Treehouse SOFTWARE

AUDITRE

Standardized Auditing Facility for ADABAS

Product Overview

TREEHOUSE SOFTWARE, INC.
2605 Nicholson Road, Suite 1230
Sewickley, PA 15143
Phone: 724.759.7070
Fax: 724.759.7067
E-mail: tsi@treehouse.com
http://www.treehouse.com

This page intentionally left blank.

Introduction

This product overview highlights the key features of **AUDITRE**. The following sections are presented:

- The Need for a Standardized ADABAS Auditing Facility
- The ADABAS Auditing Challenge
- Embedded Audit Routines are not the Solution
- The Impact of the Protection Log on ADABAS Auditing
- AUDITRE's Function
- AUDITRE Provides the Best Auditing Solution
- AUDITRE Makes ADABAS Auditing Easier
- AUDITRE Streamlines Audit Data
- AUDITRE Summarizes Update Activity
- AUDITRE Provides "After the Fact" Selective Protection Logging
- AUDITRE Benefits the Entire Organization
- Which Organizations Can Benefit from AUDITRE?
- User-Friendly Documentation
- Quick and Easy Installation
- Customer Support
- Demo Diskettes
- Conclusion

The Need for a Standardized ADABAS Auditing Facility

A variety of sources, including NATURAL programs, Direct Calls, ADASQL, etc., make updates to ADABAS files. This makes ADABAS very flexible, but introduces a serious weakness. The weakness is that there is no standardized method for auditing changes to the database from these various sources. The lack of a standardized auditing capability for ADABAS presents the auditor with many serious challenges.

The ADABAS Auditing Challenge

Compliance testing presents the biggest challenge to the organization's Auditors. Compliance testing requires the Auditor to learn if applications use the proper update procedure to change ADABAS data. Compliance testing requires a record of exactly what changes an application program made to the database and the order in which the changes took place. ADABAS provides no automatic method to generate audit data, making compliance testing difficult. This challenge leads some sites to invest substantial resources in the development of Embedded Audit Routines (EARs).

Embedded Audit Routines are not the Solution

EARs generate the data Auditors need. EARs store audit data in special files which may later be used to generate audit reports. However, EARs can cause more harm than good.

EARs increase processing overhead and DASD storage requirements, hampering the efforts of the DP department to function at maximum efficiency. EARs add overhead by requiring applications to perform additional logic to generate and store the audit data each time the user updates a record. The data these routines generate consumes space on the system until archived or erased. The resource problems caused by EARs may lead the site to use them only where absolutely necessary, or avoid EARs and hope that users do not make unauthorized changes to data.

With EARs, audit data generated by an application is only as reliable as the programmer who codes the audit routines. A dishonest programmer could code a program to generate "legitimate" audit data when "incorrect" or illegal transactions are taking place. The Auditors and security personnel may remain unaware of unauthorized database changes.

The program producing the audit report may have to access multiple audit data files to generate a single report. Each audit file may have a completely different format and content. The report program must read the different formats and produce the desired audit report. Considerable time and effort may be involved in writing the program, potentially increasing development backlog. The ADABAS Protection Log offers a better way to audit ADABAS applications.

The Impact of the Protection Log on ADABAS Auditing

With Protection Logging turned on, ADABAS automatically generates a log of all changes to the database. The intended function of the Protection Log is for Backout/Regeneration of updates (i.e., protection of the database). Use of the Protection Log information for auditing provides several benefits.

Using the Protection Log eliminates some of the programming difficulties caused by different applications updating the same database. Information about updates made by multiple applications to the same file is stored in the same place in the same format. Generating an audit report is easier and faster than it might have been with EARs.

Using the Protection Log adds little overhead to the application systems, and no overhead if the site currently uses Protection Logging. ADABAS stores the Protection Log in a compressed format, significantly reducing audit data storage requirements.

Using the Protection Log as a source of audit data ensures that reports come from reliable, complete, and unmodified data. Neither application logic nor programmer opinions determine the information gathered.

The Protection Log contains information about all updates to all ADABAS files from all sources at all times, so the Auditor can use the Protection Log data to audit any application at any time, without the knowledge of those who wrote or use the application software. This allows the Auditor to perform spot checks on any critical or trouble areas of the database as desired.

Finally, because the Protection Log is archivable, audit data can be maintained off-line indefinitely and recalled as needed. If Auditors discover a long-term fraud involving the site's Inventory Control system, the data necessary to document the history of the fraud is available on the old Protection Logs.

