed indicating that the session is currently being used and statistics may change during the reporting process.

Note: It is recommended that all users of a profile session deactivate the profile session before reporting begins on that session. If all users do not deactivate a session, statistics may change if users are actively profiling during the reporting process.

Note: If an object has been STOWed since profiling began on a session involving that object, statistics for that object may be inaccurate. If an object has been reSTOWed, use the 'RS' action to reset statistics for the object, and then activate the session. Then re-profile the object.

How to Invoke Session Reporting

Enter '?' in a 'Com' field (or press PF1 while the cursor is in a 'Com' field) on the Main "session list" menu to access a help selection window of available "session actions".

```

*** PROFILER 4.3.2 for Natural ***
  2011-06-12 16:48 Acti Mark Command ... USER24
C A P Acti _ DI Display PAYTEST
o c or _ MO Modify Page 1_ of 1
m t T Session Name _ CO Copy LastUser Description
_ P PAYROLL EXAMPLE _ PU Purge USER23 Example for
?_ P PAYROLL TEST USER27 Payroll Test

_ AC Activate
_ DA De-Activate
_ RS Reset Stats
_ AU Active User *

_ S= Stats Report
x S? Stats Help *
_ QA Applic QA *
_ CU Customized *

From PAYROLL_____ Profile/Trace/All P
Only for Profile: *
Enter-PF1---PF2---PF3---PF4 ---PF9---PF10--PF11--PF12---
Help About End Admin Backg Up Down New-S Left Right Exit

```

For a Profile session, the action 'S=' will invoke the most recently used 'Summary Report' format. For a Trace session, the action 'S=' will invoke the 'Trace Source Code Report'.

The 'QA' action is only available for Profile sessions and will invoke the 'Application QA' report.

V.2 Session Statistics Summary Reports

The 'S?' action is only available for Profile sessions and will display the following help selection window of available 'Summary Report' formats.

```

*** PROFILER 4.3.2 for Natural ***

2011-11-20 16:22   Active Profile Session: NO           USER24
C A P             Active Trace Session: NO           PAYTEST
o c o r                               Page 1_ of 1
m t T Session Name      Date      Time  Owner  LastUser Description
s? P PAYROLL           2011-06
  P PAYROLL EXAMPLE    2011-06
  P PAYROLL TEST       2011-06

                                Mark Report Format..
x S Statement Execution Summary
  C CPU Time Used Summary
  D Database Elapsed Time Summary
  N Nums,Times: Stmts, CPU, Dbase
  P Percents:  Stmts, CPU, Dbase
  G % Graphs:  Stmts, CPU, Dbase

  F Database/Work File Loops
  B Inline Sub/Non-proc'al Blocks
  R FOR and/or REPEAT Loops
  I IF and/or DECIDE Conditions
  T Percent Executed by Stmt Type

From PAYROLL_____

Enter-PF1---PF2---PF3---PF4---PF5-F12---
      Help About End Admin Backg Up Down New-S Left Right Exit

```

The two-letter action codes (e.g., 'SS', 'SC', 'SD', etc.) may be used directly in the 'Com' field to invoke the required 'Summary Report' format.

```

PRO0096: 'S'elect Object to see its Source Code Listing Report.
                                Summary Report for Session
2011-11-20 16:23   Session PAYROLL EXAMPLE           USER24 PAYTEST
Report Format S Sort Order OBJ Types _____ QA? N Page 1_ of 1
Start Library _____ Start Object _____ View/Amend Thresholds N
Total Stmt Execs                               80

s          T          Exec %Exec % Graph of
e          y          utbl -utbl Executable
l Library Object p Run Count Excs Stmt Excs Executed
- PAYTEST PAYROLL P 1 1 2 1 50.00 *****
- PAYTEST PAY0100M M 4 8 2 2 100.00 *****
- PAYTEST PAY0100P P 1 59 26 20 76.92 *****
- PAYTEST PAY0110M M 1 4 2 2 100.00 *****
- PAYTEST PAY0125M M 1 4 2 2 100.00 *****
- PAYTEST PAY0130M M 1 4 2 2 100.00 *****

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help User End Stmts CPU Dbase Up Down Sourc Left Right Exit

```

For example, in the screen above, statistics are reported for objects executed from library PAYTEST. The statistics presented in this report were stored under the session "PAYROLL EXAMPLE".

The current 'Report Format', 'Sort Order', and (shown object) 'Types' settings are shown on the next line. These values may be changed directly or by help selection windows (with PF1 or a '?'). The 'QA?' field allows direct access to the 'Application QA' report for the current session. The current and maximum page number of the report is located on the right side of this line.

The 'Start Library' and 'Start Object' fields allow the summary report to be started from a chosen point. The 'View/Amend Thresholds?' field allows selection of various values to limit the objects included in the summary report.

There are eleven Session Statistics Summary Report formats and all have a very similar layout and options. These reports provide information about NATURAL objects executed during an active profile session.

Each of these report formats shows the session, User-ID, library, object, object type, and run count (the number of times an object executed). Statistics that round to zero display as blank. For example, a CPU time of 0.00003 milliseconds displays as blank.

Each of these report formats uses the following PF keys:

Key	Function	Description
PF1	Help	Provides cursor-sensitive help information
PF2	User	Toggles between the display of the profiled library and the user who performed the profiling
PF3	End	Returns to the "session list" menu
PF4	Stmts	Changes the summary report format directly to 'S' (Statement Execution)
PF5	CPU	Changes the summary report format directly to 'C' (CPU Time Used)
PF6	Dbase	Changes the summary report format directly to 'D' (Database Elapsed Time Used)
PF7	Up	Scrolls up through the report
PF8	Down	Scrolls down through the report
PF9	Sourc	Allows the location of source code (i.e., libraries) to be changed
PF10	Left	Scrolls left through the other summary report formats
PF11	Right	Scrolls right through the other summary report formats
PF12	Exit	Exits PROFILER

Statement Execution Count Summary Report ('SS' action)

Returns statement execution statistics, percent of statements executed, and a statement graph.

CPU Time Summary Report ('SC' action)

Returns statistics on CPU time used, percent of CPU time used, and a CPU graph.

Database Elapsed Time Summary Report ('SD' action)

Returns statistics on Database Elapsed Time used, percent of Database Elapsed Time used, and a Database graph.

Number of Statements, CPU Time, Database Time Report ('SN' action)

Returns statement execution statistics, CPU time used, and Database Elapsed time used.

Percentage of Executions, CPU Time, and Database Time Report ('SP' action)

Returns percent of statements executed, percent of CPU time used, and percent of Database Elapsed time used.

Percent Graph of Executions, CPU and Database Report ('SG' action)

Returns percent graph of statements executed, percent graph of CPU time used, and percent graph of Database Elapsed time used.

Database/Work File Loops Report ('SF' action)

Returns execution counts and execution percentages for the Database Loops (i.e., READ, FIND, HISTOGRAM) and Work File Loops in each object.

Internal Subroutines/Non-Procedural Blocks Report ('SB' action)

Returns execution counts and execution percentages for the internal subroutines and non-procedural blocks (i.e., break processing) in each object. A Non-Procedural Block is one in which execution depends on an event, not on where the statement is located in an object. The non-procedural NATURAL statements are AT BREAK, AT END OF DATA, AT END OF PAGE, AT START OF DATA, AT TOP OF PAGE, BEFORE BREAK PROCESSING, IF NO RECORDS FOUND, and ON ERROR.

FOR/REPEAT Loops Report ('SR' action)

Returns execution counts and execution percentages for the FOR/REPEAT loops in each object.

IF/DECIDE Conditions Report ('SI' action)

Returns execution counts and execution percentages for the IF/DECIDE conditions in each object.

Percent Executed by Statement Type Report ('ST' action)

Returns execution percentages for the Database Loops/Work File Loops, internal subroutines/non-procedural blocks, FOR/REPEAT loops, and IF/DECIDE conditions in each object.

Each report format may be retrieved in one of four sort orders.

- Object within Library (OBJ)
Returns statistics by library/object in alphabetical order.
- Statement Execution Count (STA)
Returns statistics sorted by total statement executions in descending order.
- CPU Time (CPU)
Returns statistics sorted by CPU time in descending order.
- Database Elapsed Time (DAT)
Returns statistics sorted by Database Elapsed Time in descending order.

The objects included in these summary reports can be limited using several criteria. Enter "Y" in the 'View/Amend Thresholds?' field to invoke the Session Thresholds window.

```

PRO0096: 'S'elect Object to see its Source Code Listing Report.
                Summary Report for Session
2011-11-20 15:27      Session PAYROLL      USER24  PAYTEST
Report Format S Sort Order OBJ Types _____ QA? N Page 1__ of 1
Start Library _____ Start Object _____ View/Amend Thresholds y
                Total Stmt Execs                2263

S                T                Total Exec      Exec      %Exec      % Graph of
e                y                Stmt utbl      utbl      -utbl      Executable
l Library Object p                Count      Execs Stmt      Exec      Stmts      Statements
_ PAYTEST PAYR ■                View/Amend Thresholds
_ PAYTEST PAYO                Exclude Objects where.. ..is less than
_ PAYTEST PAYO                .. Run Count for Object _____ 0 runs
_ PAYTEST PAYO                .. Stmts Executed for Object _____ 0 statements
_ PAYTEST PAYO                .. CPU Time used by Object _____ 0.0000 msecs
_ PAYTEST PAYO                .. D'base Elapsed Time for Object _____ 0.0000 msecs

                or Object's Percentage of.. ..is less than
                .. Executable Stmts Executed ___0.00 %
                .. Total Session CPU Time used ___0.00 %
                .. Total Session D'base Elapsed ___0.00 %

Enter-PF1---PF2---
                Help User
    
```

Enter '10' in the 'Stmts Executed for Object' field and press Enter.

```

PRO0103: 5 objects EXCLUDED due to Threshold settings.
                Summary Report for Session
2011-11-20 15:29      Session PAYROLL      USER24  PAYTEST
Report Format S Sort Order OBJ Types _____ QA? N Page 1__ of 1
Start Library _____ Start Object _____ View/Amend Thresholds N
                Total Stmt Execs                2263

S                T                Total Exec      Exec      %Exec      % Graph of
e                y                Stmt utbl      utbl      -utbl      Executable
l Library Object p                Count      Execs Stmt      Exec      Stmts      Statements
_ PAYTEST PAY0100P P                1                59      26      20      76.92      *****

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
                Help User End Stmt CPU Dbase Up Down Sourc Left Right Exit
    
```

Only program PAY0100P in library PAYTEST satisfied the threshold entered. Message 'PRO0103: 5 objects EXCLUDED due to Threshold settings.' is displayed at the top of the screen.

V.2.1 Statement Execution Count Summary Report

The Statement Execution Count Summary Report lists statistics about statement executions for each object executed within a profile session that satisfies the report parameters.

This report identifies the extent to which an object has been tested and which objects have excessive statement executions. The report also helps to identify poorly-designed objects.

Use action code 'SS' in the 'Com' field for the desired Profile session on the "session list" menu.

```

PRO0096: 'S'elect Object to see its Source Code Listing Report.
                Summary Report for Session
2011-11-20 16:23      Session PAYROLL EXAMPLE      USER24  PAYTEST
Report Format S  Sort Order OBJ Types _____  QA? N  Page 1__ of 1
Start Library _____ Start Object _____  View/Amend Thresholds N
                Total Stmt Execs                80

S          T          Run          Total Exec          %Exec  % Graph of
e          Y          Count          Stmt utbl          -utbl  Executable
l Library  Object p          Execs Stmt          Exec   Exec   Executed
- NATLIB1  PROG001 P          3          114  45          40    88.9  *****
- NATLIB1  PROG090 P          1          110  55          12    21.8   **
- NATLIB1  PROG012 P          2          99   50          27    54.0  *****
- NATLIB1  PROG003 P          4          83   33           1     3.0
- NATLIB1  PROG055 P          1          76   25          15    60.0  *****
- NATLIB1  PROG110 P          1          54   97          42    43.3  *****
- NATLIB1  PROG008 P          2          23   15          11    73.3  *****

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help User End  Stmts CPU  Dbase Up  Down  Sourc Left Right Exit

```

Field	Description
Total Stmt Execs	The total number of statements executed by objects during a profiling session.
Sel	An "S" in the Sel field next to a library/object invokes the Source Code Listing Report, which lists statistics and source code for that individual object. Pressing PF3 from the Source Code Listing Report returns to this report.
User-ID (Use PF2 to toggle)	The User-ID of the user who profiled the object.
Library (Use PF2 to toggle)	The library containing the profiled object.
Object	The profiled object.
Typ	The NATURAL object type of the profiled object. Valid values are as follows:
	P Program
	N Subprogram
	S Subroutine
	M Map
	H Helproutine

(continued from the previous page)

Field	Description
Run Count	The number of times the object executed.
Total Stmt Execs	The number of statement executions for the object during a profile session.
Executbl Stmt	The number of executable statements in the object. NATURAL includes executable statements as part of its object code. PROFILER only collects statistics on these executable statements. Non-executable statements, such as comments and continuation lines, are not included in these statistics.
Executbl Stmt Exec	The number of executable statements executed at least once during object profiling and counted only once.
%Executbl Stmts Exec	This percentage points out the extent to which an object has been tested.
%Graph of Executable Statements Executed	The visual representation of the percent of Executable Statement Executions. Each * represents 10 percent or greater of the executable statements executed.

V.2.2 CPU Time Summary Report

The CPU Time Summary Report lists statistics for CPU time usage for each object executed within a profile session that satisfies the report parameters.

This report may be used to determine which objects have excessive CPU time and may be affecting system performance.

Use action code 'SC' in the 'Com' field for the desired Profile session on the "session list" menu.

```

PRO0096: 'S'elect Object to see its Source Code Listing Report.
                Summary Report for Session
2011-11-20 16:23      Session PAYROLL EXAMPLE      USER24  PAYTEST
Report Format C  Sort Order OBJ Types _____ QA? N  Page 1__ of 1
Start Library _____ Start Object _____ View/Amend Thresholds N
                Total CPU Time      142.9120 msec

s
e
l  Library  Object  y  Run  CPU  % of CPU  Graph of
   _  _  _  _  _  _  _  _  _  _  _  _
   _ PAYTEST PAYROLL P    1    2.1760    1.52
   _ PAYTEST PAY0100M M    4   38.7200   27.09  **
   _ PAYTEST PAY0100P P    1   29.0560   20.33  **
   _ PAYTEST PAY0110M M    1   26.5600   18.58   *
   _ PAYTEST PAY0125M M    1   17.3440   12.14   *
   _ PAYTEST PAY0130M M    1   29.0560   20.33  **

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  User  End  Stmt CPU  Dbase Up  Down  Sourc Left  Right Exit

```

Field	Description
Total CPU Time	The total number of milliseconds of CPU time used by all objects in the session that satisfy the report parameters.
Sel	An "S" in the Sel field next to a library/object invokes the Source Code Listing Report, which lists statistics and source code for that individual object. Pressing PF3 from the Source Code Listing Report returns to this report.
User-ID (Use PF2 to toggle)	The User-ID of the user who profiled the object.
Library (Use PF2 to toggle)	The library containing the profiled object.
Object	The profiled object.
Typ	The NATURAL object type of the profiled object. Valid values are as follows:
	P Program
	N Subprogram
	S Subroutine
	M Map
	H Helproutine

(continued from the previous page)

Field	Description
Run Count	The number of times the object executed.
CPU Time (msec)	The number of milliseconds of CPU time used by each object during a profile session.
% of CPU Time Used	The percent of the total CPU time for all objects that satisfy the report parameters used by an object.
Graph of % of CPU Time Used	The visual representation of the percent CPU time used by each object. Each * represents 10 percent or greater of the CPU time.

V.2.3 Database Elapsed Time Summary Report

The Database Elapsed Time Summary Report lists statistics for Database Elapsed Time usage for each object executed within a profile session that satisfies the report parameters.

This report may be used to determine which objects have excessive Database Elapsed Time and may be using inefficient database access methods.

Use action code 'SD' in the 'Com' field for the desired Profile session on the "session list" menu.

```

PRO0096: 'S'elect Object to see its Source Code Listing Report.
                Summary Report for Session
2011-11-20 16:23      Session PAYROLL EXAMPLE      USER24  PAYTEST
Report Format D  Sort Order OBJ  Types _____  QA? N  Page 1__ of 1
Start Library _____ Start Object _____  View/Amend Thresholds N
                Total Database Elapsed      1005.5840 msec
s                T                Database      Graph of
e                y                Elapsed      % of Dbase % of Dbase
l Library      Object p      Count      Time (msec) Elaps Time Elaps Time
- PAYTEST    PAYROLL P      1
- PAYTEST    PAY0100M M      4
- PAYTEST    PAY0100P P      1      1005.5840      100.00      *****
- PAYTEST    PAY0110M M      1
- PAYTEST    PAY0125M M      1
- PAYTEST    PAY0130M M      1

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
                Help  User  End  Stmt CPU  Dbase Up  Down  Sourc Left  Right Exit

```

Field	Description
Total Database Elapsed	The total number of milliseconds of Database Elapsed Time used by all objects in the session that satisfy the report parameters.
Sel	An "S" in the Sel field next to a library/object invokes the Source Code Listing Report, which lists statistics and source code for that individual object. Pressing PF3 from the Source Code Listing Report returns to this report.
User-ID (Use PF2 to toggle)	The User-ID of the user who profiled the object.
Library (Use PF2 to toggle)	The library containing the profiled object.
Object	The profiled object.
Typ	The NATURAL object type of the profiled object. Valid values are as follows:
	P Program
	N Subprogram
	S Subroutine
	M Map
	H Helproutine

(continued from the previous page)

Field	Description
Run Count	The number of times the object executed.
Database Elapsed Time (msec)	The number of milliseconds of Database Elapsed Time used by each object during a profile session.
% of Dbase Elaps Time	The percent of the total Database Elapsed Time for all objects that satisfy the report parameters used by an object.
Graph of % of Dbase Elaps Time	The visual representation of the Database Elapsed Time used by each object. Each * represents 10 percent or greater of the Database Elapsed Time.

V.2.4 Number of Statements, CPU Time, Database Time Report

The Number of Statements, CPU Time, Database Time Report lists statistics for number of Statement Executions, CPU Time usage, and Database Elapsed Time usage for each object executed within a profile session that satisfies the report parameters.

This report may be used to determine which objects have excessive Statement Executions, CPU Time, and/or Database Elapsed Time and may be using inefficient design or database access methods.

Use action code 'SN' in the 'Com' field for the desired Profile session on the "session list" menu.

```

PRO0096: 'S'elect Object to see its Source Code Listing Report.
                Summary Report for Session
2011-11-20 16:23      Session PAYROLL EXAMPLE      USER24  PAYTEST
Report Format N Sort Order OBJ Types _____ QA? N Page 1__ of 1
Start Library _____ Start Object _____ View/Amend Thresholds N
Total Stmt Execs 80      Exec Total CPU 142.9120
s          T          Total utbl          Database
e          Y          Stmt Stmt          CPU Time
l Library  Object p   Count  Execs Exec  Used (msec)  Elapsed
- PAYTEST  PAYROLL P    1      1    1      2.176
- PAYTEST  PAY0100M M    4      8    2      38.720
- PAYTEST  PAY0100P P    1     59  20     29.056      1005.5840
- PAYTEST  PAY0110M M    1      4    2      26.560
- PAYTEST  PAY0125M M    1      4    2      17.344
- PAYTEST  PAY0130M M    1      4    2      29.056

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10---PF11---PF12---
      Help User End  Stmt CPU  Dbase Up   Down Sourc Left Right Exit

```

Field	Description
Total Stmt Execs	The total number of statements executed by objects during a profiling session.
Total CPU	The total number of milliseconds of CPU time used by all objects in the session that satisfy the report parameters.
Sel	An "S" in the Sel field next to a library/object invokes the Source Code Listing Report, which lists statistics and source code for that individual object. Pressing PF3 from the Source Code Listing Report returns to this report.
User-ID (Use PF2 to toggle)	The User-ID of the user who profiled the object.
Library (Use PF2 to toggle)	The library containing the profiled object.
Object	The profiled object.

(continued from the previous page)

Field	Description
Typ	The NATURAL object type of the profiled object. Valid values are as follows: P Program N Subprogram S Subroutine M Map H Helproutine
Run Count	The number of times the object executed.
Total Stmt Execs	The number of statement executions for the object during a profile session.
Executbl Stmt Exec	The number of executable statements executed at least once during object profiling and counted only once.
CPU Time (msec)	The number of milliseconds of CPU time used by each object during a profile session.
Database Elapsed Time (msec)	The number of milliseconds of Database Elapsed Time used by each object during a profile session.

V.2.5 Percentage of Executions, CPU Time, and Database Time Report

The Percentage of Executions, CPU Time, and Database Time Report lists percentage statistics for Executable Statement Executed, CPU Time usage, and Database Elapsed Time usage for each object executed within a profile session that satisfies the report parameters.

This report may be used to determine which objects have insufficient statements tested, and/or excessive CPU Time, and/or excessive Database Elapsed Time and may be using inefficient design or database access methods.

Use action code 'SP' in the 'Com' field for the desired Profile session on the "session list" menu.

```

PRO0096: 'S'elect Object to see its Source Code Listing Report.
                Summary Report for Session
2011-11-20 16:23      Session PAYROLL EXAMPLE      USER24  PAYTEST
Report Format P  Sort Order OBJ Types _____  QA? N  Page 1__ of 1
Start Library _____ Start Object _____ View/Amend Thresholds N
Total Stmt Execs 80      Total CPU 142.9120
s                T                % Executable
e                y                Run   Statements
l Library  Object p   Count   Executed   % of CPU   % of Dbase
- PAYTEST  PAYROLL P     1     50.00     1.52
- PAYTEST  PAY0100M M     4    100.00    27.09
- PAYTEST  PAY0100P P     1     76.92    20.33      100.00
- PAYTEST  PAY0110M M     1    100.00    18.58
- PAYTEST  PAY0125M M     1    100.00    12.14
- PAYTEST  PAY0130M M     1    100.00    20.33

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10---PF11---PF12---
      Help User End  Stmt CPU   Dbase Up   Down Sourc Left Right Exit

```

Field	Description
Total Stmt Execs	The total number of statements executed by objects during a profiling session.
Total CPU	The total number of milliseconds of CPU time used by all objects in the session that satisfy the report parameters.
Sel	An "S" in the Sel field next to a library/object invokes the Source Code Listing Report, which lists statistics and source code for that individual object. Pressing PF3 from the Source Code Listing Report returns to this report.
User-ID (Use PF2 to toggle)	The User-ID of the user who profiled the object.
Library (Use PF2 to toggle)	The library containing the profiled object.
Object	The profiled object.

(continued from the previous page)

Field	Description
Typ	The NATURAL object type of the profiled object. Valid values are as follows: P Program N Subprogram S Subroutine M Map H Helproutine
Run Count	The number of times the object executed.
%Executbl Stmt Exec	This percentage points out the extent to which an object has been tested.
% of CPU Time Used	The percent of the total CPU time for all objects that satisfy the report parameters used by an object.
% of Dbase Elaps Time	The percent of the total Database Elapsed Time for all objects that satisfy the report parameters used by an object.

V.2.6 Percent Graph of Executions, CPU and Database Report

The Percent Graph of Executions, CPU Time, and Database Time Report lists percent graph statistics for Executable Statement Executed, CPU Time usage, and Database Elapsed Time usage for each object executed within a profile session that satisfies the report parameters.

This report may be used to determine which objects have insufficient statements tested, and/or excessive CPU Time, and/or excessive Database Elapsed Time and may be using inefficient design or database access methods.

Use action code 'SG' in the 'Com' field for the desired Profile session on the "session list" menu.

```

PRO0096: 'S'elect Object to see its Source Code Listing Report.
                Summary Report for Session
2011-11-20 16:23      Session PAYROLL EXAMPLE      USER24  PAYTEST
Report Format G  Sort Order OBJ Types _____ QA? N  Page 1__ of 1
Start Library _____ Start Object _____ View/Amend Thresholds N
Total Stmt Execs 80      Total CPU 142.9120
s                T                % Graph of      Graph of      Graph of
e                y                Executable      % of CPU      % of Dbase
l Library      Object p      Run      Count      Stmts Exec      Time Used      Elaps Time
- PAYTEST PAYROLL P      1      1      *****
- PAYTEST PAY0100M M      4      4      ***** **
- PAYTEST PAY0100P P      1      1      ***** **
- PAYTEST PAY0110M M      1      1      ***** *
- PAYTEST PAY0125M M      1      1      ***** *
- PAYTEST PAY0130M M      1      1      ***** **

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help User End  Stmt CPU  Dbase Up   Down Sourc Left Right Exit

```

Field	Description
Total Stmt Execs	The total number of statements executed by objects during a profiling session.
Total CPU	The total number of milliseconds of CPU time used by all objects in the session that satisfy the report parameters.
Sel	An "S" in the Sel field next to a library/object invokes the Source Code Listing Report, which lists statistics and source code for that individual object. Pressing PF3 from the Source Code Listing Report returns to this report.
User-ID (Use PF2 to toggle)	The User-ID of the user who profiled the object.
Library (Use PF2 to toggle)	The library containing the profiled object.
Object	The profiled object.

(continued from the previous page)

Field	Description
Typ	The NATURAL object type of the profiled object. Valid values are as follows: P Program N Subprogram S Subroutine M Map H Helproutine
Run Count	The number of times the object executed.
%Graph of Executable Statements Executed	The visual representation of the percent of Executable Statement Executions. Each * represents 10 percent or greater of the executable statements executed.
Graph of % of CPU Time Used	The visual representation of the percent CPU time used by each object. Each * represents 10 percent or greater of the CPU time.
Graph of % of Dbase Elaps Time	The visual representation of the Database Elapsed Time used by each object. Each * represents 10 percent or greater of the Database Elapsed Time.

V.2.7 Database/Work File Loops Report

The Database/Work File Loops Report lists statistics about the Database/Work File Loops for objects executed during an active profile session. This report may help identify weaknesses in test data and incorrect search criteria.

For example:

```
FIND AUTOMOBILE-FILE WITH COLOR = 'RED'
      WRITE MAKE MODEL YEAR
END-FIND
```

If there are no red automobiles on the AUTOMOBILE file, the "FIND" entry condition is tested, but the body of the "FIND" loop is never entered.

Use action code 'SF' in the 'Com' field for the desired Profile session on the "session list" menu.

```
PRO0096: 'S'elect Object to see its Source Code Listing Report.
                Summary Report for Session
2011-11-20 16:23      Session PAYROLL EXAMPLE      USER24      PAYTEST
Report Format F  Sort Order OBJ Types _____  QA? N  Page 1__ of 1
Start Library _____ Start Object _____  View/Amend Thresholds N
                Total Stmt Execs 80
S                T
e                Y      Run      Total  DB/WF  %DB/WF  DB/WF  %DB/WF
l User-ID  Object  p      Count  Loops  Exec   Loops  Exec   Bodies Bodies
- USER24  PROG001  P        3      12     6   50.00   3   25.00
- USER23  PROG002  P        1        9     9  100.00   9  100.00
- USER23  PROG003  P        2      18     9   50.00   6   33.33
- USER22  PROG004  P        4        4     4  100.00   2   50.00
- USER19  PROG005  N        1
- USER17  PROG006  N        1        1
- USER21  PROG007  P        2        8     8  100.00   4   50.00

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  User  End  Stmts CPU  Dbase Up   Down  Sourc Left  Right Exit
```

Field	Description
Total Stmt Execs	The total number of statements executed by objects during a profiling session.
Sel	An "S" in the Sel field next to a library/object invokes the Source Code Listing Report, which lists statistics and source code for that individual object. Pressing PF3 from the Source Code Listing Report returns to this report.
User-ID (Use PF2 to toggle)	The User-ID of the user who profiled the object.
Library (Use PF2 to toggle)	The library containing the profiled object.
Object	The profiled object.

(continued from the previous page)

Field	Description
Typ	The NATURAL object type of the profiled object. Valid values are as follows: P Program N Subprogram S Subroutine M Map H Helproutine
Run Count	The number of times the object executed.
Total DB/WF Loops	The number of Database (i.e., READ, FIND, HISTOGRAM) and Work File (i.e., READ WORK FILE) Loops in an object.
DB/WF Loops Exec	The number of Database and Work File Loops in an object that were tested.
%DB/WF Loops Exec	The percent of Database and Work File Loops in an object that were tested.
DB/WF Bodies Exec	The number of Database and Work File Loops in an object in which at least one statement within the body of the loop was executed. The statement that ends the loop is also considered to be part of the body of the loop.
%DB/WF Bodies Exec	The percent of Database and Work File Loops in an object in which at least one statement within the body of the loop was executed.

V.2.8 Internal Subroutines/Non-Procedural Blocks Report

The Internal Subroutines/Non-Procedural Blocks Report lists statistics about the internal subroutines and the non-procedural blocks for each object within a profile session.

This report may be used to ensure that all automatic break processing statements have executed. If break processing does not occur in an object that contains internal subroutines or non-procedural blocks, then an object has not been tested thoroughly. Test data which causes all breaks to occur and all internal subroutines to be performed should be used.

Use action code 'SB' in the 'Com' field for the desired Profile session on the "session list" menu.

```

PRO0096: 'S'elect Object to see its Source Code Listing Report.
                Summary Report for Session
2011-11-20 16:23      Session PAYROLL EXAMPLE      USER24  PAYTEST
Report Format B Sort Order OBJ Types _____ QA? N Page 1__ of 1
Start Library _____ Start Object _____ View/Amend Thresholds N
Total Stmt Execs 80      Total CPU 142.9120
S
e
l User-ID  Object  P  Run  Total  Blocks  %Blocks  Graph of
  _ USER24  PROG001  P    3    12     6    50.00  *****
  _ USER22  PROG002  P    1     9     5    55.56  *****
  _ USER23  PROG003  P    2    18    18   100.00  *****
  _ USER19  PROG004  P    4     4     2    50.00  *****
  _ USER21  PROG005  N    1     1     0     0.00
  _ USER17  PROG006  N    1    14     9    64.29  *****
  _ USER18  PROG007  P    2     8     4    50.00  *****

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10---PF11---PF12---
      Help User End  Stmts CPU  Dbase Up   Down  Sourc Left  Right Exit

```

Field	Description
Total Stmt Execs	The total number of statements executed by objects during a profiling session.
Total CPU	The total number of milliseconds of CPU time used by all objects in the session that satisfy the report parameters.
Sel	An "S" in the Sel field next to a library/object invokes the Source Code Listing Report, which lists statistics and source code for that individual object. Pressing PF3 from the Source Code Listing Report returns to this report.
User-ID (Use PF2 to toggle)	The User-ID of the user who profiled the object.
Library (Use PF2 to toggle)	The library containing the profiled object.
Object	The profiled object.

(continued from the previous page)

Field	Description
Typ	The NATURAL object type of the profiled object. Valid values are as follows: P Program N Subprogram S Subroutine M Map H Helproutine
Run Count	The number of times the object executed.
IS/NP Blocks: Total Blocks	The number of Internal Subroutines and Non-Procedural Blocks within an object.
IS/NP Blocks: Blocks Exec	The number of Internal Subroutines and Non-Procedural Blocks that were executed at least once for each object in a profile session.
IS/NP Blocks: %Blocks Exec	The percent of Internal Subroutines and Non-Procedural Blocks that were executed at least once for each object in a profile session.
IS/NP Blocks: Graph of Blocks Exec	The percent graph of Internal Subroutines and Non-Procedural Blocks that were executed at least once for each object in a profile session.

V.2.9 FOR/REPEAT Loops Report

The FOR/REPEAT Loops Report lists statistics about the number and execution percentages of the FOR and REPEAT Loops for objects executed during an active profile session.

Use action code 'SR' in the 'Com' field for the desired Profile session on the "session list" menu.

```

PRO0096: 'S'elect Object to see its Source Code Listing Report.
                Summary Report for Session
2011-11-20 16:23      Session PAYROLL EXAMPLE      USER24  PAYTEST
Report Format R  Sort Order OBJ Types _____  QA? N  Page 1__ of 1
Start Library _____ Start Object _____ View/Amend Thresholds N
Total Stmt Execs 80      Total CPU 142.9120
S
e
l  Library      Object  P  Run  Total  Loops  %Loops  Graph of
   NATLIB1     PROG001 P    2    2    2    100.00  *****
   NATLIB1     PROG002 P    4    4    4    100.00  *****
   NATLIB1     PROG003 P    1   11    6    54.55   *****
   NATLIB1     PROG004 P    1    5    4    80.00   *****
   NATLIB1     PROG005 N    6   24   20   83.33   *****
   NATLIB1     PROG006 N    2
   NATLIB1     PROG007 P    1    3
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help User  End  Stmts CPU  Dbase Up  Down  Sourc Left  Right Exit

```

Field	Description
Total Stmt Execs	The total number of statements executed by objects during a profiling session.
Total CPU	The total number of milliseconds of CPU time used by all objects in the session that satisfy the report parameters.
Sel	An "S" in the Sel field next to a library/object invokes the Source Code Listing Report, which lists statistics and source code for that individual object. Pressing PF3 from the Source Code Listing Report returns to this report.
User-ID (Use PF2 to toggle)	The User-ID of the user who profiled the object.
Library (Use PF2 to toggle)	The library containing the profiled object.
Object	The profiled object.
Typ	The NATURAL object type of the profiled object. Valid values are as follows: <ul style="list-style-type: none"> P Program N Subprogram S Subroutine M Map H Helproutine

(continued from the previous page)

Field	Description
Run Count	The number of times the object executed.
FOR/REPEAT Loops: Total Loops	The number of FOR Loops and REPEAT Loops in the object.
FOR/REPEAT Loops: Loops Exec	The number of FOR Loops and REPEAT Loops in an object which executed at least once.
FOR/REPEAT Loops: %Loops Exec	The percent of FOR Loops and REPEAT Loops in an object which executed at least once.
FOR/REPEAT Loops: Graph of Loops Exec	The percent graph of FOR Loops and REPEAT Loops in an object program which executed at least once.

V.2.10 IF/DECIDE Conditions Report

The IF/DECIDE Conditions Report lists statistics about the IF/DECIDE conditions in objects executed during an active session. This report can be used to determine if all of the conditional statements in an object have been executed.

Use action code 'S' in the 'Com' field for the desired Profile session on the "session list" menu.

```

PRO0096: 'S'elect Object to see its Source Code Listing Report.
                Summary Report for Session
2011-11-20 16:23      Session PAYROLL EXAMPLE      USER24  PAYTEST
Report Format I  Sort Order OBJ Types _____ QA? N  Page 1__ of 1
Start Library _____ Start Object _____ View/Amend Thresholds N
Total Stmt Execs 80      Total CPU 142.9120
S                T                IF / DECIDE Conditions/Bodies:
e                y                Run  Total  Conds %Cond  Total  Body %Body
l Library  Object p  Count  Conds  Exec  Exec  Body  Exec  Exec
- NATLIB1  PROG001 P    3    10    5  50.0    10    1  10.0
- NATLIB1  PROG002 P    1     1    1 100.0     2    2 100.0
- NATLIB1  PROG003 P    2     8    7  87.5     8    2  25.0
- NATLIB1  PROG004 P    4    16    6  37.5    17    2  11.7
- NATLIB1  PROG005 N    1
- NATLIB1  PROG006 N    1     4    4 100.0     4    4 100.0
- NATLIB1  PROG007 P    2     6    6 100.0     8    8 100.0

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help User  End  Stmts CPU  Dbase Up  Down  Sourc Left Right Exit

```

Field	Description
Total Stmt Execs	The total number of statements executed by objects during a profiling session.
Total CPU	The total number of milliseconds of CPU time used by all objects in the session that satisfy the report parameters.
Sel	An "S" in the Sel field next to a library/object invokes the Source Code Listing Report, which lists statistics and source code for that individual object. Pressing PF3 from the Source Code Listing Report returns to this report.
User-ID (Use PF2 to toggle)	The User-ID of the user who profiled the object.
Library (Use PF2 to toggle)	The library containing the profiled object.
Object	The profiled object.

(continued from the previous page)

Field	Description
Typ	The NATURAL object type of the profiled object. Valid values are as follows: P Program N Subprogram S Subroutine M Map H Helproutine
Run Count	The number of times the object executed.
IF/DECIDE: Total Conds	The number of IF conditions and DECIDE conditions in an object. An IF condition is counted as one IF/DECIDE condition. An ELSE condition that is part of an IF statement is not counted as a separate IF/DECIDE condition. Each VALUE/WHEN clause, except NONE IGNORE, in a DECIDE statement is counted as one IF/DECIDE condition.
IF/DECIDE: Conds Exec	The number of IF conditions and DECIDE conditions in an object which executed at least once. IF statements that are executed are counted as one Condition Executed. An ELSE statement that is part of an IF statement is not counted as a Condition Executed. Each VALUE/WHEN that is executed, except NONE IGNORE, is counted as one Condition Executed.
IF/DECIDE: %Cond Exec	The percent of IF conditions and DECIDE conditions in an object which executed at least once.
IF/DECIDE: Total Body	The number of IF bodies, ELSE bodies, and VALUE/WHEN bodies (except NONE IGNORE) in an object. A body is the action statement(s) associated with an IF condition, an ELSE condition, or a VALUE/WHEN clause (except NONE IGNORE).
IF/DECIDE: Body Exec	The number of IF Bodies, ELSE Bodies, and VALUE/WHEN Bodies (except NONE IGNORE) which executed at least once.
IF/DECIDE: %BodyS Exec	The percent of IF Bodies and DECIDE Bodies in an object which executed at least once.

V.2.11 Percent Executed by Statement Type Report

The Percent Executed by Statement Type Report lists execution percentages for the Database Loops/Work File Loops, Internal Subroutines/Non-Procedural Blocks, FOR/REPEAT loops, and IF/DECIDE conditions in objects executed during an active session. This report can be used to determine if all important statement types in an object have been executed.

Use action code 'ST' in the 'Com' field for the desired Profile session on the "session list" menu.

```

PRO0096: 'S'elect Object to see its Source Code Listing Report.
                Summary Report for Session
2011-11-20 16:23      Session PAYROLL EXAMPLE      USER24  PAYTEST
Report Format T  Sort Order OBJ Types          QA? N  Page 1__ of 1
Start Library _____ Start Object _____ View/Amend Thresholds N
Total Stmt Execs 80              Total CPU 142.9120
S                T
e                y
l Library      Object p      Run  Loops  Bodys  Block  Loops  Conds  Bodys
- USER24     PROG001 P      3   50.00  25.00  50.00  100.00  50.0  10.0
- USER23     PROG002 P      1  100.00 100.00  55.56  100.00 100.0  100.0
- USER23     PROG003 P      2   50.00  33.33  100.00  54.55  87.5  25.0
- USER22     PROG004 P      4  100.00  50.00  50.00  80.00  37.5  11.7
- USER19     PROG005 N      1
- USER17     PROG006 N      1                      64.29  100.0  100.0
- USER21     PROG007 P      2  100.00  50.00  50.00          100.0  100.0

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help User End  Stmts CPU  Dbase Up  Down  Sourc Left Right Exit

```

Field	Description
Total Stmt Execs	The total number of statements executed by objects during a profiling session.
Total CPU	The total number of milliseconds of CPU time used by all objects in the session that satisfy the report parameters.
Sel	An "S" in the Sel field next to a library/object invokes the Source Code Listing Report, which lists statistics and source code for that individual object. Pressing PF3 from the Source Code Listing Report returns to this report.
User-ID (Use PF2 to toggle)	The User-ID of the user who profiled the object.
Library (Use PF2 to toggle)	The library containing the profiled object.
Object	The profiled object.

(continued from the previous page)

Field	Description
Typ	The NATURAL object type of the profiled object. Valid values are as follows: P Program N Subprogram S Subroutine M Map H Helproutine
Run Count	The number of times the object executed.
%DB/WF Loops Exec	The percent of Database and Work File Loops in an object that were tested.
%DB/WF Bodies Exec	The percent of Database and Work File Loops in an object in which at least one statement within the body of the loop was executed.
%IS/NP Blocks: %Blocks Exec	The percent of Internal Subroutines and Non-Procedural Blocks that were executed at least once for each object in a profile session.
FOR/REPEAT Loops: %Loops Exec	The percent of FOR Loops and REPEAT Loops in an object which executed at least once.
IF/DECIDE: %Cond Exec	The percent of IF conditions and DECIDE conditions in an object which executed at least once.
IF/DECIDE: %Body Exec	The percent of IF Bodies and DECIDE Bodies in an object which executed at least once.

V.3 Source Code Listing Report for Profiled Objects

The Source Code Listing Report gives statistics for an individual NATURAL object, displays the source code for an individual NATURAL object, and lists PROFILER statistics for each executable statement in the object. The report also indicates executable statements that did not execute.

NATURAL includes executable statements as part of its object code. PROFILER only collects statistics on these executable statements. Non-executable statements such as comments and continuation lines do not have statistics but are included in the report.

To display a 'Source Code Listing Report', enter 'S' beside any object on any 'Summary Report'.

```

PRO0096: 'S'elect Object to see its Source Code Listing Report.

                Summary Report for Session
2011-11-20 15:23      Session PAYROLL      USER24  PAYTEST
Report Format S  Sort Order OBJ  Types _____  QA? N  Page 1__ of 1
Start Library _____  Start Object _____  View/Amend Thresholds N
                Total Stmt Execs                2263

S                T                Total  Exec  Exec  %Exec  % Graph of
e                y                Stmt  utbl  utbl  Stmts  -utbl  Executable
l Library      Object p      Run   Stmt  Stmt  Exec  Exec  Executed
s PAYTEST     PAYROLL P      1     1    2    1    50.00  *****
_ PAYTEST     PAY0100M M      4     8    2    2    100.00  *****
s PAYTEST     PAY0100P P      1    59   26   20    76.92  *****
_ PAYTEST     PAY0110M M      1     4    2    2    100.00  *****
_ PAYTEST     PAY0125M M      1     4    2    2    100.00  *****
_ PAYTEST     PAY0130M M      1     4    2    2    100.00  *****

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
Help User End  Stmts CPU  Dbase Up   Down Sourc Left Right Exit

```

This report will not be displayed if the object selected has been SAVED, CATALOGed, or STOWed since profiling began. Such objects are marked with a (lower case) 's' or 'c' immediately to the right of the object type. Message "PRO0094: ... has been SAVED after Profiling. Reset Stats & Re-Profile." or "PRO0095: ... has been CATALOGed after Profiling. Reset Stats & Re-Profile." will be displayed at the top of the screen. Statistics for the object should be reset using the 'RS' session action and the object should then be re-profiled.

This report can not be displayed if the source code or object code for the selected object can not be found. Such objects are marked with an (upper case) 'S' or 'C' immediately to the right of the object type. Message "PRO0091: Object Code does not exist for ... in Library" or "PRO0093: Source NOT Found in LIB ... on Use PF9 to Alter." will be displayed at the top of the screen. Use PF9 (Sourc) to temporarily amend the location of source code for this report.

The maximum number of statements for which statistics may be accumulated for an object is based on the GETMAIN size. GETMAIN values may range from 10K (312 statements) to 50K (2,018 statements). If the statement limit is reached during the execution of the Source Code Listing Report, the message ">> Statistics Limit of ... stmts Reached <<" appears in the report.

PROFILER records individual statistics for Database Elapsed Time usage for the first 44 different database accessing statements in any NATURAL object. If the 44 database access limit is reached during the execution of the Source Code Listing Report, the message ">> Statistics Limit of 44 database stmts Reached <<" appears in the report.

After these statement limits are reached, PROFILER continues to collect statistics at the object level. Total Statement Executions, CPU time, and Database Elapsed Time are still accumulated and are accurate for the object.

Statistics for NATURAL copycode are collected separately from the object source code which contains the INCLUDE statement. Statistics for copycode are collected for up to 15 copycodes per object. Copycode statistics are available as part of the NATURAL source code listing. The Source Code Listing Report expands copycode through the use of PF2. If a user attempts to display more than 15 copycodes in the Source Code Listing Report, the message, "COPYCODE STATISTICS NOT KEPT FOR MORE THAN 15 INCLUDE STATEMENTS" appears on the screen.

Press Enter to display the 'Source Code Listing Report' with statistics for each individual line.

```

2011-11-20 15:25 Profile Session PAYROLL Object Profiled by USER24
Execs Total CPU Avg CPU > .....1.....2... PAY0100P Lib PAYTEST

0010 * Program: PAY0100P
0020 DEFINE DATA LOCAL USING PAY0100L
0030 END-DEFINE
1 0.003 0.003 0040 INCLUDE PAY0100C
0050 *
1 0.005 0.005 0060 SET KEY PF8 = PGM NAMED ' +
0070 PF20 = PGM NAMED ' +
0080 *
10 0.326 0.003 0090 READ EMPLOYEES BY NAME STARTING FROM PERSON
<D'base 1005.584 100.558>
10 0.030 0.003 0100 ADD 1 TO #I
10 0.036 0.004 0110 MOVE PERSONNEL-ID TO #ID(#I)
10 0.050 0.005 0120 COMPRESS FIRST-NAME MIDDLE-NAME INTO #NAM
10 0.040 0.004 0130 MOVE DEPT TO #DEPT(#I)
10 0.036 0.004 0140 MOVE JOB-TITLE TO #TITLE(#I)

PF1 ? PF2 COPY PF3 QUIT PF4 SCAN PF5 SC= PF6 SHOW Page 1 of 5
    
```

Statistics that round to zero display as 0.000.

Field	Description
Profile Session ...	The profiled session.
Object Profiled by ...	The User-ID of the user who profiled the object.
Execs	The number of times each statement executed during a profile session.
Total CPU	The number of milliseconds of CPU Time used by each statement during a profile session.
Avg CPU	The average number of milliseconds of CPU Time used by each statement during a profile session. This value is calculated by total CPU/statement count.

(continued from the previous page)

Field	Description
> _____	<p>Allows a specific line number or one of the following commands to be entered. Most of these commands can be invoked by use of a PF key.</p> <p>COPY Show Copycode (at cursor)</p> <p>Q{UIT} Quit from Object Listing</p> <p>SC{AN}.. SCAN for text string..</p> <p>SCAN=, SC= Repeat previous SCAN</p> <p>SHOW.. Show statement options..</p> <p>T{OP}, -- Go to Top of Object</p> <p>-P, - Up one Page of Object</p> <p>+P, + Down one Page of Object</p> <p>B{OT}, ++ Go to Bottom of Object</p> <p>L{EFT}, < Scroll to Left of Page</p> <p>R{IGHT}, > Scroll to Right of Page</p> <p>EXIT Exit from PROFILER</p>
....+....1....+....2...	The column positions of the source code line.
... Lib ...	The object name and library name of the profiled object.
Line markings (>, O, S)	<p>If an executable statement has not executed, it is marked with a ">". Non-executable statements such as comments and continuation lines appear in the report listing but have no statistics and are not marked.</p> <p>If the first statement in a block of NATURAL Optimized Code has executed, it is marked with "O".</p> <p>If the text scan is used, lines containing the text are marked with an "S".</p>
0010	The NATURAL statement line number.

(continued from the previous page)

Field	Description
<D'base 1005.584 100.558>	<p>The total number of milliseconds and the average number of milliseconds of Database Elapsed Time used by a statement during a profile session.</p> <p>The average number of milliseconds of Database Elapsed Time used by each statement is calculated by Total Database Elapsed Time/Statement Count.</p> <p>If the NATURAL statement accesses the database (i.e., READ, FIND, GET, HISTOGRAM, UPDATE, DELETE, STORE, END TRANSACTION, and BACKOUT TRANSACTION), database statistics are shown in brackets on the line directly below this statement.</p>
<D'base 1005.584 100.558>	<p>The total number of milliseconds and the average number of milliseconds of Database Elapsed Time used by a statement during a profile session.</p>

The following PF-keys are provided for the Source Code Listing Report:

Key	Function	Description
PF1	HELP	Provides help information about Source Code Listing Report
PF2	COPY	Displays copycode when the cursor is on a line that contains an INCLUDE statement
PF3	Q{UIT}	Quit from Object Listing (Returns to the 'Summary Session Report' menu)
PF4	SC{AN}..	SCAN for text string.
PF5	SCAN=, SC=	Repeat previous SCAN
PF6	SHOW..	Show statement options (report can be limited to any combination of executed, un-executed, and non-executable lines)
PF7	-P, -	Pages backward
PF8	+P, +	Pages forward
PF9	B{OT}, ++	Go to Bottom of Object
PF10	L{EFT}, <	Scrolls left
PF11	R{IGHT}, >	Scrolls right
PF12	EXIT	Exits PROFILER

V.4 Source Code Listing Report for Traced Objects

The Source Code Listing Report for Traced Objects lists each statement that was executed during the Trace statistics collection. Statements are listed in the exact order in which they were executed, showing the Library name, Object name, Statement number, and source code for the statement. The source code can also be scanned for particular text. A Trace session can report up to 13,500 executed statements relating to 200 objects at most.

NATURAL includes executable statements as part of its object code. PROFILER only collects statistics on these executable statements. Non-executable statements, such as comments and continuation lines, and un-executed statements are not included in the report.

To display the 'Source Code Listing Report' for a Trace session, enter 'S=' beside the required Trace session on the main "session list" menu. If there are no traced statements for the session, the message "PRO0087: No Trace data Exists for Session {session-name}" will be displayed.

```

*** PROFILER 4.3.2 for Natural ***

2011-11-20 15:07   Active Profile Session: NO           USER24
C A P             Active Trace Session: NO           PAYTEST
o c or
m t T Session Name      Date      Time      Owner      LastUser  Description
--- P PAYROLL           2011-06-05 16:52 USER24     USER23     Example for
--- P PAYROLL TEST      2011-06-02 15:33 USER24     USER27     Payroll Test
s= T PAYROLL TRACE      2011-11-20 15:06 USER24     USER24     Payroll Trace

From PAYROLL_____ Owner _____ Profile/Trace/All A

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
Help      End      Objts      Exit

```

Pressing Enter displays the 'Source Code Listing' report for the Trace session.

The maximum number of statements for which statistics may be accumulated is 13,500. If the statement limit is reached during the execution of the Source Code Listing Report, the message "PRO0086: Trace data NOT accumulated for more than 13,500 statements." is displayed.

Statements executed from NATURAL copycode are marked with a "C" (to the left of the source code line number) and the name of the copycode object is shown immediately above the lines of executed copycode.

```

2011-11-20 15:25   Report on Trace Session PAYROLL TRACE   USER24   PAYTEST

      Lines 1 to 15 of 2045 +...3...+...4...+...   PAYROLL Lib PAYTEST
0050 FETCH 'PAY0100P'
      ...+...1...+...2...+...3...+...4...+...   PAY0100P Lib PAYTEST
0050 INCLUDE PAY0100C
      ...+...1...+...2...+...3...+...4...+...   PAY0100C Lib PAYTEST
C 0010 SET KEY PF1 = PGM NAMED 'HELP'
      ...+...1...+...2...+...3...+...4...+...   PAY0100P Lib PAYTEST
0060 REPEAT
0070 INPUT USING MAP 'PAY0100M'
      ...+...1...+...2...+...3...+...4...+...   PAY0100M Lib PAYTEST
0012 INPUT      (      IP=OFF      HE='PAY0100H'      )
0058 END
      ...+...1...+...2...+...3...+...4...+...   PAY0100P Lib PAYTEST
0080 DECIDE ON FIRST *PF-KEY
0110 VALUE 'ENTR'
0120 PERFORM INPUT-CHECK
1320 IF #VALUE = ' '
1330 REINPUT 'Please enter a Value.' MARK *#VALUE ALARM
1370 END-IF
0070 INPUT USING MAP 'PAY0100M'
      ...+...1...+...2...+...3...+...4...+...   PAY0100M Lib PAYTEST
0012 INPUT      (      IP=OFF      HE='PAY0100H'      )

Start from Line _____ and/or Scan for _____ pfl Help

```

Field	Description
Report on Trace Session ...	The name of the traced session.
Lines ... to ... of ...	The relative number of the first traced line on the screen, the last traced line on the screen, and the total number of traced lines for the session.
+...3...+...4...+...	The column positions of the source code line.
... Lib ...	The object name and library name of the traced object (whose source code lines appear immediately below).
Line markings (C, L, S, W)	<p>If the traced statement was part of copycode, it is marked with a "C".</p> <p>If the traced statement was longer than 72 characters, it is marked with an "L".</p> <p>If the text scan is used, lines containing the text are marked with an "S".</p> <p>If the object that the traced statement comes from was SAVEd after the tracing was performed, it is marked with a "W".</p>
0010	The NATURAL statement line number.
Start from Line _____	Display the trace source code report from the relative trace line number specified.
and/or Scan for _____	Scan the trace source code report for the text specified and display the page of the report holding the next occurrence of the text. The scan is an 'absolute' scan.