AUDITRE's Function

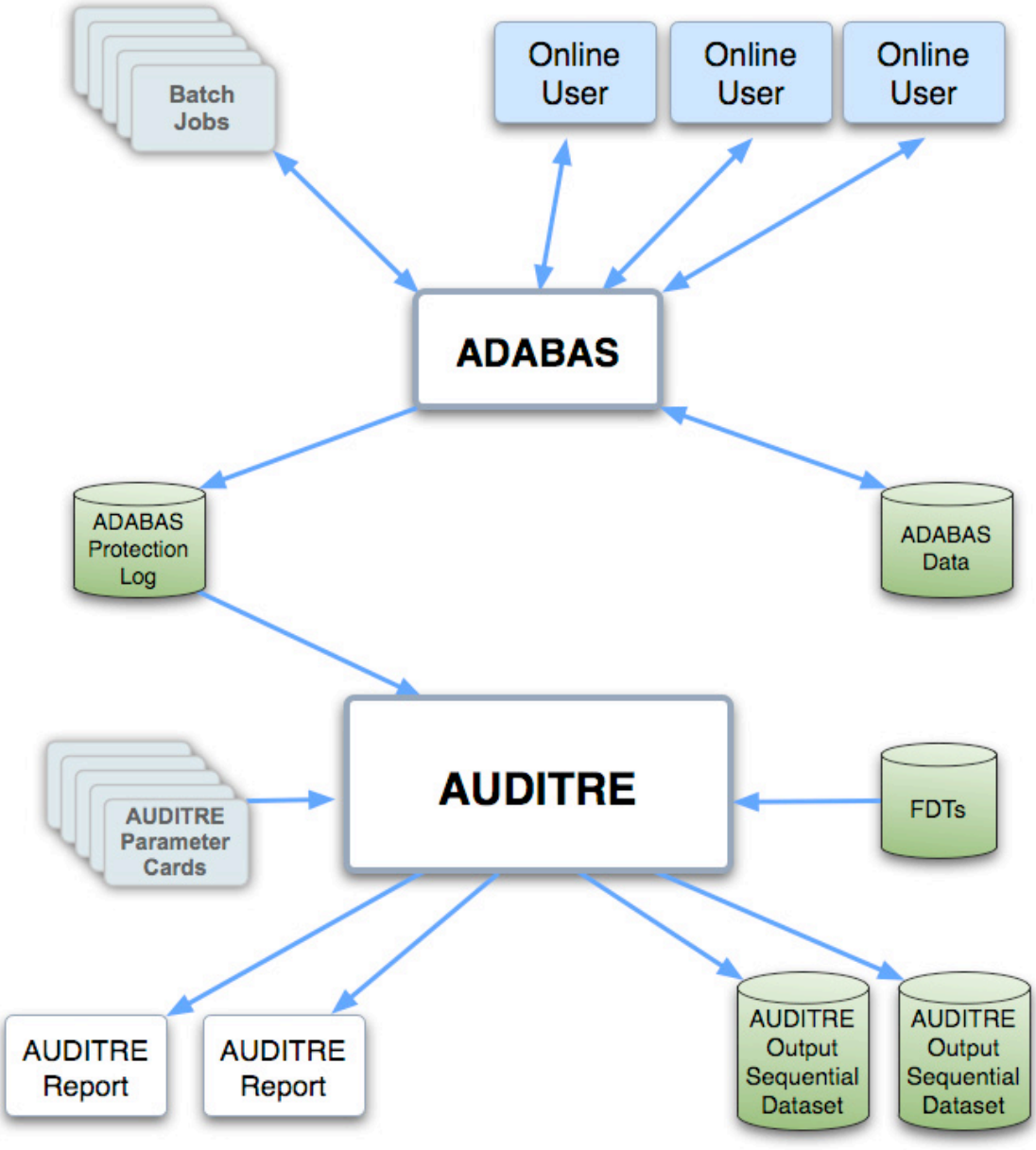
As illustrated in Figure 1, a batch job **(A)** or an on-line user **(B)** executes an ADABAS transaction. ADABAS **(C)** updates the physical database components **(D)** and writes information about the transaction to the ADABAS Protection Log **(E)**.

The information on the Protection Log **(E)** is voluminous and is not in an easily readable form. AUDITRE **(F)** processes the Protection Log data.

Taking direction from the Auditor through AUDITRE parameter cards **(G)**, and input from the Protection Log **(E)** and the Field Definition Table **(H)**, AUDITRE **(F)** makes one pass over the audit data. In this one pass, AUDITRE determines the number and kinds of transactions (Adds, Updates, or Deletes) made and produces multiple reports **(I)** or outputs the results to a sequential dataset(s) **(J)** for later processing.

FIGURE 1

AUDITRE'S FUNCTIONAL PARTS



AUDITRE Provides the Best Auditing Solution

AUDITRE requires no additional support systems such as extra programs, "audit exit routines," or specialized files. Audit reports can be generated on other CPUs at remote sites for security, if desired.