The following PF-keys are provided for the Source Code Listing Report:

Key	Function	Description
PF1	Help	Provides help information about Source Code Listing Report
PF3	End	Quit from Object Listing (Returns to the "session list" menu)
PF7	Up	Pages backward
PF8	Down	Pages forward
PF12	Exit	Exits PROFILER

V.5 Application QA Report for Profiled Libraries

The Application QA (Quality Assurance) Report returns the percentages of an application library or a range of objects in a library that has been tested. It also shows a listing of which objects have not been tested at all. The source code of these untested objects can be viewed directly from this listing. Direct access is provided to the 'Statement Execution Count' Summary report, which shows the percentage of each object that has been tested. This report may be used as a quality assurance tool to show that an application has been thoroughly tested before it is migrated to a production environment.

Use action 'QA' on the main "session list" menu or enter "Y" in the 'QA?' field on any 'Summary Report' screen to invoke the Application QA Report. The library required for the QA report will then need to be chosen, which in this case is PAYTEST.

```

Summary Report for Session
2011-11-20 15:29      Session PAYROLL      USER24  PAYTEST
Report Format S      Sort Order OBJ Types _____ QA? y Page 1__ of 1
Start Library _____ Start Object _____ View/Amend Thresholds N
      Total Stmt Execs      2263
S          T          Total Exec      %Exec % Graph of
e          y          Stmt utbl      -utbl Executable
l Library Object p      Count      Excs Stmt Exec      Excs Executed
_ PAYTEST PAY0100P P      1          59 26 20 76.92 *****

      ■      Mark Profiled Library ...
              ... for Session PAYROLL
      x PAYTEST

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10---PF11---PF12---
      Help User End      Stmts CPU      Dbase Up      Down Sourc Left Right Exit
    
```

Press Enter and the Application QA Report will be displayed.

```

PRO0096: 'S'elect Object to see its Source Code Listing Report.

Summary Report for Session
2011-11-20 15:30      Session PAYROLL      USER24  PAYTEST
QA Report:      52 Objects in PAYTEST_      View Executed
of which      6 ( 11.54% ) were Executed.      Objects? N
Objects NOT Executed starting.. _____ types.. _____ Page 1__ of 2

s          T Exec s          T Exec s          T Exec s          T Exec
e          y utbl e          y utbl e          y utbl e          y utbl
l Object p Stmt l Object p Stmt l Object p Stmt l Object p Stmt
_ CITYTAXL L 1 _ FICAM M 9 _ LIFEINSS S 3 _ PAY0120M M 3
_ CITYTAXM M 9 _ FICAP P 14 _ PAYBATCH P 10 _ PAY0120P P 30
s CITYTAXP P 13 _ FICAS S 3 _ PAYEMPL L 0 _ PAY0120T M 2
_ CITYTAXS S 3 _ KAH0100M M 2 _ PAYKH P 30 _ PAY0130P P 28
_ CITYTX2L L 0 _ KAH0100P P 26 _ PAYL L 0 _ PAY0130T M 2
_ FEDTAXL L 1 _ KAH1080 P 874 _ PAYLOCL L 0 _ PAY0140M M 2
_ FEDTAXM M 9 _ KHBIBM M 2 _ PAYROLLG C 1 _ PAY0140P P 25
_ FEDTAXP P 15 _ LIFEINSL L 1 _ PAY0100T M 2 _ PAY0140T M 2
_ FEDTAXS S 3 _ LIFEINSM M 6 _ PAY0110P P 27 _ PENSIONL L 0
_ FICAL L 1 _ LIFEINSP P 13 _ PAY0110T M 2 _ PENSIONM M 6

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10---PF11---PF12---
      Help      End      Up      Down Sourc      Exit
    
```

Field	Description
Session	The profiled session.
... Objects in {library}	The total number of objects determined by the library, starting and ending objectprogram range, User-ID, and object types specified on the pop-up window.
of which ... (...%) were Executed	The total number of objects (and the percentage) determined by the library, starting and ending object program range, User-ID, and object types specified on the pop-up window that executed at least once. This statistic indicates that at least one statement in an object program was executed.
View Executed Objects?	Allows direct access to the 'Statement Execution Count' summary report for this session and library. This report shows which objects were at least partially executed.
Objects NOT Executed starting ... types ...	The user may choose a starting object name for the QA report and/or choose which object types are to be included in the QA report.
Page ... of ...	The current and maximum page of the report.
Sel	An "S" in the Sel field next to an object invokes the Source Code Listing Report, which lists the source code for that individual object. Pressing PF3 from the Source Code Listing Report returns to this report.
Object	The profiled object.
Typ	The NATURAL object type of the unexecuted object. Valid values are as follows: P Program N Subprogram S Subroutine M Map H Helproutine
Executbl Stmt	The number of executable statements in each object. NATURAL includes executable statements as part of its object code. PROFILER only collects statistics on these executable statements. Non-executable statements, such as comments and continuation lines, are not included in these statistics.

The following PF-keys are provided for the Application QA Report:

Key	Function	Description
PF1	Help	Provides help information about the Application QA Report
PF3	End	Returns to the pop-up window
PF7	Up	Pages backward
PF8	Down	Pages forward
PF9	Sourc	Allows the location of source code (i.e., libraries) to be changed.
PF12	Exit	Exits PROFILER

This page intentionally left blank.

SECTION VI

ENHANCED REPORTING

VI.1 Introduction to Enhanced Reporting

The Enhanced Reporting facility allows a user to report on statistics that have been moved from the PROFILER repository to the PROFILER Reporting file. This facility allows a user to report on statistics that have been combined for multiple users and sessions. Enhanced reports with combined statistics help to determine if an entire application has been fully tested.

The Enhanced Reporting facility also enables a user to customize reports. The layout of the PROFILER Reporting File is provided to facilitate the development of site-specific reports which display PROFILER statistics in any desired format.

Enhanced Reporting includes the following functions:

Tag and Move Data Actions

Allows users to move PROFILER data from the PROFILER repository to the PROFILER Reporting file.

Report Parameter Set Maintenance

Allows users to create report parameter sets for the PROFILER Enhanced Reports.

Enhanced Summary Reports

Allows users to display the statistics stored on the PROFILER Reporting file.

Enhanced Customized Reports

Allows a site to write custom reports using the PROFILER-REPORTING file and PROFILER User-exit PRFUE1P.

VI.2 Tag and Move Data Actions

The Tag and Move Data Actions allow users to move PROFILER data from the PROFILER repository to the PROFILER Reporting file. This facility also allows users to delete data from the PROFILER Reporting file.

The PROFILER repository is the ADABAS file where PROFILER collects and stores statistics. The PROFILER Reporting file is a separate ADABAS file where users move data for Enhanced Reporting.

Note: When a user moves data from the PROFILER repository to the PROFILER Reporting file, the data is deleted from the PROFILER repository and may only be accessed through Enhanced Reporting.

How to Invoke Tag and Move Data Actions

Enter '?' in a 'Com' field (or press PF1 while the cursor is in a 'Com' field) on the Main "session list" menu to access a help selection window of available "session actions". Mark the 'T?' (Tag / Move) action and press enter.

```

*** PROFILER 4.3.2 for Natural ***

2011-11-20 16:48   Ati   Mark Command ...
C A P             Acti   _ DI Display           USER24
o c or           _ MO Modify           PAYTEST
m t T Session Name _ CO Copy           Page 1_ of 1
_ P PAYROLL EXAMPLE _ PU Purge           LastUser Description
?_ P PAYROLL TEST   USER23 Example for
                   USER27 Payroll Test

                   _ AC Activate
                   _ DA De-Activate
                   _ RS Reset Stats
                   _ AU Active User *
                   Mark Tag Command ...
                   _ S= Stats Report   _ TG Tag Data for Move
                   _ S? Stats Help *   _ UT UnTag Data for Move
                   _ QA Applic QA *    _ LT List Tagged Data
                   _ CU Customized *   _ MT Move Tagged Data
                   x T? Tag / Move *
                   Profile/Trace/All P

From PAYROLL_____ Only for Profile: *

Enter-PF1---PF2---PF3---PF4   --PF9---PF10--PF11--PF12---
Help About End Admin E-Rep Backg Up Down New-S Left Right Exit

```

Select the required 'Tag Command' from the small window. Alternatively, the tag command (i.e., 'TG', 'UT', 'LT', or 'MT') can be entered directly against the required session on the "session list" menu.

VI.2.1 Tag Data for Move

The Tag Data for Move (TG) function allows a user to mark records on the PROFILER repository for transfer to the PROFILER Reporting file. These records may be moved to the PROFILER Reporting file with the Move Tagged Data (MT) function.

To Tag a session's data, enter action code 'TG' next to the session that is to be moved. Enter the optional parameters User-ID, library mask, object mask, and object types in the pop-up window.

```

*** PROFILER 4.3.2 for Natural ***

2011-11-20 15:06   Active Profile Session: NO           USER24
                  Act                                PAYTEST
C A P              Tag Data for Move...                1_ of 1
o c or            Session PAYROLL
m t T Session Name ■
tg P PAYROLL      Enter optional parameters below:    le for
_ P PAYROLL TEST                                     ll Test

                  User-ID: _____
                  Library Mask: _____
                  Object Mask: _____
                  Object Types: _____

                  Mask Options:
                  * Matches any remaining characters
                  ? Matches a single character
                  . Matches any numeric character
                  @ Matches any non-numeric character

From PAYROLL_____

Enter-PF1---PF2---PF3---P-PF11--PF12---
Help          End                                Exit

```

The following PF-keys are provided for the pop-up window:

Key	Function	Description
PF1	Help	Provides help information about the pop-up window
PF3	End	Returns to the "session list" menu
PF12	Exit	Exits PROFILER

Pressing Enter executes the Tag Data for Move function. During the tagging process, the Tag Data for Move function displays the message "Tagging in Progress ... records Tagged ". **Do not press any key on the keyboard while this message appears on the screen or an error (NAT 1016) will occur.** When the Tag Data for Move function is complete, the message "PRO0136: ... records have been Tagged for Move for Session ..." is displayed on return to the "session list" menu. If some data for a session has been tagged for move, then a 't' is shown next to that session in the 'Act' column on the "session list" menu.

VI.2.2 Untag Data for Move

The Untag Data (UT) function allows a user to remove tags placed on PROFILER repository records by the Tag Data for Move (TG) function.

To Untag a session's data, enter action code 'UT' next to the session required. Enter the optional parameters User-ID, library mask, object mask, and object types in the pop-up window.

```

*** PROFILER 4.3.2 for Natural ***

2011-11-20 15:06   Active Profile Session: NO           USER24
                  Ac  █                               PAYTEST
C  A  P                               Untag Data for Move...   1_ of 1
o  c  or                               Session PAYROLL
m  t  T Session Name                               iption
ut  t  P PAYROLL   Enter optional parameters below:   le for
_  _  P PAYROLL TEST                               ll Test

                  User-ID: _____
                  Library Mask: _____
                  Object Mask:  _____
                  Object Types:  _____

                  Mask Options:
                  * Matches any remaining characters
                  ? Matches a single character
                  . Matches any numeric character
                  @ Matches any non-numeric character

From PAYROLL_____

Enter-PF1---PF2---PF3---P--PF11--PF12---
      Help      End                                     Exit

```

The following PF-keys are provided for the pop-up window:

Key	Function	Description
PF1	Help	Provides help information about the pop-up window
PF3	End	Returns to the "session list" menu
PF12	Exit	Exits PROFILER

Pressing Enter executes the Untag Data function. During the untagging process, the Untag Data function displays the message "Un-Tagging in Progress ... records Un-Tagged". **Do not press any key on the keyboard while this message appears on the screen or an error (NAT 1016) will occur.** When the Untag Data function is complete, the message "PRO0140: ... records have been Un-Tagged from Move for Session ..." is displayed on return to the "session list" menu. If some data for the session remains tagged for move, then a 't' is shown next to the session in the 'Act' column on the "session list" menu.

VI.2.3 Move Selected Tagged Data

The Move Selected Tagged Data function allows a user to transfer selected records that have been tagged on the PROFILER repository to the PROFILER Reporting file.

To move selected tagged data for a session, enter action code 'MT' next to the session required. Enter the optional parameters User-ID, library mask, and object mask in the pop-up window.

```

*** PROFILER 4.3.2 for Natural ***

2011-11-20 15:06   Active Profile Session: NO           USER24
                  Act                                PAYTEST
C A P                               Move Tagged Records...   1_ of 1
o c or                               Session PAYROLL
m t T Session Name
mt t P PAYROLL      Enter optional parameters below:       le for
_ P PAYROLL TEST                                         ll Test

                  User-ID: _____
                  Library Mask: _____
                  Object Mask: _____
                  Object Types: _____

                  Mask Options:
                  * Matches any remaining characters
                  ? Matches a single character
                  . Matches any numeric character
                  @ Matches any non-numeric character

From PAYROLL_____

Enter-PF1---PF2---PF3---P0--PF11--PF12---
      Help           End                               Exit

```

The following PF-keys are provided for the pop-up window:

Key	Function	Description
PF1	Help	Provides help information about the pop-up window
PF3	End	Returns to the "session list" menu
PF5	Exit	Exits PROFILER

Pressing Enter executes the Move Selected Tagged Data function. During the move process, the Move Selected Tagged Data function displays the message "Move of Tagged Data in Progress ... objects Moved". **Do not press any key on the keyboard while this message appears on the screen or an error (NAT 1016) will occur.** When the Move Selected Tagged Data function is complete, the message "PRO0141: Statistics for ... objects have been Moved for Session ..." is displayed on return to the "session list" menu. If some data for the session remains tagged (i.e., not moved), then a 't' is shown next to the session in the 'Act' column on the "session list" menu.

VI.2.4 List Tagged Data

The List Tagged Data function allows a user to identify records that have been tagged on the PROFILER repository for transfer to the PROFILER Reporting file.

To list tagged data for a session, enter action code 'LT' next to the session required.

Press Enter to display the List Tagged Data Report.

```

*** PROFILER 4.3.2 for Natural ***

2011-11-20 15:06   Active Profile Session: NO                USER24
C A P
o c or           List Records Tagged for Move...
m t T Session    2011-11-20 15:07 Session PAYROLL        USER24   PROLIB
lt t P PAYROLL   Library   Object   User-ID       Library   Object   User-ID
  P PAYROLL      PAYTEST  PAYROLL  USER24       PAYTEST  PAYROLL  USER24
                PAYTEST  PAYROLL  USER24
                PAYTEST  PAYROLL  USER24
                PAYTEST  PAYROLL  USER24
                PAYTEST  PAYROLL  USER24
                PAYTEST  PAYROLL  USER24

From PAYROLL_____

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help      End      Up      Down      Exit
    
```

Field	Description
Session	The name the user assigned to the session in the Define New Session function.
Library	The library containing the profiled object.
Object	The profiled object.
User-ID	The User-ID of the user who profiled the object.

The following PF-keys are provided for the List Tagged Data function:

Key	Function	Description
PF1	Help	Provides help information about List Tagged Data
PF3	End	Returns to the "session list" menu
PF7	Up	Pages backward
PF8	Down	Pages forward
PF12	Exit	Exits PROFILER

VI.3 Report Parameter Set Maintenance

The Report Parameter Set Maintenance facility allows users to create report parameter sets for the PROFILER Enhanced Reports. Most Enhanced Reports require a report parameter set. User groups and session groups are optional for Enhanced Reports.

A report parameter set defines the selection criteria to be used for PROFILER Enhanced Reports. A user group defines a set of users to be included in a report parameter set. A session group defines a set of sessions to be included in a report parameter set.

Press PF5 (E-Rep) on the main "session list" menu.

```

*** PROFILER 4.3.2 for Natural ***

2011-11-20 16:44   Active Profile Session: NO           USER24
C A P             Active Trace Session: NO           PAYTEST
o c or                                     Page 1_ of 1
m t T Session Name      Date   Time   Owner   LastUser Description
_ P PAYROLL EXAMPLE    2011-06-05 16:52 USER24   USER23   Example for
_ P PAYROLL TEST       2011-06-12 15:33 USER24   USER27   Payroll Test

From PAYROLL_____ Owner _____ Profile/Trace/All P

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
Help About End Admin E-Rep Backg Up Down New-S Left Right Exit

```

The "report parameter list" menu will be displayed.

```

*** PROFILER 4.3.2 for Natural ***

2011-11-20 16:45   Active Profile Session: NO           USER24
C A P             Active Trace Session: NO           PAYTEST
o c or                                     Page 1_ of 1
m Report Parameter  Last Mod  Owner   User [Group]  Session [Group]
_ ALL SESSIONS     2011-06-03 USER24   *             *
_ PAY-PARAM        2011-06-03 USER27   PAY-TESTERS  *
_ TEST REP PARM    2011-06-02 USER24   *             PAYROLL TESTING

From _____ Owner _____ Amend Groups? _ Amend Groups? _

Enter-PF1---PF2---PF3---PF4-- PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
Help About End Admin Profl Backg Up Down New-P Left Right Exit

```

All Enhanced Reporting functions can be accessed from this screen via "report parameter actions" or PF keys. After performing any enhanced reporting function, the user is returned by default to this menu. Basic information about the user's current PROFILER environment is shown at the head of this menu.

Enter 'Y' against the 'Amend Groups?' field to invoke 'User Group' or 'Session Group' maintenance. Refer to the **User Group Maintenance** and **Session Group Maintenance** section for more information.

Report parameters matching the 'From' and 'Owner' settings will be displayed on the "report parameter list" menu. Listed report parameters can be limited to those "owned" (i.e., defined) by a particular user by entering a value in the "Owner" field. Leave the "Owner" field blank to list report parameters "owned" by all users. The "report parameter list" can be started from a chosen position by entering a value in the "From" field. Leave the "From" field blank to list all report parameters (matching the other settings). The required page of the "report parameter list" can be entered directly in the "Page" field. The entered value must not be less than one (1) or greater than the last page of the listing.

"Report parameter actions" are two-letter acronyms and can be entered in the 'Com' field beside the report parameter that is to be processed. Entering a "?" on the 'Com' field displays the field-level help screen below for "report parameter actions" (i.e., "report parameter commands").

```

*** PROFILER 4.3.2 for Natural ***

2011-11-20 16:45 Acti ■ USER24
C A P Acti Mark Command ... PAYTEST
o c or _ DI Display Page 1_ of 1
m Report Parameter Las _ MO Modify roup] Session [Group]
?_ ALL SESSIONS 2011 _ CO Copy *
_ PAY-PARAM 2011 _ PU Purge TERS *
_ TEST REP PARM 2011 _ E= Stats Report PAYROLL TESTING
_ E? Stats Help
_ EQ Applic QA

From _____ Owner _____ Amend Groups? _ Amend Groups? _

Enter-PF1---PF2---PF3---PF4-- PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
Help About End Admin Prof1 Backg Up Down New-P Left Right Exit

```

Mark the command that is required and press enter. The marked command will be copied into the 'Com' field ready for processing.

Refer to the **Report Parameter Maintenance** section for more information about the following "report parameter actions":

- DI Display Report Parameter Definition
- MO Modify Report Parameter Definition
- CO Copy Report Parameter Definition
- PU Purge Report Parameter Definition

Refer to the **Enhanced Summary Reports** and **Enhanced Customized Reports** sections for more information about the following "report parameter actions":

- E= Statistics Report for Report Parameter (most recent report format)
- E? Statistics Help (list of available report formats)
- ES Statement Execution Count Summary Report
- EC CPU Time Used Summary Report
- ED Database Elapsed Time Summary Report
- EN Number of Statements, CPU Time, Database Summary
- EP Percentage of Executions, CPU and Database Summary
- EG Percent Graph of Executions, CPU and Database Summary
- EF Database/Work File Loops Summary Report
- EB Internal Subroutines/Non-Procedural Blocks Summary
- ER FOR/REPEAT Loops Summary Report
- EI IF/DECIDE Conditions Summary Report
- ET Percent Executed by Statement Type Summary Report
- EQ Application Quality Assurance Report for Report Parameter

The following are the standard PF-keys for the "report parameter list" menu:

Key	Function	Description
PF1	Help	Provides cursor-sensitive help information
PF2	About	Displays installation, license, and environment details (refer to the 'About' Screen section for more details)
PF3	End	Returns to the previous screen (NATURAL Main menu)
PF4	Admin	Gives access to the Administrator options (refer to the PROFILER Administration for more information)
PF5	Profl	Returns to the main "session list" menu
PF6	Backg	Gives access to the Background Monitoring facility (refer to the Background Monitoring section for more information)
PF7	Up	Scrolls up the "report parameter list"
PF8	Down	Scrolls down the "report parameter list"
PF9	New-P	Allows a new report parameter to be defined
PF10	Left	Scrolls left on the "report parameter list"
PF11	Right	Scrolls right on the "report parameter list"
PF12	Exit	Exits PROFILER and returns to the NATURAL Main menu

VI.3.1 Report Parameter Maintenance

The Report Parameter Maintenance facility allows a user to specify selection criteria to be used for Enhanced Reports. Report parameter sets are stored on the PROFILER Reporting file by a unique report parameter name.

The following "report parameter actions" make up the Report Parameter Maintenance facility:

- PF9 (New-P) Define New Report Parameter Definition
- DI Display Report Parameter Definition
- MO Modify Report Parameter Definition
- CO Copy Report Parameter Definition
- PU Purge Report Parameter Definition

These actions will be described in the following sub-sections.

VI.3.1.1 Define Report Parameters

The Define Report Parameters function allows a user to add a report parameter set. A user must first define a report parameter set before executing an Enhanced Report.

This function allows a user to specify a group of User-IDs, a group of sessions, a library or range of libraries, an object or range of objects, and object types to be used as report parameters. Users may define up to 1,000 report parameter sets.

Press PF9 (New-P) on the "Report Parameter list" menu to display the Define New Report Parameter Set screen.

```

*** PROFILER 4.3.2 for Natural ***

2011-11-20 16:45  ■
C A P              Define New Report Parameter Set...
o c or            Report Parameter Name _____
m Report Parameter
  _ ALL SESSIONS   All Users _ or User [Group] _____
  _ PAY-PARAM     All Sessions _ or Session [Group] _____
  _ TEST REP PARM

                  Library Mask * _____ Object Mask * _____
                                Object Types _____

Combine statistics
for same object/library/catalog timestamp? _

From _____      Owner _____ Amend Groups? _ Amend Groups? _

Enter-PF1---PF2---PF3---
      Help      End                                Exit
    
```

Field	Description
Report Parameter Name	The name the user assigns to the report parameter set. The name must begin with an alphanumeric character and contain 1 to 16 characters.
All Users or User [Group]	<p>The User group to be included in a report parameter set. Entering "?" in the 'User [Group]' field displays a list of defined user groups and allows a user to select a user group. When a user selects a group, it automatically appears in the field.</p> <p>A user has two options when defining User-IDs to include in a report parameter set:</p> <ul style="list-style-type: none"> • Entering "X" in the 'All Users' field • Entering a user group in the 'User [Group]' field

(continued from the previous page)

Field	Description										
All Sessions or Session [Group]	<p>The session group to be included in a report parameter set. Entering "?" in the 'Session [Group]' field displays a list of defined sessions and allows a user to select a session group. When a user selects a group, it automatically appears in the field.</p> <p>A user has two options when defining sessions to include in a report parameter set:</p> <ul style="list-style-type: none"> • Entering "X" in the 'All Sessions' field • Entering a session group in the 'Session [Group]' field 										
Library Mask	<p>The NATURAL library or libraries to be included in a report parameter set. Mask options include the following:</p> <ul style="list-style-type: none"> * Matches any character in remainder of string ? Matches any single character . Matches any numeric character @ Matches any non-numeric character 										
Object Mask	<p>The NATURAL object or objects to be included in a report parameter set.</p>										
Combine Statistics...	<p>"Y" in this field confirms that PROFILER Reporting file statistics will be combined for objects with the same name in the same library and with the same catalog timestamp.</p> <p>"N" in this field indicates that PROFILER Reporting file statistics will not be combined for objects with the same name in the same library and with the same catalog timestamp.</p>										
Object Types	<p>The NATURAL object types to be included in a report parameter set.</p> <table data-bbox="810 1444 1062 1619"> <tr> <td>P</td> <td>Program</td> </tr> <tr> <td>N</td> <td>Subprogram</td> </tr> <tr> <td>S</td> <td>Subroutine</td> </tr> <tr> <td>M</td> <td>Map</td> </tr> <tr> <td>H</td> <td>Helproutine</td> </tr> </table> <p>Press PF1 on this field to access a help selection window.</p>	P	Program	N	Subprogram	S	Subroutine	M	Map	H	Helproutine
P	Program										
N	Subprogram										
S	Subroutine										
M	Map										
H	Helproutine										

Note:	Entering a library mask and an object mask substantially reduces processing time during Enhanced Reporting.
--------------	---

The following PF-keys are provided for the Define Report Parameters screen:

Key	Function	Description
PF1	Help	Provides help information about the Define Report Parameters screen
PF3	End	Returns to the "Report Parameter list" menu
PF12	Exit	Exits PROFILER

Enter the values required and press Enter to execute the Define Report Parameter function. The message "PRO0163: New Report Parameter Set ... Defined successfully." is displayed upon return to the "report parameter list" menu to confirm that PROFILER successfully defined the report parameter set.

VI.3.1.2 Display Report Parameters

The Display Report Parameters function allows a user to display a report parameter set.

To display a report parameter set's definition, enter action code 'DI' next to the report parameter set that is to be displayed.

```

*** PROFILER 4.3.2 for Natural ***
2011-11-20 16:45 ■
C A P                Display Report Parameter Set...
o c or              Report Parameter Name PAY-PARAM
m Report Parameter
  ALL SESSIONS      All Users or User [Group] PAY-TESTERS
di PAY-PARAM        All Sessions X or Session [Group]
  TEST REP PARM

Library Mask PAY*   Object Mask *
                   Object Types PMSMH

Combine statistics
for same object/library/catalog timestamp? Y

From _____    Owner _____ Amend Groups? _ Amend Groups? _

Enter-PF1---PF2---PF3---
      Help      End                                Exit

```

Field	Description
Report Parameter Name	The name the user assigns to the report parameter set. The name must begin with an alphanumeric character and contain 1 to 16 characters.
All Users or User [Group]	The User-IDs to be included in a report parameter set. Entering "?" in the 'User [Group]' field displays a list of defined user groups and allows a user to select a user group. When a user selects a group, it automatically appears in the field. A user has three options when defining User-IDs to include in a report parameter set: <ul style="list-style-type: none"> • Entering "X" in the 'All Users' field • Entering a user group in the 'User [Group]' field
All Sessions or Session [Group]	The sessions to be included in a report parameter set. Entering "?" in the 'Session [Group]' field displays a list of defined sessions and allows a user to select a session group. When a user selects a group, it automatically appears in the field. A user has three options when defining sessions to include in a report parameter set: <ul style="list-style-type: none"> • Entering "X" in the 'All Sessions' field • Entering a session group in the 'Session [Group]' field

(continued from the previous page)

Field	Description
Library Mask	<p>The NATURAL library or libraries to be included in a report parameter set. Mask options include the following:</p> <ul style="list-style-type: none"> * Matches any character in remainder of string ? Matches any single character . Matches any numeric character @ Matches any non-numeric character
Object Mask	The NATURAL object or objects to be included in a report parameter set.
Object Types	<p>The NATURAL object types to be included in a report parameter set.</p> <p>P Program N Subprogram S Subroutine M Map H Helproutine</p> <p>Press PF1 on this field to access a help selection window.</p>
Combine Statistics...	<p>"Y" in this field confirms that PROFILER Reporting file statistics will be combined for objects with the same name in the same library and with the same catalog timestamp.</p> <p>"N" in this field indicates that PROFILER Reporting file statistics will not be combined for objects with the same name in the same library and with the same catalog timestamp.</p>

The following PF-keys are provided for the Display Report Parameters screen:

Key	Function	Description
PF1	Help	Provides help information about the Display Report Parameters screen
PF3	End	Returns to the "Report Parameter list" menu
PF12	Exit	Exits PROFILER

VI.3.1.3 Modify Report Parameters

The Modify Report Parameters function allows a user to modify a report parameter set.

To modify a report parameter set's definition, enter action code MO next to the report parameter set that is to be modified.

```

*** PROFILER 4.3.2 for Natural ***

2011-11-20 16:45  ■
C A P              Modify Report Parameter Set...
o c or            Report Parameter Name PAY-PARAM
m Report Parameter All Users x or User [Group] _____
__ ALL SESSIONS  All Sessions X or Session [Group] _____
mo PAY-PARAM
__ TEST REP PARM

Library Mask paytest_ Object Mask *_____
                          Object Types NSMH

Combine statistics
for same object/library/catalog timestamp? Y

From _____ Owner _____ Amend Groups? _ Amend Groups? _

Enter-PF1---PF2---PF3--
      Help      End                               Exit

```

Field	Description
Report Parameter Name	The name the user assigns to the report parameter set. The name must begin with an alphanumeric character and contain 1 to 16 characters.
All Users or User [Group]	<p>The User-IDs to be included in a report parameter set. Entering "?" in the 'User [Group]' field displays a list of defined user groups and allows a user to select a user group. When a user selects a group, it automatically appears in the field.</p> <p>A user has three options when defining User-IDs to include in a report parameter set:</p> <ul style="list-style-type: none"> • Entering "X" in the 'All Users' field • Entering a user group in the 'User [Group]' field
All Sessions or Session [Group]	<p>The sessions to be included in a report parameter set. Entering "?" in the 'Session [Group]' field displays a list of defined sessions and allows a user to select a session group. When a user selects a group, it automatically appears in the field.</p> <p>A user has three options when defining sessions to include in a report parameter set:</p> <ul style="list-style-type: none"> • Entering "X" in the 'All Sessions' field • Entering a session group in the 'Session [Group]' field

(continued from the previous page)

Field	Description										
Library Mask	<p>The NATURAL library or libraries to be included in a report parameter set. Mask options include the following:</p> <ul style="list-style-type: none"> * Matches any character in remainder of string ? Matches any single character . Matches any numeric character @ Matches any non-numeric character 										
Object Mask	The NATURAL object or objects to be included in a report parameter set.										
Object Types	<p>The NATURAL object types to be included in a report parameter set.</p> <table border="0"> <tr><td>P</td><td>Program</td></tr> <tr><td>N</td><td>Subprogram</td></tr> <tr><td>S</td><td>Subroutine</td></tr> <tr><td>M</td><td>Map</td></tr> <tr><td>H</td><td>Helproutine</td></tr> </table> <p>Press PF1 on this field to access a help selection window.</p>	P	Program	N	Subprogram	S	Subroutine	M	Map	H	Helproutine
P	Program										
N	Subprogram										
S	Subroutine										
M	Map										
H	Helproutine										
Combine Statistics...	<p>"Y" in this field confirms that PROFILER Reporting file statistics will be combined for objects with the same name in the same library and with the same catalog timestamp.</p> <p>"N" in this field indicates that PROFILER Reporting file statistics will not be combined for objects with the same name in the same library and with the same catalog timestamp.</p>										

The following PF-keys are provided for the Modify Report Parameters screen:

Key	Function	Description
PF1	Help	Provides help information about the Modify Report Parameters screen
PF3	End	Returns to the "Report Parameter list" menu
PF12	Exit	Exits PROFILER

Change the values required and press Enter to execute the Modify Report Parameter function. The message "PRO0165: Report Parameter Set ... Modified successfully." is displayed upon return to the "report parameter list" menu to confirm that PROFILER successfully modified the report parameter set.

VI.3.1.4 Delete Report Parameters

The Delete Report Parameters function allows a user to delete a report parameter set.

To delete a report parameter set's definition, enter action code 'PU' next to the report parameter set that is to be deleted.

```

*** PROFILER 4.3.2 for Natural ***

2011-11-20 16:45 ■
C A P                               Purge Report Parameter Set...
o c or                             Report Parameter Name PAY-PARAM
m Report Parameter                 Proceed with PURGE of this Set? _
  ALL SESSIONS                     All Users or User [Group] PAY-TESTERS
pu PAY-PARAM                       All Sessions X or Session [Group]
  TEST REP PARM

Library Mask *                      Object Mask *
                                      Object Types PNSMH

Combine statistics
for same object/library/catalog timestamp? Y

From _____ Owner _____ Amend Groups? _ Amend Groups? _

Enter-PF1---PF2---PF3---PF4-- PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
Help      End                               Exit

```

Field	Description
Report Parameter Name	The name the user assigns to the report parameter set. The name must begin with an alphanumeric character and contain 1 to 16 characters.
Proceed with PURGE of this Set?	Enter "N" or press PF3 to end the delete function and return to the "Report Parameter list". Enter "Y" to continue the process of deleting a report parameter.
All Users or User [Group]	The User-IDs to be included in a report parameter set. Entering "?" in the 'User [Group]' field displays a list of defined user groups and allows a user to select a user group. When a user selects a group, it automatically appears in the field. A user has three options when defining User-IDs to include in a report parameter set: <ul style="list-style-type: none"> • Entering "X" in the 'All Users' field • Entering a user group in the 'User [Group]' field

(continued from the previous page)

Field	Description
All Sessions or Session [Group]	<p>The sessions to be included in a report parameter set. Entering "?" in the 'Session [Group]' field displays a list of defined sessions and allows a user to select a session group. When a user selects a group, it automatically appears in the field.</p> <p>A user has three options when defining sessions to include in a report parameter set:</p> <ul style="list-style-type: none"> • Entering "X" in the 'All Sessions' field • Entering a session group in the 'Session [Group]' field
Library Mask	<p>The NATURAL library or libraries to be included in a report parameter set. Mask options include the following:</p> <ul style="list-style-type: none"> * Matches any character in remainder of string ? Matches any single character . Matches any numeric character @ Matches any non-numeric character
Object Mask	<p>The NATURAL object or objects to be included in a report parameter set.</p>
Object Types	<p>The NATURAL object types to be included in a report parameter set.</p> <p>P Program N Subprogram S Subroutine M Map H Helproutine</p> <p>Press PF1 on this field to access a help selection window.</p>
Combine Statistics...	<p>"Y" in this field confirms that PROFILER Reporting file statistics will be combined for objects with the same name in the same library and with the same catalog timestamp.</p> <p>"N" in this field indicates that PROFILER Reporting file statistics will not be combined for objects with the same name in the same library and with the same catalog timestamp.</p>

The following PF-keys are provided for the Modify Report Parameters screen:

Key	Function	Description
PF1	Help	Provides help information about the Delete Report Parameters screen
PF3	END	Returns to the Report Parameter Maintenance menu
PF12	Exit	Exits PROFILER

After entering "Y", press Enter, and the message "PRO0166: Report Parameter Set ... Deleted successfully." will be displayed upon return to the "report parameter list" menu to confirm that PROFILER successfully deleted the report parameter set.

VI.3.1.5 Copy Report Parameters

The Copy Report Parameters function allows a user to copy a report parameter set.

To copy a report parameter set's definition, enter action code 'CO' next to the report parameter set that is to be copied.

```

*** PROFILER 4.3.2 for Natural ***

2011-11-20 16:45 ■
C A P          Copy Report Parameter Set...
o c or        Report Parameter Name _____
m Report Parameter
__ ALL SESSIONS      All Users _ or User [Group] PAY-TESTERS ____
co PAY-PARAM        All Sessions X or Session [Group] _____
__ TEST REP PARM

Library Mask * _____ Object Mask * _____
                          Object Types PNSMH

Combine statistics
for same object/library/catalog timestamp? Y

From _____      Owner _____ Amend Groups? _ Amend Groups? _

Enter-PF1---PF2---PF3-
Help      End                                Exit
    
```

Field	Description
Report Parameter Name	The name the user assigns to the report parameter set. The name must begin with an alphanumeric character and contain 1 to 16 characters.
All Users or User [Group]	<p>The User-IDs to be included in a report parameter set. Entering "?" in the 'User [Group]' field displays a list of defined user groups and allows a user to select a user group. When a user selects a group, it automatically appears in the field.</p> <p>A user has three options when defining User-IDs to include in a report parameter set:</p> <ul style="list-style-type: none"> • Entering "X" in the 'All Users' field • Entering a user group in the 'User [Group]' field
All Sessions or Session [Group]	<p>The sessions to be included in a report parameter set. Entering "?" in the 'Session [Group]' field displays a list of defined sessions and allows a user to select a session group. When a user selects a group, it automatically appears in the field.</p> <p>A user has three options when defining sessions to include in a report parameter set:</p> <ul style="list-style-type: none"> • Entering "X" in the 'All Sessions' field • Entering a session group in the 'Session [Group]' field

(continued from the previous page)

Field	Description
Library Mask	<p>The NATURAL library or libraries to be included in a report parameter set. Mask options include the following:</p> <ul style="list-style-type: none"> * Matches any character in remainder of string ? Matches any single character . Matches any numeric character @ Matches any non-numeric character
Object Mask	The NATURAL object or objects to be included in a report parameter set.
Object Types	<p>The NATURAL object types to be included in a report parameter set.</p> <p>P Program N Subprogram S Subroutine M Map H Helproutine</p> <p>Press PF1 on this field to access a help selection window.</p>
Combine Statistics...	<p>"Y" in this field confirms that PROFILER Reporting file statistics will be combined for objects with the same name in the same library and with the same catalog timestamp.</p> <p>"N" in this field indicates that PROFILER Reporting file statistics will not be combined for objects with the same name in the same library and with the same catalog timestamp.</p>

The following PF-keys are provided for the Modify Report Parameters screen:

Key	Function	Description
PF1	Help	Provides help information about the Copy Report Parameters screen
PF3	End	Returns to the "Report Parameter list" menu
PF12	Exit	Exits PROFILER

Change the values required, and press Enter to execute the Copy Report Parameter function. The message "PRO0163: New Report Parameter Set ... Defined successfully." is displayed upon return to the "report parameter list" menu to confirm that PROFILER successfully defined the new report parameter set.

VI.3.2 User Group Maintenance

The User Group Maintenance facility allows a user to specify a set of users to be included in a user group. The user group may be included in a report parameter set. User groups are stored in the PROFILER Reporting file by a unique user group name.

On the "report parameter list" menu there are two 'Amend Groups?' field, one below the 'User [Group]' heading and one below the 'Session [Group]' heading.

Enter 'Y' against the 'Amend Groups?' field below the 'User [Group]' heading to invoke 'User Group' maintenance.

```

*** PROFILER 4.3.2 for Natural ***

2011-11-20 16:45   Active Profile Session: NO           USER24
C A P             Active Trace Session: NO           PAYTEST
o c or
m Report Parameter Last Mod  Owner   User [Group]  Session [Group]
--- ALL SESSIONS    2011-06-03 USER24  *            *
--- PAY-PARAM      2011-06-03 USER27  PAY-TESTERS *
--- TEST REP PARM  2011-06-02 USER24  *            PAYROLL TESTING

From _____ Owner _____ Amend Groups? y Amend Groups? _

Enter-PF1---PF2---PF3---PF4-- PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
Help About End Admin Prof1 Backg Up Down New-P Left Right Exit

```

Press Enter to display the 'Amend User Groups' window.

```

*** PROFILER 4.3.2 for Natural ***

2011-11-20 16:45   Active Profile Session: NO           Amend User Groups...
C A P             Active Trace Session: NO           Com User Groups
o c or
m Report Parameter Last Mod  Owner   User [  Com
--- ALL SESSIONS    2011-06-03 USER24  *      ___ ALL-TESTERS
--- PAY-PARAM      2011-06-03 USER27  PAY-TE *      ___ PAY-TESTERS
--- TEST REP PARM  2011-06-02 USER24  *

From _____

From _____ Owner _____ Amend Groups? y Amend Groups? _

Enter-PF1---PF2---PF3---PF4-- PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
Help End Up Down New-G Exit

```

To choose a command to process against a user group, enter '?' in any 'Com' field. Available commands are: 'DI' (Display), 'MO' (Modify), 'CO' (Copy), and 'PU' (Purge). These commands can be entered directly against any existing user group. To start the list of user groups from a particular point, enter a value in the 'From' field.

The following PF-keys are provided for the 'Amend User Groups' window:

Key	Function	Description
PF1	Help	Provides help information about the 'Amend User Groups' window
PF3	End	Returns to the "Report Parameter list" menu
PF7	Up	Pages up through the list of user groups
PF8	Down	Pages down through the list of user groups
PF9	New-G	Define a new user group
PF12	Exit	Exits PROFILER

Note: A User-ID should never be used as the name of a user group.

The following PF-keys are provided for the Define New User Group screen:

Key	Function	Description
PF1	Help	Provides help information about the Define New User Group screen
PF3	End	Returns to the 'Amend User Groups' Window
PF1 2	Exit	Exits PROFILER

Pressing Enter on the 'Define New User Group' screen defines the user group that includes the specified User-IDs and displays the message 'PRO0158: New Group ... Defined successfully.' on return to the 'Amend User Groups' window, confirming that PROFILER successfully defined the user group.

VI.3.2.2 Display User Group

The Display User Group function allows a user to display a user group.

Enter command 'DI' on the 'Amend User Groups' window next to the user group that is to be displayed.

```

*** PROFILER 4.3.2 for Natural ***

2011-11-20 16:45   Active Profile Session: NO   Amend User Groups...
C A P             Active Trace Session: NO   Com User Groups
o c or
m Report Parameter Last Mod   Owner   User [   ___ ALL-TESTERS
___ ALL SESSIONS   2011-06-03 USER24  *      di PAY-TESTERS
___ PAY-PARAM      2011-06-03 USER27  PAY-TE
___ TEST REP PARM  2011-06-02 USER24  *

                                     From _____

From _____   Owner _____   Amend Groups? y Amend Groups? _

Enter-PF1---PF2---PF3---PF4-- PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help           End           Up     Down  New-G           Exit
    
```

Press Enter to display the 'Display User Group' screen.

```

Display User Group

User Group Name PAY-TESTERS

User-IDs to Include in User Group:
USER24
USER27
USER01
USER23
USER21

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help           End           Exit
    
```

Field	Description
User Group Name	The name the user assigned to the user group.
User-IDS to Include in User Group	The User-ID or User-IDs to be included in a user group.

The following PF-keys are provided for the 'Display User Group' screen:

Key	Function	Description
PF1	Help	Provides help information about the 'Display User Group' screen
PF3	End	Returns to the 'Amend User Groups' window
PF12	Exit	Exits PROFILER

VI.3.2.4 Delete User Group

The Delete User Group function allows a user to delete a user group.

Enter command 'PU' on the 'Amend User Groups' window next to the user group that is to be deleted.

```

*** PROFILER 4.3.2 for Natural ***

2011-11-20 16:45   Active Profile Session: NO   Amend User Groups...
C A P             Active Trace Session: NO   Com User Groups
o c or
m Report Parameter Last Mod  Owner  User [   ___ ALL-TESTERS
___ ALL SESSIONS   2011-06-03 USER24  *       pu PAY-TESTERS
___ PAY-PARAM      2011-06-03 USER27  PAY-T
___ TEST REP PARM  2011-06-02 USER24  *

                                     From _____

From _____      Owner _____  Amend Groups? y Amend Groups? _

Enter-PF1---PF2---PF3---PF4-- PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help           End           Up     Down  New-G           Exit

```

Press Enter to display the 'Purge User Group' screen.

```

Purge User Group

User Group Name PAY-TESTERS

User-IDs to Include in User Group:
USER24
USER27
USER01
USER23
USER21

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help           End           Exit

```

Field	Description
User Group Name	The name the user assigned to the user group.
User-IDS to Include in User Group	The User-ID or User-IDs to be included in a user group.

The following PF-keys are provided for the 'Purge User Group' screen:

Key	Function	Description
PF1	Help	Provides help information about the 'Purge User Group' screen
PF3	End	Returns to the 'Amend User Groups' window
PF1	Exit	Exits PROFILER

Press Enter, and the message 'PRO0160: Group ... Purged successfully.' will be displayed on return to the 'Amend User Groups' window, confirming that PROFILER successfully deleted the user group.

VI.3.3 Session Group Maintenance

The Session Group Maintenance facility allows a user to specify a set of sessions to be included in a session group. The session group may be included in a report parameter or specified during Enhanced Reporting. Session groups are stored in the PROFILER Reporting file by a unique session group name.

On the "report parameter list" menu there are two 'Amend Groups?' field, one below the 'User [Group]' heading and one below the 'Session [Group]' heading.

Enter 'Y' against the 'Amend Groups?' field below the 'Session [Group]' heading to invoke 'Session Group' maintenance.

```

*** PROFILER 4.3.2 for Natural ***

2011-11-20 16:45   Active Profile Session: NO           USER24
C A P             Active Trace Session: NO           PAYTEST
o c o r
m Report Parameter Last Mod   Owner   User [Group]   Session [Group]
--- ALL SESSIONS   2011-06-03 USER24   *              *
--- PAY-PARAM     2011-06-03 USER27   PAY-TESTERS   *
--- TEST REP PARM 2011-06-02 USER24   *              PAYROLL TESTING

From _____ Owner _____ Amend Groups? _ Amend Groups? y

Enter-PF1---PF2---PF3---PF4-- PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
Help About End Admin Profl Backg Up Down New-P Left Right Exit

```

Press Enter to display the 'Amend Session Groups' window.

```

*** PROFILER 4.3.2 for Natural ***

2011-11-20 16:45   Active Profile Session: NO           Amend Session Groups...
C A P             Active Trace Session: NO           Com Session Groups
o c o r
m Report Parameter Last Mod   Owner   User [   ___ ALL SESSIONS
--- ALL SESSIONS   2011-06-03 USER24   *              ___ PAYROLL TESTING
--- PAY-PARAM     2011-06-03 USER27   PAY-
--- TEST REP PARM 2011-06-02 USER24   *

From _____

From _____ Owner _____ Amend Groups? _ Amend Groups? y

Enter-PF1---PF2---PF3---PF4-- PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
Help End Up Down New-G Exit

```

To choose a command to process against a user group, enter '?' in any 'Com' field. Available commands are: 'DI' (Display), 'MO' (Modify), 'CO' (Copy), and 'PU' (Purge). These commands can be entered directly against any existing user group. To start the list of user groups from a particular point, enter a value in the 'From' field.

The following PF-keys are provided for the 'Amend Session Groups' window:

Key	Function	Description
PF1	Help	Provides help information about the 'Amend Session Groups' window
PF3	End	Returns to the "Report Parameter list" menu
PF7	Up	Pages up through the list of session groups
PF8	Down	Pages down through the list of session groups
PF9	New-G	Define a new session group
PF12	Exit	Exits PROFILER

The following PF-keys are provided for the 'Define New Session Group' screen.

Key	Function	Description
PF1	Help	Provides help information about the 'Define New Session Group' screen
PF3	End	Returns to the 'Amend Session Groups' window
PF12	Exit	Exits PROFILER

Pressing Enter on the 'Define New Session Group' screen defines the session group that includes the specified sessions and displays the message 'PRO0158: New Group ... Defined successfully.' on return to the 'Amend Session Groups' window, confirming that PROFILER successfully defined the session group.

VI.3.3.2 Display Session Group

The Display Session Group function allows a user to display a session group.

Enter command 'DI' on the 'Amend Session Groups' window next to the user group that is to be displayed.

```

*** PROFILER 4.3.2 for Natural ***

2011-11-20 16:45   Active Profile Session: NO   Amend Session Groups...
C A P             Active Trace Session: NO   ■ Com Session Groups
o c or
m Report Parameter Last Mod  Owner   User [   ___ ALL SESSIONS
___ ALL SESSIONS   2011-06-03  USER24  *       di PAYROLL TESTING
___ PAY-PARAM      2011-06-03  USER27  PAY-TE
___ TEST REP PARM  2011-06-02  USER24  *

                                From _____

From _____      Owner _____  Amend Groups? _ Amend Groups? y

Enter-PF1---PF2---PF3---PF4-- PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help           End                Up      Down  New-G           Exit
    
```

Press Enter to display the 'Display Session Group' screen.

```

                                Display Session Group

Session Group Name PAYROLL TESTING

Sessions to Include in Session Group:
PAYROLL PHASE 1
PAYROLL PHASE 2
PAYROLL PHASE 3
PAYROLL PHASE 4

Enter-PF1---PF2---PF3---PF4-- PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help           End                Exit
    
```

Field	Description
Session Group Name	The name the user assigned to the session group.
Sessions to Include in Session Group	The session or sessions to be included in a session group.

The following PF-keys are provided for the Display Session Group screen:

Key	Function	Description
PF1	Help	Provides help information about the 'Display Session Group' screen
PF3	End	Returns to the 'Amend Session Groups' window
PF12	Exit	Exits PROFILER

VI.3.3.4 Delete Session Group

The Delete Session Group function allows a user to delete a session group.

Enter command 'PU' on the 'Amend Session Groups' window next to the user group that is to be deleted.

```

*** PROFILER 4.3.2 for Natural ***

2011-11-20 16:45   Active Profile Session: NO   Amend Session Groups...
C A P             Active Trace Session: NO   Com Session Groups
o c or
m Report Parameter Last Mod  Owner   User [   _ ALL SESSIONS
_ ALL SESSIONS    2011-06-03  USER24  *      pu PAYROLL TESTING
_ PAY-PARAM       2011-06-03  USER27  PAY-TE
_ TEST REP PARM   2011-06-02  USER24  *

                                From _____

From _____      Owner _____  Amend Groups? _ Amend Groups? y

Enter-PF1---PF2---PF3---PF4-- PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help           End                Up      Down  New-G      Exit
    
```

Press Enter to display the 'Purge Session Group' screen.

```

Purge Session Group

Session Group Name PAYROLL TESTING

Sessions to Include in Session Group:
PAYROLL PHASE 1
PAYROLL PHASE 2
PAYROLL PHASE 3
PAYROLL PHASE 4

Enter-PF1---PF2---PF3---PF4-- PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help           End                Exit
    
```

Field	Description
Session Group Name	The name the user assigned to the session group.
Sessions to Include in Session Group	The session or sessions to be included in a session group.

The following PF-keys are provided for the 'Purge Session Group' screen:

Key	Function	Description
PF1	Help	Provides help information about the 'Purge Session Group' screen
PF3	End	Returns to the 'Amend Session Groups' window
PF12	Exit	Exits PROFILER

Press Enter, and the message 'PRO0160: Group ... Purged successfully.' will be displayed on return to the 'Amend Session Groups' window, confirming that PROFILER successfully deleted the session group.

The following PF-keys are provided for the 'Copy Session Group' screen:

Key	Function	Description
PF1	Help	Provides help information about the 'Copy Session Group' screen
PF3	End	Returns to the 'Amend Session Groups' window
PF12	Exit	Exits PROFILER

When Enter is pressed, the message 'PRO0158: New Group ... Defined successfully.' is displayed on return to the 'Amend Session Groups' window, confirming that PROFILER successfully defined the session group.

VI.4 Introduction to Enhanced Reporting

The PROFILER Enhanced Reports facility allows users to display the statistics stored on the PROFILER Reporting file. Reports that display these statistics may be obtained on-line or in batch. There are three types of PROFILER Enhanced Reports: Enhanced Statistics Summary, Enhanced Source Code Listing, and Enhanced Application QA (Quality Assurance).

Enhanced Statistics Summary Reports

For a nominated "report parameter set", these reports provide information about statement executions, CPU time, Database Elapsed Time, and type of NATURAL statement executed for NATURAL objects stored on the PROFILER Reporting file. There are eleven different summary report formats available directly from the "report parameter list" menu. While viewing a particular report format, another report format can be selected using a pop-up menu or by use of PF keys:

- Statement Execution Count ('ES' action)
- CPU Time ('EC' action)
- Database Elapsed Time ('ED' action)
- Number of Statements, CPU Time, Database ('EN' action)
- Percentage of Executions, CPU and Database ('EP' action)
- Percent Graph of Executions, CPU and Database ('EG' action)
- Database/Work File Loops ('EF' action)
- Internal Subroutines/Non-Procedural Blocks ('EB' action)
- FOR/REPEAT Loops ('ER' action)
- IF/DECIDE Conditions ('EI' action)
- Percent Executed by Statement Type ('ET' action)

The scope of any of these report formats can be limited with several threshold settings. Any objects whose statistics are greater than the relevant threshold setting will be included in the summary report. Thresholds may be entered for the following object statistics:

- Run Count
- Statement Executions
- CPU Time
- Database Elapsed Time
- Percent of Statement Executions
- Percent of CPU Time
- Percent of Database Elapsed Time

The enhanced summary reports may also be sorted in one of four ways (without leaving the report):

- Object (within Library) order (ascending)
- Statement Execution Count order (descending)
- CPU Time order (descending)
- Database Elapsed Time order (descending)

Enhanced Source Code Listing Report

Displays the source code, including copycode, for an individual NATURAL object and gives statistics for run count. Also displays CPU and database usage for the object, as well as for each individual executed, un-executed, and non-executable statement within the object. The source code listing can be limited to any combination of executed, un-executed, and non-executable statements required. The source code can also be scanned for particular text.

Enhanced Application QA Report

Provides the percentage of a NATURAL library or for a range of objects in a library that has been tested. It also shows a listing of which objects have not been tested at all. The source code of these untested objects can be viewed directly from this listing. Direct access is provided to the Enhanced 'Statement Execution Count' Summary report which shows the percentage of each object tested.

Notes about Enhanced Reporting

When reporting on PROFILER statistics for numerous objects, it is recommended that reports be run in batch. The eleven Enhanced Statistics Summary Report formats show up to 500 profiled objects on-line.

The Enhanced Application QA Report shows up to 4,000 untested objects for one library on-line. The Enhanced Source Code Listing Report allows 2500 object statements and 2500 copycode statements to be displayed on-line.

In batch, an unlimited number of lines and pages may be displayed.

Database Elapsed Time represents elapsed time for calls to databases, such as ADABAS, DB2, and VSAM.

All statistics on the PROFILER reports are rounded.

For Enhanced Reporting functions that are run on-line, processing time may be reduced by entering a specific User-ID, session, and library/object range, rather than selecting all users, all sessions, and all libraries/objects.

Object code must exist on the NATURAL FUSER where PROFILER is installed. To create the object code, NATURAL objects must be STOWed before profiling.

It is recommended that the object code be cataloged in the library from which it is to be profiled. For more information, refer to the **PROFILER Statistics for Objects Migrated with SYSMAIN** section.

Source code is only necessary for the Enhanced Source Code Listing Reports.

How to Invoke Enhanced Reporting

Enter '?' in a 'Com' field (or press PF1 while the cursor is in a 'Com' field) on the Main "report parameter list" menu to access a help selection window of available "report parameter actions".

```

*** PROFILER 4.3.2 for Natural ***

2011-11-20 16:45 Acti ■ USER24
C A P Acti ■ Mark Command ... PAYTEST
o c or _ DI Display Page 1_ of 1
m Report Parameter Las _ MO Modify roup] Session [Group]
?_ ALL SESSIONS 2011 _ CO Copy *
_ PAY-PARAM 2011 _ PU Purge TERS *
_ TEST REP PARM 2011 _ E= Stats Report PAYROLL TESTING
_ E? Stats Help
_ EQ Applic QA
_ EU Customized

From _____ Owner _____ Amend Groups? _ Amend Groups? _

Enter-PF1---PF2---PF3---PF4-- PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
Help About End Admin Profl Backg Up Down New-P Left Right Exit

```

The action 'E=' will invoke the most recently used Enhanced 'Summary Report' format. The 'EQ' action will invoke the 'Application QA' report.

VI.4.1 Enhanced Statistics Summary Reports

The 'E?' action will display this help selection window of available Enhanced 'Summary Report' formats:

```

*** PROFILER 4.3.2 for Natural ***

2011-11-20 16:22 Active Profile Session: NO USER24
C Active Trace Session: NO PAYTEST
o Page 1_ of 1
m Report Parameter Last Mod Owner User [Group] Session [Group]
e? ALL SESSIONS 2011-06-03
  PAY-PARAM 2011-06-03
  TEST REP PARM 2011-06-02

Mark Report Format..
- S Statement Execution Summary
- C CPU Time Used Summary
- D Database Elapsed Time Summary
- N Num,Times: Stmt, CPU, Dbase
- P Percents: Stmt, CPU, Dbase
- G % Graphs: Stmt, CPU, Dbase

- F Database/Work File Loops
- B Inline Sub/Non-proc'al Blocks
- R FOR and/or REPEAT Loops
- I IF and/or DECIDE Conditions
- T Percent Executed by Stmt Type

From PAYROLL_____

Enter-PF1---PF2---PF3---PF4---PF12---
Help About End Admin Backg Up Down New-S Left Right Exit

```

The two-letter action codes (e.g., 'ES', 'EC', 'ED', etc.) may be used directly in the 'Com' field to invoke the required Enhanced 'Summary Report' format.

```

PRO0096: 'S'elect Object to see its Source Code Listing Report.
Enhanced Summary Report...
2011-11-20 16:23 Report Parameter PAY-PARAM USER24 PAYTEST
Report Format S Sort Order OBJ Types PNSMH QA? N Page 1_ of 1
Library Mask PAY* Object Mask * View/Amend Thresholds N
Total Stmt Execs 80

s e T Run Total Exec Exec %Exec
l Library Object p Count Stmt utbl Stmt Stmts Date Objt
- PAYTEST PAYROLL P 1 1 2 1 50.00 2011-06-06
- PAYTEST PAY0100M M 4 8 2 2 100.00 2011-06-03
- PAYTEST PAY0100P P 1 59 26 20 76.92 2011-06-05
- PAYTEST PAY0110M M 1 4 2 2 100.00 2011-06-06
- PAYTEST PAY0125M M 1 4 2 2 100.00 2011-06-01
- PAYTEST PAY0130M M 1 4 2 2 100.00 2011-06-06

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
Help User End Stmt CPU Dbase Up Down Sourc Left Right Exit

```

For example, in the screen above, statistics are reported for objects executed from library PAYTEST. The statistics presented in this report were collected using the criteria defined in report parameter "PAY-PARAM".

The current 'Report Format' and 'Sort Order' settings are shown on the next line. These values may be changed directly or by help selection windows (with PF1 or a '?'). The 'QA' field allows direct access to the 'Application QA' report for the current report parameter. The current and maximum page number of the report is located on the right side of this line.

The '(Object) Types', 'Library Mask', and 'Object Mask' fields are those defined in the report parameter "PAY-PARAM" and can not be altered directly on this screen. The 'View/Amend Thresholds?' field allows selection of various values to limit the objects included in the summary report.

There are eleven Enhanced Statistics Summary Report formats and all have a very similar layout and very similar options. These reports provide information about NATURAL objects stored on the PROFILER Reporting file.

Each of these report formats shows the library, object, object type, and run count (the number of times an object executed). Statistics that round to zero display as blank. For example, a CPU time of 0.00003 milliseconds displays as blank.

Each of these report formats uses the following PF keys:

Key	Function	Description
PF1	Help	Provides cursor-sensitive help information
PF2	User	Toggles between the display of the profiled library and the user who performed the profiling
PF3	End	Returns to the "report parameter list" menu
PF4	Stmts	Changes the summary report format directly to 'S' (Statement Execution)
PF5	CPU	Changes the summary report format directly to 'C' (CPU Time Used)
PF6	Dbase	Changes the summary report format directly to 'D' (Database Elapsed Time Used)
PF7	Up	Scrolls up through the report
PF8	Down	Scrolls down through the report
PF9	Sourc	Allows the location of source code (i.e., libraries) to be changed
PF10	Left	Scrolls left through the other summary report formats
PF11	Right	Scrolls right through the other summary report formats
PF12	Exit	Exits PROFILER

Statement Execution Count Summary Report ('ES' action)

Returns statement execution statistics, percent of statements executed, and date object was cataloged.

CPU Time Summary Report ('EC' action)

Returns statistics on CPU time used, percent of CPU time used, a CPU graph, and date object was cataloged.

Database Elapsed Time Summary Report ('ED' action)

Returns statistics on Database Elapsed Time used, percent of Database Elapsed Time used, a Database graph, and date object was cataloged

Number of Statements, CPU Time, Database Time Report ('EN' action)

Returns statement execution statistics, CPU time used, and Database Elapsed time used.

Percentage of Executions, CPU Time, and Database Time Report ('EP' action)

Returns percent of statements executed, percent of CPU time used, percent of Database Elapsed time used, and date object was cataloged

Percent Graph of Executions, CPU and Database Report ('EG' action)

Returns percent graph of statements executed, percent graph of CPU time used, percent graph of Database Elapsed time used, and date object was cataloged

Database/Work File Loops Report ('EF' action)

Returns execution counts and execution percentages for the Database Loops (i.e., READ, FIND, HISTOGRAM) and Work File Loops in each object.

Internal Subroutines/Non-Procedural Blocks Report ('EB' action)

Returns execution counts and execution percentages for the internal subroutines and non-procedural blocks (i.e., break processing) in each object and date object was cataloged. A Non-Procedural Block is one in which execution depends on an event, not on where the statement is located in an object. The non-procedural NATURAL statements are AT BREAK, AT END OF DATA, AT END OF PAGE, AT START OF DATA, AT TOP OF PAGE, BEFORE BREAK PROCESSING, IF NO RECORDS FOUND, and ON ERROR.

FOR/REPEAT Loops Report ('ER' action)

Returns execution counts and execution percentages for the FOR/REPEAT loops in each object and date object was cataloged.

IF/DECIDE Conditions Report ('EI' action)

Returns execution counts and execution percentages for the IF/DECIDE conditions in each object.

Percent Executed by Statement Type Report ('ET' action)

Returns execution percentages for the Database Loops/Work File Loops, internal subroutines/non-procedural blocks, FOR/REPEAT loops, and IF/DECIDE conditions in each object.

Each report format may be retrieved in one of four sort orders.

- **Object within Library (OBJ)**
Returns statistics by library/object in alphabetical order.
- **Statement Execution Count (STA)**
Returns statistics sorted by total statement executions in descending order.
- **CPU Time (CPU)**
Returns statistics sorted by CPU time in descending order.
- **Database Elapsed Time (DAT)**
Returns statistics sorted by Database Elapsed Time in descending order.

The objects included in these summary reports can be limited using several criteria. Enter "Y" in the 'View/Amend Thresholds?' field to invoke the Thresholds window.

```

PRO0096: 'S'elect Object to see its Source Code Listing Report.
                Enhanced Summary Report...
2011-11-20 16:23 Report Parameter PAY-PARAM                USER24  PAYTEST
Report Format S  Sort Order OBJ Types PNSMH              QA? N   Page 1__ of 1
Library Mask PAY* Object Mask *                          View/Amend Thresholds y
                Total Stmt Execs                          80

S                T                Total  Exec  Exec  %Exec
e                y                Stmt  utbl  Stmt  -utbl
l Library  Object p  Run          Stmt  utbl  Stmt  Stmts  Date Objt
_ PAYTEST  PAYR  p  Count      Execs Stmt  Exec  Exec  Cataloged
_ PAYTEST  PAY0                                View/Amend Thresholds
_ PAYTEST  PAY0  Exclude Objects where..                ..is less than
_ PAYTEST  PAY0                .. Run Count for Object                0 runs
_ PAYTEST  PAY0                .. Stmt Executed for Object                0 statements
_ PAYTEST  PAY0                .. CPU Time used by Object                0.0000 msecs
_ PAYTEST  PAY0                .. D'base Elapsed Time for Object                0.0000 msecs

or Object's Percentage of..                ...is less than
.. Executable Stmt Executed                0.00 %
.. Total Session CPU Time used                0.00 %
.. Total Session D'base Elapsed                0.00 %

Enter-PF1---PF2---
Help User

```

Enter '10' in the 'Stmts Executed for Object' field and press Enter.

```

PRO0103: 5 objects EXCLUDED due to Threshold settings.
                Enhanced Summary Report...
2011-11-20 16:23 Report Parameter PAY-PARAM                USER24  PAYTEST
Report Format S  Sort Order OBJ Types PNSMH              QA? N   Page 1__ of 1
Library Mask PAY* Object Mask *                          View/Amend Thresholds N
                Total Stmt Execs                          80

S                T                Total  Exec  Exec  %Exec
e                y                Stmt  utbl  Stmt  -utbl
l Library  Object p  Run          Stmt  utbl  Stmt  Stmts  Date Objt
_ PAYTEST  PAY0100P P  Count      Execs Stmt  Exec  Exec  Cataloged
_ PAYTEST  PAY0100P P  1          59   26   20   76.92  2011-06-05

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
Help User End  Stmts CPU  Dbase Up  Down  Sourc Left Right Exit

```

Only program PAY0100P in library PAYTEST satisfied the threshold entered. Message 'PRO0103: 5 objects EXCLUDED due to Threshold settings.' is displayed at the top of the screen.

VI.4.3.1 Enhanced Statement Execution Count Summary Report

The Enhanced Statement Execution Count Summary Report lists statistics about statement executions for each object meeting the criteria of the nominated 'report parameter set' and the threshold settings.

This report identifies the extent to which an object has been tested and which objects have excessive statement executions. The report also helps to identify poorly-designed objects.

Use action code 'ES' in the 'Com' field for the desired report parameter the "report parameter list" menu.

```

PRO0096: 'S'elect Object to see its Source Code Listing Report.
          Enhanced Summary Report...
2011-11-20 16:23 Report Parameter PAY-PARAM          USER24  PAYTEST
Report Format S Sort Order OBJ Types P NSMH        QA? N   Page 1__ of 1
Library Mask PAY* Object Mask *                    View/Amend Thresholds _
          Total Stmt Execs                          80

S          T          Run          Total Exec          Exec          %Exec
e          y          Count         Stmt  utbl          utbl          -utbl
l Library  Object  p          Execs Stmt  Stmt          Stmt          Stmts  Date Objt
- NATLIB1  PROG001  P          3         114  45          40          88.9  2011-06-09
- NATLIB1  PROG090  P          1         110  55          12          21.8  2011-06-01
- NATLIB1  PROG012  P          2          99  50          27          54.0  2011-06-07
- NATLIB1  PROG003  P          4          83  33           1           3.0  2011-06-05
- NATLIB1  PROG055  P          1          76  25          15          60.0  2011-06-05
- NATLIB1  PROG110  P          1          54  97          42          43.3  2011-06-01
- NATLIB1  PROG008  P          2          23  15          11          73.3  2011-06-03

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help User End  Stmts CPU  Dbase Up   Down Sourc Left Right Exit
    
```

Field	Description
Total Stmt Execs	The total number of statements executed by objects included in the 'report parameter set'.
Sel	An "S" in the Sel field next to a library/object invokes the Source Code Listing Report, which lists statistics and source code for that individual object. Pressing PF3 from the Source Code Listing Report returns to this report.
Library	The library containing the profiled object.
Object	The profiled object.
Typ	The NATURAL object type of the profiled object. Valid values are as follows: P Program N Subprogram S Subroutine M Map H Helproutine
Run Count	The number of times the object executed.

(continued from the previous page)

Field	Description
Total Stmt Execs	The number of statement executions for the object.
Executbl Stmt	The number of executable statements in the object. NATURAL includes executable statements as part of its object code. PROFILER only collects statistics on these executable statements. Non-executable statements, such as comments and continuation lines, are not included in these statistics.
Executbl Stmt Exec	The number of executable statements executed at least once during object profiling and counted only once.
%Executbl Stmts Exec	This percentage points out the extent to which an object has been tested.
Date Objt Cataloged	The date the object was cataloged.

VI.4.1.2 Enhanced CPU Time Summary Report

The Enhanced CPU Time Summary Report lists statistics for CPU time usage for each object meeting the criteria of the nominated 'report parameter set' and the threshold settings.

This report may be used to determine which objects have excessive CPU time and may be affecting system performance.

Use action code 'EC' in the 'Com' field for the desired report parameter the "report parameter list" menu.

```

PRO0096: 'S'elect Object to see its Source Code Listing Report.
          Enhanced Summary Report...
2011-11-20 16:23 Report Parameter PAY-PARAM          USER24  PAYTEST
Report Format C Sort Order OBJ Types PMSMH          QA? N  Page 1__ of 1
Library Mask PAY* Object Mask *                    View/Amend Thresholds _
          Total CPU Time          142.9120 msec

s          T          Graph of
e          y          Time
l Library Object p          Run          CPU          % CPU % of CPU Date Objt
_ PAYTEST PAYROLL P          1          2.1760          1.52
_ PAYTEST PAY0100M M          4          38.7200          27.09 **
_ PAYTEST PAY0100P P          1          29.0560          20.33 **
_ PAYTEST PAY0110M M          1          26.5600          18.58 *
_ PAYTEST PAY0125M M          1          17.3440          12.14 *
_ PAYTEST PAY0130M M          1          29.0560          20.33 **

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help User End Stmts CPU Dbase Up Down Sourc Left Right Exit
    
```

Field	Description
Total CPU Time	The total number of milliseconds of CPU time used by all objects that satisfy the report parameters.
Sel	An "S" in the Sel field next to a library/object invokes the Source Code Listing Report, which lists statistics and source code for that individual object. Pressing PF3 from the Source Code Listing Report returns to this report.
Library	The library containing the profiled object.
Object	The profiled object.
Typ	The NATURAL object type of the profiled object. Valid values are as follows: P Program N Subprogram S Subroutine M Map H Helproutine
Run Count	The number of times the object executed.
CPU Time (msec)	The number of milliseconds of CPU time used by each object that satisfies the report parameters.

(continued from the previous page)

Field	Description
% of CPU Time Used	The percent of the total CPU time for all objects that satisfy the report parameters used by an object.
Graph of % of CPU Time Used	The visual representation of the percent CPU time used by each object. Each * represents 10 percent or greater of the CPU time.
Date Objt Cataloged	The date the object was cataloged.

VI.4.1.3 Enhanced Database Elapsed Time Summary Report

The Enhanced Database Elapsed Time Summary Report lists statistics for Database Elapsed Time usage for each object meeting the criteria of the nominated 'report parameter set' and the threshold settings.

This report may be used to determine which objects have excessive Database Elapsed Time and may be using inefficient database access methods.

Use action code 'ED' in the 'Com' field for the desired report parameter the "report parameter list" menu.

```

PRO0096: 'S'elect Object to see its Source Code Listing Report.
          Enhanced Summary Report...
2011-11-20 16:23 Report Parameter PAY-PARAM          USER24  PAYTEST
Report Format D Sort Order OBJ Types PNSMH      QA? N   Page 1__ of 1
Library Mask PAY* Object Mask *          View/Amend Thresholds _
          Total Database Elapsed          1005.5840 msec
s          T          Database          Graph of
e          y          Elapsed          %Dbase % of Dbase Date Objt
l Library Object p          Count          Time (msec) Elaps Elaps Time Cataloged
_ PAYTEST PAYROLL P          1
_ PAYTEST PAY0100M M          4
_ PAYTEST PAY0100P P          1          1005.5840 100.00 ***** 2011-06-05
_ PAYTEST PAY0110M M          1          2011-06-06
_ PAYTEST PAY0125M M          1          2011-06-01
_ PAYTEST PAY0130M M          1          2011-06-06

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help User End Stmts CPU Dbase Up Down Sourc Left Right Exit
    
```

Field	Description										
Total Database Elapsed	The total number of milliseconds of Database Elapsed Time used by all objects that satisfy the report parameters.										
Sel	An "S" in the Sel field next to a library/object invokes the Source Code Listing Report, which lists statistics and source code for that individual object. Pressing PF3 from the Source Code Listing Report returns to this report.										
Library	The library containing the profiled object.										
Object	The profiled object.										
Typ	The NATURAL object type of the profiled object. Valid values are as follows: <table border="0"> <tr><td>P</td><td>Program</td></tr> <tr><td>N</td><td>Subprogram</td></tr> <tr><td>S</td><td>Subroutine</td></tr> <tr><td>M</td><td>Map</td></tr> <tr><td>H</td><td>Helproutine</td></tr> </table>	P	Program	N	Subprogram	S	Subroutine	M	Map	H	Helproutine
P	Program										
N	Subprogram										
S	Subroutine										
M	Map										
H	Helproutine										
Run Count	The number of times the object executed.										

(continued from the previous page)

Field	Description
Database Elapsed Time (msec)	The number of milliseconds of Database Elapsed Time used by each object that satisfies the report parameters.
% of Dbase Elaps Time	The percent of the total Database Elapsed Time for all objects that satisfy the report parameters used by an object.
Graph of % of Dbase Elaps Time	The visual representation of the Database Elapsed Time used by each object. Each * represents 10 percent or greater of the Database Elapsed Time.
Date Objt Cataloged	The date the object was cataloged.

VI.4.1.4 Enhanced Number of Statements, CPU Time, Database Time Report

The Enhanced Number of Statements, CPU Time, Database Time Report lists statistics for number of Statement Executions, CPU Time usage, and Database Elapsed Time usage for each object meeting the criteria of the nominated 'report parameter set' and the threshold settings.

This report may be used to determine which objects have excessive Statement Executions, CPU Time, and/or Database Elapsed Time and may be using inefficient design or database access methods.

Use action code 'EN' in the 'Com' field for the desired report parameter the "report parameter list" menu.

```

PRO0096: 'S'elect Object to see its Source Code Listing Report.
          Enhanced Summary Report...
2011-11-20 16:23 Report Parameter PAY-PARAM          USER24  PAYTEST
Report Format N Sort Order OBJ Types PNSMH          QA? N   Page 1__ of 1
Library Mask PAY* Object Mask *                    View/Amend Thresholds _
Total Stmt Execs 80                               Total CPU 142.9120
S          T          Exec          Database
e          y          Total utbl          Elapsed
l Library  Object  p          Run          Stmt Stmt          CPU Time
Count          Execs Exec          Used (msec)          Time (msec)
- PAYTEST  PAYROLL P          1          1  1          2.176
- PAYTEST  PAY0100M M          4          8  2          38.720
- PAYTEST  PAY0100P P          1          59 20          29.056          1005.5840
- PAYTEST  PAY0110M M          1          4  2          26.560
- PAYTEST  PAY0125M M          1          4  2          17.344
- PAYTEST  PAY0130M M          1          4  2          29.056

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help User End  Stmts CPU   Dbase Up   Down Sourc Left Right Exit
    
```

Field	Description
Total Stmt Execs	The total number of statements executed by all objects that satisfy the report parameters.
Total CPU	The total number of milliseconds of CPU time used by all objects that satisfy the report parameters.
Sel	An "S" in the Sel field next to a library/object invokes the Source Code Listing Report, which lists statistics and source code for that individual object. Pressing PF3 from the Source Code Listing Report returns to this report.
Library	The library containing the profiled object.
Object	The profiled object.
Typ	The NATURAL object type of the profiled object. Valid values are as follows: P Program N Subprogram S Subroutine M Map H Helproutine

(continued from the previous page)

Field	Description
Run Count	The number of times the object executed.
Total Stmt Execs	The number of statement executions for each object that satisfies the report parameters.
Executbl Stmt Exec	The number of executable statements executed at least once during object profiling and counted only once.
CPU Time (msec)	The number of milliseconds of CPU time used by each object that satisfies the report parameters.
Database Elapsed Time (msec)	The number of milliseconds of Database Elapsed Time used by each object that satisfies the report parameters.

VI.4.1.5 Enhanced Percentage of Executions, CPU Time, and Database Time Report

The Enhanced Percentage of Executions, CPU Time, and Database Time Report lists percentage statistics for Executable Statement Executed, CPU Time usage, and Database Elapsed Time usage for each object meeting the criteria of the nominated 'report parameter set' and the threshold settings.

This report may be used to determine which objects have insufficient statements tested, and/or excessive CPU Time, and/or excessive Database Elapsed Time and may be using inefficient design or database access methods.

Use action code 'EP' in the 'Com' field for the desired report parameter the "report parameter list" menu.

```

PRO0096: 'S'elect Object to see its Source Code Listing Report.
          Enhanced Summary Report...
2011-11-20 16:23 Report Parameter PAY-PARAM          USER24  PAYTEST
Report Format P Sort Order OBJ Types PNSMH      QA? N  Page 1__ of 1
Library Mask PAY*      Object Mask *      View/Amend Thresholds _
Total Stmt Execs 80          Total CPU 142.9120
S          T
e          y
l Library  Object  p      Run   Stmts  % of CPU % of Dbase Date Objt
Count     Executed Time Used Elaps Time Cataloged
- PAYTEST  PAYROLL  P      1     50.00   1.52          2011-06-06
- PAYTEST  PAY0100M M     4    100.00   27.09          2011-06-03
- PAYTEST  PAY0100P P     1     76.92   20.33    100.00    2011-06-05
- PAYTEST  PAY0110M M     1    100.00   18.58          2011-06-06
- PAYTEST  PAY0125M M     1    100.00   12.14          2011-06-01
- PAYTEST  PAY0130M M     1    100.00   20.33          2011-06-06

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help User End  Stmts CPU   Dbase Up   Down Sourc Left Right Exit
    
```

Field	Description
Total Stmt Execs	The total number of statements executed by all objects that satisfy the report parameters.
Total CPU	The total number of milliseconds of CPU time used by all objects that satisfy the report parameters.
Sel	An "S" in the Sel field next to a library/object invokes the Source Code Listing Report, which lists statistics and source code for that individual object. Pressing PF3 from the Source Code Listing Report returns to this report.
Library	The library containing the profiled object.
Object	The profiled object.
Typ	The NATURAL object type of the profiled object. Valid values are as follows: <ul style="list-style-type: none"> P Program N Subprogram S Subroutine M Map H Helproutine

(continued from the previous page)

Field	Description
Run Count	The number of times the object executed.
%Executable Stmts Exec	This percentage points out the extent to which an object has been tested.
% of CPU Time Used	The percent of the total CPU time for all objects that satisfy the report parameters used by an object.
% of Dbase Elaps Time	The percent of the total Database Elapsed Time for all objects that satisfy the report parameters used by an object.
Date Objt Cataloged	The date the object was cataloged.

VI.4.1.6 Enhanced Percent Graph of Executions, CPU and Database Report

The Enhanced Percent Graph of Executions, CPU Time, and Database Time Report lists percent graph statistics for Executable Statement Executed, CPU Time usage, and Database Elapsed Time usage for each object meeting the criteria of the nominated 'report parameter set' and the threshold settings.

This report may be used to determine which objects have insufficient statements tested, and/or excessive CPU Time, and/or excessive Database Elapsed Time and may be using inefficient design or database access methods.

Use action code 'EG' in the 'Com' field for the desired report parameter the "report parameter list" menu.

```

PRO0096: 'S'elect Object to see its Source Code Listing Report.
          Enhanced Summary Report...
2011-11-20 16:23 Report Parameter PAY-PARAM          USER24  PAYTEST
Report Format G Sort Order OBJ Types PNSMH          QA? N  Page 1__ of 1
Library Mask PAY* Object Mask *                    View/Amend Thresholds _
Total Stmt Execs 80                               Total CPU 142.9120
S          T          % Graph of Graph of Graph of
e          y          Run Executable % of CPU % of Dbase Date Objt
l Library  Object  p  Count Stmts Exec Time Used Elaps Time Cataloged
- PAYTEST  PAYROLL P    1 ***** **                               2011-06-06
- PAYTEST  PAY0100M M   4 ***** **                               2011-06-03
- PAYTEST  PAY0100P P    1 ***** **                               2011-06-05
- PAYTEST  PAY0110M M    1 ***** *                               2011-06-06
- PAYTEST  PAY0125M M    1 ***** *                               2011-06-01
- PAYTEST  PAY0130M M    1 ***** **                               2011-06-06

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help User End  Stmts CPU   Dbase Up   Down Sourc Left Right Exit
    
```

Field	Description
Total Stmt Execs	The total number of statements executed by all objects that satisfy the report parameters.
Total CPU	The total number of milliseconds of CPU time used by all objects that satisfy the report parameters.
Sel	An "S" in the Sel field next to a library/object invokes the Source Code Listing Report, which lists statistics and source code for that individual object. Pressing PF3 from the Source Code Listing Report returns to this report.
Library	The library containing the profiled object.
Object	The profiled object.
Typ	The NATURAL object type of the profiled object. Valid values are as follows: P Program N Subprogram S Subroutine M Map H Helproutine

(continued from the previous page)

Field	Description
Run Count	The number of times the object executed.
%Graph of Executable Statements Executed	The visual representation of the percent of Executable Statement Executions. Each * represents 10 percent or greater of the executable statements executed.
Graph of % of CPU Time Used	The visual representation of the percent CPU time used by each object. Each * represents 10 percent or greater of the CPU time.
Graph of % of Dbase Elaps Time	The visual representation of the Database Elapsed Time used by each object. Each * represents 10 percent or greater of the Database Elapsed Time.
Date Objt Cataloged	The date the object was cataloged.

VI.4.1.7 Enhanced Database/Work File Loops Report

The Enhanced Database/Work File Loops Report lists statistics about the Database/Work File Loops for objects meeting the criteria of the nominated 'report parameter set' and the threshold settings. This report may help identify weaknesses in test data and incorrect search criteria.

For example:

```
FIND AUTOMOBILE-FILE WITH COLOR = 'RED'
    WRITE MAKE MODEL YEAR
END-FIND
```

If there are no red automobiles on the AUTOMOBILE file, the "FIND" entry condition is tested but the body of the "FIND" loop is never entered.

Use action code 'EF in the 'Com' field for the desired report parameter the "report parameter list" menu.

```
PRO0096: 'S'elect Object to see its Source Code Listing Report.
          Enhanced Summary Report...
2011-11-20 16:23 Report Parameter PAY-PARAM          USER24  PAYTEST
Report Format F Sort Order OBJ Types PNSMH          QA? N   Page 1__ of 1
Library Mask PAY* Object Mask *                    View/Amend Thresholds _
Total Stmt Execs 80
S          T
e          Y
l User-ID  Object p   Run  Total  DB/WF  %DB/WF  DB/WF  %DB/WF
- USER24  PROG001 P    3    12    6    50.00    3    25.00
- USER23  PROG002 P    1     9    9   100.00    9   100.00
- USER23  PROG003 P    2    18    9    50.00    6   33.33
- USER22  PROG004 P    4     4    4   100.00    2   50.00
- USER19  PROG005 N    1
- USER17  PROG006 N    1     1
- USER21  PROG007 P    2     8    8   100.00    4   50.00

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help User End  Stmts CPU  Dbase Up   Down Sourc Left Right Exit
```

Field	Description
Total Stmt Execs	The total number of statements executed by all objects that satisfy the report parameters.
Sel	An "S" in the Sel field next to a library/object invokes the Source Code Listing Report, which lists statistics and source code for that individual object. Pressing PF3 from the Source Code Listing Report returns to this report.
Library	The library containing the profiled object.
Object	The profiled object.

(continued from the previous page)

Field	Description
Typ	The NATURAL object type of the profiled object. Valid values are as follows: P Program N Subprogram S Subroutine M Map H Helproutine
Run Count	The number of times the object executed.
Total DB/WF Loops	The number of Database (i.e., READ, FIND, HISTOGRAM) and Work File (i.e., READ WORK FILE) Loops in an object.
DB/WF Loops Exec	The number of Database and Work File Loops in an object that were tested.
%DB/WF Loops Exec	The percent of Database and Work File Loops in an object that were tested.
DB/WF Bodies Exec	The number of Database and Work File Loops in an object in which at least one statement within the body of the loop was executed. The statement that ends the loop is also considered to be part of the body of the loop.
%DB/WF Bodies Exec	The percent of Database and Work File Loops in an object in which at least one statement within the body of the loop was executed.

VI.4.1.8 Enhanced Internal Subroutines/Non-Procedural Blocks Report

The Enhanced Internal Subroutines/Non-Procedural Blocks Report lists statistics about the internal subroutines and the non-procedural blocks for objects meeting the criteria of the nominated 'report parameter set' and the threshold settings.

This report may be used to ensure that all automatic break processing statements have executed. If break processing does not occur in an object that contains internal subroutines or non-procedural blocks, then an object has not been tested thoroughly. Test data which causes all breaks to occur and all internal subroutines to be performed should be used.

Use action code 'EB' in the 'Com' field for the desired report parameter the "report parameter list" menu.

```

PRO0096: 'S'elect Object to see its Source Code Listing Report.
          Enhanced Summary Report...
2011-11-20 16:23 Report Parameter PAY-PARAM          USER24  PAYTEST
Report Format B Sort Order OBJ Types PNSMH          QA? N   Page 1__ of 1
Library Mask PAY* Object Mask * View/Amend Thresholds _
Total Stmt Execs 80 Total CPU 142.9120
S          T          Inline Subroutines/Non-Procedural Blocks:
e          y          Run   Total Blocks %Blks Graph of   Date Objt
l User-ID  Object p   Count Blocks Exec  Exec Blks Exec  Cataloged
- USER24  PROG001 P    3     12    6  50.0 ***** 2011-06-09
- USER22  PROG002 P    1     9     5  55.5 ***** 2011-06-01
- USER23  PROG003 P    2    18    18 100.0 ***** 2011-06-07
- USER19  PROG004 P    4     4     2  50.0 ***** 2011-06-05
- USER21  PROG005 N    1     1     1  100.0 ***** 2011-06-05
- USER17  PROG006 N    1    14     9  64.2 ***** 2011-06-01
- USER18  PROG007 P    2     8     4  50.0 ***** 2011-06-03

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help User End  Stmts CPU   Dbase Up   Down Sourc Left Right Exit
    
```

Field	Description
Total Stmt Execs	The total number of statements executed by all objects that satisfy the report parameters.
Total CPU	The total number of milliseconds of CPU time used by all objects that satisfy the report parameters.
Sel	An "S" in the Sel field next to a library/object invokes the Source Code Listing Report, which lists statistics and source code for that individual object. Pressing PF3 from the Source Code Listing Report returns to this report.
Library	The library containing the profiled object.
Object	The profiled object.
Typ	The NATURAL object type of the profiled object. Valid values are as follows: P Program N Subprogram S Subroutine M Map H Helproutine

(continued from the previous page)

Field	Description
Run Count	The number of times the object executed.
IS/NP Blocks: Total Blocks	The number of Internal Subroutines and Non-Procedural Blocks within an object.
IS/NP Blocks: Blocks Exec	The number of Internal Subroutines and Non-Procedural Blocks that were executed at least once for each object that satisfies the report parameters.
IS/NP Blocks: %Blocks Exec	The percent of Internal Subroutines and Non-Procedural Blocks that were executed at least once for each object that satisfies the report parameters.
IS/NP Blocks: Graph of Blocks Exec	The percent graph of Internal Subroutines and Non-Procedural Blocks that were executed at least once for each object that satisfies the report parameters.
Date Objt Cataloged	The date the object was cataloged.

VI.4.1.9 Enhanced FOR/REPEAT Loops Report

The Enhanced FOR/REPEAT Loops Report lists statistics about the number and execution percentages of the FOR and REPEAT Loops for objects meeting the criteria of the nominated 'report parameter set' and the threshold settings.

Use action code 'ER' in the 'Com' field for the desired report parameter the "report parameter list" menu.

```

PRO0096: 'S'elect Object to see its Source Code Listing Report.
          Enhanced Summary Report...
2011-11-20 16:23 Report Parameter PAY-PARAM          USER24  PAYTEST
Report Format R Sort Order OBJ Types PMSMH          QA? N  Page 1__ of 1
Library Mask PAY* Object Mask * View/Amend Thresholds _
Total Stmt Execs 80          Total CPU 142.9120
S          T          FOR / REPEAT Loops:
e          y          Run Total Loops %Loops Graph of Date Objt
l Library Object p Count Loops Exec Exec Loops Exec Cataloged
- NATLIB1 PROG001 P 2 2 2 100.00 ***** 2011-06-09
- NATLIB1 PROG002 P 4 4 4 100.00 ***** 2011-06-01
- NATLIB1 PROG003 P 1 11 6 54.55 ***** 2011-06-07
- NATLIB1 PROG004 P 1 5 4 80.00 ***** 2011-06-05
- NATLIB1 PROG005 N 6 24 20 83.33 ***** 2011-06-05
- NATLIB1 PROG006 N 2 2011-06-01
- NATLIB1 PROG007 P 1 3 2011-06-03

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help User End Stmts CPU Dbase Up Down Sourc Left Right Exit
    
```

Field	Description
Total Stmt Execs	The total number of statements executed by all objects that satisfy the report parameters.
Total CPU	The total number of milliseconds of CPU time used by all objects that satisfy the report parameters.
Sel	An "S" in the Sel field next to a library/object invokes the Source Code Listing Report, which lists statistics and source code for that individual object. Pressing PF3 from the Source Code Listing Report returns to this report.
Library	The library containing the profiled object.
Object	The profiled object.
Typ	The NATURAL object type of the profiled object. Valid values are as follows: P Program N Subprogram S Subroutine M Map H Helproutine
Run Count	The number of times the object executed.

(continued from the previous page)

Field	Description
FOR/REPEAT Loops: Total Loops	The number of FOR Loops and REPEAT Loops in the object.
FOR/REPEAT Loops: Loops Exec	The number of FOR Loops and REPEAT Loops in an object which executed at least once.
FOR/REPEAT Loops: %Loops Exec	The percent of FOR Loops and REPEAT Loops in an object which executed at least once.
FOR/REPEAT Loops: Graph of Loops Exec	The percent graph of FOR Loops and REPEAT Loops in an object which executed at least once.
Date Objt Cataloged	The date the object was cataloged.

VI.4.3.10 Enhanced IF/DECIDE Conditions Report

The Enhanced IF/DECIDE Conditions Report lists statistics about the IF/DECIDE conditions in objects meeting the criteria of the nominated 'report parameter set' and the threshold settings. This report can be used to determine if all of the conditional statements in an object have been executed.

Use action code 'EI' in the 'Com' field for the desired report parameter the "report parameter list" menu.

```

PRO0096: 'S'elect Object to see its Source Code Listing Report.
          Enhanced Summary Report...
2011-11-20 16:23 Report Parameter PAY-PARAM          USER24  PAYTEST
Report Format I Sort Order OBJ Types PMSMH          QA? N  Page 1__ of 1
Library Mask PAY* Object Mask * View/Amend Thresholds _
Total Stmt Execs 80 Total CPU 142.9120
S          T          IF / DECIDE Conditions/Bodies:
e          y          Run Total Conds %Cond Total Body %Body
l Library Object p    Count Conds Exec Exec Body Exec Exec
- NATLIB1  PROG001 P    3      10      5  50.0    10      1  10.0
- NATLIB1  PROG002 P    1       1      1 100.0     2      2 100.0
- NATLIB1  PROG003 P    2       8      7  87.5     8      2  25.0
- NATLIB1  PROG004 P    4      16      6  37.5    17      2  11.7
- NATLIB1  PROG005 N    1
- NATLIB1  PROG006 N    1       4      4 100.0     4      4 100.0
- NATLIB1  PROG007 P    2       6      6 100.0     8      8 100.0

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help User End  Stmts CPU  Dbase Up  Down Sourc Left Right Exit
    
```

Field	Description
Total Stmt Execs	The total number of statements executed by all objects that satisfy the report parameters.
Total CPU	The total number of milliseconds of CPU time used by all objects that satisfy the report parameters.
Sel	An "S" in the Sel field next to a library/object invokes the Source Code Listing Report, which lists statistics and source code for that individual object. Pressing PF3 from the Source Code Listing Report returns to this report.
Library	The library containing the profiled object.
Object	The profiled object.
Typ	The NATURAL object type of the profiled object. Valid values are as follows: P Program N Subprogram S Subroutine M Map H Helproutine
Run Count	The number of times the object executed.

(continued from the previous page)

Field	Description
IF/DECIDE: Total Conds	The number of IF conditions and DECIDE conditions in an object. An IF condition is counted as one IF/DECIDE condition. An ELSE condition that is part of an IF statement is not counted as a separate IF/DECIDE condition. Each VALUE/WHEN clause, except NONE IGNORE, in a DECIDE statement is counted as one IF/DECIDE condition.
IF/DECIDE: Conds Exec	The number of IF conditions and DECIDE conditions in an object which executed at least once. IF statements that are executed are counted as one Condition Executed. An ELSE statement that is part of an IF statement is not counted as a Condition Executed. Each VALUE/WHEN that is executed, except NONE IGNORE, is counted as one Condition Executed.
IF/DECIDE: %Cond Exec	The percent of IF conditions and DECIDE conditions in an object which executed at least once.
IF/DECIDE: Total Body	The number of IF bodies, ELSE bodies, and VALUE/WHEN bodies (except NONE IGNORE) in an object. A body is the action statement(s) associated with an IF condition, an ELSE condition, or a VALUE/WHEN clause (except NONE IGNORE).
IF/DECIDE: Body Exec	The number of IF Bodies, ELSE Bodies, and VALUE/WHEN Bodies (except NONE IGNORE) which executed at least once.
IF/DECIDE: %BodyS Exec	The percent of IF Bodies and DECIDE Bodies in an object which executed at least once.

VI.4.3.11 Enhanced Percent Executed by Statement Type Report

The Enhanced Percent Executed by Statement Type Report lists execution percentages for the Database Loops / Work File Loops, Internal Subroutines / Non-Procedural Blocks, FOR/REPEAT loops, and IF/DECIDE conditions in objects meeting the criteria of the nominated 'report parameter set' and the threshold settings. This report can be used to determine if all important statement types in an object have been executed.

Use action code 'ET' in the 'Com' field for the desired report parameter the "report parameter list" menu.

```

PRO0096: 'S'elect Object to see its Source Code Listing Report.
          Enhanced Summary Report...
2011-11-20 16:23 Report Parameter PAY-PARAM          USER24  PAYTEST
Report Format T Sort Order OBJ Types PMSMH          QA? N  Page 1__ of 1
Library Mask PAY*      Object Mask *      View/Amend Thresholds _
Total Stmt Execs 80          Total CPU 142.9120
S          T          %DB/Work %IS/NP %F/R %IF/DECIDE
e          y          Run  Loops  Bodys  Block  Loops  Conds  Bodys
l Library  Object  p  Count  Exec  Exec  Exec  Exec  Exec  Exec
- USER24  PROG001 P    3  50.00  25.00  50.00  100.00  50.0  10.0
- USER23  PROG002 P    1  100.00 100.00  55.56  100.00 100.0  100.0
- USER23  PROG003 P    2   50.00  33.33  100.00  54.55  87.5  25.0
- USER22  PROG004 P    4  100.00  50.00  50.00  80.00  37.5  11.7
- USER19  PROG005 N    1
- USER17  PROG006 N    1          64.29          100.0  100.0
- USER21  PROG007 P    2  100.00  50.00  50.00          100.0  100.0

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help User End  Stmts CPU  Dbase Up  Down Sourc Left Right Exit
    
```

Field	Description
Total Stmt Execs	The total number of statements executed by all objects that satisfy the report parameters.
Total CPU	The total number of milliseconds of CPU time used by all objects that satisfy the report parameters.
Sel	An "S" in the Sel field next to a library/object invokes the Source Code Listing Report, which lists statistics and source code for that individual object. Pressing PF3 from the Source Code Listing Report returns to this report.
Library	The library containing the profiled object.
Object	The profiled object.
Typ	The NATURAL object type of the profiled object. Valid values are as follows: P Program N Subprogram S Subroutine M Map H Helproutine

(continued from the previous page)

Field	Description
Run Count	The number of times the object executed.
%DB/WF Loops Exec	The percent of Database and Work File Loops in an object that were tested.
%DB/WF Bodies Exec	The percent of Database and Work File Loops in an object in which at least one statement within the body of the loop was executed.
%IS/NP Blocks: %Blocks Exec	The percent of Internal Subroutines and Non-Procedural Blocks that were executed at least once for each object that satisfies the report parameters.
FOR/REPEAT Loops: %Loops Exec	The percent of FOR Loops and REPEAT Loops in an object which executed at least once.
IF/DECIDE: %Cond Exec	The percent of IF conditions and DECIDE conditions in an object which executed at least once.
IF/DECIDE: %Body Exec	The percent of IF Bodies and DECIDE Bodies in an object which executed at least once.

VI.4.2 Enhanced Source Code Listing Report

The Enhanced Source Code Listing Report gives statistics for an individual NATURAL object, displays the source code for an individual NATURAL object, and lists combined PROFILER statistics for each executable statement in the object. PROFILER statistics meeting the criteria defined for the chosen 'report parameter' are combined for each executable statement of the object. The report also indicates executable statements that did not execute.

NATURAL includes executable statements as part of its object code. PROFILER only collects statistics on these executable statements. Non-executable statements, such as comments and continuation lines, do not have statistics but are included in the report.

To display an Enhanced 'Source Code Listing Report' enter 'S' beside any object on any Enhanced 'Summary Report'.

```

PRO0096: 'S'elect Object to see its Source Code Listing Report.
          Enhanced Summary Report...
2011-11-20 16:23 Report Parameter PAY-PARAM          USER24  PAYTEST
Report Format S Sort Order OBJ Types PNSMH          QA? N  Page 1__ of 1
Library Mask PAY* Object Mask *                    View/Amend Thresholds _
          Total Stmt Execs                          2263

          Exec  $Exec
          utbl  -utbl
          Stmt  Stmt  Stmts  Date Objt
          Exec  Exec  Exec  Cataloged
1 Library  Object p  Count  Execs Stmt  Exec  Exec  Exec  Cataloged
- PAYTEST  PAYROLL P    1      1    2    1  50.00  2011-06-06
- PAYTEST  PAY0100M M    4      8    2    2  100.00  2011-06-03
s PAYTEST  PAY0100P P    1     59  26   20  76.92  2011-06-05
- PAYTEST  PAY0110M M    1      4    2    2  100.00  2011-06-06
- PAYTEST  PAY0125M M    1      4    2    2  100.00  2011-06-01
- PAYTEST  PAY0130M M    1      4    2    2  100.00  2011-06-06

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help User End  Stmts CPU  Dbase Up   Down Sourc Left Right Exit

```

This report will not be displayed if the object selected has been SAVED, CATALOGed, or STOWed since profiling began. Such objects are marked with a (lower case) 's' or 'c' immediately to the right of the object type. Message "PRO0094: ... has been SAVED after Profiling. Reset Stats & Re-Profile." or "PRO0095: ... has been CATALOGed after Profiling. Reset Stats & Re-Profile." will be displayed at the top of the screen. Statistics for the object should be regenerated and then moved to the PROFILER Reporting file again.

This report can not be displayed if the source code or object code for the selected object can not be found. Such objects are marked with an (upper case) 'S' or 'C' immediately to the right of the object type. Message "PRO0091: Object Code does not exist for ... in Library" or "PRO0093: Source NOT Found in LIB ... on Use PF9 to Alter." will be displayed at the top of the screen. Use PF9 (Sourc) to temporarily amend the location of source code for this report.

The maximum number of statements for which statistics may be accumulated for an object is based on the GETMAIN size. GETMAIN values may range from 10K (312 statements) to 50K (2,018 statements). If the statement limit is reached during the execution of the Source Code Listing Report, the message ">> Statistics Limit of ... stmts Reached <<" appears in the report.

PROFILER records individual statistics for database elapsed time usage for the first 44 different database accessing statements in any NATURAL object. If the 44 database access limit is reached during the execution of the Source Code Listing Report, the message ">> Statistics Limit of 44 database stmts Reached <<" appears in the report.

After these statement limits are reached, PROFILER continues to collect statistics at the object level. Total Statement Executions, CPU time, and Database Elapsed Time are still accumulated and are accurate for the object.

Statistics for NATURAL copycode are collected separately from the object source code which contains the INCLUDE statement. Statistics for copycode are collected for up to 15 copycodes per object. Copycode statistics are available as part of the NATURAL source code listing. The Source Code Listing Report expands copycode through the use of PF2. If a user attempts to display more than 15 copycodes in the Source Code Listing Report, the message, "COPYCODE STATISTICS NOT KEPT FOR MORE THAN 15 INCLUDE STATEMENTS" appears on the screen.

Press Enter to display the Enhanced 'Source Code Listing Report' with statistics for each individual line.

2011-11-20 15:25 Report Parameter PAY-PARAM			Cataloged on 2011-06-06	
Execs	Total	CPU	Avg	CPU
				> _____1.....2... PAY0100P Lib PAYTEST
				0010 * Program: PAY0100P
				0020 DEFINE DATA LOCAL USING PAY0100L
				0030 END-DEFINE
1	0.003	0.003		0040 INCLUDE PAY0100C
				0050 *
1	0.005	0.005		0060 SET KEY PF8 = PGM NAMED ' +
				0070 PF20 = PGM NAMED ' +
				0080 *
10	0.326	0.003		0090 READ EMPLOYEES BY NAME STARTING FROM PERSON
<D'base	1005.584	100.558		>
10	0.030	0.003		0100 ADD 1 TO #I
10	0.036	0.004		0110 MOVE PERSONNEL-ID TO #ID(#I)
10	0.050	0.005		0120 COMPRESS FIRST-NAME MIDDLE-NAME INTO #NAM
10	0.040	0.004		0130 MOVE DEPT TO #DEPT(#I)
10	0.036	0.004		0140 MOVE JOB-TITLE TO #TITLE(#I)
PF1 ? PF2 COPY PF3 QUIT PF4 SCAN PF5 SC= PF6 SHOW Page 1 of 5				

Statistics that round to zero display as 0.000.

Field	Description
Report Parameter ...	The chosen 'report parameter set'.
Cataloged on ...	The date the object was last cataloged.
Execs	The number of times each statement executed (based on the PROFILER statistics stored on the PROFILER Reporting file which meet the criteria defined in the chosen 'report parameter set').
Total CPU	The number of milliseconds of CPU Time used by each statement (based on the PROFILER statistics stored on the PROFILER Reporting file which meet the criteria defined in the chosen 'report parameter set').

(continued from the previous page)

Field	Description
Avg CPU	The average number of milliseconds of CPU Time used by each statement (based on the PROFILER statistics stored on the PROFILER Reporting file which meet the criteria defined in the chosen 'report parameter set'). This value is calculated by total CPU / statement count.
> _____	<p>Allows a specific line number or one of the following commands to be entered. Most of these commands can be invoked by use of a PF key.</p> <p>COPY Show Copycode (at cursor)</p> <p>Q{UIT} Quit from Object Listing</p> <p>SC{AN}.. SCAN for text string..</p> <p>SCAN=, SC= Repeat previous SCAN</p> <p>SHOW.. Show statement options..</p> <p>T{OP}, -- Go to Top of Object</p> <p>-P, - Up one Page of Object</p> <p>+P, + Down one Page of Object</p> <p>B{OT}, ++ Go to Bottom of Object</p> <p>L{EFT}, < Scroll to Left of Page</p> <p>R{IGHT}, > Scroll to Right of Page</p> <p>EXIT Exit from PROFILER</p>
....+....1....+....2...	The column positions of the source code line.
... Lib ...	The object name and library name of the profiled object.
Line markings (>, O, S)	<p>If an executable statement has not executed, it is marked with a ">". Non-executable statements, such as comments and continuation lines, appear in the report listing but have no statistics and are not marked with an asterisk.</p> <p>If the first statement in a block of NATURAL Optimized Code has executed, it is marked with "O".</p> <p>If the text scan is used, lines containing the text are marked with an "S".</p>
0010	The NATURAL statement line number.

(continued from the previous page)

Field	Description
<D'base 1005.584 100.558>	<p>The total number of milliseconds and the average number of milliseconds of Database Elapsed Time used by a statement (based on the PROFILER statistics stored on the PROFILER Reporting file which meet the criteria defined in the chosen 'report parameter set').</p> <p>The average number of milliseconds of Database Elapsed Time used by each statement is calculated by Total Database Elapsed Time/Statement Count.</p> <p>If the NATURAL statement accesses the database (i.e., READ, FIND, GET, HISTOGRAM, UPDATE, DELETE, STORE, END TRANSACTION, and BACKOUT TRANSACTION), database statistics are shown in brackets on the line directly below this statement.</p>

The following PF-keys are provided for the Enhanced Source Code Listing Report:

Key	Function	Description
PF1	HELP	Provides help information about Enhanced Source Code Listing Report
PF2	COPY	Displays copycode when the cursor is on a line that contains an INCLUDE statement
PF3	Q{UIT}	Quit from Object Listing (returns to the Enhanced 'Summary Report' menu)
PF4	SC{AN}..	SCAN for text string..
PF5	SCAN=, SC=	Repeat previous SCAN
PF6	SHOW..	Show statement options (report can be limited to any combination of executed, un-executed, and non-executable lines)
PF7	-P, -	Pages backward
PF8	+P, +	Pages forward
PF9	B{OT}, ++	Go to Bottom of Object
PF10	L{EFT}, <	Scrolls left
PF11	R{IGHT}, >	Scrolls right
PF12	EXIT	Exits PROFILER

VI.4.3 Enhanced Application QA Report

The Enhanced Application QA (Quality Assurance) Report returns the percentages of an application library or a range of objects in a library that has been tested. It also shows a listing of which objects have not been tested at all. The source code of these untested objects can be viewed directly from this listing. Direct access is provided to the Enhanced 'Statement Execution Count' Summary report which shows the percentage of each object which has been tested. This report may be used as a quality assurance tool to show that an application has been thoroughly tested before it is migrated to a production environment.

Use action 'EQ' on the main "report parameter list" menu or enter "Y" in the 'QA?' field on any Enhanced 'Summary Report' screen to invoke the Enhanced Application QA Report. The library required for the QA report will then need to be chosen, in this case PAYTEST.

```

PRO0096: 'S'elect Object to see its Source Code Listing Report.
                Enhanced Summary Report...
2011-11-20 16:23 Report Parameter PAY-PARAM          USER24  PAYTEST
Report Format S Sort Order OBJ Types P NSMH        QA? y  Page 1__ of 1
Library Mask PAY* Object Mask *                   View/Amend Thresholds _
      Total Stmt Execs                2263

S          T          Total Exec      Exec      %Exec
e          y          Stmt utbl      utbl      -utbl
l Library  Object  p          Count    Execs Stmt  Exec  Stmt  Date Objt
_ PAYTEST  PAY0100P P          1         59  26   20   76.92 2011-06-06
                ■
                Mark Profiled Library ...
                ... for Session PAYROLL
                x PAYTEST

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  User  End  Stmt CPU  Dbase Up  Down  Sourc Left Right Exit
    
```

Press Enter and the Enhanced Application QA Report will be displayed.

```

PRO0096: 'S'elect Object to see its Source Code Listing Report.
                Summary Report for Session
2011-11-20 15:30 Session PAYROLL          USER24  PAYTEST
QA Report:      52 Objects in PAYTEST_        View Executed
of which       6 ( 11.54% ) were Executed.   Objects? N
Objects NOT Executed starting.. _____ types.. _____ Page 1__ of 2

S          T Exec  S          T Exec  S          T Exec  S          T Exec
e          y utbl e          y utbl e          y utbl e          y utbl
l Object  p Stmt l Object  p Stmt l Object  p Stmt l Object  p Stmt
_ CITYTAXL L 1 _ FICAM   M 9 _ LIFEINSS S 3 _ PAY0120M M 3
_ CITYTAXM M 9 _ FICAP   P 14 _ PAYBATCH P 10 _ PAY0120P P 30
s CITYTAXP P 13 _ FICAS   S 3 _ PAYEMPL L 0 _ PAY0120T M 2
_ CITYTAXS S 3 _ KAH0100M M 2 _ PAYKH   P 30 _ PAY0130P P 28
_ CITYTX2L L 0 _ KAH0100P P 26 _ PAYL    L 0 _ PAY0130T M 2
_ FEDTAXL L 1 _ KAH1080 P 874 _ PAYLOCL L 0 _ PAY0140M M 2
_ FEDTAXM M 9 _ KHBIBM  M 2 _ PAYROLLG C 1 _ PAY0140P P 25
_ FEDTAXP P 15 _ LIFEINSL L 1 _ PAY0100T M 2 _ PAY0140T M 2
_ FEDTAXS S 3 _ LIFEINSM M 6 _ PAY0110P P 27 _ PENSIONL L 0
_ FICAL   L 1 _ LIFEINSP P 13 _ PAY0110T M 2 _ PENSIONM M 6

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help      End                Up      Down  Sourc      Exit
    
```

Field	Description
Report Parameter	The chosen 'report parameter set' chosen from the "report parameter list" menu.
... Objects in {library}	The total number of objects for the library and for the criteria defined in the chosen 'report parameter set'.
of which ... (...%) were Executed	The total number of objects (and the percentage) for the library and for the criteria defined in the chosen 'report parameter set' that executed at least once. This statistic indicates that at least one statement in an object was executed.
View Executed Objects?	Allows direct access to the Enhanced 'Statement Execution Count' summary report for this 'report parameter set' and library. This report shows which objects were at least partially executed.
Objects NOT Executed for Library Mask ... Types ...	The chosen 'report parameter' may limit this report with a Library Mask and by object types.
Page ... of ...	The current and maximum page of the report.
Sel	An "S" in the Sel field next to an object invokes the Enhanced Source Code Listing Report, which lists the source code for that individual object. Pressing PF3 from the Enhanced Source Code Listing Report returns to this report.
Object	The profiled object.
Typ	The NATURAL object type of the unexecuted object. Valid values are as follows: P Program N Subprogram S Subroutine M Map H Helproutine
Executbl Stmt	The number of executable statements in each object. NATURAL includes executable statements as part of its object code. PROFILER only collects statistics on these executable statements. Non-executable statements, such as comments and continuation lines, are not included in these statistics.

The following PF-keys are provided for the Enhanced Application QA Report:

Key	Function	Description
PF1	Help	Provides help information about the Enhanced Application QA Report
PF3	End	Returns to the pop-up window
PF7	Up	Pages backward
PF8	Down	Pages forward
PF9	Sourc	Allows the location of source code (i.e., libraries) to be changed.
PF12	Exit	Exits PROFILER

SECTION VII

BACKGROUND MONITORING

VII.1 Overview of Background Monitoring

Users normally activate PROFILER at the start of a profiling session and deactivate PROFILER at the end of profiling. Although PROFILER activation and deactivation are very simple, some sites prefer a more automated approach.

The Background Monitoring function allows a profiling session to be automatically activated for a user by the user or by a third party, based on library/object masks, NATURAL object types, and date/time at log on to a NATURAL library.

PROFILER requires the user to supply a password to access the Background Monitoring function. The default password is "PROFPROF". The default may be changed after PROFILER is installed. The new password must be within 1 - 8 alphanumeric characters. The Background Monitoring password is kept separately from the PROFILER Administration password.

The Background Monitoring function allows the authorized user to set parameters that control PROFILER activation and deactivation when a user logs on to a NATURAL library. Parameters such as library/object masks, object types, and date/time determine the statistics to be collected. These parameters are written to the PROFILER repository as Background Monitoring records.

An existing profiling session must be specified when a Background Monitoring record is defined. The specified session must have been previously defined using the Define New Session function. This session is activated during Background Monitoring. The same profiling session may be specified for multiple Background Monitoring records.

The activation parameters specified during session definition, except for the ADABAS password, will become the default activation parameters for the session when it is activated by the Background Monitoring function. These default activation parameters may be overridden by specifying new values for those parameters on the Background Monitoring record. These new values will be in effect only when the session is activated by that particular Background Monitoring record and will not modify the values stored in the session definition.

If the ADABAS file where PROFILER statistics are stored is protected by ADABAS security, the ADABAS password must have been entered during session definition using the Define New Session function or modified using the Modify Session Definition function. PROFILER obtains the ADABAS password from the session definition record when a session is activated as a result of Background Monitoring.

There are two types of Background Monitoring records: User and Library.

User Background Monitoring records are based on a single User-ID, and there may be only one user record per User-ID. Activation/deactivation based on a User Background Monitoring record occurs at logon to a NATURAL library if the library and current time are within the range of the activation parameters specified on the User Background Monitoring

record. Statistics collection will be based on activation parameters specified on the User Background Monitoring record.

Library Background Monitoring records are based on a single NATURAL library, and there may be only one library record per NATURAL library. Activation/deactivation based on a Library Background Monitoring record occurs at logon to the library if the current time is within the range of the activation parameters specified on the Library Background Monitoring record. Statistics collection will be based on activation parameters specified on the Library Background Monitoring record. If a session is activated through Background Monitoring, the PROFILER Trace function will include or exclude objects listed in the Session Definition for the session being activated.

At logon to a NATURAL library, a PROFILER front-end program checks to see if a user has an active profiling session. Activation or deactivation based on a Background Monitoring record is prioritized by PROFILER as follows:

- Activation by a user through the use of the Activate Session function takes priority over activation based on a Background Monitoring record. If a user has an active session as a result of the Activate Session function, PROFILER does not access the Background Monitoring function. If a user attempts to activate a session using the Activate Session function and the session is already active due to Background Monitoring, the user receives the message "PRO0031: You have ... active already. You can NOT activate another session. ".
- Activation/deactivation as a result of a User Background Monitoring record takes priority over activation/deactivation based on a Library Background Monitoring record.
- If a user does not have an active session or if a user has an active session as a result of the Background Monitoring function, PROFILER looks for a User Background Monitoring record. If a User Background Monitoring record exists for the user and if the library and current time are within the range of the activation parameters specified on the user record, activation occurs or continues; otherwise, deactivation occurs.
- If a User Background Monitoring record is not found for the user or if the library and current time are not within the range of the activation parameters specified on the user record, PROFILER looks for a Library Background Monitoring record. If a Library Background Monitoring record exists and if the library and current time are within the range of the activation parameters specified on the library record, activation occurs or continues; otherwise, deactivation occurs.
- If none of the situations above are encountered, PROFILER passes control directly to the standard LOGON program.

A user may deactivate the user's own session using the Deactivate Session function if the session was activated with the Activate Session function or with the Background Monitoring function.

When a profiling session is deleted using the Delete Existing Session function, any Background Monitoring records that use the session will also be deleted.

When a Background Monitoring record is deleted using the Delete Background Monitoring Record function, any users who have active profiling sessions as a result of that Background Monitoring record will remain active until they issue the NATURAL LOGON command.

Background Monitoring records are accessed by PROFILER at NATURAL LOGON in on-line and batch modes. Maintenance on a Background Monitoring record may only be done in on-line mode.

Reporting on statistics collected as a result of Background Monitoring occurs through the use of the PROFILER Reporting and Enhanced Reporting facilities based on the session specified on the Background Monitoring record.

VII.2 Background Monitoring "Session List" Menu

Pressing PF6 on the PROFILER Main "session list" menu displays the Background Monitoring pop-up window.

```

*** PROFILER 4.3.2 for Natural ***

2011-11-2016:44   Active Profile Session: PAYROLL           USER24
C A P           Active Trace Session: NO           PAYTEST
o c o r
m t T Session Name      Date      Time   Owner   LastUser Description
-----
* P PAYROLL            2011-06-12 15:33 USER24   USER24   Payroll Test
_ P PAYROLL EXAMPLE    2011-06-12 15:47 USER24   USER24   Payroll Exam
  █

                Password required for
                Background Monitoring
                Profile Sessions:
                Current Password _____

                Optional:
                New Password _____
                Confirm New _____

From PAYROLL_____ Owner _____ Profile/Trace/All P

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help About End Admin Backg Up Down New-S Left Right Exit
    
```

If a user does not enter a password, PROFILER displays the message "PRO0062: Password must be supplied." in the top of the screen. If a user enters an invalid password, the message "PRO0065: Password you have supplied is NOT correct." or the message "PRO0063: Password must be alphanumeric." appears in the top of the screen.

The following PF-keys are provided for the Background Monitoring pop-up window:

Key	Function	Description
PF3	End	Returns to the "session list" menu
PF12	Exit	Exits PROFILER

This pop-up window allows a user to change the PROFILER Background Monitoring password. A user is required to enter the current password (Current Password) and the new password (New Password). The first time the password is changed, "PROFPROF" should be entered for the current password. A user must also enter a NEW PASSWORD.

If "New Password" and "Confirm New" do not match, then the message "PRO0066: New Password and Confirmation do NOT match." will displayed in the top of the screen. The user should type "New Password" and "Confirm New" again and press enter. If "New Password" and "Confirm New" do match, then the message "PRO0067: Password change was Successful." will displayed in the top of the screen.

The Background Monitoring "session list" menu is displayed after the correct password is entered.

```

*** PROFILER 4.3.2 for Natural ***
2011-11-2016:44   Active Profile Session: PAYROLL           USER24
C               Active Trace Session: NO             PAYTEST
o   User or Background Monitoring                   Page 1_ of 1
m   Library Profile Sessions      Date      Time   Owner Start Mon. Stop Mon.  Lib.Mask
Obj.Mask
_   U USER24  PAYROLL              2011-06-12 15:33 USER24  2011-06-14 2011-06-20 PAYTEST  PAY*
_   L PAYTEST  PAYROLL EXAMPLE    2011-06-12 15:47 USER24  2011-06-16 2011-06-20 *      *

```

This is the main menu for 'Background Monitoring' in PROFILER. All 'Background Monitoring' functions can be accessed from this screen via "session actions" or PF keys. After performing any function, the user is returned by default to this menu. Basic information about the user's current PROFILER environment is shown at the head of this menu.

If the current user has a PROFILER session active, the session name is displayed to the right of "Active Profile Session". For example, the screen above displays "PAYROLL" in the session name field, indicating that USER24 has a profile session named "PAYROLL" active. "NO" in this field indicates that no profile session is active for the current user.

If the user has an active Trace session, the name of the session will appear to the right of "Active Trace Session". If no Trace session is active, this field will display "NO".

Sessions matching the 'From' and 'Owner' settings will be displayed on the Background Monitoring "session list" menu.

Listed sessions can be limited to those "owned" (i.e., defined) by a particular user by entering a value in the "Owner" field. Leave the "Owner" field blank to list 'Background Monitoring' sessions "owned" by all users.

The Background Monitoring "session list" can be started from a chosen position by entering a value in the "From" field. Leave the "From" field blank to list all sessions matching the other settings.

The required page of the Background Monitoring "session list" can be entered directly in the "Page" field. The entered value must not be less than one (1) or greater than the last page of the listing.

"Session actions" are two-letter acronyms and can be entered in the 'Com' field beside the session that is to be processed. (Please note that actions 'AU', 'S?', 'QA' and 'CU' actions are only valid for Profile sessions).

Entering a "?" on the 'Com' field displays the field-level help screen below for Background Monitoring "session actions" (i.e., "session commands").

```

*** PROFILER 4.3.2 for Natural ***

2011-11-2016:44   Active Profile Session: PAYROLL           USER24
C                Active Trace Session: NO           PAYTEST
o                User or Background Monitoring       Page 1_ of 1
m                Library Profile Sessions           Date      Time      Owner   Start Mon. Stop Mon.  Lib.Mask Obj.Mask
?_ U USER24    PAYROLL                2011-11-2015:33 USER24   2011-06-14 2011-06-20 PAYTEST  PAY*
__ L PAYTEST   PAYROLL EXAMPLE 2011-11-2015:47 USER24   2011-06-16 2011-06-20 *      *

                ■
                Mark Command ...
                _ DI Display
                _ MO Modify
                _ PU Purge

                From PAYROLL_____ Owner _____

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
                Help About End Admin Prof1 Up Down New-S Left Right Exit

```

Mark the command that is required and press enter. The marked command will be copied into the 'Com' field ready for processing.

The following "session actions" will be described later in this section:

- DI Display Background Monitoring Session Definition
- MO Modify Background Monitoring Session Definition
- PU Purge Background Monitoring Session Definition

The following are the standard PF-keys for the PROFILER Background Monitoring "session list" menu:

Key	Function	Description
PF1	Help	Provides cursor-sensitive help information
PF2	About	Displays installation, license, and environment details
PF3	End	Exits PROFILER and returns to the NATURAL Main menu
PF4	Admin	Gives access to the Administrator options (refer to the PROFILER Administration section for more information)
PF6	Profl	Gives access to normal Profile and Trace Session functions (refer to the Main "Session List" Menu section for more information)
PF7	Up	Scrolls up the "session list"
PF8	Down	Scrolls down the "session list"
PF9	New-S	Allows a new Background Monitoring session to be defined
PF10	Left	Scrolls left on the "session list"
PF11	Right	Scrolls right on the "session list"
PF12	Exit	Exits PROFILER and returns to the NATURAL Main menu

VII.3 Define Background Monitoring Record

The Define Background Monitoring Record function allows a user to add a new background monitoring record based on a User-ID or a library. Background Monitoring records control PROFILER activation and deactivation at logon to a NATURAL library.

Parameters, such as library/object masks, object types, start date/time, and stop date/time, may be specified to restrict statistics collection. An existing profiling session must be specified when defining the Background Monitoring record. The session must have been previously defined using the Define New Session function.

Press PF9 (New-S) on the Background Monitoring "session list" menu to invoke the 'Define New Background Monitoring Session' screen.

```

PRO0013: Use PF5 to access Included/Excluded Objects.
*** PROFILER 4.3.2 for Natural ***

2011-11-2016:44 Active Profile Session: PAYROLL USER24
Act

C Define New Background Monitoring Details
o User or Backgro Monitoring will be for u User-ID user27__
m Library Profile Session payroll_____ Status
Desc

U USER24 PAYROLL
L PAYTEST PAYROLL

User Date Time Version
Defined Last
Backgrnd

Lib.Mask Obj.Mask & Types Source
Defined Last DBID
Backgrnd FUSER
Monitor... ADABAS Password
from _____ to _____
YYYY MM DD HH:II YYYY MM DD HH:II

From PAYROLL_____

Enter-PF1---PF2---PF3---P
Help End Objts Exit

```

For the field 'Monitoring will be for...', enter either 'U' (for User-ID based monitoring) or 'L' (for Library based monitoring). If a user does not enter 'U' or 'L', the message " PRO0068: Must be "L"ibrary or "U"ser-Id." displays at the top of the screen.

The next field (to the the right of the 'U'/'L' selection) requires a valid User-ID or Library to be entered (depending on whether 'U' or 'L' was chosen). If left blank, the message "PRO0069: A value MUST be provided." will be displayed. Wildcards (e.g., '*') may NOT be used in the User-ID or Library.

An existing profile session name must be provided in the field 'Session'. If a user enters a session that has not been defined, the message "PRO0070: No existing Profile Session with this name." displays at the top of the screen. If a Background Monitoring record has already been defined for the chosen User-ID or Library, the message "PRO0077: Session {other-session-name} already has Monitoring defined for {chosen-user/library}" will be displayed.

After supplying the 'U'/'L' selection, the User-ID or Library, and the session name, press enter and the current settings for that session will be displayed. The default activation parameters are those specified for the existing profile session. These parameters may be modified for Background Monitoring.

```

PRO0013: Use PF5 to access Included/Excluded Objects.
*** PROFILER 4.3.2 for Natural ***

2011-11-2016:44 Active Profile Session: PAYROLL USER24
Act
C Define New Background Monitoring Details
o Monitoring will be for U User-ID USER27__
m Library Profile Session PAYROLL_____ Status ACTIVE
Desc Getting Started with PROFILER 4.3.1
(Test 20 new Payroll objects. Give
efficiency results to supervisor. Give
QA results to Quality Assurance team.
Test will be conducted by Mary B.

User Date Time Version
Defined USER24 2011-06-12 15:33 431
Last USER24 2011-06-12 15:35
Backgrnd USER24 2011-06-12 16:45

Lib.Mask Obj.Mask & Types Source
Defined PAYTEST PAY* PNSMH DBID 1
Last PAYTEST PAY* PNSMH FUSER 29
Backgrnd PAYTEST_ PAY*_____ PNSMH > _____
Monitor... ADABAS Password
from _____ to _____
YYYY MM DD HH:II YYYY MM DD HH:II

From PAYROLL_____
Enter-PF1---PF2---PF3---P-
Help End Objts Exit
    
```

Field	Description
Session	The name of the existing profile session upon which this Background Monitoring record is based. This session gets activated during Background Monitoring.
Library Mask	The NATURAL library or libraries to be profiled. Mask options include the following: * Matches any character in remainder of string. ? Matches any single character. . Matches any numeric character. @ Matches any non-numeric character.
Object Mask	The NATURAL object or objects to be profiled.
Object Types	The NATURAL object types to be profiled. Press PF1 to access a help selection window.
Monitor from {date time}	The date (yyyymmdd)/time (hhmm) PROFILER is to start collecting statistics.
Monitor to {date time}	The date (yyyymmdd)/time (hhmm) PROFILER is to stop collecting statistics.

The following PF-keys are provided for the Define New Background Monitoring Details screen:

Key	Function	Description
PF1	Help	Provides help information for the Define Background Monitoring Details screen
PF3	End	Returns to the Background Monitoring "session list" menu
PF5	Objts	Allows Included/Excluded Objects to be viewed
PF12	Exit	Exits PROFILER

VII.4 Display Background Monitoring Record

The Display Background Monitoring Record function allows a user to display information about a previously defined User or Library Background Monitoring record.

Enter action code "DI" next to the required record on the Background Monitoring "session list" menu.

```

*** PROFILER 4.3.2 for Natural ***

2011-11-2016:46   Active Profile Session: PAYROLL           USER24
C                 Active Trace Session: NO              PAYTEST
o   User or Background Monitoring                       Page 1_ of 1
m   Library Profile Sessions      Date      Time   Owner  Start Mon.  Stop Mon.  Lib.Mask
Obj.Mask
___ U USER24  PAYROLL              2011-06-12 15:33 USER24  2011-06-14 2011-06-20 PAYTEST  PAY*
di U USER27  PAYROLL              2011-06-12 15:33 USER24  2011-06-20 2011-06-27 PAYTEST  PAY*
___ L PAYTEST  PAYROLL EXAMPLE    2011-06-12 15:47 USER24  2011-06-16 2011-06-20 *          *

From PAYROLL_____ Owner _____

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
Help About End Admin   Profl Up   Down New-S Left Right Exit
    
```

Press Enter and the Display Background Monitoring Details screen is displayed.

```

PRO0013: Use PF5 to access Included/Excluded Objects.
*** PROFILER 4.3.2 for Natural ***

2011-11-2016:46   Active Profile Session: PAYROLL           USER24
Act
C                 Display Background Monitoring Details
o   User or Backgro Monitoring will be for U User-ID USER27
m   Library Profile Session PAYROLL           Status ACTIVE
Desc Getting Started with PROFILER 4.3.1
      (Test 20 new Payroll objects. Give
di U USER24  PAYROLL efficiency results to supervisor. Give
      QA results to Quality Assurance team.
___ L PAYTEST  PAYROLL Test will be conducted by Mary B.

User      Date      Time   Version
Defined USER24  2011-11-20 15:33  431
Last     USER24  2011-11-20 15:35
Backgrnd USER24  2011-11-20 16:45

Lib.Mask  Obj.Mask & Types  Source
Defined PAYTEST  PAY*             PNSMH DBID 1
Last     PAYTEST  PAY*             PNSMH FUSER 29
Backgrnd PAYTEST  PAY*             PNSMH

Monitor...
From PAYROLL_____ from 2011-06-20 01:00 to 2011-06-27 23:00

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
Help      End      Objts      Exit
    
```

Field	Description
Session	The name of the existing profile session upon which this Background Monitoring record is based. This session gets activated during Background Monitoring.
Library Mask	The NATURAL library or libraries to be profiled. Mask options include the following: * Matches any character in remainder of string ? Matches any single character . Matches any numeric character @ Matches any non-numeric character
Object Mask	The NATURAL object or objects to be profiled.
Object Types	The NATURAL object types to be profiled.
Monitor from {date time}	The date (yyyymmdd)/time (hhmm) PROFILER is to start collecting statistics.
Monitor to {date time}	The date (yyyymmdd)/time (hhmm) PROFILER is to stop collecting statistics.

The following PF-keys are provided for the Display Background Monitoring Details screen.

Key	Function	Description
PF1	Help	Provides help information for the Display Background Monitoring Record
PF3	End	Returns to the Background Monitoring "session list" menu
PF5	Objts	Allows Included/Excluded Objects to be viewed
PF12	Exit	Exits PROFILER

VII.5 Modify Background Monitoring Record

The Modify Background Monitoring Record function allows a user to modify information about a previously defined User or Library Background Monitoring record.

Enter action code "MO" next to the required record on the Background Monitoring "session list" menu .

```

*** PROFILER 4.3.2 for Natural ***
2011-11-2016:47 Active Profile Session: PAYROLL USER24
C Active Trace Session: NO PAYTEST
o User or Background Monitoring Page 1_ of 1
m Library Profile Sessions Date Time Owner Start Mon. Stop Mon. Lib.Mask
Obj.Mask
___ U USER24 PAYROLL 2011-06-12 15:33 USER24 2011-06-14 2011-06-20 PAYTEST PAY*
mo U USER27 PAYROLL 2011-06-12 15:33 USER24 2011-06-20 2011-06-27 PAYTEST PAY*
___ L PAYTEST PAYROLL EXAMPLE 2011-06-12 15:47 USER24 2011-06-16 2011-06-20 * *

From PAYROLL_____ Owner _____

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
Help About End Admin Profl Up Down New-S Left Right Exit
    
```

Press Enter and the Modify Background Monitoring Details screen displays:

```

PRO0013: Use PF5 to access Included/Excluded Objects.
*** PROFILER 4.3.2 for Natural ***

2011-11-2016:47 Active Profile Session: PAYROLL USER24
Act
C  Modify Background Monitoring Details
o User or Backgro Monitoring will be for U User-ID USER27
m Library Profile Session PAYROLL_____ Status ACTIVE
Desc Getting Started with PROFILER 4.3.1
(Test 20 new Payroll objects. Give
efficiency results to supervisor. Give
QA results to Quality Assurance team.
Test will be conducted by Mary B.

User Date Time Version
Defined USER24 2011-11-20 15:33 431
Last USER24 2011-11-20 15:35
Backgrnd USER24 2011-11-20 16:45

Lib.Mask Obj.Mask & Types Source
Defined PAYTEST PAY* PNSMH DBID 1
Last PAYTEST PAY* PNSMH FUSER 29
Backgrnd PAYTEST_ PAY*_____ PNSMH > _____
Monitor... ADABAS Password
from 2011 06 20 01 00 to 2011 06 27 23 00
YYYY MM DD HH:II YYYY MM DD HH:II

From PAYROLL_____

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
Help End Objts Exit
    
```

For the field 'Monitoring will be for...', enter either 'U' (for User-ID based monitoring) or 'L' (for Library based monitoring). If a user does not enter 'U' or 'L', the message " PRO0068: Must be "L"ibrary or "U"ser-Id." displays at the top of the screen.

The next field (to the right of the 'U'/'L' selection) requires a valid User-ID or Library to be entered (depending on whether 'U' or 'L' was chosen). If left blank, the message "PRO0069: A value MUST be provided." will be displayed. Wildcards (e.g., '*') may NOT be used in the User-ID or Library.

An existing profile session name must be provided in the field 'Session'. If a user enters a session that has not been defined, the message "PRO0070: No existing Profile Session with this name." displays at the top of the screen. If a Background Monitoring record has already been defined for the chosen User-ID or Library, the message "PRO0077: Session {other-session-name} already has Monitoring defined for {chosen-user/library}" will be displayed.

After modifying the 'U'/L' selection, the User-ID or Library, or the session name, press enter, and the current settings for that session will be displayed. The other activation parameters may then be modified.

Field	Description
Session	The name of the existing profile session upon which this Background Monitoring record is based. This session gets activated during Background Monitoring.
Library Mask	The NATURAL library or libraries to be profiled. Mask options include the following: * Matches any character in remainder of string ? Matches any single character . Matches any numeric character @ Matches any non-numeric character
Object Mask	The NATURAL object or objects to be profiled.
Object Types	The NATURAL object types to be profiled.
Monitor from {date /time}	The date (yyyymmdd)/time (hhmm) PROFILER is to start collecting statistics.
Monitor to {date /time}	The date (yyyymmdd)/time (hhmm) PROFILER is to stop collecting statistics.

The following PF-keys are provided for the Modify Background Monitoring Details screen:

Key	Function	Description
PF1	Help	Provides help information for the Modify Background Monitoring Details screen
PF3	End	Returns to the Background Monitoring "session list" menu
PF5	Objts	Allows Included/Excluded Objects to be viewed
PF12	Exit	Exits PROFILER

When a user presses Enter, message "PRO0074: Background Monitor for PAYROLL Modified successfully by USER24" is displayed to confirm that PROFILER successfully modified the Background Monitoring Record.

VII.6 Delete Background Monitoring Record

The Delete Background Monitoring Record function allows a user to delete a Background Monitoring record. When a Background Monitoring record is deleted, any users who have active profiling sessions as a result of that Background Monitoring record will remain active until they issue the NATURAL LOGON command.

Enter action code "PU" on the Background Monitoring "session list" menu.

```

*** PROFILER 4.3.2 for Natural ***

2011-11-2016:48  Active Profile Session: PAYROLL          USER24
C                Active Trace Session: NO             PAYTEST
o                User or Background Monitoring        Page 1_ of 1
m                Library Profile Sessions             Date      Time      Owner  Start Mon. Stop Mon.  Lib.Mask
Obj.Mask

_ U USER24  PAYROLL          2011-06-12 15:33 USER24  2011-06-14 2011-06-20 PAYTEST  PAY*
pu U USER27  PAYROLL          2011-06-12 15:33 USER24  2011-06-20 2011-06-27 PAYTEST  PAY*
_ L PAYTEST  PAYROLL EXAMPLE  2011-06-12 15:47 USER24  2011-06-16 2011-06-20 *      *

                From PAYROLL_____ Owner _____

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10---PF11---PF12---
Help About End Admin Profl Up Down New-S Left Right Exit
    
```

Press Enter to display the Delete Background Monitoring Details screen.

```

PRO0013: Use PF5 to access Included/Excluded Objects.
*** PROFILER 4.3.2 for Natural ***

2011-11-2016:48  Active Profile Session: PAYROLL          USER24
C                Act
o                Delete Background Monitoring Details
m                Monitoring will be for U User-ID USER27
                Session PAYROLL          Status ACTIVE
                Desc Getting Started with PROFILER 4.3.1
                (Test 20 new Payroll objects. Give
                efficiency results to supervisor. Give
                QA results to Quality Assurance team.
                Test will be conducted by Mary B.

                User      Date      Time      Version
                Defined USER24  2011-06-12 15:33  431
                Last    USER24  2011-06-12 15:35
                Backgrnd USER24  2011-06-12 16:45

                Lib.Mask  Obj.Mask & Types  Source
                Defined PAYTEST  PAY*      PNSMH  DBID 1
                Last    PAYTEST  PAY*      PNSMH  FUSER 29
                Backgrnd PAYTEST  PAY*      PNSMH

                Monitor...
                From PAYROLL_____ from 2011-06-20 01:00 to 2011-06-27 23:00

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10---PF11---PF12---
Help      End      Objts      Exit
    
```

Pressing PF3 ends the delete function and returns to the Background Monitoring "session list" menu.

Pressing Enter continues the process of deleting the Background Monitoring record.

The following PF-keys are provided for the Delete Background Monitoring Details screen:

Key	Function	Description
PF1	Help	Provides help information for the Delete Background Monitoring Details screen
PF3	End	Returns to the Background Monitoring "session list" menu
PF5	Objts	Allows Included/Excluded Objects to be viewed
PF12	Exit	Exits PROFILER

After pressing Enter, message "PRO0075: Background Monitor for PAYROLL Deleted successfully by USER24" is displayed, confirming that PROFILER successfully deleted the Background Monitoring record.

This page intentionally left blank.

SECTION VIII

CLARIFICATION OF PROFILER STATISTICS DURING REPORTING

This section explains how PROFILER collects and reports on statistics for various NATURAL statements and reporting conditions. This section also describes how PROFILER operates with the NATURAL OPTIMIZER COMPILER and how CPU time is calculated in Batch/TSO, COM-LETE/TPF, and CICS.

VIII.1 NATURAL Statements Coded on the Same Line

If multiple NATURAL statements are coded on the same line, they have the same line number in the NATURAL object code. PROFILER accumulates statistics for each statement, but statistics are totaled by line number and are displayed by line number on the Source Code Listing Report and the Enhanced Source Code Listing Report.

VIII.2 Statement Statistics Limits

PROFILER collects statistics on executable statements only. Non-executable statements, such as comments and continuation lines, are not included in the statistics but are included in the Source Code Listing Report and the Enhanced Source Code Listing Report.

The maximum number of statements for which statistics may be accumulated for an object is based on the PROFILER GETMAIN size. The default GETMAIN size is 28K, which allows for statistics for 1,080 executable statements. The GETMAIN size may be changed with a zap (refer to the **PROFILER Installation** section for more information). GETMAIN values may range from 10K (312 statements) to 50K (2,018 statements). If the statement limit is ever reached a message appears in the Source Code Listing Report and the Enhanced Source Code Listing Report.

PROFILER records statistics for database elapsed time for the first 44 different database accessing statements in any NATURAL object. If the 44 database access limit is reached, the message ">> Statistics Limit of 44 database stmts Reached <<" appears in the Source Code Listing Report and the Enhanced Source Code Listing Report.

After statement limits and/or database access statement limits are reached, PROFILER continues to collect object statistics. Total Statement Executions, CPU time, and Database Elapsed Time are still accumulated and are accurate for the object.

Statistics for NATURAL copycode are collected separately from the object which contains the INCLUDE statement. Statistics for copycode are collected for up to 15 copycodes per object. For objects that contain more than 15 copycodes, results may be unpredictable. Copycode statistics are available as part of the NATURAL source code listing. The Source Code Listing Report expands copycode through the use of a PF-key. The Enhanced Source Code Listing Report displays copycode after the corresponding INCLUDE statement in the object source code.

If NATURAL has been invoked with the parameter MT=0, PROFILER stops recording statistics after one hour of CPU time.

VIII.3 **Statistics for Objects Migrated with SYSMAIN**

For best results, object code should be CATALOGed in the library where it will be profiled. If a NATURAL object is CATALOGed in one library, then moved to another library, statistics relate to the original library when the object is profiled. When N₂O or SYSMAIN is used to migrate an object from one library to another, the object code contains the original library unless the object is reCATALOGed in the new library.

For example, program PAY0001P is CATALOGed in library PAYTEST. Program PAY0001P is migrated to library PAYPROD and is not reCATALOGed. When program PAY0001P in library PAYPROD is profiled, statistics indicate that PAY0001P was executed from library PAYTEST.

VIII.4 **NATURAL STEPLIB Objects**

When executing objects from a NATURAL STEPLIB, PROFILER records the objects with the STEPLIB Library-ID.

VIII.5 **Statistics for PROLIB, SYS, and SYSTEM Objects**

PROFILER prevents the profiling of objects executed from library PROLIB (e.g., PROFILER code). PROFILER also prevents profiling of objects executed from any library beginning with SYS (e.g., Software AG code).

VIII.6 **Statistics for PREDICT "Free" and "Automatic" Verification Rules Included in NATURAL Maps**

If a NATURAL map contains a PREDICT "free" or "automatic" verification rule, an additional executable statement is included in the map. Under the RULEVAR for that rule is the statement INCDIC <rulename>. The PREDICT rule does not become part of the source code but becomes part of the object code at compile time. The PREDICT rule retains its original statement line numbers from the PREDICT verification rule editor in the NATURAL object code. This may cause rule line numbers to be the same as map line numbers in PROFILER's reports.

Statistics for statements with the same line number are accumulated and reported by line number. Statements in a map which are not executable may have execution statistics shown beside them in the report. This indicates that PREDICT verification rules with the same line numbers as un-executable map statements were executed.

VIII.7 **Statistics for Subroutine Statements**

The DEFINE SUBROUTINE statement does not display individual statistics in the Source Code Listing Report and the Enhanced Source Code Listing Report because it is not an executable statement. The PERFORM statement will show statistics if executed.

PRO0101: Statements shown: Executed, Un-executed and Non-Executable.						
2011-11-2016:25 Profile Session TESTPROF			Object Profiled by USER24			
Execs	Total	CPU Avg	CPU >1.....2...	PAYPROG6	Lib PAYTEST
				0110	DEFINE	SUBROUTINE PAYIT
1	0.013	0.013		0120	WRITE	'THIS IS PAYIT'
1	0.004	0.004		0130	END-SUBROUTINE	
1	0.000	0.000		0140	END	
PF1 ?	PF2 COPY	PF3 QUIT	PF4 SCAN	PF5 SC=	PF6 SHOW	Page 1 of 5

VIII.8 Statistics for IF/ELSE/END-IF

Statistics for the NATURAL IF/ELSE/END-IF conditional statement sets may not always report statistics for the ELSE statement. This is because the ELSE may act as a branch at the end of the "IF true condition" or as a marker to start the "IF false condition". The following explanation shows how the IF true/false condition is executed.

IF conditional statements consist of the following parts:

- IF statement
- true condition
- ELSE statement
- false condition
- END-IF statement

When the IF condition is true, statistics on the Source Code Listing Report and the Enhanced Source Code Listing Report appear as follows:

```

PRO0101: Statements shown: Executed, Un-executed and Non-Executable.
2011-11-2016:25 Profile Session TESTPROF          Object Profiled by USER24

  Execs Total CPU Avg CPU > _____ .....1.....2... PAYPROG6 Lib PAYTEST
    1      0.010  0.010  0040 MOVE 'A' TO #X
    1      0.003  0.003  0050 IF #X = 'A'
    1      0.069  0.069  0060 WRITE 'THIS IS A TEST'
    1      0.001  0.001  0070 ELSE
                          > 0080 WRITE 'THIS IS B TEST'
                          0090 END-IF

PF1 ?      PF2 COPY  PF3 QUIT  PF4 SCAN  PF5 SC=   PF6 SHOW   Page 1 of 5
  
```

In the example above, the IF statement and the true statement each display statistics. The ELSE statement signifies the end of the IF true condition and displays statistics. In this case, the ELSE is a branch around the false condition to the END-IF. The run count for the ELSE statement will be the same as the run count for the prior executable (true) statement in the IF condition.

When the IF condition is false, statistics on the Source Code Listing Report and the Enhanced Source Code Listing Report appear as follows:

```

PRO0101: Statements shown: Executed, Un-executed and Non-Executable.
2011-11-2016:25 Profile Session TESTPROF          Object Profiled by USER24

  Execs Total CPU Avg CPU > _____ .....1.....2... PAYPROG6 Lib PAYTEST
    1      0.010  0.010  0040 MOVE 'A' TO #X
    1      0.003  0.003  0050 IF #X = 'A'
                          > 0060 WRITE 'THIS IS A TEST'
                          > 0070 ELSE
    1      0.069  0.069  0080 WRITE 'THIS IS B TEST'
                          0090 END-IF

PF1 ?      PF2 COPY  PF3 QUIT  PF4 SCAN  PF5 SC=   PF6 SHOW   Page 1 of 5
  
```

In the example above, the IF statement shows statistics, while the true statement and ELSE statement do not. The false statement following the ELSE has statistics reported. In this case, the ELSE is a marker to signal the start of the false condition. The NATURAL Dispatcher uses this marker to locate the false portion. Therefore, the ELSE is not an executable statement.

The END-IF statement may be viewed as a marker, and is therefore not an executable statement.

Note that on the NATURAL Statement Type Reports and the Enhanced NATURAL Statement Type Reports for IF/DECIDE CONDITIONS, an IF statement and its related ELSE statement are counted as one IF condition. For IF/DECIDE bodies, the action statement(s) associated with each IF statement is counted as one IF body and the action statement(s) associated with each ELSE statement is also counted as one IF body.

VIII.9 Statistics for Database/Work File Statements

In the 'Database and/or Work File' Report Format (within the 'Summary Report for (Profile) Session' action command), the Bodies Executed count is incremented if there is at least one statement inside of the loop (other than the IF NO RECORDS FOUND clause) that executes. The IF NO RECORDS FOUND clause does not count as a database body but does count as one internal subroutine/non-procedural block.

Examples:

/*PROGRAM1: in this example there is one database/work file loop, but no database/work file bodies.

```
FIND EMPL WITH NAME = #NAME
  IF NO RECORDS FOUND
    WRITE 'NO RECORDS FOUND'
  ESCAPE
  END-NOREC
END-FIND
```

/*PROGRAM2: in this example there is one database/work file loop, and one database/work file body.

```
FIND EMPL WITH NAME = #NAME
  IF NO RECORDS FOUND
    WRITE 'NO RECORDS FOUND'
  ESCAPE
  END-NOREC
  WRITE 'FOUND A RECORD'
END-FIND
```

In the 'Database and/or Work File' Report Format, statistics for PROGRAM1 and PROGRAM2 would appear as follows:

Summary Report for Session									
2011-11-2010:24		Session TESTPROF			USER24 PAYTEST				
Report Format F		Sort	Order	OBJ	Types	QA? N	Page 1__ of 1		
Start Library NATLIB1_		Start Object		View/Amend			Thresholds N		
Total Stmt Execs 80		Total CPU 142.9120							
S		T		Database and/or Work Files:					
e	l	Library	Object	p	Run	Total	Loops	%Loops	Bodies
					Count	Loops	Exec	Exec	%Bodies
									Exec
		_ NATLIB1	PROGRAM1	P	1	1	1	100.00	
		_ NATLIB1	PROGRAM2	P	1	1	1	100.00	1 100.00

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
 Help User End Stmt CPU Dbase Up Down Sourc Left Right Exit

VIII.10 Statistics for DECIDE Statements

In the 'IF/DECIDE Conditions/Bodies' Report Format (within the 'Summary Report for (Profile) Session' action command), for IF/DECIDE CONDITIONS, each VALUE/WHEN clause (except NONE IGNORE) is counted as one DECIDE condition; for IF/DECIDE bodies, the action statement(s) associated with each VALUE/WHEN clause (except NONE IGNORE) is counted as one DECIDE body. The NONE VALUE statement is not counted if its action is IGNORE.

On the Source Code Listing Report and the Enhanced Source Code Listing Report screens, statistics for the first VALUE statement of a DECIDE statement are included in the statistics for the DECIDE STATEMENT.

```

PRO0101: Statements shown: Executed, Un-executed and Non-Executable.
2011-11-2016:25 Profile Session TESTPROF          Object Profiled by USER24

  Execs Total CPU Avg CPU > .....+.....1.....+.....2... PAYPROG8 Lib PAYTEST
                                0010 DEFINE DATA LOCAL
                                0020 1 #X (A1)
                                0030 END-DEFINE
                                0040 *
      1      0.014  0.014  0050 DECIDE ON FIRST VALUE OF #X
                                0060 VALUE 'A'
                                > 0070 FETCH 'PRO0000P'
      1      0.003  0.003  0080 VALUE 'B'
                                > 0090 FETCH 'PRO1000P'
      1      0.001  0.001  0100 VALUE 'C'
                                > 0110 FETCH 'PRO2000P'
      1      0.001  0.001  0120 VALUE 'D'
                                > 0130 FETCH 'PRO3000P'
      1      0.001  0.001  0140 NONE VALUE
      1      0.068  0.068  0150 WRITE 'ENTER A VALID CODE'
                                0160 END-DECIDE
                                0170 *
      1      0.000  0.000  0180 END

PF1 ?      PF2 COPY  PF3 QUIT  PF4 SCAN  PF5 SC=   PF6 SHOW   Page 1 of 5

```

The NONE VALUE statement with an IGNORE action has no statistics. The NONE VALUE statement with an action does have statistics.

Note that there may be differences in the IF/DECIDE bodies statistics between the IF/DECIDE Conditions report and the Enhanced IF/DECIDE Conditions report. This occurs because of a difference in the processing of the VALUE IGNORE clause between the two reports. On the IF/DECIDE Conditions report, the next statement following a VALUE IGNORE clause is counted as an IF/DECIDE body. This extra calculation does not occur on the Enhanced IF/DECIDE Conditions report.

VIII.11 Statistics for DEFINE DATA Statements

On the Source Code Listing Report and the Enhanced Source Code Listing Report, statistics are reported for a DEFINE DATA statement with a GLOBAL clause.

```
PRO0101: Statements shown: Executed, Un-executed and Non-Executable.
2011-11-2016:25 Profile Session TESTPROF          Object Profiled by USER24

  Execs Total CPU Avg CPU > _____ .....1.....2... PAYPROG9 Lib PAYTEST
    1      0.008  0.008  0010 DEFINE DATA GLOBAL
                          0020 USING PAY0000G
                          0030 END-DEFINE

PF1 ?      PF2 COPY  PF3 QUIT  PF4 SCAN  PF5 SC=  PF6 SHOW  Page 1 of 5
```

Statistics are not reported for a DEFINE DATA statement with any other clause, as shown below.

```
PRO0101: Statements shown: Executed, Un-executed and Non-Executable.
2011-11-2016:25 Profile Session TESTPROF          Object Profiled by USER24

  Execs Total CPU Avg CPU > _____ .....1.....2... PAYPROG8 Lib PAYTEST
                          0010 DEFINE DATA LOCAL
                          0020 1 #X (A1)
                          0030 END-DEFINE

PF1 ?      PF2 COPY  PF3 QUIT  PF4 SCAN  PF5 SC=  PF6 SHOW  Page 1 of 5
```

VIII.12 Statistics for FIND Statements

In the 'Database and/or Work File' Report Format (within the 'Summary Report for (Profile) Session' action command), the FIND/END-FIND statement, including the SORTED BY clause, the RETAIN clause, the WHERE clause, and the IF NO RECORDS FOUND clause, is considered a database/work file loop.

If the FIND/END-FIND entry condition is never tested (e.g., the FIND/END-FIND is in a subroutine that is never performed), the 'Total (DB/WF) Loops' count is incremented, but the '(DB/WF) Loops Exec' and the '(DB/WF) Bodies Exec' counts are not incremented.

If the FIND/END-FIND entry condition is tested, but the body of the loop is never entered, the 'Total (DB/WF) Loops' and the '(DB/WF) Loops Exec' counts are incremented.

If the FIND/END-FIND entry condition is tested and the body of the loop is entered, the 'Total (DB/WF) Loops', the '(DB/WF) Loops Exec', and the '(DB/WF) Bodies Exec' counts are all incremented.

FIND FIRST, FIND UNIQUE, and FIND NUMBER statements are not processed by the Database/Work File Loops Report because they do not initiate processing loops.

In the 'Inline Subroutines/Non-Procedural Blocks' Report Format (within the 'Summary Report for (Profile) Session' action command), the IF NO RECORDS FOUND clause of the FIND statement is considered a non-procedural block and is counted in the 'Total (IS/NP) Blocks' field.

If the FIND statement that contains the IF NO RECORDS FOUND clause is executed, the '(IS/NP) Blocks Exec' count is incremented only if no records are found. The IF NO RECORDS FOUND clause is not considered to be executable. However, the END-NOREC statement is executable.

On the Source Code Listing Report, statistics are reported for a FIND statement with an IF NO RECORDS FOUND clause.

PRO0101: Statements shown: Executed, Un-executed and Non-Executable.									
2011-11-2016:25 Profile Session TESTPROF					Object Profiled by USER24				
Execs	Total	CPU	Avg	CPU	>1.....2...	PROG090	Lib	PAYTEST
1	0.075	0.075			0080	INPUT #NAME			
2	0.042	0.042			0090	FIND EMPL. WITH NAME = #NAME			
<D'Base 745.776 372.888>									
					0100	IF NO RECORDS FOUND			
1	0.011	0.011			0110	WRITE 'NO RECORDS FOUND'			
1	0.005	0.005			0120	ESCAPE			
1	0.006	0.006			0130	END-NOREC			
	0.009	0.009			0140	WRITE 'FOUND RECORD'			
1	0.003	0.003			0150	END-FIND			
1	0.000	0.000			0160	END			
PF1 ? PF2 COPY PF3 QUIT PF4 SCAN PF5 SC= PF6 SHOW Page 1 of 5									

VIII.13 Statistics for CALL Statements

When a non-NATURAL module is called from a NATURAL object, the NATURAL CPU time required to complete the CALL is recorded for the CALL statement. This CPU time is shown beside the CALL statement in the Source Code Listing Report and the Enhanced Source Code Listing Report. The CPU time is added to the total CPU time for the object that contains the CALL statement.

There is an Administrator option which allows the user to be asked if SYSRDC is to be turned off whenever a session is deactivated. The user may want SYSRDC to remain on for other tracing or debugging software. By default, this option is turned off. That is, SYSRDC will be turned off when a session is deactivated without asking the user first.

VIII.14 Statistics for AT END OF PAGE Statements

If an object contains an AT END OF PAGE Statement, run count for the object will be incremented when the AT END OF PAGE condition is encountered at the end of the object. The AT END OF PAGE condition is checked once after the END Statement. For example, when an object that contains an AT END OF PAGE Statement is executed one time, the run count will be 2.

VIII.15 Statistics for END Statements

CPU time for END statements can vary depending on whether an END statement results in control being passed back to NATURAL. END statements that cause an object to complete and control to be passed back to NATURAL may have higher CPU than END statements for objects where control returns to another NATURAL object (e.g., sub-programs).

If Background Monitoring is active for a user and the user leaves the library being monitored, the last END statement executed does not get counted in the Statement Count.

VIII.16 Statistics for TERMINATE Statements

When PROFILER encounters a TERMINATE statement in a NATURAL object, the PROFILER session is terminated. PROFILER statistics include the TERMINATE statement.

Because the user exited NATURAL, the user's session is closed and must be restarted when the user reenters NATURAL. It is recommended that users convert TERMINATE statements to comment lines before profiling objects that contain such statements.

VIII.17 Defining User Group Names

A User-ID should never be used as the name of a user group. If a user group is the same as a User-ID, unexpected results may occur during PROFILER Enhanced Reporting.

VIII.18 NATURAL OPTIMIZER COMPILER (NOC)

PROFILER may be used with objects that have been CATALOGed in a NATURAL OPTIMIZER COMPILER (NOC) environment. However, NOC generates direct machine code, which often executes as a group of statements in one dispatching action (e.g., the NATURAL Dispatcher and PROFILER do not get control at the start of every statement). The effect is that a group of source statements is generated into a block of contiguous object code. PROFILER attributes the NOC execution and the CPU time for the group of statements against the first source code statement in the group. This statement is identified by an "O" immediately to the left of the source code line number in the Source Code Listing Report and the Enhanced Source Code Listing Report. Other statements in the NOC group of code will not show accumulated statistics and are marked with a ">" immediately to the left of the source code line number, as though not executed.

PROFILER is an excellent tool for measuring the effectiveness of the NATURAL OPTIMIZER COMPILER. However, for detailed object analysis during application development, NOC should be disabled. NOC may be disabled during a NATURAL session by entering the NATURAL command GLOBALS MCG=OFF. Objects may then be reSTOWed and profiled with NOC disabled. Once the objects have been profiled, enable NOC by issuing the command GLOBALS MCG=ON, and reSTOW the objects.

VIII.19 How PROFILER Calculates CPU Time

To calculate CPU time, PROFILER queries the CPU timer under which NATURAL is running. PROFILER measures CPU time by issuing the TTIMER macro, which assembles to an SVC call. On entry to the PROFILER assembler module, which calculates PROFILER statistics, PROFILER issues the TTIMER SVC, which returns the CPU time remaining before the CPU timer expires.

PROFILER has done this same calculation on its previous exit from the assembler module. PROFILER then subtracts the current time remaining from the previous time remaining. The difference is the CPU time spent executing the NATURAL statement plus the *overhead* of issuing the TTIMER SVC and a few other machine instructions. PROFILER estimates this *overhead* before starting a session and subtracts the overhead during CPU time calculations to obtain statement CPU time.

If NATURAL is invoked with the parameter MT=0, which normally suppresses CPU timing checks in NATURAL, PROFILER detects this and issues a CPU timer to expire after 3600 CPU seconds. In this case, PROFILER stops recording statistics for a session after one hour of CPU time.

In Batch/TSO and COM-LETE/TPF, each user runs in the user's own TCB, which makes CPU calculation very straightforward and accurate.

In the CICS environment, since all users share a single task and a single CPU timer, CPU calculation is more involved. CICS continuously runs a CPU timer. This is the timer used by PROFILER in CICS. In a CICS environment, PROFILER is able to accurately measure CPU time for a NATURAL statement, unless a different user gets dispatched during the execution of that statement. When another user is given control of the CPU, PROFILER detects this and will indicate in its reports that the CPU time for the given statement may be skewed and may not accurately reflect the true resource usage for a particular object. Detected situations that may cause this include:

- The NATURAL screen I/O count for the user has changed
- The NATURAL database call count for the user has changed
- The NATURAL total roll count for the user has changed
- A NATURAL thread switch has occurred
- The terminal ID of the user has changed

In all cases under CICS, PROFILER accurately records the following statistics:

- all object and Statement Execution Counts
- all Database Elapsed Time
- almost all statement CPU time

In non-CICS environments, the CPU time recorded against each statement is accurate.

Although the system's CPU reporting is very good, it is not exact. If the same job is run multiple times, slightly different CPU times may be recorded for each run. For example, one run might take 60.00 milliseconds, another run might take 59.33 milliseconds, and another run might take 61.20 milliseconds.

Much depends on system load, the speed of the system, and the number of interrupts relative to the resolution and frequency of updates to the system CPU counters. Jobs may also incur CPU time in processing interrupts for other address spaces.

When timings vary, a small sample will show greater variability than a large sample. For samples of this kind, the variability typically decreases at a rate that is inversely proportional to the square root of the number of samples. In other words, the variability seems to decrease rapidly at first as the number of samples increases, and then it levels off. The sample variation will seem large when:

- The number of samples is small.
- The time taken to execute a NATURAL statement is small in comparison to the effective CPU timer resolution (i.e., how often the CPU timer is updated). This means that NATURAL OPTIMIZED statements, which take very little time, show greater variations.
- Samples are executed on a faster machine, which spends a smaller portion of its time updating the CPU timer.

VIII.20 Object CPU Time vs. Statement CPU Time

The total CPU time for an object is stored in the PROFILER repository in units of 64 microseconds. Statement CPU time is stored in microseconds. CPU time is stored differently for objects and statements in order to allow large CPU times for long-running objects. Because of the difference in the way that CPU time is stored, the total CPU time for an object may vary from the sum of individual statement CPU times within a given object. CPU times on the PROFILER reports are displayed in milliseconds.

Database Elapsed Time is stored in units of 16 microseconds for objects and for statements. Database Elapsed Time is displayed on PROFILER reports in milliseconds.

VIII.21 Considerations When Using the NATURAL Review Data Collector Interface for Collection of PROFILER Statistics

The NATURAL Review Data Collector Interface (RDC) from Software AG is used to collect PROFILER statistics. PROFILER makes use of the internal trace call-statement tracing option of RDC to call PROFILER during the execution of any NATURAL object. Because of this, it is recommended that a user not issue the NATURAL command "%TRI-" or "SET CONTROL TRI-" while a profiling session is active or no profiling data will be recorded. When activating/deactivating profiling sessions, PROFILER automatically issues the appropriate "%TRI+/%TRI-" command.

There is an Administration option which allows the user to be asked if SYSRDC is to be turned off whenever a session is deactivated. The user may want SYSRDC to remain on for other tracing or debugging software. By default, this option is turned off. That is, SYSRDC will be turned off when a session is deactivated without asking the user first.

VIII.22 Processing of Object Catalog Timestamps on the Enhanced Source Code Listing Report

The Enhanced Source code Listing Report uses data that is stored in the PROFILER Reporting file. The PROFILER Reporting file has statement records that contain statistics about NATURAL statements collected during profiling for a User-ID/library/object/object catalog timestamp session. The PROFILER Reporting file also has object records for each library/object/object catalog timestamp which contain information, such as statement numbers, executable statements and NATURAL statement types, about each profiled object. These statement and object records are related by library/object/object catalog timestamp.

Input parameters to the Enhanced Source code Listing Report consist of library, object, session/session group/all sessions, and User-ID/user group/all users. The Enhanced Source code Listing displays NATURAL source code and profiling statistics related to this source code. Since the only object source code available to the report is that residing on the FUSER file, the Enhanced Source code Listing Report selects the object data on the PROFILER Reporting file with an object catalog timestamp that is closest to the save timestamp of the NATURAL source object on the FUSER file.

One problem that may occur is that the catalog timestamp of the NATURAL object may be greater than the save timestamp of the NATURAL object (e.g., in a CATALOG situation). It is also possible that a user wants to match old object data with newer NATURAL source code.

The Enhanced Source code Listing Report will attempt to match object data to the most recent statement data when a specific user or a specific session is selected as report parameters.

The Enhanced Source code Listing Report will obtain object data on the PROFILER Reporting file in the following order of priority when all users and all sessions are selected as report parameters:

- OBJECT data whose catalog timestamp matches the save timestamp of the NATURAL source object
- OBJECT data whose catalog timestamp is greater than, but closest to, the save timestamp of the NATURAL source object
- OBJECT data whose catalog timestamp is closest to the save timestamp of the NATURAL source object

The Enhanced Source code Listing Report will then obtain statement data based on the catalog timestamp of the selected object data.

If no object data is found to match the NATURAL source object, PROFILER displays the message "OBJECT DATA NOT FOUND ON REPORTING FILE" and does not display the Enhanced Source code Listing Report.

If the catalog timestamp of the object data matches the save timestamp of the NATURAL source object, PROFILER displays the Enhanced Source Code Listing Report.

If the catalog timestamp of the object data is greater than the save timestamp of the NATURAL source object, PROFILER displays the message "SOURCE AND OBJECT HAVE DIFFERENT TIMESTAMPS. STATISTICS MAY NOT BE ACCURATE FOR THIS SOURCE" on the Enhanced Source code Listing Report.

If the save timestamp of the NATURAL source object is greater than the catalog timestamp of the object data, PROFILER displays the message "OBJECT HAS BEEN SAVED SINCE DATA WAS COLLECTED. STATISTICS MAY NOT BE ACCURATE FOR THIS SOURCE" on the Enhanced Source code Listing Report.

Once the object data that is the best match for the NATURAL source object has been selected, PROFILER looks for corresponding statement data on the PROFILER Reporting file. If there is no statement data to match the object data, PROFILER displays the message "NO STATEMENT DATA WAS FOUND TO MATCH CRITERIA" and does not display the Enhanced Source code Listing Report. This could happen if the user profiled an object, moved the statistics to the PROFILER Reporting file, subsequently SAVED and CATALOGED the object, re-profiled the object, and moved the new statistics to the PROFILER Reporting file. To see the old statistics, the user must use one of the other Enhanced Reports that lists statistics by catalog timestamp.

If a user selects an object from one of the other Enhanced Reports, the Enhanced Source code Listing Report will report differently. This is because the user is provided with a list of object catalog timestamps and can select a specific version. The library, object, catalog timestamp, session/session group/all sessions and User-ID/user group/all users are passed to the Enhanced Source code Listing Report. In this way, the user may choose to see mismatched information. Warning messages will be displayed if the catalog timestamp of the object data and the save timestamp of the NATURAL source object do not match.

VIII.23 PROFILER Statistics for NATURAL Object Types

An asterisk in the Object Type field on a PROFILER Enhanced Report indicates that no object record exists on the PROFILER Reporting file for the object whose statistics are being displayed. Statistics could be unreliable. The statistics for this object should be deleted from the PROFILER Reporting file using the Delete Selected Data function. The session should be reactivated, the object should be re-profiled, and the statistics should be tagged and moved to the PROFILER Reporting file.

VIII.24 PROFILER Sessions - Individual Versus Shared

Many applications require more than one person to test the objects involved. PROFILER has the flexibility to allow either an individual user or multiple users to log statistics under a single session when testing an application.

The following example of the development of a new Payroll application illustrates the difference between individual and shared sessions.

The project leader (User 1) must fully test each of the objects which make up the Payroll application. User 1 plans to have two other programmers, User 2 and User 3, help with the testing using PROFILER. Assume that all of the objects, named PAY1000P through PAY3999P, are stored in a library called PAYROLL. After User 1 divides the objects among the three testers, User 1 has two options for collecting PROFILER statistics on the Payroll application:

1. Each programmer may log statistics into individual sessions by User-ID.
2. Each programmer may log statistics into a shared session by User-ID.

Either method is acceptable, depending on the goal of the testing efforts. A discussion of each method follows.

VIII.24.1 Individual Sessions

If User 1 decides to have each programmer log statistics using individual sessions, then User 1, User 2, and User 3 would each define a new session using the Define New Session (PF9) function. In the example below, User 1 defines a session and names it PAYROLL1. Similarly, User 2 and User 3 define sessions named PAYROLL2 and PAYROLL3.

Session Name:	PAYROLL1	PAYROLL2	
	PAYROLL3		
User:	User 1	User 2	User 3
Library Mask:	PAYROLL	PAYROLL	PAYROLL
Object Mask:	PAY1*	PAY2*	PAY3*

When User 1 activates session PAYROLL1, PROFILER collects statistics whenever User 1 executes objects in the range PAY1000P through PAY1999P in library PAYROLL.

The programmers then execute their group of Payroll objects and collect PROFILER statistics in their three individual sessions: PAYROLL1, PAYROLL2, and PAYROLL3. Statistics are not accumulated for objects executed outside the range of the user's object and library masks. When finished, the programmers may deactivate their sessions. A user may then review the statistics in each of the three individual sessions, one section at a time, by using the Session Reporting facility. The Enhanced Reporting facility may be used to combine and report on statistics for multiple users and multiple sessions.

Alternatively, any of the three programmers may use library mask = PAYROLL and object mask = PAY* to collect statistics for the objects they are testing. For example, User 1 may collect statistics on Programs PAY1000P-PAY1999P by executing objects in this range.

VIII.24.2 Shared Session

Another approach to testing the Payroll application is to define a single session and name it PAYTEST. User 1, User 2, and User 3 may each activate the PAYTEST session, specify a range of objects to be profiled, and test their objects in the PAYROLL library. Statistics will be collected in a single session by User-ID for all object executions by each user. A user may review the results separated by User-ID using the Session Reporting facility. A user may combine and review the statistics for multiple users using the Enhanced Reporting facility.

Session Name:	PAYTEST		
User:	User 1	User 2	User 3
Library Mask:	PAYROLL	PAYROLL	PAYROLL
Object Mask:	PAY1*	PAY2*	PAY3*

Each programmer may activate the same session concurrently. When users have finished testing, they may deactivate their own execution of the session without affecting the other users. Each user may reset session statistics based on session, User-ID, library, object, and/or object types.

VIII.25 How To Use Enhanced Reporting

The PROFILER Enhanced Reporting facility allows statistics which have been collected in the PROFILER repository to be tagged and moved to the PROFILER Reporting file where they can be combined in a variety of ways for flexible reporting.

The following is a sample scenario of how the PROFILER Enhanced Reporting Facility might be used by a site to ensure that an application has been thoroughly tested.

The ABC Company has completed a new personnel system which is ready for quality assurance testing. A project leader and three application programmers have been working on this project. The project leader wants to ensure that every line of NATURAL code has been tested. The project leader needs to provide management with a report that proves the testing has been completed.

The three application programmers have used PROFILER to individually test their NATURAL objects as follows:

PROGRAMMER1	PROGRAMMER2	PROGRAMMER3
USER-ID1	USER-ID2	USER-ID3
SESSION1	SESSION2	SESSION3
LIBRARY1	LIBRARY1	LIBRARY1
PROGRAM1	PROGRAM1	PROGRAM6
PROGRAM2	PROGRAM2	PROGRAM7
PROGRAM3	PROGRAM3	PROGRAM8
PROGRAM4	PROGRAM4	PROGRAM9
PROGRAM5	PROGRAM5	PROGRAMA
PROGRAM6	PROGRAMB	PROGRAMG
PROGRAM7	PROGRAMC	PROGRAMH
PROGRAM8	PROGRAMD	PROGRAMI
PROGRAM9	PROGRAME	PROGRAMJ
PROGRAMA	PROGRAMF	PROGRAMK

Statistics from this testing are stored in the PROFILER repository and may be displayed individually using the PROFILER Reporting facility. To display the statistics combined on one report, they must be moved to the PROFILER Reporting file.

SECTION IX

PROFILER ADMINISTRATION

IX.1 Introduction to PROFILER Administration

The PROFILER Administration facility is designed to be used by the Database Administrator (DBA) or PROFILER Administrator. This facility allows a DBA or PROFILER Administrator to reset active users, to remove PROFILER records from the database and ADABAS file on which they reside, and to display the site's PROFILER environment.

PROFILER requires a password to access the PROFILER Administration function. The default password is "PROFPROF". The site may change this default after PROFILER is installed. The new password must be from 1 - 8 alphanumeric characters in length.

Pressing PF4 on the PROFILER Main menu displays this pop-up window.

```
*** PROFILER 4.3.2 for Natural ***
2011-11-20 16:44 Active Profile Session: PAYROLL USER24
C A P Active Trace Session: NO PAYTEST
o c or Page 1_ of 1
m t T Session Name Date Time Owner LastUser Description
--- * P PAYROLL 2011-06-12 15:33 USER24 USER24 Payroll Test
--- P PAYROLL EXAMPLE 2011-06-12 15:47 USER24 USER24 Payroll Exam
      ■
      Password required for
      Administrator Functions:
      Current Password _____
      Optional:
      New Password _____
      Confirm New _____
      Administrator Functions are...
      > Reset Active Users for a Profile Session
      > Purge All Profile Sessions for ALL Users
      > Purge All Trace Sessions for ONE User
      > Purge All Trace Sessions for ALL Users
      > Move Tagged Data for ALL Profiler Sessions
      > Ask User about SYSRDC
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help About End Admin Backg Up Down New-S Left Right Exit
```

If a user does not enter a password, PROFILER displays the message "PRO0062: Password must be supplied." at the top of the screen. If a user enters an invalid password, the message "PRO0065: Password you have supplied is NOT correct." or the message "PRO0063: Password must be alphanumeric." appears at the top of the screen.

The following PF-keys are provided for the PROFILER Administration pop-up window:

Key	Function	Description
PF3	End	Returns to the PROFILER Main menu
PF12	Exit	Exits PROFILER

This pop-up window allows a user to change the PROFILER Administration password. A user is required to enter the current password (Current Password) and the new password (New Password) and confirm the new password (Confirm New). The first time the password is changed, "PROFPROF" should be entered for the current password.

If "New Password" and "Confirm New" do not match, then the message "PRO0066: New Password and Confirmation do NOT match." will be displayed at the top of the screen. The user should type "New Password" and "Confirm New" again and press enter. If "New Password" and "Confirm New" do match, then the message "PRO0067: Password change was Successful." will be displayed at the top of the screen.

After the correct current password is entered or after a successful password change, the four Administration options become available for selection. These four options are described on the following pages.

```

Password required for
Administrator Functions:
Current Password

Optional:
New Password
Confirm New

■
Select Administrator Function...
_ Reset Active Users for PAYROLL_____
_ Purge All Profile Sessions for ALL Users
_ Purge All Trace Sessions for USER24__
_ Purge All Trace Sessions for ALL Users
_ Move Tagged Data for ALL Profile Sessions
_ Ask User about SYSRDC? NO_
    
```

IX.2 Reset Active Users

The Reset Active Users function should be used to remove one or more active user indicators.

PROFILER activates a GETMAIN area for each user who has an active session. If a user terminates a NATURAL session using a NATURAL TERMINATE statement, or if the system crashes, that user's GETMAIN area is no longer available. Statistics are no longer collected, but an indicator that a user is active may remain. When a user reenters PROFILER, this indicator is normally reset.

If for some reason, the user cannot reenter PROFILER, it may be necessary for the DBA or PROFILER Administrator to reset the active user indicator to inactive.

For example, a user has a profiling session active and the system crashes. The user leaves for vacation. The user is no longer collecting PROFILER statistics, but PROFILER indicates this user is still active. Since the user is on vacation and cannot reenter PROFILER, the DBA may reset the active user using the System Administration Reset Active Users function.

Note: An active user should only be reset when circumstances prohibit the user from reentering PROFILER.

The Reset Active Users function provides a list of active users on a session, the date and time of activation, the library and object masks, and the object types in use.

```

Password required for
Administrator Functions:
Current Password

Optional:
New Password
Confirm New

■ Select Administrator Function...
s Reset Active Users for PAYROLL
_ Purge All Profile Sessions for ALL Users
_ Purge All Trace Sessions for USER24__
_ Purge All Trace Sessions for ALL Users
_ Move Tagged Data for ALL Profile Sessions
_ Ask User about SYSRDC? NO_

```

Select the "Reset Active Users" option and enter the required Session Name. The session name will default to your most recently used session.

Valid sessions are sessions that have been defined to PROFILER. If a user does not enter a session, PROFILER displays the message "PRO0028: Session Name MUST be provided." at the top of the screen. When a user enters an undefined session, PROFILER displays the message "PRO0070: No existing Profile Session with this name." at the top of the screen.

The following PF-keys are provided for the pop-up window:

Key	Function	Description
PF3	End	Returns to the PROFILER Main menu
PF12	Exit	Exits PROFILER

If no users are active on a session, the message "PRO0109: No users currently ACTIVE on Profile Session" appears at the top of the screen.

Entering a valid session name in the pop-up window and pressing Enter displays the Reset Active Users screen.

```

Reset Active Users
2011-06-12 16:45 PAYROLL          USER24  PAYTEST

S
e
l
Active User      Active Date      Active Time      Library Mask      Object Mask      Object Types
- USER20        1999-03-19 09:45 PAYTEST          PAY*             PNSMH
- USER21        1999-03-19 07:20 PAYTEST          PAY1000P        P
- USER22        1999-03-19 09:54 PAYTEST          PAY3*           M
- USER23        1999-02-19 09:17 PAYTEST          PAY?000P        P
- USER24        1999-03-19 08:41 PAYTEST          PAY4000N        N
- USER25        1999-03-19 06:36 PAYTEST          PAY*             PNSMH

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Reset End          Up      Down          Exit
    
```

Field	Description
Sel	Any non-blank character in this field indicates that the active user is to be reset.
Active User	The User-ID of the active user for the session.
Active Date	The date the user activated the session.
Active Time	The time the user activated the session.
Library Mask	The NATURAL library mask the user specified at session activation.
Object MASK	The NATURAL object mask the user specified at session activation.
Object Types	The NATURAL object types the user specified at session activation. Valid values are as follows: P Program N Subprogram S Subroutine M Map H Helproutine

The following PF-keys are provided for the Reset Active Users screen:

Key	Function	Description
PF1	Help	Provides help information about Reset Active Users
PF2	Reset	Resets all users marked with any non-blank character
PF3	End	Returns to the PROFILER Main menu
PF7	Up	Pages backward
PF8	Down	Pages forward
PF12	Exit	Exits PROFILER

Pressing Enter or PF8 on the last page of the report returns to the first page of the report.

Pressing PF2 after marking user(s) displays the Reset Active Users confirmation pop-up window. If no users were marked, PROFILER will display the message "PRO0110: No users have been MARKED for "Reset". Mark User(s) and Press PF2."

```

      ■
      ■
                                PAYROLL
      User(s) currently Active for Session
      2 User(s) have been Marked for RESET..
      ..Proceed with RESET? y

      - USER20  2011-03-19 09:45 PAYTEST  PAY*      PNSMH
      x USER21  2011-03-19 07:20 PAYTEST  PAY1000P P
      - USER22  2011-03-19 09:54 PAYTEST  PAY3*     M
      x USER23  2011-02-19 09:17 PAYTEST  PAY?000P P
      - USER24  2011-03-19 08:41 PAYTEST  PAY4000N N
      - USER25  2011-03-19 06:36 PAYTEST  PAY*      PNSMH

      Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10---PF11---PF12---
      Help  Reset End                               Up    Down                               Exit
  
```

Entering "N" cancels the Reset Active Users function and returns to the list of Active Users.

Entering "Y" and pressing Enter continues the Reset Active Users function. The user will then be returned to the Administration options window and the message "PRO0111: 2 Active Users have been "Reset" for Session PAYROLL" will be shown at the top of the screen.

IX.3 Purge All Profile Sessions

The Purge All Profile Sessions function may be run on the PROFILER repository to remove all Profile Sessions. It does not remove PROFILER objects from the NATURAL library PROLIB. All Profile session definitions and corresponding statistics are deleted. This function may be run in batch if desired. The Purge All Profile Sessions function may be used to delete PROFILER data from a database if a site is moving PROFILER records from that database to another.

Note: If a large number of PROFILER records are to be deleted, it is recommended that this function be performed in batch mode. Refer to "Using PROFILER in Batch" for details.

```

Password required for
Administrator Functions:
Current Password

Optional:
New Password
Confirm New

■ Select Administrator Function...
_ Reset Active Users for PAYROLL_____
s Purge All Profile Sessions for ALL Users
_ Purge All Trace Sessions for USER24__
_ Purge All Trace Sessions for ALL Users
_ Move Tagged Data for ALL Profile Sessions
_ Ask User about SYSRDC? NO_
    
```

Select the "Purge All Profile Sessions for ALL Users" option. The following confirmation window will be displayed.

```

2011-06-12 16:47          USER24  PAYTEST
Purge ALL Profile Sessions for ALL Users

53 Profile Sessions will be purged,
along with 'Password', 'Maintenance',
'Detail', 'Tag', 'Background' and
'User' PROFILER records..

..Proceed with PURGE? _
    
```

Entering "N" or pressing PF3 cancels the Purge All Profile Sessions function and returns to the PROFILER Administration options window.

Entering "Y" and pressing Enter continues the Purge All Profile Sessions function.

The following PF-keys are provided for the Purge All Profile Sessions confirmation window:

Key	Function	Description
PF1	Help	Provides help information about Purge All Profile Sessions
PF3	END	Cancels the removal of records and returns to the PROFILER Administration menu
PF12	Exit	Exits PROFILER

After entering "Y", press Enter and message "PRO0112: 53 Sessions Purged. 834 PROFILER records deleted in total." will be displayed upon returning to the Administration options window.

IX.4 Purge All Trace Sessions for One User

```

Password required for
Administrator Functions:
Current Password

Optional:
New Password
Confirm New

■ Select Administrator Function...
_ Reset Active Users for PAYROLL_____
_ Purge All Profile Sessions for ALL Users
s Purge All Trace Sessions for USER24__
_ Purge All Trace Sessions for ALL Users
_ Move Tagged Data for ALL Profile Sessions
_ Ask User about SYSRDC? NO_

```

Select the "Purge All Trace Sessions for One User" option and enter a User-ID. If no User-ID is entered, message "PRO0069: A value MUST be provided." will be displayed at the top of the screen. After providing a User-ID, press Enter and all Trace sessions for that user will be purged. Message "PRO0107: 13 Sessions Purged for User USER24." will be displayed upon returning to the Administration options window.

IX.5 Purge All Trace Sessions for All Users

```

Password required for
Administrator Functions:
Current Password

Optional:
New Password
Confirm New

■ Select Administrator Function...
_ Reset Active Users for PAYROLL_____
_ Purge All Profile Sessions for ALL Users
_ Purge All Trace Sessions for USER24__
s Purge All Trace Sessions for ALL Users
_ Move Tagged Data for ALL Profile Sessions
_ Ask User about SYSRDC? NO_

```

Select the "Purge All Trace Sessions for All Users" option. Press Enter and all Trace sessions for all users will be purged. Message "PRO0108: 57 Sessions Purged for All Users." will be displayed upon returning to the Administration options window.

IX.6 Move Tagged Data for All Profile Sessions

```

Password required for
Administrator Functions:
Current Password

Optional:
New Password
Confirm New

■

Select Administrator Function...
_ Reset Active Users for PAYROLL_____
_ Purge All Profile Sessions for ALL Users
_ Purge All Trace Sessions for USER24__
_ Purge All Trace Sessions for ALL Users
s Move Tagged Data for ALL Profile Sessions
_ Ask User about SYSRDC? NO_

```

Select the "Move Tagged Data for ALL Profile Sessions" option. Press Enter and all tagged data for all Profile sessions will be moved to the Enhanced Reporting repository. Message "PRO0141: Statistics for ... objects have been Moved for ALL Sessions." will be displayed upon returning to the Administration options window.

IX.7 Ask User about SYSRDC

SYSRDC is the 'Review Data Collector' functionality provided by Software AG. This option allows the user to be asked if SYSRDC is to be turned off whenever a session is deactivated. The user may want SYSRDC to remain on for other tracing or debugging software. By default, this option is turned off. That is, SYSRDC will be turned off when a session is deactivated, without asking the user first.

```

      Password required for
      Administrator Functions:
      Current Password

      Optional:
      New Password
      Confirm New
■
Select Administrator Function...
_ Reset Active Users for PAYROLL_____
_ Purge All Profile Sessions for ALL Users
_ Purge All Trace Sessions for USER24__
_ Purge All Trace Sessions for ALL Users
_ Move Tagged Data for ALL Profile Sessions
s Ask User about SYSRDC? NO_
```

Select the "Ask User About SYSRDC" option and enter 'YES' or 'NO'.

If 'YES' is entered, whenever a session is deactivated, the user will be asked if SYSRDC is to be turned off (or left on).

If 'NO' is entered, whenever a session is deactivated, SYSRDC will be turned off without reference to the user.

'NO' is the default setting when PROFILER is installed.

SECTION X

USING PROFILER IN BATCH

X.1 Introduction to PROFILER in Batch

PROFILER includes batch and on-line components, providing users with a high level of flexibility. Statistics collected during batch processing are saved in the same PROFILER repository, by session name and User-ID, as those collected during on-line processing. All statistics may be retrieved using the PROFILER on-line and batch reporting facilities.

Profiling/tracing batch objects with PROFILER is very similar to profiling/tracing on-line applications. The batch job activates a profile or trace session, executes the NATURAL object(s) to be monitored by PROFILER and deactivates the session. Optionally, the batch job may perform other functions, such as resetting session statistics, moving PROFILER statistics to the Enhanced Reporting File, or generating reports.

When a profile or trace session is activated as a batch job, PROFILER collects statistics for the batch NATURAL objects that are executed during the same batch job (i.e., the same NATURAL session). A profile or trace session must be activated before a NATURAL object is executed in order for statistics collection to occur. Statistics are only collected for the objects executed during the batch job in which the profile or trace session is activated.

A profile or trace session that is activated during a batch job should be deactivated at the end of the batch job. Statistics are not collected after the batch job has ended. But if the session is not deactivated, PROFILER may contain indicators that the session is active.

Note: PROFILER permits statistics for batch objects to be viewed on-line, as well as in batch reports. PROFILER also permits statistics for on-line activity to be viewed on-line and in batch reports. Batch and on-line statistics may also be viewed in the same report.
--

X.2 Activate/Deactivate Session in Batch

The Batch Activate Session function allows a user to define activation parameters and start a profiling session in order to collect statistics during the execution of a batch NATURAL object.

The Batch Deactivate Session function allows a user to stop the collection of PROFILER statistics after the execution of a batch NATURAL object.

A user may not designate additional objects to be included or excluded when activating a session in batch. However, the PROFILER Trace function will include or exclude objects listed in the Session Definition for the session being activated.

Sample MVS JCL

The following JCL is an example of typical JCL used to support the Activate and Deactivate Session functions in batch for MVS. Actual JCL differs, depending on processing environment factors.

Required parameters are enclosed {...} while optional parameters are enclosed [...].

```
//PROBATCH      JOB (20000),'PROFILER SESSION'
//*
//PROFILE       EXEC  NATURAL
//*
//CMPRT01      DD   SYSOUT=*
//CMSYNIN      DD   *
LOGON USERLIB /* USER LIBRARY WHICH CONTAINS THE OBJECT TO BE PROFILED
PRF AC {Session Name} [Library Mask] [Object Mask] [Object Types] [Password]
USERPGM      /* THE OBJECT TO BE PROFILED
PRF DA {Session Name}
FIN
/*
//
```

Sample VM EXECs

The following EXEC is an example of a typical EXEC used to support the Activate and Deactivate Session functions in batch for VM. The actual EXEC differs depending on processing environment factors.

```
/* Sample REXX EXEC for using PROFILER in batch */
address 'COMMAND'
'FILEDEF CMSYNIN DISK PROF CMSYNIN A'
'FILEDEF CMPRINT PRINTER'

'EXEC NATURAL'
exit
```

Sample PROF CMSYNIN A

Required parameters are enclosed {...} while optional parameters are enclosed [...].

```
LOGON USERLIB (USER LIBRARY WHICH CONTAINS THE OBJECT TO BE PROFILED)
PRF AC {Session Name} [Library Mask] [Object Mask] [Object Types] [Password]
USERPGM      (THE OBJECT TO BE PROFILED)
PRF DA {Session Name}
FIN
```

Activate Profile Session

When activating a profile session in batch, it is necessary to replace the required and optional parameters after the 'PRF AC' command with values suitable to the user's purpose. Parameters should be provided in the exact order described, separated by the delimiter character (the default is comma: ",").

Command:	PRF AC										
Parameters:	Session Name,Library Mask,Object Mask,Object Types, ADABAS Password										
<u>Parameter Format:</u>											
Session Name	The 16-character alphanumeric field that represents the profiled session. {Required}										
Library Mask	An 8-character alphanumeric field that defines the NATURAL library or libraries to be profiled.										
Object Mask	An 8-character alphanumeric field that defines the NATURAL object or objects to be profiled. Refer to the Activate Session sub-section of the Session Maintenance and Execution section for applicable Library/Object masks.										
Object Types	A 5-character alphabetic field that contains the NATURAL object types to be profiled. Valid values are as follows: <table> <tr> <td>P</td> <td>Program</td> </tr> <tr> <td>N</td> <td>Subprogram</td> </tr> <tr> <td>S</td> <td>Subroutine</td> </tr> <tr> <td>M</td> <td>Map</td> </tr> <tr> <td>H</td> <td>Helproutine</td> </tr> </table> <p>One or more object types may be specified. If no object type is specified, all object types are automatically selected.</p>	P	Program	N	Subprogram	S	Subroutine	M	Map	H	Helproutine
P	Program										
N	Subprogram										
S	Subroutine										
M	Map										
H	Helproutine										
ADABAS Password	An 8-character password required only if the PROFILER File is protected by ADABAS Security.										
Activate Example:	LOGON USERLIB PRF AC PAYSESSION,PAYTEST,PAY0000P,P PAYSESSION will be activated to collect statistics for program PAY0000P in library PAYTEST.										

Deactivate Profile Session

When deactivating a profile session in batch, it is necessary to replace the session name with a value suitable to the user's purpose. The named, currently active profile session will be deactivated.

Command : PRF DA {Session Name}
 Deactivate Example: LOGON USERLIB
 PRF DA PAYSESSION

The user's active session will be deactivated to stop the collection of statistics.

Activate Trace Session

To run the Activate Trace Session function in batch, the following parameters are used:

Command: PRF AC
 Parameters: Session Name,Library Mask,Object Mask,Statement Types,ADABAS password,Starting Statement,Ending Statement.

Parameter Format:

Session Name The 16-character alphanumeric field that represents the profiled session. {Required}

Library Mask An 8-character alphanumeric field that defines the NATURAL library or libraries to be traced.

Object Mask An 8-character alphanumeric field that defines the NATURAL object or objects to be traced.

Refer to the **Activate Session** sub-section of the **Session Maintenance and Execution** section for applicable Library/Object masks.

Statement Types An 8-character alphabetic field that contains the NATURAL statement types to be traced. Valid values are as follows:

- D Database
- I I/O
- X Calls
- C CallNat
- P Perform
- F Fetch
- R For/Repeat
- O Other

One or more statement types may be specified. If no statement type is specified, all statement types are automatically selected.

ADABAS Password An 8-character password required only if the PROFILER File is protected by ADABAS Security.

Starting/Ending Statement If only one library/object is to be traced, the 'starting' and 'ending' statement numbers may be specified in order to reduce the amount of Trace output.

Activate Trace Example: LOGON USERLIB
 PRF AC PAYTRACE,PAYTEST,PAY*

A trace session will be activated to collect Trace data for all objects beginning with PAY and library PAYTEST.

Deactivate Trace Session

To run the Deactivate Trace Session function in batch, replace the session name with a value suitable to the user's purpose. The named, currently active trace session will be deactivated:

Command: PRF DA {Session Name}

Deactivate Trace Example: LOGON USERLIB
PRF DA PAYTRACE

The user's active session will be deactivated to stop the collection of statistics.

X.3 Reset Session Statistics in Batch

The Reset Session Statistics function allows a user to reset profile or trace session statistics, but not affect the session definition.

Sample MVS JCL

The following JCL is an example of typical JCL used to support the Reset Session Statistics function in batch for MVS. Actual JCL differs, depending on processing environment factors.

```
//PROBATCH JOB (20000),'PROFILER SESSION'
//*
//PROFILE EXEC NATURAL
//*
//CMPRT01 DD SYSOUT=*
//CMSYNIN DD *
LOGON USERLIB
PRF RS {Session Name} [User-ID] [Library Mask] [Object Mask] [Object Types]
FIN
/*
//
```

Sample VM EXEC

The following EXEC is an example of a typical EXEC used to support the Reset Session Statistics function in batch for VM. Actual EXEC differs depending on processing environment factors.

```
/* Sample REXX EXEC for using PROFILER in batch*/
address 'COMMAND'
'FILEDEF CMSYNIN DISK PROF CMSYNIN A'
'FILEDEF CMPRINT PRINTER'

'EXEC NATURAL'
exit
```

Sample PROF CMSYNIN A:

```
LOGON USERLIB
PRF RS {Session Name} [User-ID] [Library Mask] [Object Mask] [Object Types]
FIN
```

Reset Profile Session Statistics

When resetting profile session statistics in batch, it is necessary to replace the required and optional parameters with values suitable to the user's purpose in the exact order described, separated by the delimiter character (the default is comma ",").

Command: PRF RS
Parameters: Session Name, User-ID, Library Mask, Object Mask, Object Types

Parameter Format:Required Parameters:

Session Name The 16-character alphanumeric field that represents the profiled session.

Optional Parameters:

User-ID An 8-character field that represents the user who profiled the object.

Library Mask An 8-character alphanumeric field that defines the profiled library or libraries.

Object Mask An 8-character alphanumeric field that defines the profiled object or objects.

Object Types A 5-character alphabetic field that contains the NATURAL object types to be profiled. Valid values are as follows:

P	Program
N	Subprogram
S	Subroutine
M	Map
H	Helproutine

One or more object types may be specified. If no object type is specified, all object types are automatically selected.

Reset Session Statistics Example: LOGON USERLIB
PRF RS PAYSESSION, USER24, PAYTEST, PAY1000P
Statistics for program PAY1000P in library PAYTEST in PAYSESSION for USER24 will be reset.

Reset Trace Session Statistics

To reset trace session statistics in batch, replace the session name with a value suitable to the user's purpose. The named trace session will have all its statistics reset:

Command: PRF RS {Session Name}
Deactivate Trace Example: LOGON USERLIB
PRF RS PAYTRACE

The user's active session will be deactivated to stop the collection of statistics.

X.4 Tagged Data Batch Operations

X.4.1 Tag Data For Move

The Tag Data for Move (TG) function allows a user to mark records on the PROFILER repository for transfer to the PROFILER Reporting file. These records may be moved to the Reporting file with the Move Tagged Data (MT) function.

Sample MVS JCL

The following JCL is an example of JCL used to support the Tag Data function in batch for MVS. Actual JCL differs depending on processing environment factors.

```
//PROBATCH      JOB (20000),'PROFILER SESSION'
//*
//PROFILE       EXEC  NATURAL
//*
//CMPRT01      DD   SYSOUT=*
//CMSYNIN      DD   *
LOGON USERLIB
PRF TG {Session Name|*},[User-ID|*],[Library Mask|*],[Object Mask|*],[Object Types|*]
FIN
/*
//
```

Sample VM EXECs

The following EXEC is an example of a typical EXEC used to support the Tag Data For Move function in batch for VM. Actual EXEC differs depending on processing environment factors.

```
/* Sample REXX EXEC for using PROFILER in batch */
address 'COMMAND'
'FILEDEF CMSYNIN DISK PROF CMSYNIN A'
'FILEDEF CMPRINT PRINTER'

'EXEC NATURAL'
exit
```

Sample PROF CMSYNIN A:

```
LOGON USERLIB
PRF TG {Session Name|*},[User-ID|*],[Library Mask|*],[Object Mask|*],[Object Types|*]
FIN
```

Command: PRF TG

Parameters: Session Name,User-ID,Library Mask,Object Mask,Object Types

Parameter Format:

Required Parameters:

Session Name The 16-character alphanumeric field that represents the profiled session.

Optional Parameters:

User-ID An 8-character field that represents the user who profiled the object. An astrisk (*) may also be used to specify all users (this is the default).

Library Mask An 8-character alphanumeric field that defines the profiled library or libraries. An astrisk (*) may also be used to specify all libraries (this is the default).

Object Mask An 8-character alphanumeric field that defines the profiled object or objects. An astrisk (*) may also be used to specify all Objects (this is the default).

Object Types A 5-character alphabetic field that contains the NATURAL object types to be profiled. Valid values are as follows:

P	Program
N	Subprogram
S	Subroutine
M	Map
H	Helproutine
*	All Objects

One or more object types may be specified. If no object type is specified, all object types are automatically selected.

Tag Data For Move Example: LOGON USERLIB
PRF TG PAYSESSION,USER24,PAYTEST,PAY1000P

Statistics for program PAY1000P in library PAYTEST in PAYSESSION for USER24 will be tagged for move.

X.4.2 Move All/Selected Tagged Data in Batch

The Move All/Selected Tagged Data functions allow a user to transfer records that have been tagged on the PROFILER repository to the PROFILER Reporting file.

Sample MVS JCL

The following JCL is an example of typical JCL used to support the Move All/Selected Tagged Data functions in batch for MVS. Actual JCL differs depending on processing environment factors.

```
//PROBATCH JOB (20000),'PROFILER SESSION'  
//*  
//PROFILE EXEC NATURAL  
//*  
//CMPRT01 DD SYSOUT=*  
//CMSYNIN DD *  
LOGON USERLIB  
PRF MT {Session Name*}, [Library Mask], [Object Mask], [User-ID]  
FIN  
/*  
//
```

Sample VM EXECs

The following EXEC is an example of a typical EXEC used to support the Move All/Selected Tagged Data functions in batch for VM. Actual EXEC differs depending on processing environment factors.

```
/* Sample REXX EXEC for using PROFILER in batch */  
address 'COMMAND'  
'FILEDEF CMSYNIN DISK PROF CMSYNIN A'  
'FILEDEF CMPRINT PRINTER'  
  
'EXEC NATURAL'  
exit
```

Sample PROF CMSYNIN A:

```
LOGON USERLIB  
PRF MT {Session Name*}, [Library Mask], [Object Mask], [User-ID]  
FIN
```

Move All Tagged Data

When resetting profile session statistics in batch, it is necessary to replace the

Command: PRF RS
 Parameters: Session Name, User-ID, Library Mask, Object Mask, Object Types

When moving all tagged data in batch, it is necessary to replace the required and optional parameters with values suitable to the user's purpose in the exact order described, separated by the delimiter character (the default is comma ",").

Command: PRF MT
 Parameters: *, Library Mask, Object Mask, User-ID

All records that have been tagged on the PROFILER repository will be transferred to the PROFILER Reporting file.

Move Selected Tagged Data

When moving selected tagged data in batch, it is necessary to replace the required and optional parameters with values suitable to the user's purpose in the exact order described, separated by the delimiter character (the default is comma ",").

Command: PRF MT
 Parameters: Session Name, Library Mask, Object Mask, User-ID

Parameter Format:

Session Name	The 16-character alphanumeric field that represents the profiled session. {Required}
Library Mask	An 8-character alphanumeric field that defines the profiled library or libraries.
Object Mask	An 8-character alphanumeric field that defines the profiled object or objects.
User-ID	An 8-character field that represents the user who profiled the object.

Move Selected Tagged Data Example: LOGON USERLIB
 PRF MT PAYSESSION, PAYTEST, PAY0000P, USER24

Data for program PAY0000P in library PAYTEST in PAYSESSION for USER24 will be moved from the PROFILER Reporting file to the PROFILER Reporting file.

X.5 Batch Reports

All of the on-line reports may be run in batch. To execute PROFILER reports in batch, use a working copy of a NATURAL PROC.

Batch reporting is recommended when reports access statistics for a large number of objects.

Sample MVS JCL

The following JCL is an example of JCL used to support the Batch Reporting process for MVS. Actual JCL differs, depending on processing environment factors.

Note: The following parameters in the JCL/EXEC below must be replaced with the values suitable to the user's purpose.

```
//PROBATCH JOB (20000),'PRO BATCH REPORT'
//*
//REPORT EXEC NATURAL
//*
//CMPRT01 DD SYSOUT=*
//CMSYNIN DD *
LOGON USERLIB
PRF {Session Action} {Session Name} [Sort Order] [Object Types] [Starting Library]
    [Starting Object] [Ending Object] [User-ID] [Thresholds]...7
FIN
/*
//
```

Sample VM EXECs

The following EXEC is an example of an EXEC used to support the Batch Reporting process for VM. The EXEC should be tailored to accommodate site-specific needs.

```
/* Sample REXX EXEC for PROFILER batch reports */
address 'COMMAND'
'FILEDEF CMSYNIN DISK PROF CMSYNIN A'
'FILEDEF CMPRINT PRINTER'

'EXEC NATURAL'
exit
```

Sample PROF CMSYNIN A

```
LOGON PROLIB
PRF {Session Action} {Session Name} [Sort Order] [Object Types] [Starting Library]
    [Starting Object] [Ending Object] [User-ID] [Thresholds]...7
```

FINSort datasets must be specified in the MVS JCL or the VM EXEC when the Session Statistics Summary Reports and the Enhanced Session Statistics Summary Reports are executed.

When submitting a PROFILER report in batch, it is necessary to replace the report parameters in the exact order described, separated by the delimiter character (the default is comma ","). Optional parameters may be omitted but must be designated with the delimiter character if other parameters follow.

The first parameter in the example below is session name PAYSESSION followed by the delimiter character, which is a comma. The second parameter is sort order PGM followed by a comma. The next four commas represent four optional parameters not entered. USER24 is the last parameter.

```
LOGON USERLIB
PRF SS PAYSESSION,OBJ,,,,USER24
```

Multiple reports may be run in the same job stream.

X.6 Batch Report Programs and Parameters

X.6.1 Session Statistics Summary Reports

There are eleven Session Statistics Summary Report formats. These reports provide information about NATURAL objects executed during an active profile session.

- Statement Execution Count ('SS' action)
- CPU time ('SC' action)
- Database Elapsed Time ('SD' action)
- Number of Statements, CPU Time, Database ('SN' action)
- Percentage of Executions, CPU and Database ('SP' action)
- Percent Graph of Executions, CPU and Database ('SG' action)
- Database/Work File Loops ('SF' action)
- Internal Subroutines/Non-Procedural Blocks ('SB' action)
- FOR/REPEAT Loops ('SR' action)
- IF/DECIDE Conditions ('SI' action)
- Percent Executed by Statement Type ('ST' action)

Statement Execution Count Summary Report ('SS' action)

Returns statement execution statistics, percent of statements executed, and a statement graph.

CPU Time Summary Report ('SC' action)

Returns statistics on CPU time used, percent of CPU time used, and a CPU graph.

Database Elapsed Time Summary Report ('SD' action)

Returns statistics on Database Elapsed Time used, percent of Database Elapsed Time used, and a Database graph.

Number of Statements, CPU Time, Database Time Report ('SN' action)

Returns statement execution statistics, CPU time used, and Database Elapsed time used.

Percentage of Executions, CPU Time, and Database Time Report ('SP' action)

Returns percent of statements executed, percent of CPU time used, and percent of Database Elapsed time used.

Percent Graph of Executions, CPU and Database Report ('SG' action)

Returns percent graph of statements executed, percent graph of CPU time used, and percent graph of Database Elapsed time used.

Database/Work File Loops Report ('SF' action)

Returns execution counts and execution percentages for the Database Loops (i.e., READ, FIND, HISTOGRAM) and Work File Loops in each object.

Internal Subroutines/Non-Procedural Blocks Report ('SB' action)

Returns execution counts and execution percentages for the internal subroutines and non-procedural blocks (i.e., break processing) in each object. A Non-Procedural Block is one in which execution depends on an event, not on where the statement is located in an object. The non-procedural NATURAL statements are AT BREAK, AT END OF DATA, AT END OF PAGE, AT START OF DATA, AT TOP OF PAGE, BEFORE BREAK PROCESSING, IF NO RECORDS FOUND, and ON ERROR.

FOR/REPEAT Loops Report ('SR' action)

Returns execution counts and execution percentages for the FOR/REPEAT loops in each object.

IF/DECIDE Conditions Report ('SI' action)

Returns execution counts and execution percentages for the IF/DECIDE conditions in each object.

Percent Executed by Statement Type Report ('ST' action)

Returns execution percentages for the Database Loops/Work File Loops, internal subroutines/non-procedural blocks, FOR/REPEAT loops, and IF/DECIDE conditions in each object.

The remaining parameters are the same for all these report formats.

Command: PRF {Session Action}

Parameters: Session Name,Sort Order,Object Types,Starting Library,
Starting Object,Ending Object,User-ID,Thresholds

Required Parameters:

Session Name The 16-character name that represents the profiled session.

Sort Order A 3-character alphabetic field that represents the report sort order. Valid sort orders are:

OBJ (Library/Object)
STA (Statement Execution Count)
CPU (CPU time)
DAT (Database Elapsed Time)

Optional Parameters:

Starting Library An 8-character field that represents the profiled library.

Starting Object	An 8-character field that represents the profiled object with which reporting is to begin.										
Ending Object	An 8-character field that represents the profiled object with which reporting is to end.										
User-ID	An 8-character field that represents the user who profiled the object.										
Object Types	<p>A 5-character alphabetic field that contains the NATURAL object types to be profiled. Valid values are as follows:</p> <table border="0"> <tr><td>P</td><td>Program</td></tr> <tr><td>N</td><td>Subprogram</td></tr> <tr><td>S</td><td>Subroutine</td></tr> <tr><td>M</td><td>Map</td></tr> <tr><td>H</td><td>Helproutine</td></tr> </table> <p>One or more object types may be specified. If no object type is specified, all object types are automatically selected.</p>	P	Program	N	Subprogram	S	Subroutine	M	Map	H	Helproutine
P	Program										
N	Subprogram										
S	Subroutine										
M	Map										
H	Helproutine										
Thresholds	<p>Seven thresholds may be supplied to limit the objects included in the report. Objects will be excluded from the report if their statistics are less than the supplied threshold values. The seven thresholds relate to:</p> <ul style="list-style-type: none"> • Run Count for Object • Statements Executed for Object • CPU Time used by Object (msecs) • Database Elapsed Time for Object (msecs) • Object's Percentage of Executable Statements Executed • Object's Percentage of Total Session CPU Time used • Object's Percentage of Total Session Database Elapsed 										
Report Example:	<pre>LOGON USERLIB PRF SS PAYSESSION,STA,,,,,USER24</pre> <p>The Statement Execution Count Summary Report will be executed for session PAYSESSION, for USER24, for all profiled libraries and object types, and sorted in statement execution order.</p>										

X.6.2 Source Code Listing Report for Profiled Objects

This report gives statistics for an individual NATURAL object, displays the source code for an individual NATURAL object, and lists PROFILER statistics for each executable statement in the object.

Command: PRF SL

Parameters: Session Name, User-ID, Library, Object, Source DBID, Source FUSER, Source Library, copycode STEPLIB

Required Parameters:

Session Name The 16-character name that represents the profiled session.

User-ID An 8-character field that represents the user who profiled the object.

Library An 8-character field that represents the profiled library.

Object An 8-character field that represents the profiled object.

Optional Parameters:

Source DBID The number of the database where PROFILER accesses NATURAL source code.

This field should not be specified unless source code resides on a different database than the database where PROFILER is installed.

Source FUSER The number of the FUSER where PROFILER accesses NATURAL source code.

This field should not be specified unless source code resides on a different FUSER than the FUSER where PROFILER is installed.

Source Library An 8-character field that represents the library that contains the NATURAL source code.

This field should not be specified unless source code resides in a different library than the library which contains the profiled object code.

Copycode STEPLIB An additional library where PROFILER should look for copycode source if the copycode is not found in the same library as the object source.

Report Example: LOGON USERLIB
PRF SL PAYSESSION,USER24,PAYTEST,PAY2000P

The Source Code Listing Report will be executed for session PAYSESSION, USER24, the library PAYTEST, and the program PAY2000P.

X.6.3 Source Code Listing Report for Traced Objects

This report displays the Trace data that has been collected for a Trace session.

To run the Trace Source Code Report in batch, the parameters for PROFILER Batch Reports are used, along with the following additional parameters:

Command: PRF SL

Parameters: Session Name, Source Code DBID and FUSER

Required Parameters:

Session Name The 16-character name that represents the Trace session.

Optional Fields:

Source Code DBID Identifies the database in which source code is to be retrieved.

Source Code FUSER Identifies the FUSER in which source code is to be retrieved.

X.6.4 Application QA Report

This report returns the percentages of an application library or a range of objects in a library tested and the names of objects un-tested during a profile session.

Command: PRF QA

Parameters: Session Name, Library, Starting Object, Ending Object, User-ID, Object Types

Required Parameters:

Session Name The 16-character name that represents the profiled session.

Library An 8-character field that represents the profiled library.

Optional Parameters:

Starting Object An 8-character field that represents the profiled object with which reporting is to begin.

Ending Object An 8-character field that represents the profiled object with which reporting is to end.

User-ID An 8-character field that represents the user who profiled the object.

Object Types A 5-character field that represents all NATURAL object types to be reported. Valid object types are:

- P Program
- M Map
- H Helproutine
- N Subprogram
- S Subroutine

Object types may be in any order and in any combination in the Object Type field. At least one object type is required.

Report Example: LOGON USERLIB
PRF QA PAYSESSION,PAYTEST,,,,PNS

The Application QA Report will be executed for session PAYSESSION, all users, library PAYTEST, and all programs, subprograms, and subroutines.

X.7 Batch Enhanced Report Programs and Parameters

X.7.1 Enhanced Session Statistics Summary Reports.

There are eleven Enhanced Session Statistics Summary Report formats. These reports provide information about NATURAL objects executed during an active profile session. Profile statistics have to be moved from the PROFILER repository to the Enhanced Reporting repository to make use of these Enhanced Reports.

- Statement Execution Count ('ES' action)
- CPU time ('EC' action)
- Database Elapsed Time ('ED' action)
- Number of Statements, CPU Time, Database ('EN' action)
- Percentage of Executions, CPU and Database ('EP' action)
- Percent Graph of Executions, CPU and Database ('EG' action)
- Database/Work File Loops ('EF' action)
- Internal Subroutines/Non-Procedural Blocks ('EB' action)
- FOR/REPEAT Loops ('ER' action)
- IF/DECIDE Conditions ('EI' action)
- Percent Executed by Statement Type ('ET' action)

Statement Execution Count Summary Report ('ES' action)

Returns statement execution statistics, percent of statements executed, and a statement graph.

CPU Time Summary Report ('EC' action)

Returns statistics on CPU time used, percent of CPU time used, and a CPU graph.

Database Elapsed Time Summary Report ('ED' action)

Returns statistics on Database Elapsed Time used, percent of Database Elapsed Time used, and a Database graph.

Number of Statements, CPU Time, Database Time Report ('EN' action)

Returns statement execution statistics, CPU time used, and Database Elapsed time used.

Percentage of Executions, CPU Time, and Database Time Report ('EP' action)

Returns percent of statements executed, percent of CPU time used, and percent of Database Elapsed time used.

Percent Graph of Executions, CPU and Database Report ('EG' action)

Returns percent graph of statements executed, percent graph of CPU time used, and percent graph of Database Elapsed time used.

Database/Work File Loops Report ('EF' action)

Returns execution counts and execution percentages for the Database Loops (i.e., READ, FIND, HISTOGRAM) and Work File Loops in each object.

Internal Subroutines/Non-Procedural Blocks Report ('EB' action)

Returns execution counts and execution percentages for the internal subroutines and non-procedural blocks (i.e., break processing) in each object. A Non-Procedural Block is one in which execution depends on an event, not on where the statement is located in an object. The non-procedural NATURAL statements are AT BREAK, AT END OF DATA, AT END OF PAGE, AT START OF DATA, AT TOP OF PAGE, BEFORE BREAK PROCESSING, IF NO RECORDS FOUND, and ON ERROR.

FOR/REPEAT Loops Report ('ER' action)

Returns execution counts and execution percentages for the FOR/REPEAT loops in each object.

IF/DECIDE Conditions Report ('EI' action)

Returns execution counts and execution percentages for the IF/DECIDE conditions in each object.

Percent Executed by Statement Type Report ('ET' action)

Returns execution percentages for the Database Loops/Work File Loops, internal subroutines/non-procedural blocks, FOR/REPEAT loops, and IF/DECIDE conditions in each object.

The remaining parameters are the same for all these report formats.

Command: PRF {Report Parameter Action}
Parameters: Report Parameter Name, Sort Order, Thresholds

Required Parameters:

Report Parameter Name The 16-character name that represents the Report Parameter.

Sort Order A 3-character alphabetic field that represents the report sort order. Valid sort orders are:

- OBJ (Library/Object)
- STA (Statement Execution Count)
- CPU (CPU time)
- DAT (Database Elapsed Time)

Optional Parameters:

Thresholds

Seven thresholds may be supplied to limit the objects included in the report. Objects will be excluded from the report if their statistics are less than the supplied threshold values. The seven thresholds relate to:

- Run Count for Object
- Statements Executed for Object
- CPU Time used by Object (msecs)
- Database Elapsed Time for Object (msecs)
- Object's Percentage of Executable Statements Executed
- Object's Percentage of Total Report Parameter CPU Time used
- Object's Percentage of Total Report Parameter Database Elapsed

Report Example:

```
LOGON USERLIB  
PRF ES PAY-PARAM,STA
```

The Enhanced Statement Execution Count Summary Report will be executed for report parameter PAY-PARAM sorted in statement execution count order.

X.7.2 Enhanced Source Code Listing Report

This report gives statistics for an individual NATURAL object, displays the source code for an individual NATURAL object, and lists PROFILER statistics for each executable statement in the object.

Command: PRF EL

Parameters: Report Parameter Name, Library, Object, Source DBID, Source FUSER, Source Library, copycode STEPLIB

Required Parameters:

Report Parameter Name The 16-character name that represents the Report Parameter.

Library An 8-character field that represents the profiled library.

Object An 8-character field that represents the profiled object.

Optional Parameters:

Source DBID The number of the database where PROFILER accesses NATURAL source code.

This field should not be specified unless source code resides on a different database than the database where PROFILER is installed.

Source FUSER The number of the FUSER where PROFILER accesses NATURAL source code.

This field should not be specified unless source code resides on a different FUSER than the FUSER where PROFILER is installed.

Source Library An 8-character field that represents the library that contains the NATURAL source code.

This field should not be specified unless source code resides in a different library than the library which contains the profiled object code.

Copycode STEPLIB An additional library where PROFILER should look for copycode source if the copycode is not found in the same library as the object source.

Report Example: LOGON USERLIB
PRF EL PAY-PARAM,PAYTEST,PAY2000P

The Enhanced Source Code Listing Report will be executed for report parameter PAY-PARAM for the library PAYTEST and program PAY2000P.

X.7.3 Enhanced Application QA Report

This report returns the percentages of an application library or a range of objects in a library tested and the names of objects un-tested during a profile sessions included in the chosen report parameter.

Command: PRF EQ
Parameters: Report Parameter Name, Library, Starting Object, Ending Object

Required Parameters:

Report Parameter Name The 16-character name that represents the profiled Report Parameter.

Library An 8-character field that represents the profiled library.

Optional Parameters:

Starting Object An 8-character field that represents the profiled object with which reporting is to begin.

Ending Object An 8-character field that represents the profiled object with which reporting is to end.

Report Example: LOGON USERLIB
PRF EQ PAY-PARAM,PAYTEST

The Enhanced Application QA Report will be executed for report parameter PAY-PARAM for library PAYTEST.

This page intentionally left blank.

SECTION XI

EDUCATION AND EXAMPLES

XI.1 Introduction

This section provides instructions detailing the use of PROFILER to assist in:

- Quality Assurance
- Performance Analysis
- Debugging
- Application Testing
- Education and Evaluation

PROFILER can help pinpoint poor coding techniques, compare different coding techniques, learn how NATURAL statements work internally, and identify insufficient test data.

Various factors in the application environment determine which coding techniques and NATURAL statements are the most efficient. It is the programmer's responsibility to understand the environment and the data being used in order to write efficient NATURAL code. Using PROFILER in a development environment allows a programmer to try different NATURAL statements and alternative coding techniques to determine which are best for a specific application.

Examples in this section also show a programmer how to use the PROFILER reports to identify when insufficient test data has been used. Learning to use the PROFILER reports to determine which statements in an object were not executed leads to the development of data which tests every condition in an object.

Such education and testing leads to better code, better educated programmers, thoroughly tested programs, and less maintenance once an object is migrated to a production environment.

XI.2 Quality Assurance

Quality Assurance (QA) is one of the primary uses for PROFILER. This is achieved using the 'Summary Report' and 'Application QA Report' for a Profile Session. The Application QA Report shows the percentage of an application tested, while the Summary Report shows the percentage of each object in the library tested. The following Application QA Report indicates that 11.54% of the objects in the library have been tested.

```

2011-06-10 16:38      Session PAYROLL      USER24  PAYTEST
QA Report:      52 Objects in PAYTEST_      View Executed
of which      6 ( 11.54% ) were Executed.      Objects? N
Objects NOT Executed starting..      types..      Page 1__ of 2

S      T Exec| S      T Exec| S      T Exec| S      T Exec
e      y utbl| e      y utbl| e      y utbl| e      y utbl
l Object p Stmt| l Object p Stmt| l Object p Stmt| l Object p Stmt
- CITYTAXL L 1 | - FICAM M 9 | - LIFEINSS S 3 | - PAY0120M M 3
- CITYTAXM M 9 | - FICAP P 14 | - PAYBATCH P 10 | - PAY0120P P 30
s CITYTAXP P 13 | - FICAS S 3 | - PAYEMPL L 0 | - PAY0120T M 2
- CITYTAXS S 3 | - KAH0100M M 2 | - PAYKH P 30 | - PAY0130P P 28
- CITYTX2L L 0 | - KAH0100P P 26 | - PAYL L 0 | - PAY0130T M 2
- FEDTAXL L 1 | - KAH1080 P 874 | - PAYLOCL L 0 | - PAY0140M M 2
- FEDTAXM M 9 | - KHBIBM M 2 | - PAYROLLG C 1 | - PAY0140P P 25
- FEDTAXP P 15 | - LIFEINSL L 1 | - PAY0100T M 2 | - PAY0140T M 2
- FEDTAXS S 3 | - LIFEINSM M 6 | - PAY0110P P 27 | - PENSIONL L 0
- FICAL L 1 | - LIFEINSP P 13 | - PAY0110T M 2 | - PENSIONM M 6

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10---PF11---PF12---
      Help      End      Up      Down      Sourc      Exit
    
```

The %Executbl Stmt Exec (Percent of Executable Statements Executed) on the Summary Report shows the percentage of each object that has been tested. Note that only 76.92% of the executable statements in program PAY0100P were executed.

```

PRO0096: 'S'elect Object to see its Source Code Listing Report.

                Summary Report for Session
2011-06-10 16:23      Session PAYROLL      USER24  PAYTEST
Report Format S Sort Order OBJ Types      QA? N Page 1__ of 1
Start Library      Start Object      View/Amend Thresholds N

      Total Stmt Execs      2263      Exec %Exec % Graph of
s      T      Run      Total Exec utbl -utbl Executable
e      y      Count      Stmt utbl Stmt Stmts Statements
l Library Object p      Execs Stmt Exec Exec Executed
- PAYTEST PAYROLL P      1      1 2 1 50.00 *****
- PAYTEST PAY0100M M      4      8 2 2 100.00 *****
- PAYTEST PAY0100P P      1      159 104 80 76.92 *****
- PAYTEST PAY0110M M      1      4 2 2 100.00 *****
- PAYTEST PAY0125M M      1      4 2 2 100.00 *****
- PAYTEST PAY0130M M      1      4 2 2 100.00 *****

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10---PF11---PF12---
      Help User End Stmt CPU Dbase Up Down Sourc Left Right Exit
    
```

The Source Code Listing Report may be used to determine which statements in program PAY0100P were not executed. Identifying statements which did not execute may pinpoint poor coding techniques and insufficient test data.

In the screen excerpt, executable statements that did not execute are marked with a ">".

```

PRO0101: Statements shown: Executed, Un-executed and Non-Executable.
2011-06-10 16:29 Profile Session PAYROLL          Object Profiled by USER24

  Execs Total CPU Avg CPU > _____ .....1.....2... PAY0100P Lib PAYTEST
                                0230 01 #CODE (A1)
                                0240 END-DEFINE
                                0250 *
                                0260 INPUT #SUPER-KEY
                                0270 *
1      5.874  5.874  0280 FIND EMPLOYEES WITH
<D'base 31.328 31.328>
                                0290 PERSONNEL-ID = #SUPER-KEY
                                > 0300 IF *NUMBER = 0
                                > 0310 WRITE 'NO RECORDS FOUND ON EMPLOYEE FILE
                                > 0320 ESCAPE BOTTOM
                                0330 END-IF
                                > 0340 INPUT USING MAP 'EMP0001P'
                                > 0350 END-FIND

PF7 -P  PF8 +P  PF9 BOT  PF10 <  PF11 >  PF12 EXIT  Page 2 of 5

```

Lines 0300, 0310, 0320, 0340 and 0350 were never executed because the FIND loop was never entered. PERSONNEL-ID did not equal #SUPER-KEY. This report identifies a poor coding technique. Code to warn a user that no records have been found has been placed inside of a loop that will not be entered if no records are found. When program PAY0100P executed, the INPUT statement was not executed and the message "NO RECORDS FOUND ON EMPLOYEE FILE" was not displayed.

The program PAY0100P may be re-coded as follows to correct this logic error.

```

PRO0101: Statements shown: Executed, Un-executed and Non-Executable.
2011-06-10 16:29 Profile Session PAYROLL          Object Profiled by USER24

  Execs Total CPU Avg CPU > _____ .....1.....2... PAY0100P Lib PAYTEST
                                0230 01 #CODE (A1)
                                0240 END-DEFINE
                                0250 *
                                0260 INPUT #SUPER-KEY
                                0270 *
1      5.075  5.075  0280 FIND EMPLOYEES WITH
<D'base 36.128 36.128>
                                0290 PERSONNEL-ID = #SUPER-KEY
                                0300 IF NO RECORDS FOUND
1      2.059  2.059  0310 WRITE 'NO RECORDS FOUND ON EMPLOYEE FILE
1      0.000  0.000  0320 ESCAPE BOTTOM
                                > 0330 END-NOREC
                                > 0340 INPUT USING MAP 'EMP0001P'
                                > 0350 END-FIND

PF7 -P  PF8 +P  PF9 BOT  PF10 <  PF11 >  PF12 EXIT  Page 2 of 5

```

The IF NO RECORDS FOUND clause is executed once if no records are found. Then, if the message "NO RECORDS FOUND ON EMPLOYEE FILE" is displayed and the INPUT statement is not executed, the programmer knows that #SUPER-KEY did not contain a valid PERSONNEL-ID.

This may point to a problem in the way that #SUPER-KEY was constructed or may point to insufficient test data.

XI.3 Performance Analysis

The Session Statistics Summary Reports provide the tools to measure the performance of an application. A project manager or Database Administrator (DBA) may easily identify which objects use the most resources. The user may toggle between the following eleven reports:

- Statement Execution Summary
- CPU Time Used Summary
- Database Elapsed Time Summary
- Number of Statements, CPU Time, Database Time
- Percentage of Statements Executed, CPU and Database Time Used
- Percent Graph of Statements Executed, CPU and Database Time Used
- Database/Work File Loops
- Inline Subroutine/Non-procedural Blocks
- FOR and/or REPEAT Loops
- IF and/or DECIDE Conditions
- Percent Executed by Statement Type

Each report may be presented in one of four orders:

- Library/Object
- Statement Execution Count
- CPU time
- Database Elapsed Time

For example, the following report shows the Statement Execution Count Summary Report sorted in library/object (OBJ) order.

```

PRO0096: 'S'elect Object to see its Source Code Listing Report.

                Summary Report for Session
2011-06-10 16:23      Session PAYROLL      USER24  PAYTEST
Report Format S  Sort Order OBJ Types _____ QA? N  Page 1__ of 1
Start Library _____ Start Object _____ View/Amend Thresholds N

                Total Stmt Execs          2263      Exec  %Exec  % Graph of
                T                               Total Exec utbl  -utbl Executable
e               y                               Stmt utbl Stmt  Stmts
l Library  Object p   Count      Execs Stmt Exec  Exec  Executed
- PAYTEST  PAYROLL P     1           1    2    1   50.00  *****
- PAYTEST  PAY0100M M     4           8    2    2  100.00  *****
- PAYTEST  PAY0100P P     1          159  104   80   76.92  *****
- PAYTEST  PAY0110M M     1           4    2    2  100.00  *****
- PAYTEST  PAY0125M M     1           4    2    2  100.00  *****
- PAYTEST  PAY0130M M     1           4    2    2  100.00  *****

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  User  End  Stmts CPU  Dbase Up   Down  Sourc Left  Right Exit
    
```

Pressing PF5 toggles to the CPU Time Summary Report.

```

PRO0096: 'S'elect Object to see its Source Code Listing Report.

                Summary Report for Session
2011-06-12 10:17  Session PAYROLL                USER24  PAYTEST
Report Format C  Sort Order OBJ Types _____ QA? N  Page 1__ of 1
Start Library _____ Start Object _____ View/Amend Thresholds N

                Total CPU Time                142.9120 msecs

S
e
l  Library  Object  p  Run  CPU  % of CPU  Graph of
   _  _  _  _  _  _  _  _  _  _  _  _
   _  _  _  _  _  _  _  _  _  _  _  _
   _  _  _  _  _  _  _  _  _  _  _  _
   _  _  _  _  _  _  _  _  _  _  _  _
   _  _  _  _  _  _  _  _  _  _  _  _
   _  _  _  _  _  _  _  _  _  _  _  _
   _  _  _  _  _  _  _  _  _  _  _  _

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help User End  Stmts CPU  Dbase Up  Down  Sourc Left Right Exit
    
```

Pressing PF10 (Left) and PF11 (Right) allows you to toggle or cycle through all eleven report formats.

```

PRO0096: 'S'elect Object to see its Source Code Listing Report.

                Summary Report for Session
2011-06-12 10:18  Session PAYROLL                USER24  PAYTEST
Report Format N  Sort Order OBJ Types _____ QA? N  Page 1__ of 1
Start Library _____ Start Object _____ View/Amend Thresholds N

Total Stmt Execs 580                Exec Total CPU 142.9120
S          T          Database
e          y          Elapsed
l  Library  Object  p  Run  Stmt Stmt  CPU Time
   _  _  _  _  _  _  _  _  _  _  _  _
   _  _  _  _  _  _  _  _  _  _  _  _
   _  _  _  _  _  _  _  _  _  _  _  _
   _  _  _  _  _  _  _  _  _  _  _  _
   _  _  _  _  _  _  _  _  _  _  _  _
   _  _  _  _  _  _  _  _  _  _  _  _
   _  _  _  _  _  _  _  _  _  _  _  _

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help User End  Stmts CPU  Dbase Up  Down  Sourc Left Right Exit
    
```

Excessive CPU Time

The CPU Time Summary Report indicates that the CPU time for program PROG001 is 91.16 milliseconds, which is 50.32% of the total CPU time for the profiling session.

```

PRO0096: 'S'elect Object to see its Source Code Listing Report.

                Summary Report for Session
2011-06-12 10:17      Session PAYROLL      USER24  PAYTEST
Report Format C  Sort Order OBJ Types _____ QA? N  Page 1__ of 1
Start Library _____ Start Object _____ View/Amend Thresholds N
                Total CPU Time      142.9120 msec

s          T
e          y
l Library  Object  p      Run      CPU      % of CPU      Graph of
                Count      Time (msec)  Time Used      % of CPU
                p      Count      Time (msec)  Time Used      Time Used.

s PAYTEST  PROG001  P          1          91.16      50.32      *****
_ PAYTEST  PROG002  P          1          50.00      27.60      **
_ PAYTEST  PROG003  P          1          10.00      5.52
_ PAYTEST  PROG004  P          1          10.00      5.52
_ PAYTEST  PROG005  P          1          20.00      11.04      *

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help User  End  Stmt CPU  Dbase Up  Down  Sourc Left  Right Exit
    
```

The Sel field on the CPU Time Summary Report allows a programmer to view a source listing of the object to see which statements accounted for the most CPU time.

```

PRO0101: Statements shown: Executed, Un-executed and Non-Executable.
2011-06-12 10:18 Profile Session PAYROLL      Object Profiled by USER24

Execs Total CPU Avg CPU > _____ .....1.....2... PROG001  Lib PAYTEST
                0010 DEFINE DATA LOCAL
                0020 1 #I (P3)
                0030 1 #A (A1/1:20)
                0040 END-DEFINE
                0050 *
                21 33.577 1.599 0060 FOR #I 1 20
                20 31.446 1.572 0070 ASSIGN #A (#I) = 4
                20 26.137 1.307 0080 END-FOR
                1 0.000 0.000 0090 END

PF7 -P  PF8 +P  PF9 BOT  PF10 <  PF11 >  PF12 EXIT  Page 1 of 1
    
```

In this simple program, the 3 statements in the FOR loop used 91.16 milliseconds of CPU time and accounted for all of the CPU time in program PROG001. PROFILER allows a programmer to code and test alternate methods for accomplishing the same task (in this case, populating array #A).

Other coding techniques that populate array #A are shown below.

```

PRO0101: Statements shown: Executed, Un-executed and Non-Executable.
2011-06-12 10:19 Profile Session PAYROLL      Object Profiled by USER24

Execs Total CPU Avg CPU > _____ .....1.....2... PROG01A  Lib PAYTEST
                0010 DEFINE DATA LOCAL
                0020 1 #A (A1/1:20)
                0030 END-DEFINE
                0040 *
                1 2.007 2.007 0050 ASSIGN #A(1:20) = 4
                1 0.000 0.000 0060 END

PF7 -P  PF8 +P  PF9 BOT  PF10 <  PF11 >  PF12 EXIT  Page 1 of 1
    
```

In program PROG01A, the code to populate array #A used 2.007 milliseconds of CPU time, compared to 91.16 milliseconds used by the FOR loop in PROG001.

```

PRO0101: Statements shown: Executed, Un-executed and Non-Executable.
2011-06-12 10:20 Profile Session PAYROLL          Object Profiled by USER24

  Execs Total CPU Avg CPU > _____ .....1.....2... PROG01B  Lib PAYTEST
                                0010 DEFINE DATA LOCAL
                                0020 1 #A (A1/1:20)
                                0030 END-DEFINE
                                0040 *
      1      2.642  2.642  0050 ASSIGN #A(*) = 4
      1      0.000  0.000  0060 END

PF7 -P   PF8 +P   PF9 BOT  PF10 <   PF11 >   PF12 EXIT   Page 1 of 1

```

In program PROG01B, the code to populate array #A used 2.642 milliseconds of CPU time.

In this test, when using three different methods for populating an array, PROG01A would have been the best choice for the programmer, saving 89.153 milliseconds of CPU time.

Excessive Database Elapsed Time

The Number of Statements, CPU Time, Database Time Report shows that PROG20P used over 3700 milliseconds of Database Elapsed Time, indicating a possible coding problem.

```

PRO0096: 'S'elect Object to see its Source Code Listing Report.

                          Summary Report for Session
2011-06-12 10:21          Session PAYROLL          USER24  PAYTEST
Report Format N  Sort Order OBJ  Types _____  QA? N   Page 1__ of 1
Start Library PAYTEST_  Start Object PROG20P_  View/Amend Thresholds N
Total Stmt Execs 580          Exec  Total CPU 142.9120
S          T          Total utbl          Database
e          y          Stmt Stmt          CPU Time          Elapsed
l Library  Object  p          Count          Execs Exec          Used (msec)          Time (msec)

s PAYTEST  PROG20P  P          1          303 364          1.14          3750.11

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
Help User  End  Stmts CPU  Dbase Up  Down  Sourc Left  Right Exit

```

Looking at the source code listing for program PROG20P may indicate the reason for the high Database Elapsed Time.

```

PRO0101: Statements shown: Executed, Un-executed and Non-Executable.
2011-06-12 10:22 Profile Session PAYROLL          Object Profiled by USER24

  Execs Total CPU Avg CPU > _____ .....1.....2... PROG20P  Lib PAYTEST
                                0010 DEFINE DATA LOCAL
                                0020 1 #TOTAL-EMPLOYEES (P7)
                                0030 END-DEFINE
                                0040 *
                                0050 *
101   730.607   7.234   0060 READ (100) EMPLOYEES
<D'base 3750.112 37.130>
100   245.660   2.457   0070   WHERE BIRTH > 500101
100   148.743   1.487   0080   ADD 1 TO #TOTAL-EMPLOYEES
1     2.655     2.655   0090 END-READ
1     0.000     0.000   0100 DISPLAY #TOTAL-EMPLOYEES
                                0110 END

PF7 -P   PF8 +P   PF9 BOT  PF10 <   PF11 >   PF12 EXIT   Page 1 of 1

```

If the intention of the programmer who coded program PROG20P was to read only 100 records, the READ statement with a WHERE clause may have been a poor choice to use. When a processing limit is specified in a READ statement with a WHERE clause, records that are rejected as a result of the WHERE clause are not counted against the limit. The program reads records until it finds 100 records which satisfy the condition specified in the WHERE clause. This may cause a considerable number of records to be read. In this example, 208 records were actually read. Since this is not readily apparent to the programmer, the high Database Elapsed Time is an indication that this program may have a problem. A better choice for the programmer who wanted to read only 100 records may be the following code:

```

PRO0101: Statements shown: Executed, Un-executed and Non-Executable.
2011-06-12 10:23 Profile Session PAYROLL          Object Profiled by USER24

  Execs Total CPU Avg CPU > _____ .....1.....2... PROG21P  Lib PAYTEST
                                0010 DEFINE DATA LOCAL
                                0020 1 #TOTAL-EMPLOYEES (P7)
                                0030 END-DEFINE
                                0040 *
                                0050 *
101   464.725   4.601   0060 READ (100) EMPLOYEES
<D'base 1947.552 19.283>
100   144.593   1.446   0070   ACCEPT IF BIRTH > 500101
46    119.316   2.594   0080   ADD 1 TO #TOTAL-EMPLOYEES
46    63.640   1.383   0090 END-READ
1     2.018     2.018   0100 DISPLAY #TOTAL-EMPLOYEES
1     0.000     0.000   0110 END

PF7 -P   PF8 +P   PF9 BOT  PF10 <   PF11 >   PF12 EXIT   Page 1 of 1

```

PROG21P reads 101 records, processes 46 records, and saves 1,802.56 milliseconds of Database Elapsed Time.

XI.4 Debugging

The Source Code Listing Report is an effective tool for debugging a NATURAL object. It displays the number of times each NATURAL statement executed and how much CPU time and Database Elapsed Time each statement used. It also shows which statements were not executed.

PRO0101: Statements shown: Executed, Un-executed and Non-Executable.				2011-06-12 10:24 Profile Session PAYROLL		Object Profiled by USER24	
Execs	Total CPU	Avg CPU	>1.....2...	PAYPROG4	Lib	PAYTEST
				0010	DEFINE DATA LOCAL USING PAYEMP		
				0020	END-DEFINE		
101	22.038	.218		0030	READ (100) EMPLOYEES-VIEW		
<D'base	230.592	2.283>					
100	1.051	0.011		0040	IF BIRTH < 430101		
9	0.766	0.085		0050	WRITE 'EMPLOYEE ELIGIBLE FOR PENSION'		
				0060	PERSONNEL-ID FULL-NAME		
9	0.045	0.005		0070	IF DEPT = 'ADM'		
			>	0080	WRITE 'EMPLOYEE WILL KEEP CONTRIBUTIONS'		
				0090	END-IF		
PF7 -P				PF8 +P		PF9 BOT	
PF10 <				PF11 >		PF12 EXIT	
Page				1		of 1	

The IF statement at line 0070 was tested 9 times but the condition DEPT = "ADM" was never met. Therefore, the WRITE statement at line 0080 was never executed. This is indicated by the '>' beside line 0080.

XI.5 Application Testing

The NATURAL Statement Type Reports assist in application testing. There are five types of reports:

- Database/Work File Loops
- Inline Subroutine/Non-procedural Blocks
- FOR and/or REPEAT Loops
- IF and/or DECIDE Conditions
- Percent Executed by Statement Type

Each report indicates the number of occurrences of the statement type in an object and the percent of the statement type tested.

```

PR00096: 'S'elect Object to see its Source Code Listing Report.

                Summary Report for Session
2011-06-12 10:25      Session TESTPROF      USER24  PAYTEST
Report Format I  Sort Order OBJ  Types _____  QA? N  Page 1__ of 1
Start Library _____  Start Object _____  View/Amend Thresholds N

S          T          IF / DECIDE Conditions/Bodies:
e          y          Run  Total  Conds %Cond  Total  Body %Body
l Library  Object  p    Count  Conds  Exec  Exec   Body  Exec  Exec
- NATLIB1  PAY1000P P      3     10     5  50.0    10     1  10.0
- NATLIB1  PAY2000P P      1      1     1 100.0     2     2 100.0
- NATLIB1  PAY3000P P      2      8     7  87.5     8     2  25.0
- NATLIB1  PAY4000P P      4     16     6  37.5    17     2  11.8
- NATLIB1  PAY5000P N      1
- NATLIB1  PAY6000P N      1      4     4 100.0     4     4 100.0
- NATLIB1  PAY7000P P      2      6     6 100.0     8     8 100.0

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  User  End  Stmts CPU  Dbase Up    Down  Sourc Left  Right Exit
    
```

The IF DECIDE Conditions Report displays the following statistics in each object: the number of IF/DECIDE conditions, the Percent of Execution of the IF/DECIDE conditions, the number of IF/DECIDE Bodies, and the Percent of Execution of the IF/DECIDE Bodies.

The View/Amend Thresholds option may also be used as an application testing tool to ensure that application resource usage falls within acceptable limits.

XI.6 Education and Evaluation

The Source Code Listing Report and the Enhanced Source Code Listing Report enable programmers to determine which NATURAL statements are the most efficient in a particular NATURAL object.

EXAMINE vs. FOR

The following screen excerpts use the Source Code Listing Report to compare the EXAMINE and the FOR statements.

EXAMINE:

```

PRO0101: Statements shown: Executed, Un-executed and Non-Executable.
2011-06-12 10:26 Profile Session PAYROLL          Object Profiled by USER24

  Execs Total CPU Avg CPU > _____ .....1.....2...  PROG100P Lib PAYTEST
                                0010 DEFINE DATA LOCAL
                                0020 1 #ARRAY (A5/50)
                                0030 1 #NUMBER(P2)
                                0040 END-DEFINE
      1      0.099  0.099  0050 EXAMINE #ARRAY(*) FOR 'MONEY'
      1      0.002  0.002  0060 IF #NUMBER = 0
      1      0.069  0.069  0070 WRITE 'NO MONEY FOUND'
      1      0.000  0.000  0080 ELSE
                                > 0090 WRITE 'YOU ARE RICH!'
                                0100 END-IF

PF7 -P   PF8 +P   PF9 BOT  PF10 <   PF11 >   PF12 EXIT   Page 1 of 1

```

FOR:

```

PRO0101: Statements shown: Executed, Un-executed and Non-Executable.
2011-06-12 10:27 Profile Session PAYROLL          Object Profiled by USER24

  Execs Total CPU Avg CPU > _____ .....1.....2...  PROG101P Lib PAYTEST
                                0010 DEFINE DATA LOCAL
                                0020 1 #ARRAY (A5/50)
                                0030 1 #NUMBER(P2)
                                0040 1 #I      (P2)
      51      0.285  0.006  0060 FOR #I = 1 TO 50
      50      0.337  0.007  0070 IF #ARRAY(#I) = 'MONEY'
                                > 0080 WRITE 'YOU ARE RICH!'
                                > 0090 ADD 1 TO #NUMBER
                                > 0100 ESCAPE BOTTOM
                                0110 END-IF
                                0120 END-FOR
      1      0.002  0.002  0130 IF #NUMBER = 0
      1      0.069  0.069  0140 WRITE 'NO MONEY FOUND'
                                0150 END-IF

PF7 -P   PF8 +P   PF9 BOT  PF10 <   PF11 >   PF12 EXIT   Page 1 of 1

```

The EXAMINE statement was more efficient in this test. It used 0.099 milliseconds of CPU Time, compared to 0.622 milliseconds used by the FOR statement. It took 1 statement execution to use the EXAMINE statement and 101 to use the FOR/IF statements. Both programs obtained the same results.

NATURAL Internals

The Source Code Listing Report may be used to teach programmers how certain NATURAL statements work internally. The following example illustrates how the FOR statement works.

```

PRO0101: Statements shown: Executed, Un-executed and Non-Executable.
2011-06-12 10:28 Profile Session PAYROLL          Object Profiled by USER24

  Execs Total CPU Avg CPU > _____ .....+.....1.....+.....2...  PROG102P Lib PAYTEST
                                0010 DEFINE DATA LOCAL
                                0020 1 #I (P3)
                                0030 END-DEFINE
    100   169.454   1.695   0040 FOR #I 1 99
     99   357.720   3.613   0050   FETCH RETURN 'PROG30P'
     99   187.903   1.898   0060 END-FOR
     1     0.000   0.000   0070 END

PF7 -P   PF8 +P   PF9 BOT  PF10 <   PF11 >   PF12 EXIT   Page 1 of 1

```

Even though the FOR statement specifies 99 iterations of the loop, NATURAL must execute the FOR statement 100 times to verify that it has satisfied the loop criteria. A common programming problem is to define the loop control variable (#I) as "P2" and specify the loop criteria as "#I 1 99". Because #I is incremented each time the FOR loop is processed, #I becomes 100 and a NAT1301 (Intermediate result too large) is received. Using PROFILER to learn how NATURAL statements work internally may eliminate this kind of programming error.

COMPUTE vs. ARITHMETIC

In the following screen excerpts, the Source Code Listing Report shows that the use of the COMPUTE statement was nearly 4 times faster than the use of 4 arithmetic statements to accomplish the same function.

```

PRO0101: Statements shown: Executed, Un-executed and Non-Executable.
2011-06-12 10:29 Profile Session PAYROLL          Object Profiled by USER24

  Execs Total CPU Avg CPU > _____ .....1.....2...  PROG103P Lib PAYTEST
                                0010 DEFINE DATA LOCAL
                                0020 1 #I (P3.2)
                                0030 1 #J (P3.2)
                                0040 1 #K (P3.2)
                                0050 1 #L (P3.2)
                                0060 END-DEFINE
                                0070 *
1          2.655    2.655    0080 ADD #I #I GIVING #J
1          2.011    2.011    0090 SUBTRACT #K FROM #J
1          2.023    2.023    0100 MULTIPLY #J BY #L GIVING #J
1          2.024    2.024    0110 DIVIDE #K INTO #J GIVING #J
                                0120 *
1          0.000    0.000    0130 END

PF7 -P   PF8 +P   PF9 BOT  PF10 <   PF11 >   PF12 EXIT   Page 1 of 1

```

```

PRO0101: Statements shown: Executed, Un-executed and Non-Executable.
2011-06-12 10:30 Profile Session PAYROLL          Object Profiled by USER24

  Execs Total CPU Avg CPU > _____ .....1.....2...  PROG104P Lib PAYTEST
                                0010 DEFINE DATA LOCAL
                                0020 1 #I (P3.2)
                                0030 1 #J (P3.2)
                                0040 1 #K (P3.2)
                                0050 1 #L (P3.2)
                                0060 END-DEFINE
                                0070 *
1          2.019    2.019    0080 COMPUTE #J = (#I + #I - #K)
                                0090 * #L / #K
1          0.000    0.000    0100 *
                                0110 END

PF7 -P   PF8 +P   PF9 BOT  PF10 <   PF11 >   PF12 EXIT   Page 1 of 1

```

FIND (1) vs. READ (1)

The following screen excerpts illustrate the difference between the use of the FIND (1) statement and the READ (1) statement. The programmer wants to locate one record on the EMPLOYEES file that satisfies the search criteria of "BIRTH >= 500101". The programmer codes one program using a FIND (1) statement, one program using a READ (1) statement, and then compares the two programs.

```

PRO0101: Statements shown: Executed, Un-executed and Non-Executable.
2011-06-12 10:31 Profile Session PAYROLL          Object Profiled by USER24

  Execs Total CPU Avg CPU > _____ .....1.....2...  PROG40P  Lib PAYTEST
2    688.704 344.352 0010 FIND (1) EMPLOYEES
<D'base 700.080 350.040>
                                0020 WITH BIRTH >= 500101
1          2.059    2.059    0030 DISPLAY BIRTH
1          0.000    0.000    0040 END-FIND
1          0.000    0.000    0050 DISPLAY *NUMBER (0010)
1          0.000    0.000    0060 END

PF7 -P   PF8 +P   PF9 BOT  PF10 <   PF11 >   PF12 EXIT   Page 1 of 1

```

The FIND (1) statement builds an ISN list of all records that satisfy the search criteria even though only one record is displayed. The number of records placed into the ISN list may be displayed in *NUMBER. The FIND (1) statement then processes the loop statements until it determines that it has satisfied the processing limit.

In this example PROG40P, only one record is displayed in the FIND (1) loop, but *NUMBER contains a value of 1105, indicating that 1,105 records which satisfied the search criteria were found.

The READ (1) statement in PROG41P begins reading with the first record that is equal to or greater than the search criteria and stops reading when it determines that it has satisfied the processing limit of (1). In this example, only one record is displayed in the READ (1) loop and *COUNTER contains a value of 1, indicating that 1 record which satisfied the search criteria was found.

PRO0101: Statements shown: Executed, Un-executed and Non-Executable.					
2011-06-12 10:32 Profile Session PAYROLL				Object Profiled by USER24	
Execs	Total CPU	Avg CPU	>1.....2...	PROG41P Lib PAYTEST
2	63.881	31.941	0010	READ (1)	EMPLOYEES
<D'base	67.952	33.976>			
			0020	WITH BIRTH = 500101	
1	2.059	2.059	0030	DISPLAY BIRTH	
1	0.000	0.000	0040	END-READ	
1	0.000	0.000	0050	DISPLAY *COUNTER (0010)	
1	0.000	0.000	0060	END	
PF7 -P PF8 +P PF9 BOT PF10 < PF11 > PF12 EXIT Page 1 of 1					

PROFILER has illustrated how the FIND (1) and the READ (1) statements work internally. Knowing the difference could save a tremendous amount of CPU Time and Database Elapsed Time in many cases. In this example, use of the READ (1) statement displayed the same record as the FIND (1) statement, but saved 624.823 milliseconds of CPU Time and 632.128 milliseconds of Database Elapsed Time.

SECTION XII

INSTALLATION

XII.1 Introduction to Installation

GENERAL

This section describes the installation process for PROFILER and is divided into two subsections, one for each operating system: OS and VM. PROFILER may be installed in the following environments: z/OS, z/VSE and z/VM.

PROFILER requires no zaps to any operating system, teleprocessing system, or to ADABAS. Use of the NATURAL Review Data Collector Interface (RDC) is required. The Review Data Collector Interface is part of the NATURAL nucleus.

PROFILER will only run under NATURAL versions 4.2.2 and above and can only profile programs that have been cataloged under NATURAL 2.2 and above. PROFILER does not collect statistics on programs cataloged under NATURAL 2.1 and below.

The PROFILER NATURAL modules must reside on the FUSER that contains the object code to be profiled. PROFILER needs to be installed in each NATURAL where profiling will occur.

PROFILER OVERHEAD

There is some overhead when a profiling session is active. This overhead affects only the user who has activated profiling. The overhead is greater when a large number of NATURAL programs are run (this includes maps, programs, etc.). When a user has an active session, PROFILER is invoked every time any NATURAL module is executed by the active user. PROFILER writes collected statistics to an ADABAS file if the executing module matches the defined library/program masks and object types specified by a user at session activation.

A PROFILER assembler program responsible for collecting statistics is invoked by the NATURAL Review Data Collector Interface (RDC). When no Profiling session is active, there is no overhead caused by PROFILER unless the user has requested that RDC statement trace remain active after deactivating a profiling session; then the overhead is approximately 12 instructions per NATURAL statement (not including NATURAL's RDC processing).

BACKGROUND

A PROFILER assembler program accumulates statistics in a GETMAINed area. The assembler program writes records to an ADABAS file used later when running the PROFILER report programs. The statistics are available to any user, not just the user who activated PROFILER.

The maximum number of statements for which statistics may be accumulated for a program is based on the PROFILER GETMAIN size. The default GETMAIN size is 28K, which allows for statistics for 1,080 executable statements. The GETMAIN size may be changed with definition in the PROFILER configuration . GETMAIN values may range from 10K (312 statements) to 50K (2,018 statements).

If the PROFILER Trace Option is to be used (default is **enabled**), PROFILER will perform an additional GETMAIN of 10K per user. This 10K should not be included when calculating the GETMAIN size zap to PROFINIT (described above).

By default, PROFILER acquires its GETMAIN storage from the NATURAL User Buffer (USERBUF). Therefore, unless PROFILER has been configured to acquire its GETMAIN from a different source, NATURAL must be configured such that the User Buffer area (USERBUF) is large enough to accommodate PROFILER's GETMAIN size. PROFILER also requires a NATURAL DATSIZE (local buffer area) of at least 90K and RDSIZE of at least 2.

PROFILER requires a NATURAL DATSIZE (local buffer area) of at least 90K and a DSIZE (debug buffer area) of at least 2K.

PROFILER statistics may be stored on the NATURAL FUSER file where PROFILER is installed or on a separate ADABAS file with the same layout (FDT) as the FUSER. A second ADABAS file is required for PROFILER Enhanced Reporting.

The NATURAL object code to be profiled must reside on the database and NATURAL FUSER file where PROFILER is installed. NATURAL source code is not needed to profile an application. NATURAL source code is needed for the Program Listing Report and Enhanced Program Listing Report, which display source code. NATURAL source code may reside on a different DBID and FUSER than the DBID and FUSER where NATURAL object code is being profiled.

XII.2 Z/OS Installation

PROFILER is distributed on a cartridge with standard labels or via the internet.

Dataset	Description	Contents
PROFILER.Vvrs.SOURCE	Source Library	JCL members and macros*
PROFILER.Vvrs.LOAD	Load Library	Members which must be link- edited with the NATURAL nucleus*
PROFILER.Vvrs.SYSOBJH	SYSOBJH Dataset	PROFILER NATURAL modules
PROFILER.Vvrs.REPDDE	DDE Dataset	PROFILER-REPORTING file DDE
PROFILER.Vvrs.FDT	FDT Dataset	PROFILER-REPORTING file FDT
PROFILER.Vvrs.README	README File	

* Contents listed on the following pages.

The Source Library contains the following:

File	Description
JOSLOAD	Sample NATURAL SYSOBJH load JCL.
JOSDDE	Sample SYSDICBE JCL to load PROFILER-REPORTING file DDE.
JOSCFG	Sample JCL to assemble and link the PROFILER configuration.
PROFCFG	Macro to define the PROFILER configuration
PROFCONF	Default PROFILER configuration
JOSRDC1	Sample JCL to change the internal name of TSIRDC from RDCEX1 to RDCEX2 or RDCEX3.
PROFREP	PROFILER-REPORTING file field descriptions.

The Load Library contains the following:

Please note: the xx at the end of the module names represents the NATURAL version and release.

Member	Description
PROF4Nxx	Profiler module specific for the NATURAL version
TSIRDCxx	Treehouse Interface with NATURAL Review Data Collector.
TSICIADA	Profiler modulesADABAS calling routine for CICS environments.
PROFCONF	Default PROFILER configuration

Installation Steps - OS

The following steps are required for the installation of PROFILER. It is important to use this list as a guide to be sure that the installation is complete. Each step is covered in detail in this section.

1. Load datasets from either a web/email or tape distribution.
2. Apply PROFILER Authorization Zap
3. Specify your PROFILER configuration using the PROFCFG macro (optional)
4. Use the Natural Object Handler Utility (SYSOBJH) to load the PROFILER NATURAL Modules
5. Load the PROFILER-REPORTING File DDE (optional)
6. Establish the ADABAS file Where PROFILER Statistics Will Be Stored
7. Establish the ADABAS file for PROFILER Enhanced Reporting
8. Increase Sizes of Partition or Region and Possibly the NATURAL Thread, and USERBUF
9. Assemble NATPARMs and Relink the NATURAL Nucleus
10. Apply NATURAL Zap(s) for PROFILER Statistics Collection (if necessary)
11. Create Reentrant ADALINKs
12. Recycle CICS and COM-LETE/TPF (for CICS and COM-LETE/TPF Installations only)
13. Verify the PROFILER Installation

XII.2.1 Load Datasets From Either a Web/Email or Tape Distribution

XII.2.1.1 Loading Datasets From a Web or Email Distribution

If installing from a cartridge, skip to **Section XII.2.1.2 Load Datasets From a Tape Distribution**.

You must have an FTP server running on your mainframe and a FTP client running on your PC in order to transfer these files.

PROFILER is distributed as a ZIP file containing the following files:

File	Description
PRF.Vvrs.RN.PDF	PROFILER Release Notes
PRFVvrs.PDF	PROFILER Manual
PRFVvrs.SYSOBJH.XMT	NATURAL source/object code in SYSOBJH format.
PRFVvrs.SRCE.XMT	PDS containing source code for
PRFVvrs.LOAD.XMT	PDS containing load modules for PROFILER
README.TXT	Readme file
PRFVvrs.FDT	ADACMP cards to create an FDT
PRFVvrs.REPDDE	PROFILER-REPORTING DDE

Installation procedure:

Summary of installation procedure:

- 1) Allocate datasets
- 2) Load the source/load and SYSOBJH datasets
- 3) Load the PROFILER-REPORTING FDT
- 4) Load the PROFILER-REPORTING DDE
- 5) PROFILER Installation

Allocate Datasets

Allocate the following datasets:

Dataset	DCB Information
TEMP.PR.F.Vvrs. SRCE	RECFM=FB,LRECL=80,BLKSIZE=3120 SPACE=(CYL,(4,2)),DSORG=PS
TEMP.PR.F.Vvrs. LOAD	RECFM=FB,LRECL=80,BLKSIZE=3120 SPACE=(CYL,(6,3)),DSORG=PS
TEMP.PR.F.Vvrs.SYSOBJH	RECFM=FB,LRECL=80,BLKSIZE=3120 SPACE=(CYL,(2,3)),DSORG=PS
TEMP.PR.F.Vvrs.SYSOBJH.FIX	RECFM=FB,LRECL=80,BLKSIZE=3120 SPACE=(CYL,(2,3)),DSORG=PS (this dataset is only necessary if PRFVvrs.SYSOBJH.FIX.XMT was supplied with your distribution)
TEMP.PR.F.Vvrs.FDT	RECFM=FB,LRECL=80,SPACE=(TRK,(1,1))
PROFILER.Vvrs.ZAPS	RECFM=FB,LRECL=80,BLKSIZE=8000 SPACE=(TRK,(5,3,2)),DSORG=PO
PROFILER.Vvrs.REPDDE	RECFM=FB,LRECL=4624,DSORG=PS, SPACE=(TRK,(20,10))
PROFILER.Vvrs.README	RECFM=FB,LRECL=80,BLKSIZE=8000, SPACE=(CYL,(1,1)),DSORG=PS
PROFILER.Vvrs.FIX.INSTALL	RECFM=FB,LRECL=80,BLKSIZE=8000, SPACE=(CYL,(1,1)),DSORG=PS

Load the Source/Load and NATLOAD Datasets

In BINARY mode transfer file:

PRFVvrs.SRCE.XMT to TEMP.PRF.Vvrs.SRCE
 PRFVvrs.LOAD.XMT to TEMP.PRF.Vvrs.LOAD
 PRFVvrs.SYSOBJH. XMT to TEMP.PRF.Vvrs.SYSOBJH

Once the binary transfers are complete, issue the following commands from the TSO command line:

Command	Restore parameters
receive indataset('temp.prf.Vvrs.srce')	Dsname('profiler.Vvrs.source')
receive indataset('temp.prf. vrs.load')	Dsname('profiler.Vvrs.load')
receive indataset('temp.prf.Vvrs.sysobjh')	Dsname('profiler.Vvrs.SYSOBJH')

Once all datasets have been transmitted and successfully received, the TEMP.* datasets may be deleted.

In ASCII mode, transfer README.TXT to PRF.Vvrs.README and FIX.INSTALL.TXT to PRF.Vvrs.FIX.INSTALL.

Load the PROFILER-REPORTING FDT

Transfer the FDT in ASCII mode to the TEMP.PRF.Vvrs.FDT dataset.

Execute ADACMP on the FDT dataset to convert them to ADABAS ADACMP compressed format. Sample Z/OS(MVS) JCL to convert ADAWAN cards to ADABAS ADACMP compressed format.

```
//PRFINSTL JOB (nnn), 'INSTALL PROFILER',CLASS=A
//*****
//CMPADM EXEC PGM=ADARUN
//STEPLIB DD DISP=SHR,DSN=ADABAS.LOAD <=== ADABAS LOAD
//*
//DDASSOR1 DD DISP=SHR,DSN=ADABAS.ASSOR1 <=== ASSO
//DDDATAR1 DD DISP=SHR,DSN=ADABAS.DATAR1 <=== DATA
//DDTEMPR1 DD DISP=OLD,DSN=ADABAS.TEMPR1 <=== TEMP
//DDSORTR1 DD DISP=OLD,DSN=ADABAS.SORTR1 <=== SORT
//DDDRUCK DD SYSOUT=*
//DDPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//DDEBAND DD DUMMY
//DDAUSBA DD DSN=PROFILER.Vvrs.FDT,
// DISP=(NEW,CATLG,DELETE),
// SPACE=(TRK,(1,1),RLSE),
// UNIT=SYSDA
//DDFEHL DD DUMMY
//DDCARD DD *
ADARUN PROG=ADACMP,MODE=MULTI,SVC=###,DEVICE=8390,DBID=###
/*
//DDKARTE DD DSN=TEMP.PRF.V422.FDT,DISP=SHR
```

Load the PROFILER-REPORTING DDE

Transfer the REPDDE in ASCII mode to the PROFILER.Vvrs.REPDDE dataset.

PROFILER installation

Skip to **Section XII.2.1.3 Install Zaps and Fixes.**

XII.2.1.2 Load datasets from a tape distributionCopy Source, Load and NATURAL Modules

The PROFILER tape volume serial number is PRFvrs. The tape contains nine datasets, as indicated below, to assist in the installation and operation of PROFILER.

Location Library	Library Dataset	Member Contents	Generated Format	Storage Requirements
1st dataset on tape PROFILER.Vvrs.SOURCE	Source Library	JCL members*	IEBCOPY format	1 cylinder 3390 disk space, or equivalent
2nd dataset on tape PROFILER.vrs .LOAD	Load Library	Members which must be link-edited with the NATURAL nucleus*	IEBCOPY format	3 cylinders 3390 disk space, or equivalent
3rd dataset on tape PROFILER.VvrsVvrs.SYSOBJH	SYSOBJH Dataset	PROFILER NATURAL modules	NATURAL SYSOBJH format	5 cylinders 3390 disk space, or equivalent
4th dataset on tape PROFILER.Vvrs.REPDDE	DDE Dataset	PROFILER-REPORTING file DDE	PREDICT SYSDICBE UNLOAD format	1 cylinder 3390 disk space, or equivalent
5th dataset on tape PROFILER.Vvrs.FDT	FDT Dataset	PROFILER-REPORTING file FDT	ADABAS ADACMP format	1 cylinder 3390 disk space, or equivalent
6th dataset on tape PROFILER.Vvrs.README	README File		IEBGENER format	10 tracks 3390 disk space, or equivalent
7th dataset on tape PROFILER.Vvrs.FIX.INSTALL	Fix Installation Instructions	Fix Installation Instructions	IEBGENER	10 tracks 3390 disk space, or equivalent

Copy the PROFILER datasets from the RELEASE tape to disk using the IEBCOPY utility. The following is sample OS JCL:

```
//COPYJOB      JOB      ...
//COPYALL     EXEC     PGM=IEBCOPY
//SYSPRINT    DD      SYSOUT=A
//SRCET DD     DSN=PROFILER.VVVS.SOURCE,DISP=(OLD,PASS),
//              UNIT=TAPE,VOL=SER= PRVVVS,
//              LABEL=(1,SL)
//*
//LOADT DD     DSN=PROFILER.VVVS.LOAD,DISP=(OLD,PASS),
//              UNIT=TAPE,VOL=SER= PRVVVS,
//              LABEL=(2,SL)
//*
//SYSOBJHT    DD     DSN=PROFILER.VVVS.SYSOBJHT,DISP=(OLD,PASS),
//              UNIT=TAPE,VOL=SER= PRVVVS,
//              LABEL=(3,SL)
//*
//SRCE DD     DSN=PROFILER.VVVS.SOURCE,DISP=(NEW,CATLG),
//              UNIT=uuuu,VOL=SER=vvvvvv,
//              SPACE=(CYL,(1,,3)),
//              DCB=(RECFM=FB,LRECL=80,BLKSIZE=6000)
//*
//LOAD DD     DSN=PROFILER.VVVS.LOAD,DISP=(NEW,CATLG),
//              UNIT=uuuu,VOL=SER=vvvvvv,
//              SPACE=(CYL,(3,,3)),
//              DCB=(RECFM=U,BLKSIZE=6447)
//*
//SYSOBJH     DD     DSN=PROFILER.VVVS.SYSOBJH,DISP=(NEW,CATLG),
//              UNIT=uuuu,VOL=SER=vvvvvv,
//              SPACE=(CYL,(5,5)),
//              DCB=(LRECL=256,RECFM=PS,BLKSIZE=2564)
//*
//SYSIN DD     *
COPY  INDD=SRCET,OUTDD=SRCE31
COPY  INDD=LOADT,OUTDD=LOAD31
COPY  INDD=SYSOBJHT,OUTDD=SYSOBJH
//*
```

Where: uuuu DASD device type
vvvvvv Volume/serial number for PROFILER libraries

Copy the following datasets from the RELEASE tape to disk using the IEBGENER utility. The following is sample OS JCL:

```
//COPYJOB      JOB
//LODDDE      EXEC PGM=IEBGENER
//SYSPRINT    DD SYSOUT=*
//*
//SYSUT1     DD DISP=SHR,UNIT=TAPE,DISP=SHR,LABEL=(8,SL),
//            VOL=SER=PRFvrs,DSN=PROFILER.Vvrs.REPDDE
//*
//SYSUT2     DD DSN=PROFILER.Vvrs.REPDDE,
//            DISP=(NEW,CATLG,DELETE),SPACE=(TRK,(1,15)),
//            UNIT=uuuu,VOL=SER=vvvvvv
//SYSIN      DD DUMMY
//*
//LODFDT     EXEC PGM=IEBGENER
//SYSPRINT    DD SYSOUT=*
//*
//SYSUT1     DD DISP=SHR,UNIT=TAPE,DISP=SHR,LABEL=(9,SL),
//            VOL=SER=PRFvrs,DSN=PROFILER.Vvrs.FDT
//*
//SYSUT2     DD DSN=PROFILER.Vvrs.FDT,
//            DISP=(NEW,CATLG,DELETE),SPACE=(TRK,(1,15)),
//            UNIT=uuuu,VOL=SER=vvvvvv
//SYSIN      DD DUMMY
//*
//LODRM      EXEC PGM=IEBGENER
//SYSPRINT    DD SYSOUT
//*
//SYSUT1     DD DISP=SHR,UNIT=TAPE,DISP=SHR,LABEL=(10,SL),
//            VOL=SER= PRFvrs,DSN=PROFILER.Vvrs.README
//*
//SYSUT2     DD DSN=PROFILER.Vvrs.README,
//            DISP=(NEW,CATLG,DELETE),SPACE=(TRK,(1,15)),
//            UNIT=uuuu,VOL=SER=vvvvvv
//SYSIN      DD DUMMY
//*
//LODFI      EXEC PGM=IEBGENER
//SYSPRINT    DD SYSOUT=*
//*
//SYSUT1     DD DISP=SHR,UNIT=TAPE,DISP=SHR,LABEL=(11,SL),
//            VOL=SER= PRFvrs,DSN=PROFILER.Vvrs.FIX.INSTALL
//*
//SYSUT2     DD DSN=PROFILER.Vvrs.FIX.INSTALL,
//            DISP=(NEW,CATLG,DELETE),SPACE=(TRK,(1,15)),
//            UNIT=uuuu,VOL=SER=vvvvvv
//SYSIN      DD DUMMY
//*
```

Where: uuuu DASD device type
 vvvvvv Volume/serial number for PROFILER libraries

XII.2.1.3 Install zaps and fixes

This distribution of PROFILER may contain additional zaps and fixes that were created after the initial release date. Before continuing the installation, please refer to the PRF.Vvrs.README for a description of any zaps/fixes available. The PRF.Vvrs.FIX.INSTALL dataset contains information on installing these fixes.

XII.2.2 Apply PROFILER Authorization Zap

Before proceeding with the PROFILER installation, apply a zap to set the expiration date for a trial or a sold site. This zap is necessary to run PROFILER. All zaps supplied must be applied to load modules and are necessary for the proper functioning of PROFILER.

XII.2.3 Define your PROFILER configuration using the PROFCFG macro

PROFCFG WORKBUFF=USERBUF,	X
WORKSIZE=28,	X
TRACE=YES,	X
ADALNK=ADALNKR,	X
CICSLNK=ADACICST	

Note:

The PROFCONF load module delivered was generated with the defaults shown above.

WORKBUFF WORK BUFFER LOCATION
 USERBUF
 ZSIZE
 GETMAIN (NON-CICS ONLY)

By default, PROFILER GETMAINS its statistics accumulation area from the NATURAL User Buffer (USERBUF). If the default installation of PROFILER interferes with the use of the NATURAL User Buffer by another system installed under NATURAL, WORKBUFF=ZSIZE should be defined.

PROFILER can also be configured to allocate its statistics gathering area outside of NATURAL as a separate GETMAIN, **unless PROFILER is running under CICS**.

To configure PROFILER to allocate its statistics gathering buffer outside of NATURAL as a separate GETMAIN set WORKBUFF=GETMAIN.

WORKSIZE SIZE OF WORK BUFFER (IN K)
 28

The maximum number of statements for which statistics may be accumulated for a program is based on the PROFILER GETMAIN size. The default GETMAIN size is 28K, which allows for the accumulation of statistics for 1,080 executable statements per object. The GETMAIN size may be changed adjusting WORKSIZE. The values may range from 10 (312 statements) to approximately 50 (2,018 statements).

TRACE	TRACE SUBSYSTEM USAGE YES NO SAVE APPROX 10K. OF STORAGE PROFILER requires an additional 10K of GETMAIN storage per user in order to operate the Trace Subsystem. If the Trace Subsystem is NOT going to be used at a site, define TRACE=NO.
ADALNK	Name of the reentrant ADALNK created for PROFILER ADALNKR
CICSLINK	NAME OF ADABAS LINK ROUTINE FOR CICS ADACICS Must match the value of the ENTPT keyword of the LGBLSET definitions for CICS.
UBS	SIZE OF USER BUFFER FOR ADABAS CALLS Only needed if a V7 ADALNK is in use!!! 512 In order to separate the User-ID for PROFILER transactions from the User-ID of the application, PROFILER provides its own ADABAS UB block, including the USERINFO area. This data is generally included in the statistics accumulation GETMAIN area. PROFILER allows 512 bytes for the UB block, which includes 112 bytes for the MODIFIED DSECT, 64 bytes for the UB, and 336 bytes for the USERINFO area. If a USERINFO area of more than 336 bytes is required, apply a zap to PROFINIT to cause a GETMAIN to be performed for the UB area. The value should be large enough to accommodate the MODIFIED DSECT, the UB area, and the USERINFO area.

All other keywords of the PROFCFG macro must only be modified if requested by Treehouse support.

Sample JCL for the assembly and link of PROFCONF is provided in member JOSCFG.

XII.2.4 NATLOAD the PROFILER NATURAL Modules

The PROFILER NATURAL modules must be installed under NATURAL 4.2.2 or above.

If NATURAL SECURITY SYSTEM (NSS) is installed, define the library PROLIB. PROLIB should be defined with the PREDICT XREF feature set to OFF. The PROFILER NATLOAD dataset contains no XREF data.

Execute the NATURAL LOAD program to load the PROFILER NATURAL to the desired FUSER file on each database where PROFILER will be run. The NATLOAD process loads programs into the following NATURAL libraries:

PROLIB
SYSTEM

Source and object code for the program 'PRF' are loaded in library SYSTEM. Source and object code for the programs 'PRFUEUID', 'PRFUE01P' and 'PRFUE99P' are loaded in library PROLIB. The source code for these programs (PRFUEUID, PRFUE01P and PRFUE99P) may be modified. All other programs, which are object code only, are loaded into library PROLIB.

The following is sample JCL (located in "PROFILER.Vvrs.SOURCE(JOSLOAD)") to load NATURAL modules for PROFILER.

```
//* LOAD PROFILER NATURAL MODULES
/*
//SYSOBJH EXEC NATURAL
//CMWKF01 DD DSN=PROFILER.Vvrs.SYSOBJH,DISP=SHR
//CMPRINT DD SYSOUT=*
//DDCARD DD DISP=SHR,DSN=... (...)
//CMSYNIN DD *
SYSOBJH
LOAD * LIB PROLIB
FIN
```

The sample JCL assumes a cataloged procedure named "NATURAL" exists in the system to invoke NATURAL. This JCL is provided for reference only and will not execute as displayed.

XII.2.4.1 Copy the PROFILER NATURAL Library (Optional)

NOTE: This is an optional step. If PROFILER will be executed from the PROLIB library, skip to the **SYSMAIN Module from SYSEXT to SYSTEM** section.

PROFILER can be executed from any library starting with PROLIB. By default, PROFILER is installed in the PROLIB library. It may be necessary to run PROFILER from a different library if sharing an FUSER between different versions of NATURAL. In this case, it is necessary to create one or more additional libraries (one for each version of NATURAL using the FUSER). The instructions below explain how to move PROFILER.

- 1) Using SYSMAIN copy the entire contents of the PROLIB library to a new library. The new library name must start with PROLIB.
- 2) Using SYSMAIN, copy the USR subprograms below from SYSEXT (on the FNAT) to the library created in step 1. The FNAT used must be the same version as the NATURAL nucleus that will be used with this library.

USR0010N
USR0050N
USR1025N
USR2004N

- 3) Skip to the **Installing the LOGON Front-End** section

XII.2.7.2 SYSMAIN Modules from SYSEXT to SYSTEM

PROFILER uses several standard Software AG utility 'USR' subprograms. If not already in library SYSTEM, the following NATURAL modules need to be SYSMAIN copied from NATURAL library SYSEXT (in the FNAT file) to library SYSTEM (in the FUSER file):

USR0010N
USR0050N
USR1025N
USR2004N

XII.2.4.3 Installing the LOGON Front-End

Sites where NATURAL Security is not installed:

PROFILER uses a front-end to the LOGON command so that it may activate or deactivate sessions defined by the Background Monitoring facility. The following steps are necessary to install the LOGON front-end:

1. Using SYSMAIN, copy program PRFLOGON and subprograms PRFBCKN and PRFUEUID from PROLIB library to SYSLIB library on the FNAT file.
2. Using SYSMAIN, rename the LOGON program to XLOGON in SYSLIB library on the FNAT file.
3. Using SYSMAIN, rename PRFLOGON to LOGON in SYSLIB library on the FNAT file.
4. Copy USR0050N from the SYSEXT library on the FNAT to the SYSLIB library on the FNAT.

When the user enters the LOGON command, PROFILER's LOGON front-end receives control, determines whether a session should be activated or deactivated based on Background Monitoring, performs the activation or deactivation if necessary, then FETCHes the Software AG LOGON program.

Sites Where NATURAL Security is Installed:

At a site where NATURAL Security is installed, PROFILER uses subprogram PRFBCKN at LOGON time to activate and deactivate sessions defined by the Background Monitoring function. The following steps are necessary to install this subprogram:

1. Using SYSMAIN, copy subprograms PRFBCKN and PRFUEUID from PROLIB library to SYSLIB library on the FNAT file.
2. Copy subprogram USR0050N from SYSEXT library (on the FNAT file) to SYSLIB library on the FNAT file.
3. Include a CALLNAT to PRFBCKN in the NATURAL Security user-exit LOGONEX1 after the INPUT statement. PRFBCKN requires the application name, #I-APPL, which has been passed to LOGONEX1 as a parameter. The CALLNAT should be coded as follows:

```
IF #I-APPL NE ' '  
    CALLNAT 'PRFBCKN' #I-APPL  
END-IF
```

When the user enters the LOGON command, subprogram PRFBCKN receives control, determines whether a session should be activated or deactivated based on Background Monitoring, and performs the activation or deactivation if necessary.

XII.2.4.4 Installing the FIN Front-End

PROFILER uses a front-end to the FIN command to deactivate any session that is active (whether through Background Monitoring or regular activation) before the user leaves NATURAL. The following steps are necessary to install the FIN front-end:

1. Modify the control and execute entries for the FIN command in NATTEXT (refer to the NATURAL DBA Guide or the Installation And Operations Manual) and NATTXT2, renaming the FIN command to XFIN for both the Control command and the Execute command.
2. Use SYSMAIN to rename PRFFIN (located in PROLIB library) to FIN, and move the renamed module from PROLIB library to SYSTEM library on the FUSER file.

When the user enters the FIN command, PROFILER's FIN front-end receives control, performs the steps necessary to deactivate an open session, then STACKs the Software AG FIN program.

XII.2.4.5 User-Exits

PROFILER provides a user-exit facility to allow control to be passed to a user-written NATURAL program.

Source code for the user-exit routines may be found in the PROLIB library. The user-exit routines must be STOWed in library PROLIB or in a STEPLIB accessible from PROLIB. The following are the available user-exits:

PROFILER Termination Exit - PRFUE99P

This user-exit program is invoked during the termination of PROFILER. To exit from PROFILER, the user may press PF12 from most screens or PF3 from the PROFILER Main Menu. This user-exit may be modified to provide an alternative exit procedure.

PROFILER Customized Reports Exit - PRFUE01P

This user-exit program is invoked when selecting the Customize Reports function from the Enhanced Reporting Menu. A site may use this user-exit to access site-specific customized reports. Customized reports may be written using the PROFILER-REPORTING file, which is included on the PROFILER Installation tape. A description of the fields on this file are also included on the Installation tape.

PROFILER Customized Background Monitoring User-Id Exit – PRFUEUID

This user-exit program is invoked when background monitoring obtains a user-id. A unique site-specific user-id may be set by this exit; however, use of this exit is not typical and the default of *INIT-USER is appropriate for most sites.

XII.2.4.6 Step Library changes where NATURAL SECURITY is Installed

Where NATURAL SECURITY is installed:

1. Move program PRF from SYSTEM library to PROLIB library.
2. Make PROLIB a step library of any library that will be using PROFILER.

Note: If People Protected libraries are used, the step library on the Special Link must include PROLIB.

3. Ensure program PRF is NOT defined as the automatic startup program for any of these libraries.

XII.2.5 Load the PROFILER-REPORTING File DDE (optional)

Loading the DDE for the PROFILER-REPORTING file into PREDICT is an optional step, necessary only if the site intends to write custom reports utilizing statistics saved on the PROFILER-REPORTING file. This DDE is not required by PROFILER.

Execute the SYSDICBE LOAD function to load the DDE for the PROFILER-REPORTING file from the fourth file on the tape into PREDICT.

Sample JCL (located in "PROFILER.Vvrs. SOURCE(JOSDDE)") to load the PROFILER-REPORTING file DDE follows:

```

// *LOAD PROFILER-REPORTING DDE
// *
//LOAD EXEC NATURAL, PARM= 'IM=D, MT=0, MADIO=0'
//CMWKFO1 DD DSN=PROFILER.Vvrs.REPDDE, VOL=SER=PRFvrs, DISP=OLD,
// UNIT=TAPE, LABEL=(8, SL)
//CMSYNIN DD *
LOGON SYSDICBE
MENU
LOAD ALL, REPLACE=Y
FIN
/ *

```

The sample JCL assumes a cataloged procedure named "NATURAL" exists in the system to invoke NATURAL. This JCL is provided for reference only and **will not** execute as displayed.

Once the DDE has been loaded into PREDICT, users may generate a DDM which can be referenced in their custom report programs.

Note: Care must be taken to ensure that custom report programs written by the site do not modify the statistics stored in the PROFILER-REPORTING file. Modification of the statistics in the PROFILER-REPORTING file could corrupt the integrity of PROFILER Enhanced Reports and result in unexpected errors.

XII.2.6 Establish the ADABAS file Where PROFILER Statistics Will Be Stored

PROFILER statistics may be stored on the NATURAL FUSER file where PROFILER is installed or on a separate ADABAS file with the same layout (FDT) as the FUSER file.

Select one of the following options:

NATURAL's FUSER File

Edit the NATPARM module for the NATURAL where PROFILER will be installed. Insert one NTFILE macro invocation after the last line in the NTPARM macro but before the END statement. Code the NTFILE invocation as follows:

NATURAL 4.2.2 and above

```
NTLFILE 120,xxx,yyy           FUSER file
```

Where:

xxx = The DBID of the FUSER file
 yyy = The file number of the FUSER file

Note: Logical file ID 120 is used internally by PROFILER. It will not affect any file number 120 on the database.

Separate ADABAS File

Create an ADABAS file with the same layout (FDT) as the NATURAL FUSER file where PROFILER is installed. The file should be loaded empty. The following space estimates represent a PROFILER system managing approximately 5,000 profiled programs. The space required varies based on the number of profiling sessions, the number of programs profiled in each session, and the size of each program.

PROFILER FILE	COMMENTS	SPACE EST.
PROFILER-FILE	Any unused ADABAS file number may be used	DSSIZE=38 (cylinders) UISIZE=10B NISIZE=20B

Edit the NATPARM module for the NATURAL where PROFILER is installed. Insert one NTFILE macro invocation after the last line in the NTPARM macro but before the END statement. Code the NTFILE invocation as follows:

NATURAL 4.2.2 and above

```
NTLFILE 120,xxx,yyy           PROFILER-FILE
```

Where:

xxx = The DBID of the PROFILER-FILE
 yyy = The file number of the PROFILER-FILE

Note: Logical File ID 120 is used internally by PROFILER. It will not affect any file number 120 on the database.

XII.2.7 Establish the ADABAS file For PROFILER Enhanced Reporting

PROFILER statistics may be moved to the PROFILER-REPORTING file which is a separate ADABAS file. This file should only be used by PROFILER.

The seventh file on the installation tape contains the ADABAS ADACMP compressed format dataset for the PROFILER-REPORTING file. This dataset may be input into the ADABAS ADALOD utility directly. Optionally, the ADAWAN cards may be generated from the PREDICT DDE created earlier (refer to **Load the PROFILER-REPORTING File DDE** sub-section), and then 'punched' to the appropriate dataset. The file may then be loaded empty.

The following space estimates represent a PROFILER Enhanced Reporting system managing approximately 5000 profiled programs. The space required varies based on the number of profiling sessions, the number of programs profiled in each session, and the size of each program.

PROFILER FILE	COMMENTS	SPACE EST.
PROFILER-REPORTING	Any unused ADABAS file number may be used	DSSIZE=38 (cylinders) UISIZE=10B NISIZE=20B

Edit the NATPARM module for the NATURAL where PROFILER is installed. Insert one NTFILE macro invocation after the last line in the NTPARM macro but before the END statement. Code the NTFILE invocation as follows:

NATURAL 4.2.2 and above

NTLFILE 122,xxx,yyy

PROFILER-REPORTING

Where:

xxx The DBID of the PROFILER-REPORTING
yyy The file number of the PROFILER-REPORTING

Note: Logical ID 122 is used internally by PROFILER. It does not affect any file number 122 on the database.

XII.2.8 Increase Sizes of Partition or Region and Possibly the NATURAL Thread, and USERBUF

Additional storage is required for NATURAL with PROFILER. PROFILER's GETMAIN size affects NATURAL storage requirements. For more information about the PROFILER GETMAIN size, refer to the **Apply GETMAIN Zap to PROFINIT** sub-section.

Increase USERBUF Size

It may be necessary to increase the USERBUF parameter. By default, PROFILER acquires its GETMAIN storage from the NATURAL User Buffer, unless PROFILER has been configured to acquire the storage from a different location. For more information about the USERBUF parameter, refer to the **Assemble NATPARMS and Relink the NATURAL Nucleus** sub-section.

Increase Size of Partition or Region for Batch/TSO

The additional storage requirements for NATURAL are approximately 15K for the PROFILER assembler programs, and additional storage based on the GETMAIN size for each profiling session.

Increase Thread Size for CICS

The additional storage requirements for NATURAL are approximately 15K for the PROFILER assembler programs, and additional storage based on the GETMAIN size for each profiling session. It may be necessary to increase the NATURAL CICS thread size. Use the NATURAL BUS utility to monitor thread usage and to reclaim unused space from other buffers.

XII.2.9 Assemble NATPARMs and Relink the NATURAL Nucleus

Assemble the NATPARMs:

Assemble NATPARM, including the following statements, along with any statements currently included in the existing NATPARM module.

CSTATIC=(PROFINIT,PROFCLOS,PROFDRV)

USERBUF=GETMAIN SIZE + 10K (if Trace is to be used) + 1K

(This parameter is required unless PROFILER has been configured to acquire GETMAIN storage from a source other than the NATURAL USERBUF. USERBUF must be at least equal to the GETMAIN size + 10K (for Trace) + 1K, but may be set to a larger value.)

DATSIZE=90

(Minimum required for running PROFILER)

RDCSIZE=2

(Enable NATURAL RDC)

NTRACE NATPROX

(Enable NATURAL RDC statement tracing)

For more information about GETMAIN size, refer to the Apply GETMAIN Zap to PROFINIT sub-section.

<p>Note: When assembling the NATPARM module with the USERBUF parameter specified, the following message will appear in the listing:</p> <p>USERBUF PARAMETER IS NO LONGER USED FOR THE DATA COLLECTOR. PLEASE USE RDCSIZE INSTEAD.</p> <p>This message may be ignored.</p>

<p>Note: The USERBUF parameter is required unless PROFILER has been configured to acquire its GETMAIN size from a source other than the NATURAL user buffer.</p>

If the NATPARMs are link edited by themselves, a non-zero condition code is received on the link-edit step with unresolved references for PROFINIT, PROFCLOS, and PROFDRV.

<p>Note: The definition of ITRACE=ON is no longer needed. Customers upgrading from previous Profiler versions should remove this specification to avoid unneeded overhead.</p>

Relink the NATURAL Nucleus

Include the following statement with the dataset names:

```
//PROFLIB DD DISP=SHR,DSN=PROFILER.Vvrs.LOAD
```

Include the following statements at the end of the INCLUDEs before NATLAST:

```

INCLUDE PROFLIB(PROF4Nnn)      where nn represents the NATURAL version
INCLUDE PROFLIB(TSIRDCnn)
INCLUDE PROFLIB(PROFCONF)      (Configuration may be customized)
INCLUDE PROFLIB(TSICIADA)      for CICS environment only
INCLUDE CICS LIB(DFHEAI)        ditto
INCLUDE CICS LIB(DFHEAI0)      ditto

```

Note: If a site is using a shared NATURAL nucleus, PROF4Nnn, PROFCONF, and TSIRDCnn must be included in the shared portion. If Profiler is installed under CICS with a shared nucleus, modules TSICIADA (from the PROFILER load library), DFHEAI, and DFHEAI0 (both from the CICS load library) must also be included in the shared portion.

Note: PROFILER uses the NATURAL Review Data Collector Interface (RDC). PROFILER module TSIRDC is configured with an internal name of RDCEX1 so that NATURAL will call the PROFILER statistics gathering subsystem as RDC Exit 1. TSIRDC can optionally be relinked to change the internal name from RDCEX1 to RDCEX2 or RDCEX3 (see member JOSRDC1 on the PROFILER source library) if the name RDCEX1 interferes with the installation of another RDC based product. If the following warning is displayed in the link output, JOSRDC1 should be used: IEW2480W A711 EXTERNAL SYMBOL RDCEX1 OF TYPE LD WAS ALREADY DEFINED AS A SYMBOL OF TYPE LD IN SECTION *module_name*.

XII.2.10 Considerations of Using the NATURAL RDC Interface

- Required for PROFILER
- Interfaces with the NATURAL Review Data Collector feature
- Requires NATPARM parameter NTTRACE NATPROX to enable RDC statement tracing, and RDCSIZE of at least 2

XII.2.11 Create Reentrant ADALINKs

PROFILER maintains statistics through an independent link to the ADABAS address space by issuing an ADABAS OPEN command with a newly-generated and unique ADABAS User-ID. This separate connection needs to be established because the transaction logic used by PROFILER to maintain statistics would normally conflict with that being used in the NATURAL application being profiled.

Creating a Reentrant ADALNKR for Batch/TSO and COM-LETE/TPF is documented in the Software AG manual for ADABAS under the chapter "Installing ADABAS with TP monitors".

<p>Note: If Treehouse Software, Inc.'s SECURITRE is installed, ADALNKR must be linked with the current link exits (TSIUEX1G / TSIUEX2G).</p>

Under CICS, the ADABAS calls from PROFILER are routed through the special purpose module TSICIADA, which calls the normal ADABAS 8 CICS link routine modules. If the LGBLSET definitions for the CISGBL are not using the default value ENTPT=ADACICS the configuration for PROFILER has to be adjusted so that the value for the keyword CICSLINK= in PROFCFG define the same value as ENTPT= in LGBLSET:

XII.2.12 Recycle CICS and COM-LETE/TPF (for CICS and COM-LETE/TPF Installation only)

Recycle CICS and COM-LETE/TPF to begin using PROFILER in these respective environments.

XII.2.13 Verify the PROFILER Installation

PROFILER includes a special program (PRFVRFY), which verifies that PROFILER is properly installed. PRFVRFY ensures that NATURAL is properly configured for PROFILER, that the PROFILER modules are in place, that the proper statistics file is installed and accessible, and that the PROFILER Statistics Collection zap(s) are in place. It then simulates a profiling session, verifying that statistics can be collected and stored in the statistics file. It also provides diagnostic messages, which identify possible installation problems and suggest how to resolve them.

To verify that PROFILER is installed correctly, invoke NATURAL and LOGON to PROLIB. Then type PRFVRFY and press Enter. PRFVRFY checks the installation and displays an error report or the message "No errors have been encountered in the PROFILER installation.". When PRFVRFY detects an installation error, a screen similar to the following example is displayed.

```

2011-11-20 11:38 PROFILER Installation Verification USER24  PROLIB
The following errors have been found in the PROFILER installation:

PRO0014: PROFILER requires NATURAL V4.2.2 or above. Contact System
Administrator.
PRO0015: NTFILE Macro NOT included in NATPARM Module. Contact System Admin.

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10---PF11--PF12-
--
Help          End                                Exit

```

PRFVRFY uses a profiling session to verify the statistics collection process. The session used has an internal session number of 63. If this session is already in use at a site, PRFVRFY displays message PRO0128 or PRO0129. A site should delete this session to continue the installation verification process.

These are the errors that can be detected during installation verification:

```

PRO0014: PROFILER requires NATURAL V4.2.2 or above. Contact System Administrator.
PRO0015: NTFILE Macro NOT included in NATPARM Module. Contact System Admin.
PRO0017: PROFILER for NATURAL Trial EXPIRED. Contact System Administrator.
PRO0018: PROFILER for NATURAL License is INVALID. Contact System Administrator.
PRO0019: Response ... for "PROFINIT" from {object}. Contact System Admin.
PRO0036: Response ... for "TREPREF1" from {object}. Contact System Admin.
PRO0041: Response ... for "PROFCLOS" from {object}. Contact System Admin.
PRO0119: PRFVRFY must be run from Library "PROLIB".
PRO0120: PROFILER Assembler Modules are NOT Version 4.3.1
PRO0121: Module PROFCLOS or PROFDRV is NOT Linked with NATURAL.
PRO0122: Unable to Collect Statistics. Reason Unknown.
PRO0124: Wrong Operating System Driver linked with NATURAL.
PRO0125: PROFCLOS, PROFDRV, or CMDBGEX were NOT found in CSTATIC List.
PRO0126: PROFILER "ZAP" NOT applied to NATURAL.
PRO0127: Warning: Unable to Verify that PROFILER "ZAP" has been applied to NATURAL.
PRO0128: Session {63} is already ACTIVE. Verify CANNOT continue.
PRO0129: Session {63} already has Statistics. Verify CANNOT continue.
PRO0130: Unable to obtain ACTIVE Session Information.

```

For more information on installation error codes, refer to **Appendix A, PROFILER Messages**.

XII.3 VM (SP, XA, ESA) Installation

PROFILER is distributed on a 3490 cartridge in an unlabeled tape format.

The tape contains five datasets. The first dataset contains both source and object (TEXT) files in VM (SP, XA, ESA) TAPE DUMP format for PROFILER. The following files are included:

Member	Type	Description
JVMLOAD	EXEC	Sample NATLOAD exec.
JVMDDDE	EXEC	Sample SYSDICBE exec to load PROFILER-REPORTING file DDE.
JVMGET1	NATZAP	Sample zap to change the PROFILER default GETMAIN area.
JVMGET2	NATZAP	Sample zap to increase the statistics accumulation GETMAIN size.
JVMGET3	NATZAP	Sample zap to increase the UB area GETMAIN size.
JVMTRA1	NATZAP	Sample zap to disable the PROFILER Trace option.
PROFREP	DOC	PROFILER-REPORTING file field descriptions.
PROFCLOS	TEXT	Deactivates profiling session for user.
PROFDRV	TEXT	Supports operating system-specific functions, such as obtaining and freeing storage.
PROFINIT	TEXT	Sets up profiling session, activates profiling session, and communicates with NATURAL programs.
PROFRDC	TEXT	Accumulates statistics during an active profiling session.
TSIRDC	TEXT	Treehouse Software interface program for NATURAL's Review Data Collector exit.
PROFVDRV	TEXT	Supports VM operating system-specific functions.
PROTRACE	TEXT	Accumulates Trace statements during an active Trace session.

The third dataset contains the NATLOAD for the NATURAL modules in NATUNLD format. If the dataset is to be placed onto disk, it requires 6 cylinders of 3380 disk space.

The fourth dataset contains the DDE dataset for the PROFILER-REPORTING file in PREDICT SYSDICBE UNLOAD format. If the DDE dataset is to be placed onto disk, it requires 1 cylinder of 3380 disk space.

The fifth dataset contains the FDT dataset for the PROFILER-REPORTING file in ADABAS ADACMP format. If the FDT dataset is to be placed onto disk, it requires 1 cylinder of 3380 disk space.

Installation Steps - VM

The following steps are required for the installation of PROFILER. It is important to use this list as a guide to be sure that the installation is complete. Each step is covered in detail in this section.

1. Allocate CMS Mini-disk Space
2. Load to Allocated Mini-disk
3. Apply PROFILER Authorization Zap
4. Apply GETMAIN Zap to PROFINIT (optional)
5. Trace Subsystem Installation (optional)
By default the PROFILER Trace option is enabled.
6. Apply TSIRDC TWA Zap
7. NATLOAD the PROFILER NATURAL Modules
8. Load the PROFILER-REPORTING File DDE (optional)
9. Establish the ADABAS file Where PROFILER Statistics Will Be Stored
10. Establish the ADABAS file for PROFILER Enhanced Reporting
11. Change and Assemble NATPARMs, Apply Zap(s) for PROFILER Statistics Collection, and Relink NATURAL Executable Modules
12. Apply Zap(s) for PROFILER Statistics Collection
13. Increase Size of the Virtual Machine
14. Verify the PROFILER Installation

XII.3.1 Allocate CMS Mini-disk Space

Installation of PROFILER requires the equivalent of a two-cylinder 3380 mini-disk formatted in 1024 byte blocks.

For FBA allocation, use 2,000 (512K) blocks.

Sample CMS Command to format a mini-disk:

```
FORMAT 301 f (where "301" is the address and "f" is the disk file mode)
```

XII.3.2 Load to Allocated Mini-disk

Load the required files from the release tape to the assigned mini-disk.

Sample CMS Command to load the PROFILER files for NATURAL 3.1.6 to disk:

```
TAPE LOAD * * f EOF 1 (where "f" is the disk file mode)
```

Sample CMS Command to load the PROFILER files for NATURAL 4.1 to disk:

```
TAPE FSF 1
```

```
TAPE LOAD * * f EOF 2 (where "f" is the disk file mode)
```

XII.3.3 Apply PROFILER Authorization Zap

Before proceeding with the PROFILER installation, apply a zap to set the expiration date for a trial site or a zap to set a sold site. This zap is necessary to run PROFILER. There may also be corrections or fixes that must be applied in the form of zaps to the distributed code. These zaps are supplied by Treehouse Software and its affiliates via zap letters, which are shipped with the distribution tape. Copy the zap statements, exactly as given, into a file with fixed-length 80-byte records and a file type of "NATZAP". Then, to apply the zap, use the command "NATZAP fm (INPUT filename NOAUTO)", where fm is the file mode of the module being zapped.

XII.3.4 Apply GETMAIN Zap to PROFINIT (optional)

By default, PROFILER allocates its statistics gathering area from the NATURAL User Buffer (USERBUF). This can be changed with an optional zap (see member JVMGET1 on the installation tape).

(Optional) Apply the zap to increase or decrease the statistics accumulation GETMAIN size

The maximum number of statements for which statistics may be accumulated for a program is based on the GETMAIN size. The default GETMAIN size is 28K, which allows for the accumulation of statistics for 1,080 executable statements. The GETMAIN size may be changed with a zap to PROFINIT. GETMAIN values may range from 10K (312 statements) to 50K (2,019 statements).

If a GETMAIN size other than 28K is desired for statistics accumulation, modify the REP value in the JVMGET2 NATZAP file to the hexadecimal representation of the desired GETMAIN size. For example, if 50K is desired, use C800 as the REP value as shown. If 10K is desired, use 2800 as the REP value (provided as sample zap JVMGET2):

```
NAME PROFINIT PROFINIT
VER 007A 7000
REP 007A C800
```

(Optional) Apply the zap to increase the UB area GETMAIN size

This step is only necessary if the site has other products that require space in the USERINFO area.

In order to separate the User-ID for PROFILER transactions from the User-ID of the application, PROFILER provides its own ADABAS UB block, including the USERINFO area. This data is generally included in the statistics accumulation GETMAIN area.

PROFILER allows 512 bytes for the UB block, which includes 112 bytes for the MODIFIED DSECT, 64 bytes for the UB, and 336 bytes for the USERINFO area. If a USERINFO area of more than 336 bytes is required, apply a zap to PROFINIT to cause a GETMAIN to be performed for the UB area. The REP value should be large enough to accommodate the MODIFIED DSECT, the UB area, and the USERINFO area. The following example changes the default value from 512 bytes to 1,024 bytes, which would allow a USERINFO area of 848 bytes (provided as sample zap JVMGET3).

```
NAME PROFINIT PROFINIT
VER 007C 0200
REP 007C 0400
```

XII.3.5 Trace Subsystem Installation (optional)

PROFILER requires an additional 10K of GETMAIN storage per user in order to operate the Trace Subsystem. A zap flag in the assembler module PROFINIT is preset to indicate that PROFILER should perform the GETMAIN for this additional 10K. If the Trace Subsystem is NOT going to be used at a site, the GETMAIN flag should be set OFF by applying the following zap to PROFINIT (provided as sample zap JVMTRA1):

```
NAME PROFINIT PROFINIT
VER 007E 0F
REP 007E 00
```

No additional steps are required for the installation of the Trace Subsystem.

Note: By default, the PROFILER Trace option is enabled.
--

XII.3.6 Apply TSIRDC TWA Zap

Note: If TRIM or SECURITRE are installed, this step should be skipped.

Apply the following zap to TSIRDC

```
NAME TSIRDC TSIRDC
VER 1C E8
REP 1C D5
```

XII.3.7 NATLOAD the PROFILER NATURAL Modules

The PROFILER NATURAL modules may be installed under NATURAL 4.2.2 or above.

If NATURAL SECURITY SYSTEM (NSS) is installed, define the library PROLIB. PROLIB should be defined with the PREDICT XREF feature set to OFF. The PROFILER NATLOAD dataset contains no XREF data.

Execute the NATURAL NATLOAD program or make any needed changes and use JVMLOAD EXEC to load the PROFILER NATURAL modules from the second file on the tape to the desired FUSER file on each database where PROFILER will be run.

The NATLOAD process loads programs into the following NATURAL libraries: PROLIB and SYSTEM

Source and object code for the program PROFILER are loaded in the library SYSTEM. Source and object code for the programs PROUE01P and PROUE99P are loaded in the library PROLIB. The source code for these programs (PROUE01P and PROUE99P) may be modified. All other programs, which are object code only, are loaded into the library PROLIB.

The following sample NATLOAD related EXEC and NATURAL commands to load the NATURAL modules are in file JVMLOAD EXEC:

```

/* LOAD NATURAL 2 MODULES FOR PROFILER */
dfttape   = 'TAP1'           /* tape unit */
dftmodname = 'NATURALx'     /* NATURAL module name */
dftprint  = 'TERMINAL'     /* FILEDEF options for print */
Arg tape modname print
If tape   = '' Then tape   = dfttape
If modname = '' Then modname = dftmodname
If print  = '' Then print  = dftprint
Address 'COMMAND'
/* Is tape attached? Rewind tape. */
Do Forever
  Parse Var tape x 'TAP' y
  If x = '' & Index('0123456789ABCDEF',y) > 0 Then Do
    'TAPE REW ('tape
    If rc = 0 Then Leave
  End
  Say 'Enter the tape unit id (TAPn, or QUIT to quit):'
  Pull tape
  If tape = 'QUIT' Then Exit 8
End
/* Get NATURAL module name */
Do Forever
  If modname ^= '' Then Do
    Address 'CMS' 'STATE' modname 'MODULE *'
    If rc = 0 Then Leave
  End
  Say 'Enter the name of your NATURAL module file (or QUIT to
quit):'
  Pull modname
  If modname = 'QUIT' Then Exit 8
End
/* Setup to call NATURAL */
'ERASE PROUT1 CMSYNIN A'
'EXECIO 1 DISKW PROUT1 CMSYNIN A 0 F 80 (STRING B'
'EXECIO 1 DISKW PROUT1 CMSYNIN A (FINIS STRING FIN'
'FILEDEF * CLEAR'
'FILEDEF CMWKF01' tape 'NL 2 (RECFM VB LRECL 254 BLKSIZE 2540'
'FILEDEF CMSYNIN DISK PROUT1 CMSYNIN A'
'FILEDEF CMPRINT' print
/* NATLOAD file */
Say 'Starting NATLOAD for PROFILER NATURAL modules.'
'EXEC' modname 'BATCH STACK=(NATLOAD) WORK=(OS) AUTO=OFF',
'MADIO=0 IM=D MT=0 EJ=OFF'
If rc ^= 0 Then Do
  Say 'NATLOAD for PROFILER NAT modules failed, return code =' rc'.'
  Exit rc
End
Say 'PROFILER NATURAL modules loaded.'
'FILEDEF * CLEAR'
'ERASE PROUT1 CMSYNIN A'
'TAPE RUN ('tape

```

XII.3.7.1 SYSMAIN Modules from SYSEXT to SYSTEM

PROFILER uses several standard Software AG utility 'USR' subprograms. If not already in library SYSTEM, the following NATURAL modules need to be SYSMAIN copied from NATURAL library SYSEXT (in the FNAT file) to library SYSTEM (in the FUSER file):

USR0010N
USR0050N
USR1025N
USR2004N

XII.3.7.2 Installing the LOGON Front-End

Sites where NATURAL Security is not installed:

PROFILER uses a front-end to the LOGON command so that it may activate or deactivate sessions defined by the Background Monitoring facility. The following steps are necessary to install the LOGON front-end:

1. Using SYSMAIN, copy program PRFLOGON and subprograms PRFBCKN and PRFUEUID from PROLIB library to SYSLIB library on the FNAT file.
2. Using SYSMAIN, rename the LOGON program to XLOGON in SYSLIB library on the FNAT file.
3. Using SYSMAIN, rename PRFLOGON to LOGON in SYSLIB library on the FNAT file.
4. Copy USR0050N from the SYSEXT library on the FNAT to the SYSLIB library on the FNAT.

When the user enters the LOGON command, PROFILER's LOGON front-end receives control, determines whether a session should be activated or deactivated based on Background Monitoring, performs the activation or deactivation if necessary, then FETCHes the Software AG LOGON program.

Sites where NATURAL Security is installed:

At a site where NATURAL Security is installed, PROFILER uses subprogram PRFBCKN at LOGON time to activate and deactivate sessions defined by the Background Monitoring function. The following steps are necessary to install this subprogram:

1. Using SYSMAIN, copy subprograms PRFBCKN and PRFUEUID from PROLIB library to SYSLIB library on the FNAT file. Copy subprogram USR0050N from SYSEXT library (on the FNAT file) to SYSLIB library on the FNAT file.
2. Include a CALLNAT to PRFBCKN in the NATURAL Security user-exit LOGONEX1 after the INPUT statement. PRFBCKN requires the application name, #I-APPL, which has been passed to LOGONEX1 as a parameter. The CALLNAT should be coded as follows:

```
IF #I-APPL NE ' '  
  CALLNAT 'PRFBCKN' #I-APPL  
END-IF
```

3. Copy USR0050N from the SYSEXT library on the FNAT to the SYSLIB library on the FNAT.

When the user enters the LOGON command, subprogram PRFBCKN receives control, determines whether a session should be activated or deactivated based on Background Monitoring, and performs the activation or deactivation if necessary.

XII.3.7.3 Installing the FIN Front-End

PROFILER uses a front-end to the FIN command to deactivate any session that is active (whether through Background Monitoring or regular activation) before the user leaves NATURAL. The following steps are necessary to install the FIN front-end:

1. Modify the control and execute entries for the FIN command in NATTEXT (refer to the NATURAL Operations Manual), renaming the FIN command to XFIN for both the Control command and the Execute command.
2. Use SYSMAIN to rename PRFFIN (located in PROLIB library) to FIN and move the renamed module from PROLIB library to SYSTEM library on the FUSER file.

When the user enters the FIN command, PROFILER's FIN front-end receives control, performs the steps necessary to deactivate an open session, then STACKs the Software AG FIN program.

XII.3.7.4 User-Exits

PROFILER provides a user-exit facility to allow control to be passed to a user-written NATURAL program.

Source code for the user-exit routines may be found in the PROLIB library. The user-exit routines must be STOWed in library PROLIB or in a STEPLIB accessible from PROLIB. The following are the available user-exits:

PROFILER Termination Exit - PRFUE99P

This user-exit program is invoked during the termination of PROFILER. To exit from PROFILER, the user may press PF12 from most screens or PF3 from the PROFILER Main menu. This user-exit may be modified to provide an alternative exit procedure.

PROFILER Customized Reports Exit - PRFUE01P

This user-exit program is invoked when selecting the Customize Reports function from the Enhanced Reporting menu. A site may use this user-exit to access site-specific customized reports. Customized reports may be written using the PROFILER-REPORTING file, which is included on the PROFILER Installation tape. A description of the fields on this file is also included on the Installation tape.

PROFILER Customized Background Monitoring User-Id Exit – PRFUEUID

This user-exit program is invoked when background monitoring obtains a user-id. A unique site-specific user-id may be set by this exit; however, use of this exit is not typical and the default of *INIT-USER is appropriate for most sites.

XII.3.7.5 Step Library changes where NATURAL SECURITY is Installed

Where NATURAL SECURITY is installed:

1. Move program PRF from SYSTEM library to PROLIB library.
2. Make PROLIB a step library of any library that will be using PROFILER.
3. Ensure program PRF is NOT defined as the automatic startup program for any of these libraries.

XII.3.8 Load the PROFILER-REPORTING File DDE (optional)

Loading the DDE for the PROFILER-REPORTING file into PREDICT is an optional step, necessary only if the site intends to code custom reports which display statistics from the PROFILER-REPORTING file. The DDE is not required by PROFILER.

Execute the SYSDICBE LOAD function to load the DDE for the PROFILER-REPORTING file from the third file on the tape into PREDICT. The DDE was unloaded from PREDICT. The following sample SYSDICBE related EXEC and NATURAL commands to load the PROFILER-REPORTING file DDE are in file JVMDDE EXEC:

```

/* LOAD DDE for PROFILER-REPORTING file*/
dfttape   = 'TAP1'                /* tape unit */
dftmodname = 'NATURALx'          /* NATURAL module name */
dftprint   = 'TERMINAL'          /* FILEDEF options for print */
Arg tape modname print
If tape   = '' Then tape   = dfttape
If modname = '' Then modname = dftmodname
If print  = '' Then print  = dftprint
Address 'COMMAND'
/* Is tape attached? Rewind tape. */
Do Forever
  Parse Var tape x 'TAP' y
  If x = '' & Index('0123456789ABCDEF',y) > 0 Then Do
    'TAPE REW ('tape
    If rc = 0 Then Leave
  End
  Say 'Enter the tape unit id (TAPn, or QUIT to quit):'
  Pull tape
  If tape = 'QUIT' Then Exit 8
End
/* Get NATURAL module name */
Do Forever
  If modname ^= '' Then Do
    Address 'CMS' 'STATE' modname 'MODULE *'
    If rc = 0 Then Leave
  End
  Say 'Enter the name of your NATURAL module file (or QUIT to quit):'
  Pull modname
  If modname = 'QUIT' Then Exit 8
End
/* Setup to call NATURAL */
'ERASE PROUT1 CMSYNIN A'
'EXECIO 1 DISKW PROUT1 CMSYNIN A 0 F 80 (STRING LOGON SYSDICBE'
'EXECIO 1 DISKW PROUT1 CMSYNIN A (STRING MENU'
'EXECIO 1 DISKW PROUT1 CMSYNIN A (STRING LOAD ALL,REPLACE=Y'
'EXECIO 1 DISKW PROUT1 CMSYNIN A (FINIS STRING FIN'
'FILEDEF * CLEAR'
'FILEDEF CMWKF01' tape 'NL 3 (RECFM VB LRECL 254 BLKSIZE 2540'
'FILEDEF CMSYNIN DISK PROUT1 CMSYNIN A'
'FILEDEF CMPRINT' print
/* Load INPL file */
Say 'Starting SYSDICBE LOAD for PROFILER-REPORTING file'
'EXEC' modname 'BATCH',
      'WORK=(OS) AUTO=OFF MADIO=0 IM=D MT=0 EJ=OFF'
If rc ^= 0 Then Do
  Say 'Load for PROFILER-REPORTING file failed, return code =' rc'.'
  Exit rc
End
Say 'PROFILER-REPORTING DDE loaded.'
'FILEDEF * CLEAR'
'ERASE PROUT1 CMSYNIN A'
'TAPE RUN ('tape

```

Once the DDE has been loaded into PREDICT, users may generate a DDM that can be referenced in their custom report programs.

Note: Care must be taken to ensure that custom report programs written by the site do not modify the statistics stored in the PROFILER-REPORTING file. Modification of the statistics in the PROFILER-REPORTING file could corrupt the integrity of PROFILER Enhanced Reports and result in unexpected errors.

XII.3.9 Establish the ADABAS File Where PROFILER Statistics Will Be Stored

PROFILER statistics may be stored on the NATURAL FUSER file where PROFILER is installed or on a separate ADABAS file with the same layout (FDT) as the NATURAL FUSER file.

Select one of the following options:

FUSER File

Edit the NATPARM module for the NATURAL under which PROFILER will execute. Insert one NTFILE macro invocation after the last line in the NTPARM macro but before the END statement. Code the NTFILE invocation as follows:

NATURAL 4.2.2 and above

```
NTLFILE 120,xxx,yyy                                FUSER File
```

Where:

xxx = The DBID of the FUSER file
yyy = The file number of the FUSER file

Note: Logical ID 120 is used internally by PROFILER. It does not affect any file number 120 on the database.

Separate ADABAS File

Create an ADABAS file with the same layout (FDT) as the NATURAL FUSER where PROFILER is installed. The file should be loaded empty. The following space estimates represent a PROFILER system managing approximately 5,000 profiled programs. The space required varies based on the number of profiling sessions, the number of programs profiled in each session, and the size of each program.

PROFILER FILE	COMMENTS	SPACE EST.
PROFILER-FILE	Any unused ADABAS file number may be used	DSSIZE=38 (cylinders) UISIZE=10B NISIZE=20B

Edit the NATPARM module for the NATURAL under which PROFILER will execute. Insert one NTFILE macro invocation after the last line in the NTPARM macro but before the END statement. Code the NTFILE invocation as follows:

NATURAL 4.2.2 and above

```
NTLFILE 120,xxx,yyy                                PROFILER-FILE
```

Where:

xxx = The DBID of the PROFILER-FILE
yyy = The file number of the PROFILER-FILE

Note: Logical ID 120 is used internally by PROFILER. It will not affect any file number 120 on the database.

XII.3.10 Establish the ADABAS File For PROFILER Enhanced Reporting

PROFILER statistics may be moved to the PROFILER-REPORTING file, which is a separate ADABAS file. This file should only be used by PROFILER.

The fourth dataset on the installation tape contains the ADABAS ADACMP compressed format dataset for the PROFILER-REPORTING file. This dataset may be input into the ADABAS ADALOD utility directly. Optionally, the ADAWAN cards may be generated from the PREDICT DDE created earlier (refer to **Load the PROFILER-REPORTING File DDE** sub-section) and then punched to the appropriate dataset. The file may then be loaded empty.

The following space estimates represent a PROFILER Enhanced Reporting system managing approximately 5,000 profiled programs. The space required varies based on the number of profiling sessions, the number of programs profiled in each session, and the size of each program.

PROFILER FILE	COMMENTS	SPACE EST.
PROFILER-REPORTING	Any unused ADABAS file number may be used	DSSIZE=38 (cylinders) UISIZE=10B NISIZE=20B

Edit the NATPARM module for the NATURAL under which PROFILER will execute. Insert one NTFILE macro invocation after the last line in the NTPARM macro but before the END statement. Code the NTFILE invocation as follows:

NATURAL 4.2.2 and above

NTLFILE 122,xxx,yyy

PROFILER-REPORTING

Where:

xxx = The DBID of the PROFILER-REPORTING
yyy = The file number of the PROFILER-REPORTING

Note: Logical ID 122 is used internally by PROFILER. It does not affect any file number 122 on the database.

XII.3.11 Change and Assemble NATPARMs, Apply Zap(s) for PROFILER Statistics Collection, and Relink NATURAL Executable Modules

Assemble the NATPARMs:

Assemble NATPARM, including the following statements, along with any statements currently included in the existing NATPARM module.

CSTATIC=(PROFINIT, PROFCLOS, PROFDRV)

USERBUF=GETMAIN SIZE + 10K (if Trace is to be used) + 1K

(This parameter is required unless PROFILER has been configured to acquire GETMAIN storage from a source other than the NATURAL USERBUF. USERBUF must be at least equal to the GETMAIN size + 10K (for Trace) + 1K, but may be set to a larger value.)

DATSIZE=90

(Minimum required for running PROFILER)

RDSIZE=2

(Enable NATURAL RDC)

ITRACE=ON

(Activates NATURAL's internal trace function).

NTTRACE NATPROX

(Enable NATURAL RDC statement tracing)

For more information about GETMAIN size, refer to the **Apply GETMAIN Zap to PROFINIT** sub-section.

Note: If the NATPARMs are link-edited by themselves, a non-zero condition code is received on the link-edit step with unresolved references for PROFINIT, PROFCLOS, and PROFDRV.

- **Note:** When assembling the NATPARM module with the USERBUF parameter specified, the following message will appear in the listing:

USERBUF PARAMETER IS NO LONGER USED FOR THE DATA COLLECTOR.

PLEASE USE RDCSIZE INSTEAD.

These message may be ignored.

XII.3.12 Considerations of Using the RDC Interface

- Required PROFILER
- Interfaces with the NATURAL Review Data Collector feature
- Requires NATPARM parameter NTTRACE NATPROX to enable RDC statement tracing, and RDSIZE of at least 2

XII.3.13 Increase Size of the Virtual Machine

If a site is not running NATURAL from a DCSS (Discontiguous Shared Segment), then increase the size of each virtual machine that will be running PROFILER. A machine size of 4M should be sufficient.

To increase the size of the virtual machine, issue the following command:

```
DEF STOR 4M
```

If a site is running NATURAL from a DCSS, the DCSS needs to be saved at 2M or higher. See a VM System Administrator on loading a DCSS for NATURAL.

XII.3.14 Verify the PROFILER Installation

PROFILER includes a special program (PRFVRFY), which verifies that PROFILER is properly installed. PRFVRFY ensures that NATURAL is properly configured for PROFILER, that the PROFILER modules are in place, that the proper statistics file is installed and accessible, and that the PROFILER Statistics Collection zap(s) are in place. It then simulates a profiling session, verifying that statistics can be collected and stored in the statistics file. It also provides diagnostic messages, which identify possible installation problems and suggest how to resolve them.

To verify that PROFILER is installed correctly, invoke NATURAL and LOGON to PROLIB. Then type PRFVRFY and press Enter. PRFVRFY checks the installation and displays an error report or the message "No errors have been encountered in the PROFILER installation.". When PRFVRFY detects an installation error, a screen similar to the following example is displayed.

```

2011-11-20 11:38 PROFILER Installation Verification USER24  PROLIB

The following errors have been found in the PROFILER installation:

PRO0014: PROFILER requires NATURAL V4.2.2 or above. Contact System Administrator.
PRO0015: NTFILE Macro NOT included in NATPARM Module. Contact System Admin.

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help           End                               Exit

```

PRFVRFY uses a profiling session to verify the statistics collection process. The session used has an internal session number of 63. If this session is already in use at a site PRFVRFY displays message PRO0128 or PRO0129. A site should delete this session to continue the installation verification process.

These are the errors that can be detected during installation verification:

```

PRO0014: PROFILER requires NATURAL V4.2.2 or above. Contact System Administrator.
PRO0015: NTFILE Macro NOT included in NATPARM Module. Contact System Admin.
PRO0017: PROFILER for NATURAL Trial EXPIRED. Contact System Administrator.
PRO0018: PROFILER for NATURAL License is INVALID. Contact System Administrator.
PRO0019: Response ... for "PROFINIT" from {object}. Contact System Admin.
PRO0036: Response ... for "TREPFL" from {object}. Contact System Admin.
PRO0041: Response ... for "PROFCLOS" from {object}. Contact System Admin.
PRO0119: PRFVRFY must be run from Library "PROLIB".
PRO0120: PROFILER Assembler Modules are NOT Version 4.3.1
PRO0121: Module PROFCLOS or PROFDRV is NOT Linked with NATURAL.
PRO0122: Unable to Collect Statistics. Reason Unknown.
PRO0123: No Operating System Driver linked with NATURAL.
PRO0124: Wrong Operating System Driver linked with NATURAL.
PRO0125: PROFCLOS, PROFDRV, or CMDBGEX were NOT found in CSTATIC List.
PRO0126: PROFILER "ZAP" NOT applied to NATURAL.
PRO0127: Warning: Unable to Verify that PROFILER "ZAP" has been applied to NATURAL.
PRO0128: Session {63} is already ACTIVE. Verify CANNOT continue.
PRO0129: Session {63} already has Statistics. Verify CANNOT continue.
PRO0130: Unable to obtain ACTIVE Session Information.

```

For more information on installation error codes, refer to **Appendix A, PROFILER Messages**.

APPENDIX A

PROFILER MESSAGES

PROFILER Installation Verification Messages

PROFILER includes a special program (PROVRFY), which verifies that PROFILER is properly installed. PROVRFY ensures that NATURAL is properly configured for PROFILER, that the PROFILER modules are in place, that the proper statistics file is installed and accessible, and that the PROFILER Statistics Collection zap(s) is in place. It then simulates a profiling session, verifying that statistics can be collected and stored in the statistics file. It also provides diagnostic messages, which identify possible installation problems.

PROVRFY Messages

PRO0015: NTFILE Macro NOT included in NATPARM Module. Contact System Administrator.

- PROFILER requires the NTFILE/LFILE 120 definition in the NATURAL parameters. Refer to the **Establish the ADABAS File where PROFILER Statistics will be Stored** section for more information.
-

PRO0017: PROFILER for NATURAL Trial EXPIRED: ... , Contact System Administrator.

- The trial zap to PROFINIT has expired. Contact Treehouse Software to obtain a new trial zap.
-

PRO0018: PROFILER for NATURAL License is INVALID. Contact System Administrator.

- The authorization zap received from Treehouse Software is incorrect for this CPU-ID. Verify that the authorization zap was applied correctly, and contact Treehouse Software if there is still a problem.
-

PRO0019: Response ... for "PROFINIT" from Contact System Admin.

- PROFINIT produced error message "nnn". PROFINIT response codes are included in this appendix.
-

PRO0119: PRFVRFY must be run from Library "PROLIB*".

- PROVRFY must be installed in and executed from library PROLIB*.
-

PRO0120: PROFILER Assembler Modules are NOT Version 4.3.2

- The PROFILER assembler modules are linked with NATURAL, but they are not from version 4.3.2 of PROFILER. Verify that the PROFILER load library in the NATURAL link-edit is a PROFILER 4.3.2 load library.
-

PRO0121: Module PROFCLOS is NOT Linked with NATURAL.

- PROFCLOS has not been linked with this NATURAL nucleus. Verify that the link-edit of NATURAL includes the PROFCLOS module.
-

PRO0121: Module PROFDRV is NOT Linked with NATURAL.

- PROFDRV has not been linked with this NATURAL nucleus. Verify that the link-edit of NATURAL includes the PROFDRV module.
-

PRO0122: Unable to Collect Statistics. Reason Unknown.

- PROFILER is unable to collect statistics. Verify that RDCSIZE is specified in the NATPARMS and that TSIRDCnn and PROF4Nxx are installed properly.
-

PRO0123: No Operating System Driver linked with NATURAL.

- Either PROFMDRV (MVS) or PROFVDRV (VM) has not been linked with this NATURAL nucleus. Verify that the link-edit of NATURAL includes either the PROFMDRV OR PROFVDRV module.
-

PRO0124: Wrong Operating System Driver linked with NATURAL.

- MVS: The PROFVDRV module has been link-edited with NATURAL. Remove PROFVDRV from the link-edit list, and add PROFMDRV in its place.

VM: The PROFMDRV module has been link-edited with NATURAL. Remove PROFMDRV from the link-edit list, and add PROFVDRV in its place.
-

PRO0125: PROFCLOS, PROFDRV, or CMDBGEX were NOT found in CSTATIC List.

- PROFINIT, PROFCLOS, and PROFDRV must be included in the CSTATIC list. Verify that these modules are in the CSTATIC list, that the NATPARMs have been assembled since these modules were added to the CSTATIC list, and that the correct NATPARMs have been linked with the NATURAL nucleus.
-

PRO0128: Session ... is already ACTIVE. Verify CANNOT continue.

- PROVRFY is unable to confirm successful installation when a session is active for the user running PROVRFY. Deactivate the currently active session and rerun PROVRFY.
-

PRO0129: Session ... already has Statistics. Verify CANNOT continue.

- PROVRFY uses the session internally numbered 63 to collect statistics. Session 63, named xxxxxx, already exists on this PROFILER file. A site should delete this session to continue the installation verification process.
-

PRO0130: Unable to obtain ACTIVE Session Information.

- The PROFILER assembler modules are unable to locate the work area that contains the information about the user's currently active session. Contact Treehouse Software immediately.
-

PROFILER Internal Errors

All ADABAS response codes greater than 0 generated on calls made by PROFILER will be displayed on the system console log.

The detection of an internal error during the course of execution of any of the PROFILER assembler modules results in either a return code or an abend code. In the case of assembler return codes, the code is returned to a NATURAL module, which then displays one of the following messages:

PRO0019: Response ... for "PROFINIT" from {object} . Contact System Admin.

PRO0041: Response ... for "PROFCLOS" from {object}. Contact System Admin.

Depending on the severity of the internal error message received, pressing Enter after receiving the message either exits PROFILER or allows processing to continue.

PROFCLOS Codes

Return codes from PROFCLOS:

Code	Explanation
4	Session close failed, no profiling session was active. Action: Contact Treehouse Software, Inc.
8	Session close failed, the PROFILER work area could not be found. Action: Contact Treehouse Software, Inc.

Abend codes from PROFCLOS:

Code	Explanation
SOC3	An internal error occurred. Action: Contact the PROFILER Administrator to investigate the cause of the error.
S878	The PROFILER work area FREEMAIN failed. Action: Contact Treehouse Software, Inc. with the DUMP generated.

PROFINIT Codes

Return codes from PROFINIT:

Code	Explanation
4	Session initialization failed, the module PROFDRV was not found in the CSTATIC list. Action: Contact Treehouse Software, Inc.
8	Session initialization failed. For Batch/TSO and VM, the GETMAIN for the PROFILER work area failed, probably due to insufficient virtual memory in the region. Action: For Batch/TSO, contact the Database Administrator to increase the region parameter on the NATURAL Execution Statement. For VM, increase virtual storage.
12	Session initialization failed. PROFDRV was not linked with the NATURAL nucleus. Action: Contact the PROFILER Administrator to investigate a probable installation problem.
16	Session initialization failed. The load of ADALNKR failed; probably due to either insufficient virtual memory in the region or the load module could not be found or loaded from the STEPLIB. Action: Contact the PROFILER Administrator to investigate a probable installation problem.
20	Session initialization failed. The passed parameter containing the session number was either not present, not in the range of 1 through 64 if the FUSER was being used for collecting statistics, or not in the range of 1 through 255 if the alternate FUSER was being used for collecting statistics. Action: Contact Treehouse Software, Inc.
24	Session initialization failed. The address of the NATURAL User Buffer could not be determined. Action: Contact the PROFILER Administrator to investigate a probable installation problem.
32	Session initialization failed. The USERBUF parameter was not large enough. Action: Contact the NATURAL Administrator to ensure the parameter USERBUF=GETMAIN size + 1K was used and that the NATURAL Thread is large enough.
36	Session initialization failed; non-zero response code returned for ADABAS call. The response code is displayed on the console log. Action: Contact the database administrator and investigate the reason for the response code.
44	GETMAIN failed for PROFILER work area. This is probably due to insufficient virtual memory in the region. Action: Contact the NATURAL Administrator to increase the region parameter on the NATURAL Execution Statement.

Code	Explanation
48	PROFILER trial period has expired. Action: Contact Treehouse Software.
52	Invalid PROFILER passcode detected. Action: Contact Treehouse Software.
56	The PROFILER modules linked with NATURAL are the not the correct modules for the version of NATURAL in use. Action: Contact the NATURAL administrator to install the correct version of PROFILER. Refer to the installation section of this manual for more information on linking the NATURAL nucleus.
60	An internal error caused PROFILER to be unable to locate the NATURAL CSTATIC table. Action: Contact Treehouse Software.
100	The NATURAL Administrator has deactivated the TRACE facility. Action: Contact the NATURAL administrator to have TRACE turned on.

PROFILER NATURAL Errors

PROFILER has a standard error-trapping program. When a NATURAL error occurs, the following message will be displayed.

PRO0020: Error NAT.... in Object ... on line (....) Contact System Administrator.

Pressing Enter after receiving a NATURAL error message returns the user to the PROFILER Main menu.

Occasionally, a NATURAL error message that is not trapped by the PROFILER error program is displayed in a pop-up window. For example, this may occur if non-numeric data is entered in a numeric field. The error message may be too long to completely display in the pop-up window, but the error message number is visible.

NATURAL Program Return Codes

Code	Explanation
NAT0888	Storage overflow of the DATSIZE buffer during program execution. Action: Increase the DATSIZE (local buffer area) to at least 90k.
NAT1016	Program interrupted due to attention interrupt. This indicates that the user interrupted the PROFILER NATURAL program via an attention interrupt (e.g., pressing a key while the program was executing). This error can occur during the execution of any of the following PROFILER Enhanced Reporting Data Request Facility functions: <ul style="list-style-type: none"> • Tag Data for Move • Untag Data • Move All Tagged Data • Move Selected Tagged Data Action: Perform the PROFILER function again to commence processing from the point at which the error occurred. Do not press any keys on the terminal until PROFILER indicates that processing has completed.
NAT3009	The last transaction has been backed out of the database. This indicates that an ADABAS timeout has occurred and that the Non-Activity Time Limit has been exceeded. This error will occur when the first call to ADABAS is done after the timeout occurs. Action: Check user programs for a logical transaction that does not contain an END TRANSACTION statement.
NAT3021	Invalid CID value was detected. Action: Under CICS, ensure that ADALNC/LNKOLSC has the SAP parameter set to 'YES'. Under Batch/TSO and COM-LETE/TPF, ensure that a reentrant ADALNKR has been created.
NAT3049	Compressed Record too long for internal buffer. This indicates that the blocksize of the device type on which PROFILER is running is too small. PROFILER requires at least 3,628 bytes to create its data record. Action: Contact Treehouse Software, Inc.
NAT3061	An error was detected in the search buffer. This indicates the installation step <i>Establish the ADABAS File for PROFILER Enhanced Reporting</i> has not been performed. Action: Contact the System Administrator.
PRO0015	NTFILE Macro NOT included in NATPARAM Module. Contact System Administrator. This indicates the installation step <i>Establish the ADABAS File Where PROFILER Statistics Will Be Stored</i> has not been performed. Action: Contact the System Administrator.

PROFILER Messages and Errors: Complete Listing

PRO0001	PF Key ... is NOT Valid here.
PRO0002	No Sessions Found {starting from ...} {for owner ...}.
PRO0003	More than ... sessions found. Only the first ... will be shown.
PRO0004	Selection criteria altered. ... Key not processed.
PRO0005	Command is Invalid (Type "?" for Help).
PRO0006	Use <enter> to process selected session. (Don't use ... Key.)
PRO0007	Selection criteria altered. {... Key and} Selected session(s) not processed.
PRO0008	Only select ONE session at a time.
PRO0009	This is the first page. ... Key not processed.
PRO0010	This is the last page. ... Key not processed.
PRO0011	Warning - Session is ACTIVE. Press <enter> to execute your command.
PRO0012	Mark only ONE selection.
PRO0013	Use PF5 to access Included/Excluded Objects.
PRO0014	PROFILER requires NATURAL V4.2.2 or above. Contact System Administrator.
PRO0015	NTFILE Macro NOT included in NATPARM Module. Contact System Administrator.
PRO0016	Another User has SESSION CONTROL record on HOLD. Try again in a moment.
PRO0017	PROFILER for NATURAL Trial EXPIRED ... , Contact System Administrator.
PRO0018	PROFILER for NATURAL License is INVALID. Contact System Administrator.
PRO0019	Response ... for "PROFINIT from Contact System Admin.
PRO0020	Error NAT.... in Object ... on line (....) Contact System Administrator.
PRO0021	Both Library and Object must be entered.
PRO0022	Mask may NOT be used for Included/Excluded Object.
PRO0023	At least one Type must be specified.
PRO0024	Type ... need only be specified ONCE.
PRO0025	Database Id must be in range 1 to 65535, or 0 for "default".
PRO0026	FUSER File Number must be in range 1 to 65535, or 0 for "default".
PRO0027	Session ... Modified Successfully.
PRO0028	Session Name MUST be provided.
PRO0029	Session Name has been previously defined.
PRO0030	This Library is NOT Available for PROFILING.
PRO0031	You have ... active already. You can NOT activate another session.
PRO0032	Statistics exist: <pf3> & reset stats (RS) OR <enter> & add to them.
PRO0033	Objects TAGGED for MOVE to "Enhanced Reporting". Activation MAY cause stats to change.
PRO0034	Maximum of 84 users already active: <pf3> and use "AU" to List Active Users.
PRO0035	Warning: Other Users already active on Session. Use "AU" to list them.
PRO0036	Response ... for "TREPREF1" from {object}. Contact System Admin.

PRO0037	Session ... Activated successfully for ...
PRO0038	Can NOT Activate. Session is ALREADY active.
PRO0039	Can NOT De-Activate. Session is NOT active for you.
PRO0040	Session ... De-Activated successfully for ...
PRO0041	Response ... for "PROFCLOS" from {object}. Contact System Admin.
PRO0042	"Format" or "Page" altered. ... Key not processed.
PRO0043	"Format" or "Page" altered. {... Key and} Selected object(s) not processed.
PRO0044	Sort/Type/Start/Threshold altered. ... Key not processed.
PRO0045	Sort/Type/Start/Threshold altered. {... Key and} Selected object(s) not processed.
PRO0046	Use <enter> to process selected object. (Don't use ... Key.)
PRO0047	Mark at least ONE selection.
PRO0048	No "Profiled Objects" Found starting from ... with Types ...
PRO0049	Page ... is NOT within valid range (i.e., 1 to ...).
PRO0050	Report Format is already ... {pf-key} ignored.
PRO0051	... Objects to be Reported. Only ... can be shown here.
PRO0052	... is being executed in TEST mode. NO Stores/Updates will be applied.
PRO0053	Session is currently ACTIVE.
PRO0054	"Page" altered. ... Key not processed.
PRO0055	"Page" altered. {... Key and} Selected session(s) not processed.
PRO0056	Session converted OK.
PRO0057	Header converted but NO Control rec.
PRO0058	Maximum ... sessions used. Delete one before adding another.
PRO0059	Session Name MUST begin with an alphanumeric character.
PRO0060	Session Name can NOT be "NO". (Reserved Word.)
PRO0061	New Session ... Defined successfully for ...
PRO0062	Password must be supplied.
PRO0063	Password must be alphanumeric.
PRO0064	Password record MISSING. Contact Systems Admin.
PRO0065	Password you have supplied is NOT correct.
PRO0066	New Password and Confirmation do NOT match.
PRO0067	Password change was Successful.
PRO0068	Must be "L"ibrary or "U"ser-Id.
PRO0069	A value MUST be provided.
PRO0070	No existing Profile Session with this name.
PRO0071	"Monitor From" and "Monitor to" must be in format: YYYY MM DD HH II.
PRO0072	"Monitor to" must be later than "Monitor From".
PRO0073	"Monitor From" must be in the Future.
PRO0074	Background Monitor for ... Modified successfully by ...
PRO0075	Background Monitor for ... Deleted successfully by ...
PRO0076	Background Monitor for ... Defined successfully by ...
PRO0077	Session ... already has Monitoring defined for ...
PRO0078	Must be "P" (Profile only), "T" (Trace only), or "A" (All sessions).
PRO0079	Must be "P" (Profile), or "T" (Trace).

PRO0080 Must be a valid statement number.

PRO0081 Statement range only allowed when a SINGLE object is being Traced.

PRO0082 Start of range is Greater than end.

PRO0083 Command ONLY available to Session Owner (i.e., ...).

PRO0084 Can NOT Purge an Active Session. De-Activate Session and then Purge.

PRO0085 Session data for ... Purged successfully by ...

PRO0086 Trace data NOT accumulated for more than ... statements.

PRO0087 No Trace data Exists for Session ...

PRO0088 Statistics exist: <pf3> to retain stats OR <enter> to remove them.

PRO0089 Line ... is NOT within valid range (i.e., 1 to ...).

PRO0090 Text ... Not Found starting from line ...

PRO0091 Object Code does not exist for ... in Library ...

PRO0092 No Profile Statistics exist for ... and Library ...

PRO0093 Source NOT Found in LIB ... on Use PF9 to Alter.

PRO0094 ... has been SAVED after Profiling. Reset Stats & Re-Profile.

PRO0095 ... has been CATALOGed after Profiling. Reset Stats & Re-Profile.

PRO0096 "S"elect Object to see its Source Code Listing Report.

PRO0097 Place cursor on Name of Copycode before using ...

PRO0098 Command ... does not apply to Trace sessions.

PRO0099 Enter string you want to SCAN for... (Leave blank to Quit.)

PRO0100 At least ONE type of statement must be shown.

PRO0101 Statements shown: {Executed}, {Un-executed}, {Non-Executable}.

PRO0102 Answer MUST be "Y" or "N".

PRO0103 ... objects EXCLUDED due to Threshold settings.

PRO0104 Threshold MUST be in the range ... to ...

PRO0105 Use <enter> to process selection. (Don't use ... Key.)

PRO0106 Only ONE selection at a time.

PRO0107 ... Sessions Purged for User ...

PRO0108 ... Sessions Purged for ALL Users.

PRO0109 No users currently ACTIVE on Profile Session ...

PRO0110 No users have been MARKED for "Reset". Mark User(s) and Press PF2.

PRO0111 ... Active Users have been "Reset" for Session ...

PRO0112 ... Sessions Purged. ... PROFILER records deleted in total.

PRO0113 Statistics for ... Reset successfully by ...

PRO0114 Warning - "Reset" may stop "tagged" objects being moved to "Enhanced Reporting".

PRO0115 ... "Detail" Records DELETED for Session ...

PRO0116 You may limit Reset of Stats by User, Library, Object, and/or Type.

PRO0117 Use "*" if ALL cases of this criteria are to have Statistics Reset.

PRO0118 Session ... Purged. ... PROFILER records deleted in total.

PRO0119 PRFVRFY must be run from Library "PROLIB".

PRO0120 PROFILER Assembler Modules are NOT Version 4.3.1.

PRO0121 Module ... is NOT Linked with NATURAL.

PRO0122 Unable to Collect Statistics. ITRACE=ON must be specified in the NATPARAMs.

PRO0124	Wrong Operating System Driver linked with NATURAL.
PRO0125	PROFCLOS, PROFDRV, or CMDBGEX were NOT found in CTSTATIC List.
PRO0126	PROFILER "ZAP" NOT applied to NATURAL.
PRO0127	Warning: Unable to Verify that PROFILER "ZAP" has been applied to NATURAL.
PRO0128	Session ... is already ACTIVE. Verify CANNOT continue.
PRO0129	Session ... already has Statistics. Verify CANNOT continue.
PRO0130	Unable to obtain ACTIVE Session Information.
PRO0131	Invalid Object Type ... specified.
PRO0132	Invalid Statement Type ... specified.
PRO0133	A Library name must be provided for this Report.
PRO0134	An Object name must be provided for this Report.
PRO0135	A User-ID must be provided for this Report.
PRO0136	... records have been Tagged for Move for Session ...
PRO0137	All matching records ALREADY Tagged for Move for Session ...
PRO0138	NO matching records found for Tag/Untag for Session ...
PRO0139	NO Tagged Data for this Session.
PRO0140	... records have been Un-Tagged from Move for Session ...
PRO0141	Statistics for ... objects have been Moved for Session ...
PRO0142	NO Profile Sessions have Tagged Data.
PRO0143	Wildcards (*, ?, @, .) NOT Allowed for Library/User-ID.
PRO0144	... must be defined as a step library of ...
PRO0145	No Report Params Found [starting from ...] [for owner ...].
PRO0146	Use <enter> to process selected report param. (Don't use ... Key.)
PRO0147	Selection criteria altered. ... Key [and Selected report param(s)] not processed.
PRO0148	Only select ONE report param at a time.
PRO0149	"Page" altered. ... Key and [Selected report param(s)] not processed.
PRO0150	More than ... report params found. Only the first ... will be shown.
PRO0151	No Groups Found [starting from ...].
PRO0152	Use <enter> to process selected group. (Don't use ... Key.).
PRO0153	Selection criteria altered. ... Key [and Selected group(s)] not processed.
PRO0154	Only select ONE group at a time.
PRO0155	More than ... groups found. Only the first ... will be shown.
PRO0156	At least one entry must be provided.
PRO0157	This value provided more than once.
PRO0158	New Group ... Defined successfully.
PRO0159	Group ... Modified successfully.
PRO0160	Group ... Purged successfully.
PRO0161	Mark "All Users", or Enter a User-ID, or Enter a User Group.
PRO0162	Mark "All Sessions", or Enter a Session Name, or Enter a Session Group.
PRO0163	New Report Parameter Set ... Defined successfully.
PRO0164	Command ONLY available to Parameter Owner (ie. ...).
PRO0165	Report Parameter Set ... Modified Successfully.
PRO0166	Report Parameter Set ... Deleted Successfully.

This page intentionally left blank.

APPENDIX B

COMMON PROBLEMS

Problem: **SOC4 OR NAT0954 AT SESSION ACTIVATION**

Explanation: A dump may show the ADABAS response 22 - Invalid command detected.

This is most likely a Batch/TSO and COM-LETE/TPF user installation problem that occurred during the creation of a reentrant ADALNKR. Ensure that the instruction at label NONRENT has been changed from:

```
LA RE,NOTRENT
to:
L RE,24(R1)
```

If the 'LA' is not changed to 'L', this problem occurs.

Problem: **THE SOURCE CODE LISTING REPORT INDICATES THAT EXECUTABLE STATEMENTS ARE NOT BEING EXECUTED OR THAT NON-EXECUTABLE STATEMENTS ARE BEING EXECUTED**

Explanation: The Source Code Listing Report or the Enhanced Source Code Listing Report may mark a statement that has executed with a '>', indicating that the statement did not execute or may show statistics for a non-executable statement or for an executable statement that did not execute. This may occur for several reasons:

1. The object has been re-STOWed since statistics calculation began. The Source Code Listing Report displays a warning if this is the case. Reset the statistics for the object; re-profile the object; re-run the Source Code Listing Report.
2. The object has been re-STOWed since statistics were moved to the PROFILER Reporting file. Delete the data from the PROFILER Reporting file; re-profile the object; move the data to the Reporting file; re-run the Enhanced Source Code Listing Report.
3. The site is using the NATURAL OPTIMIZER COMPILER (NOC). Using PROFILER with the NOC is documented in the **Clarification of PROFILER Statistics During Reporting** sub-section of the **Session Reporting** section.

Problem: PROFILER DOES NOT COLLECT STATISTICS

Explanation: This may occur for several reasons:

1. If there is a conflict between TSIRDC and another module trying to use the NATURAL exit RDCEX1,2, or 3.
2. if TSIRDC or PROFRDC are not linked with the NATURAL nucleus.
3. if the NATPARM parameter RDCSIZE is not set to at least 2.

Refer to the Installation Section of this manual for more information.

Problem: PROFILER STATISTICS INDICATE THAT AN OBJECT WAS EXECUTED FROM A WRONG LIBRARY

Explanation: This occurs when an object is CATALOGed in one library and moved to another library using SYSMAIN (no CATALOG done). When SYSMAIN is used to move an object from one library to another, the object code contains the original library unless the object is reCATALOGed in the new library. An object should be CATALOGed in the library from which it is to be profiled.

Problem: STATISTICS ARE DIFFERENT FOR THE SAME NATURAL STATEMENT

Explanation: For more information about the differences in statistics for the same NATURAL statement, refer to the **Clarification of PROFILER Statistics During Reporting (PROFILER CPU Calculation)** sub-section of the **Session Reporting** section.

Problem: S322 COMPLETION CODE DURING BATCH PROCESSING

Explanation: This error may occur when a specified CPU time limit has been exceeded during long-running batch processing. Increase the Job Time or Step Time parameters.

Index

A

Activate..... I-15, II-5, II-6, II-7, III-3, IV-2,
IV-3, IV-11, IV-19, IV-20, IV-23,
VII-2, X-2, X-3, X-4, A-9
ADABAS I-5, I-6, I-7, I-8, I-9, I-10, IV-3,
IV-5, IV-11, IV-13, IV-22, V-3, VI-2,
VI-38, VII-1, IX-1, X-3, X-4, XII-1, XII-2,
XII-4, XII-9, XII-14, XII-20, XII-21, XII-25,
XII-27, XII-28, XII-30, XII-39, XII-40, A-1,
A-7, B-1
ADACMP XII-9, XII-21, XII-27, XII-40
ADALNC A-7
ADALNKR XII-25, A-5, A-7, B-1
ADALOD XII-21, XII-40
ADAWAN XII-21, XII-40
Administrator I-15, I-21, III-4, VI-9, VII-6,
VIII-7, IX-1, IX-3, XI-4, XII-26, XII-42,
XII-43, A-1, A-4, A-5, A-6, A-7, A-8
Application Profiling I-1, I-2
AT BREAK V-7, VI-42, X-12, X-18
AT END V-7, VI-42, VIII-7, X-12, X-18
AT START V-7, VI-42, X-12, X-18
AT TOP V-7, VI-42, X-12, X-18
Authorization XII-4, XII-12, XII-28,
XII-29, A-1
Average CPU V-32, VI-68

B

Background monitoring I-7, I-15, II-1,
III-1, III-4, IV-1, IV-19, VI-9, VII-1,
VII-2, VII-3, VII-4, VII-5, VII-6, VII-7,
VII-8, VII-9, VII-10, VII-11, VII-12,
VII-13, VII-14, VII-15, VIII-7, XII-17,
XII-18, XII-34, XII-35, XII-36
Batch I-8, VIII-1, VIII-8, IX-6, X-1, X-2,
X-5, X-8, X-10, X-11, X-15, X-17, XII-22,
XII-25, A-5, A-7, B-1
BEFORE BREAK ... V-7, VI-42, X-12, X-18
Bottom of Object II-12, V-33, V-34,
VI-68, VI-69
BUS XII-22

C

Cartridge XII-3, XII-27
CATALOG B-2
Catalog Timestamp I-10, VIII-10
CATALOGed V-31, VI-66, VIII-2,
VIII-8, VIII-11, A-10, B-2
CICS VIII-1, VIII-9, XII-4, XII-13,
XII-22, XII-25, A-7

CMDBGEX XII-26, XII-43, A-2, A-11
COM-LETE VIII-1, VIII-8, XII-4,
XII-25, A-7, B-1
configuration XII-4, XII-13
Convert VIII-7
Copy Report VI-8, VI-9, VI-20, VI-21
Copy Session I-15, III-3, IV-2, VI-35, VI-36
Copy Session Group VI-35, VI-36
Copy User Group VI-28
Copycode II-12, V-32, V-33, VI-67,
VI-68, VIII-1, X-14, X-20, A-10
CPU Time I-9, V-6, V-7, V-11, V-12,
V-15, V-16, V-17, V-19, V-20, V-23,
V-25, V-27, V-29, VI-41, VI-42, VI-46,
VI-47, VI-50, VI-51, VI-52, VI-54,
VI-55, VI-58, VI-60, VI-62, VI-64, X-11,
X-12, X-17
CSTATIC XII-23, XII-41, A-2, A-5
Customized Report I-7, I-15, III-4, VI-1,
VI-8, XII-18, XII-36

D

Database Elapsed Time I-8, I-9, I-20,
II-10, II-11, III-4, IV-10, IV-14, IV-18,
IV-23, IV-28, V-1, V-2, V-3, V-6, V-7,
V-13, V-14, V-15, V-16, V-17, V-18,
V-19, V-20, V-32, V-34, VI-8, VI-37,
VI-38, VI-41, VI-42, VI-48, VI-49,
VI-50, VI-51, VI-52, VI-53, VI-54,
VI-55, VI-67, VI-69, VIII-1, VIII-9,
VIII-10, X-11, X-12, X-13, X-17, X-18,
X-19, XI-4, XI-7, XI-8, XI-9, XI-14
Database/Work File Loops ... I-9, III-4, V-1,
V-7, V-21, VI-8, VI-37, VI-42, VI-56,
VIII-6, X-11, X-12, X-17, XI-4, XI-10
DATSIZE XII-2, XII-23, XII-41, A-7
DB/WF Bodies Exec V-22, V-30,
VI-57, VI-65
DB/WF Loops Exec V-22, V-30, VI-57,
VI-65
DB2 I-9, V-3, VI-38
DDE XII-3, XII-4, XII-9, XII-19, XII-21,
XII-27, XII-28, XII-38, XII-40
Deactivate I-15, II-8, III-3, IV-2, IV-15,
IV-24, IV-25, VII-2, X-2, X-4, X-5, X-6,
A-3, A-9, A-10
Debugging I-2, I-3, IV-1, XI-1, XI-9
DECIDE
%Body Exec V-30, VI-65
%Cond Exec ... V-28, V-30, VI-63, VI-65
Body Exec V-28, VI-63
Conds Exec V-28, VI-63

Total Body V-28, VI-63
 Total Conds V-28, VI-63
 DEFINE DATA VIII-6
 Define Report..... VI-10, VI-11, VI-12
 Define Session..... VI-30
 Define Session Group..... VI-30
 DEFINE SUBROUTINE VIII-2
 Define User Group..... VI-23
 Delete Report..... VI-17, VI-19
 Delete Session..... VI-34
 Delete Session Group..... VI-34
 Delete User Group..... VI-27
 Dispatcher..... I-6, VIII-3, VIII-8
 Display Report VI-8, VI-9, VI-13, VI-14
 Display Session I-15, III-3, IV-2, IV-7, VI-32
 Display Session Group VI-32
 Display User Group VI-25
 Down one Page II-12, V-33, VI-68
 DSECT..... XII-14, XII-30
 DSIZE XII-2

E

Education..... I-2, I-4, XI-1, XI-11
 END Statements VIII-7
 Enhanced Reporting I-6, I-10, II-1, II-15,
 IV-29, VI-1, VI-2, VI-7, VI-11, VI-23, VI-29,
 VI-30, VI-37, VI-38, VI-39, VII-3, VIII-8,
 VIII-13, VIII-14, IX-7, X-1, X-17, XII-2,
 XII-4, XII-18, XII-21, XII-28, XII-36,
 XII-40, A-7, A-8, A-10
 Enhanced Summary Reports..... VI-1, VI-8
 Environment.. I-5, I-11, I-15, I-18, II-2, III-1,
 III-2, III-4, IV-1, V-37, VI-7, VI-9, VI-70,
 VII-5, VII-6, VIII-8, VIII-9, IX-1, X-2, X-5,
 X-7, X-8, X-10, XI-1
 Evaluation I-2, I-4, XI-1, XI-11
 Executable Statement.... V-10, V-16, V-18,
 V-38, VI-45, VI-51, VI-53, VI-71, XI-2
 Exit from PROFILER..... II-12, V-33, VI-68
 EXPIRED I-21, XII-26, XII-43, A-1, A-8

F

FDT.. XII-2, XII-3, XII-9, XII-20, XII-27, XII-
 39
 FIND..... I-4, II-13, V-7, V-21, V-22, V-34,
 VI-42, VI-56, VI-57, VI-69, VIII-4, VIII-6,
 VIII-7, X-12, X-17, XI-3, XI-13, XI-14
 FNAT..... XII-16, XII-17, XII-34, XII-35
 FOR Loops..... XI-4, XI-10
 %Loops Exec... V-26, V-30, VI-61, VI-65
 Graph of Loops Exec..... V-26, VI-61
 Loops Exec..... V-26, VI-61
 Total Loops..... V-26, VI-61
 Front-end VII-2, XII-17, XII-18, XII-34,
 XII-36

G

GETMAIN V-31, VI-66, VIII-1, IX-3,
 XII-2, XII-13, XII-14, XII-22, XII-23,
 XII-27, XII-28, XII-30, XII-31, XII-41, A-5
 Graph of % of CPU Time Used..... V-12,
 V-20, VI-47, VI-55
 Graph of % of Dbase Elaps Time V-14,
 V-20, VI-49, VI-55

I

IF
 %Body Exec V-30, VI-65
 %Cond Exec.... V-28, V-30, VI-63, VI-65
 Body Exec V-28, VI-63
 Conds Exec V-28, VI-63
 Total Body V-28, VI-63
 Total Conds V-28, VI-63
 IF NO RECORDS FOUND V-7, VI-42, VIII-
 4, VIII-6, VIII-7, X-12, X-18, XI-3
 Inline Subroutines
 %Blocks Exec.. V-24, V-30, VI-59, VI-65
 Blocks V-24, VI-59
 Graph of Blocks Exec..... V-24, VI-59
 Total Blocks V-24, VI-59
 Invoking PROFILER I-14, II-2, II-7, III-2,
 X-2, X-3, X-4, X-5, X-6, X-7, X-8, X-9,
 X-10, X-12, X-13, X-14, X-15, X-16,
 X-18, X-19, X-20, X-21, XII-15, XII-19,
 XII-37

J

JCL X-2, X-5, X-7, X-8, X-10, XII-3,
 XII-9, XII-10, XII-11, XII-15, XII-19
 JOSCFGA XII-14

L

Left..... II-12, V-33, VI-68
 Library mask IV-9, IV-13, IV-17, IV-22,
 IV-28, VI-3, VI-4, VI-5, VI-11, VIII-13, IX-4
 License I-15, III-4, VI-9, VII-6
 Limits V-32, VI-67, VIII-1, XI-10
 Line markings V-33, V-36, VI-68
 List Active Users I-15, III-3, A-8
 List Tagged VI-6
 LNKOLSC A-7
 LOAD XII-3, XII-9, XII-10, XII-15,
 XII-19, XII-24, XII-29, XII-33, XII-38
 LOGON I-7, II-7, VII-2, VII-3, VII-14,
 X-2, X-3, X-4, X-5, X-6, X-7, X-8, X-9,
 X-10, X-13, X-14, X-16, X-19, X-20,
 X-21, XII-17, XII-19, XII-26, XII-34,
 XII-35, XII-38, XII-43

M

Messages I-21, VIII-11, XII-26, XII-43,
A-1, A-4, A-8
 Milliseconds V-6, V-11, V-12, V-13,
V-14, V-15, V-16, V-17, V-19, V-23,
V-25, V-27, V-29, V-32, V-34, VI-41,
VI-46, VI-48, VI-49, VI-50, VI-51, VI-52,
VI-54, VI-58, VI-60, VI-62, VI-64, VI-67,
VI-68, VI-69, VIII-9, VIII-10, XI-6, XI-7,
XI-8, XI-11, XI-14
 Modify Report VI-8, VI-9, VI-15, VI-16,
VI-19, VI-21
 Modify Session I-15, III-3, IV-2, IV-11,
VI-33, VII-1
 Modify Session Group VI-33
 Modify User Group VI-26
 Monitor from VII-8, VII-11, VII-13
 Monitor to VII-8, VII-11, VII-13, A-9
 Move Tagged Data VI-1, VI-2, VI-3,
VI-5, IX-7, X-9, A-7

N

NAT0888 Error A-7
 NAT0954 Error B-1
 NAT1016 Error VI-3, VI-4, VI-5, A-7
 NAT3009 Error A-7
 NAT3021 Error A-7
 NAT3049 Error A-7
 NAT3061 Error A-7
 NATLOADXII-3, XII-4, XII-9, XII-15, XII-27,
XII-28, XII-32, XII-33
 NATPARM XII-20, XII-21, XII-23, XII-24,
XII-26, XII-39, XII-40, XII-41, XII-42,
XII-43, A-1, A-7, A-8
 NATUNLD XII-27
 NATURAL 2.1 XII-1
 NATURAL 2.2 XII-1, XII-29
 NATURAL Optimized Compiler ... I-8, II-12,
V-33, VI-68
 NATURAL SECURITY XII-15, XII-17,
XII-19, XII-32, XII-34, XII-35, XII-37
 New Parameter Set VI-9, VI-10
 New Session I-15, II-3, III-4, IV-2, IV-3,
IV-6, VI-6, VI-29, VI-30, VI-31, VII-1,
VII-6, VII-7, VIII-13, A-9
 NOC I-8, II-12, V-33, VI-68
 Non-executable V-2, V-34, VI-38, VI-69,
B-1
 Non-procedural Blocks
 %Blocks Exec . V-24, V-30, VI-59, VI-65
 Blocks V-24, VI-59
 Graph of Blocks Exec V-24, VI-59
 Total Blocks V-24, VI-59
 NTFILE XII-20, XII-21, XII-26, XII-39,
XII-40, XII-43, A-1, A-7, A-8

NTRACE XII-23, XII-24, XII-41, XII-42

O

Object Mask X-3, X-4
 Object Name V-33, V-36, V-38, VI-68
 Objects Not Executed V-38, VI-71
 ON ERROR V-7, VI-42, X-12, X-18
 Optimized Compiler .. I-8, II-12, V-33, VI-68
 Overview I-5, VII-1
 Owned I-14, III-1, III-2, III-3, VI-7, VII-5

P

Password IV-3, IV-11, VII-4, VII-5, IX-1,
IX-2, X-2, X-3, X-4, A-9
 Percent Graph I-8, III-4, V-1, V-7, V-19,
VI-8, VI-37, VI-42, VI-54, X-11, X-12,
X-17, XI-4
 Performance I-2, I-3, I-11, XI-1, XI-4
 PREDICT VIII-2, XII-9, XII-15, XII-19,
XII-21, XII-27, XII-32, XII-38, XII-40
 PRF Command I-14, II-2, II-7, III-2, X-2,
X-3, X-4, X-5, X-6, X-7, X-8, X-9, X-10,
X-12, X-13, X-14, X-15, X-16, X-18,
X-19, X-20, X-21, XII-15, XII-19, XII-37
 PRFBCKN XII-17, XII-34, XII-35
 PRFFIN XII-18, XII-36
 PRFLOGON XII-17, XII-34
 PRFVRFY XII-26, XII-43, A-1, A-10
 Problems I-2, XII-26, XII-43, A-1
 Production I-2, I-11, V-37, VI-70, XI-1
 PROF4Nnn XII-24
 PROF4Nxx XII-4
 PROFCFG XII-4, XII-13
 PROFCLOS.. XII-23, XII-26, XII-27, XII-41,
XII-43, A-2, A-4, A-9, A-11
 PROFCONF XII-2, XII-24
 PROFDRV XII-23, XII-26, XII-27,
XII-41, XII-43, A-2, A-5, A-11
 Profile Sessions I-14, I-15, III-1, III-3,
IV-1, IV-2, IV-3, IV-5, IV-6, IV-8, IV-9,
IV-10, IV-12, IV-13, IV-14, IV-16, IV-17,
IV-18, IV-19, IV-21, IV-22, IV-23, IV-27,
IV-28, V-4, V-5, VII-5, IX-6, IX-7, A-11
 PROFILER-REPORTING VI-1, XII-3,
XII-4, XII-9, XII-18, XII-19, XII-21,
XII-27, XII-28, XII-36, XII-38, XII-40
 PROFINIT XII-2, XII-14, XII-22, XII-23,
XII-26, XII-27, XII-28, XII-30, XII-31,
XII-41, XII-43, A-1, A-2, A-4, A-5, A-8
 PROFMDRV A-2
 PROFRDC XII-27
 PROFVDRV XII-27, A-2

PROLIB..... IV-19, V-3, VIII-2, IX-6, X-10,
 XII-15, XII-17, XII-18, XII-19, XII-26,
 XII-32, XII-34, XII-35, XII-36, XII-37,
 XII-43, A-1, A-10
 PROTRACE..... XII-27
 Purge I-15, II-17, III-3, IV-15, IV-18,
 VI-8, VI-9, VI-22, VI-27, VI-29, VI-34,
 VII-6, IX-6, IX-7, A-10
 Purge All IX-6, IX-7

Q

QA.....I-7, I-9, I-15, I-19, II-10, II-14, III-3,
 III-4, V-1, V-2, V-3, V-4, V-5, V-37,
 V-38, V-39, VI-37, VI-38, VI-39, VI-40,
 VI-70, VI-72, VII-5, X-16, X-21, XI-2
 Quality Assurance.... I-2, I-7, I-9, I-15, I-19,
 II-10, III-4, V-1, V-37, VI-8, VI-37,
 VI-70, XI-1, XI-2

R

RDCVIII-10, XII-1, XII-23, XII-24,
 XII-41, XII-42
 RDCEX1XII-3, XII-24
 REPEAT Loops.....XI-4, XI-10
 %Loops Exec...V-26, V-30, VI-61, VI-65
 Graph of Loops Exec..... V-26, VI-61
 Loops Exec..... V-26, VI-61
 Total Loops V-26, VI-61
 Repeat previous SCAN..... II-12, V-33,
 V-34, VI-68, VI-69
 Report Parameter Set.... VI-1, VI-7, VI-10,
 VI-12, VI-16, VI-19, VI-21, A-11
 Reset Active Users IX-3, IX-4, IX-5
 Reset Statistics I-15, III-3, IV-26, IV-29
 Review Data Collector VIII-10, IX-8,
 XII-1, XII-4, XII-24, XII-27, XII-42
 Right.....II-12, V-33, VI-68
 Run Count..... V-1, V-10, V-12, V-14, V-16,
 V-18, V-20, V-22, V-24, V-26, V-28,
 V-30, VI-37, VI-44, VI-46, VI-48, VI-51,
 VI-53, VI-55, VI-57, VI-59, VI-60,
 VI-62, VI-65, X-13, X-19

S

S322 Error..... B-2
 SAP..... A-7
 SAVEd V-31, V-36, VI-66, VIII-11
 SCAN for text..... II-12, V-33, V-34, VI-68,
 VI-69
 Scroll.....II-12, V-33, VI-68
 Session Actions I-17, III-1, III-2, III-3,
 III-4, V-4, VI-2, VII-5, VII-6
 Session Commands..... I-17, III-3, VII-6
 Session Group VI-7, VI-29, VI-30, VI-31,
 VI-32, VI-33, VI-34, VI-35, VI-36, A-11

Session List Menu I-5, I-7, I-13, I-14,
 I-15, I-17, I-20, I-21, II-2, II-3, II-4, II-5,
 II-7, II-8, II-9, II-11, II-15, II-17, III-1,
 III-2, III-3, III-4, III-5, IV-2, IV-3, IV-6,
 IV-10, IV-14, IV-18, IV-23, IV-25,
 IV-29, IV-30, IV-31, V-1, V-4, V-6, V-9,
 V-11, V-13, V-15, V-17, V-19, V-21,
 V-23, V-25, V-27, V-29, V-35, V-36,
 V-37, VI-2, VI-3, VI-4, VI-5, VI-6, VI-7,
 VI-9, VII-4, VII-5, VII-6, VII-7, VII-9,
 VII-10, VII-11, VII-12, VII-13, VII-14, VII-15
 settings and options III-2
 SOC4 Error B-1
 Sort Order I-19, II-10, V-5, VI-40, X-10,
 X-12, X-18
 Source Code Listing VI-58
 Source Code Listing Report..... I-10, II-2,
 II-11, II-12, IV-3, IV-5, IV-11, V-2, V-3,
 V-9, V-11, V-13, V-15, V-17, V-19,
 V-21, V-23, V-25, V-27, V-29, V-31,
 V-32, V-34, V-35, V-36, V-38, VI-38,
 VI-44, VI-46, VI-48, VI-50, VI-52,
 VI-54, VI-56, VI-58, VI-60, VI-62,
 VI-64, VI-66, VI-67, VI-69, VI-71,
 VIII-1, VIII-2, VIII-3, VIII-5, VIII-6,
 VIII-7, VIII-8, VIII-10, VIII-11, X-14,
 X-15, X-20, XI-2, XI-9, XI-11, XI-12,
 XI-13, A-10, B-1
 Source DBIDIV-5, IV-9, IV-13, IV-17,
 IV-22, IV-28, X-14, X-20
 Source FUSERX-14, X-20
 Source Library X-14, X-20, XII-3
 Start Library I-19, II-10, V-5
 Start Object..... I-19, II-10, V-5
 Statement Execution Summary I-19,
 II-13, XI-4
 statement optionsII-12, V-33, V-34,
 VI-68, VI-69
 statements executed..... IV-10, IV-14,
 IV-18, IV-23, IV-28, V-6, V-7, V-9, V-10,
 V-15, V-16, V-17, V-19, V-20, V-21,
 V-23, V-25, V-27, V-29, VI-41, VI-42,
 VI-44, VI-45, VI-50, VI-51, VI-52,
 VI-54, VI-55, VI-56, VI-58, VI-60,
 VI-62, VI-64, X-11, X-12, X-17
 Statistics Help I-15, II-9, III-4, VI-8
 Statistics Report..... I-15, III-4, VI-8
 STEPLIBVIII-2, X-14, X-20, XII-18,
 XII-36, A-5
 Subroutines (Inline)
 %Blocks Exec..V-24, V-30, VI-59, VI-65
 Blocks V-24, VI-59
 Graph of Blocks Exec.....V-24, VI-59
 Total Blocks V-24, VI-59
 Summary Reports..... V-1, V-5, VI-37,
 VI-40, X-10, X-11, X-17, XI-4

SYSDICBE XII-3, XII-9, XII-19, XII-27,
XII-38
 SYSEXT XII-16, XII-17, XII-34, XII-35
 SYSMAIN V-3, VI-38, VIII-2, XII-16,
XII-17, XII-18, XII-34, XII-35, XII-36, B-2
 SYSRDC VIII-7, VIII-10, IX-8
 SYSTEM..... VIII-2, XII-15, XII-16, XII-18,
XII-19, XII-32, XII-34, XII-36, XII-37

T

Tag Data VI-3, VI-4, A-7
 Tag Data For Move X-7
 Tagged Data Batch Operations X-7
 Tape XII-9, XII-10, XII-11, XII-18,
XII-19, XII-21, XII-27, XII-29, XII-30,
XII-32, XII-33, XII-36, XII-38, XII-40
 TERMINATE Statements VIII-7, IX-3
 Testing..... I-2, I-4, XI-1, XI-10
 Threshold Statistics I-9
 Top of Object..... II-12, V-33, VI-68
 Total DB/WF Loops V-22, VI-57
 Total Statement Executions V-32,
VI-67, VIII-1
 Trace Sessions..... I-3, I-6, I-14, III-1, III-3,
IV-1, IV-3, IV-5, IV-9, IV-10, IV-13,
IV-14, IV-17, IV-18, IV-19, IV-22,
IV-28, IV-30, IX-7, A-10
 Treehouse Software I-1, XII-27, XII-29,
A-1, A-2, A-3, A-4, A-5, A-7
 TREPREF1 XII-26, XII-43, A-8
 TSIRDC XII-3, XII-4, XII-24, XII-27
 TSO VIII-1, VIII-8, XII-22, XII-25, A-5,
A-7, B-1

U

UB XII-14, XII-27, XII-30
 un-executed..... V-2, V-34, V-35, VI-38,
VI-69
 Unexecuted V-38, VI-71
 Untag Data VI-4, A-7
 Up one Page II-12, V-33, VI-68
 User Group..... VI-7, VI-13, VI-22, VI-23,
VI-24, VI-25, VI-26, VI-27, VI-28, VIII-8,
A-11
 User Interface I-12
 USERBUF XII-2, XII-4, XII-13, XII-22,
XII-23, XII-30, XII-41, A-5
 User-Exits XII-18, XII-36
 USERINFO XII-14, XII-30
 USR0050N XII-17, XII-35
 USR1025N III-2

V

Verify XII-4, XII-26, XII-28, XII-43,
A-1, A-2, A-3, A-11
 View Executed Objects V-38, VI-71
 View/Amend Thresholds I-19, II-10,
II-13, V-5, V-8, VI-40, VI-43, XI-10
 VM X-2, X-5, X-7, X-8, X-10, XII-1,
XII-27, XII-28, XII-42, A-2, A-5
 VSAM I-9, V-3,
VI-38

Z

z/OS XII-3
 Zaps XII-4, XII-12, XII-22, XII-23,
XII-28, XII-29, XII-30, XII-41