Further, AUDITRE requires minimal installation and training time for the DBA and other users. Creation of new reports requires coding only a few simple parameter statements. For instance, coding a report to show all updates to file 27 would take only a few parameter statements like these:

```
REPORT  HEADING='UPDATES TO FILE 2789'
INCLUDE FNR=2789...
AUDIT           AA*,BB,DD,CC...
```

AUDITRE produces clear, organized reports of database modification activity including only the information desired. AUDITRE can base reports on files, fields, sets of fields, and logically related updated files. Reports contain only the files and fields desired. One pass of the Protection Log allows AUDITRE to generate multiple reports simultaneously.

AUDITRE promotes the integrity and safety of the ADABAS-related applications by reporting on changes to NATURAL programs, by reporting changes to FUSER files for fields LJ and LK (NATURAL source lines).

For maximum long-term benefit, selected data or reports can be archived to tape or disk for processing later. This allows audits to occur "after the fact" for certain critical systems if Auditors later discover a problem.

AUDITRE Makes ADABAS Auditing Easier

AUDITRE makes audits easier. Consider a Protection Log containing over one million records. ADABAS stored these records in a compressed, non-standard form, which makes them difficult to read. The Log contains all updates for all files for all fields (entire records) whether all fields changed or not. Records on the log are ordered chronologically, and are difficult to sort because ADABAS stored them in the non-standard form.

Various ADABAS utilities display Before and/or After Images of updated records as one large field in character and in hex. However, determining which field(s) changed from such a "dump" is difficult, especially when updates cause different-sized values, or when fields change to/from null values. For example, examine the following Before and After Images:

BEFORE:

```
 F5  DINERS CLUB &     AMERICAN EXPRESS
 -    ┐  AMOCO        ┐   BANKERS LIFE & C
ASUALTY      BRIGHAM YOUNG D Y CATTLE
 MORGAN GUARANTY TRUST, N.Y.
```

AFTER:

```
  F5  DINERS CLUB &     AMERICAN EXPRESS
 -    ┐  AMOCO     |  ┐   BANKERS LIFE & C
ASUALTY       BRIGHAM YOUNG D Y CATTLE
  MORGAN GUARANTY TRUST, N.Y.
```

The above dump reveals some useful, readable information. Changes to alphanumeric fields can sometimes be detected visually. A "hex" dump, shown below, might be more useful, especially for the packed and binary numbers. However, decompressing the following hex data without the file layout (FDT) or "sight decompressing" the data if the file layout is known, can be difficult and error-prone.

BEFORE:

```
040186F5020CC4C9D5C5D9E240C3D3E4C203500F03060F11C1D4C5D9C9C3C1D540C5E7D7D9C5E2E2
03600F03025F0206C1D4D6C3D602400403333F025F010118C2C1D5D2C5D9E240D3C9C6C5405040C3
C1E2E4C1D3E3E8010435000F0EC2D9C9C7C8C1D440E8D6E4D5C705C402E807C3C1E3E3D3C503150F
1CD4D6D9C7C1D540C7E4C1D9C1D5E3E840E3D9E4E2E36B40D54BE84B
```

AFTER:

```
040186F5020CC4C9D5C5D9E240C3D3E4C203500F03060F11C1D4C5D9C9C3C1D540C5E7D7D9C5E2E2
03600F03025F0206C1D4D6C3D602400404444F025F010118C2C1D5D2C5D9E240D3C9C6C5405040C3
C1E3E4C1D3E3E801050135000F0EC2D9C9C7C8C1D440E8D6E4D5C705C402E807C3C1E3E3D3C50315
0F1CD4D6D9C7C1D540C7E4C1D9C1D5E3E840E3D9E4E2E36B40D54BE84B
```

AUDITRE Streamlines Audit Data

Suppose a programmer writes code to decompress and display an updated record. This display program has only limited use. When thousands of records change on each file, printing Before and After images of the changed records in chronological order is unacceptable. A printout of several thousand pages might have to be searched to find the Before and After images for a single updated record. Investigating the changes to thousands of records could take hours. Consider a large record like this one:

```
AN = 293874628                                  CUST-ACCOUNT-NUMBER
CL = 1500.00                                    CREDIT-LIMIT
CS = SEWICKLEY, PA 15143                        CUST-CITY-ST-ZIP
CN = JOHN DOE                                   CUSTOMER-NAME
CD = 700 MAIN STREET                            CUSTOMER-ST-ADDRESS
FY = 18.00                                      INT-RATE-YEARLY
OCC = 3                                         OTHER-CARDS-COUNT
OC  1 = DINERS CLUB                             OTHER-CARDS
OC  2 = AMERICAN EXPRESS                        OTHER-CARDS
OC  3 = VISA                                    OTHER-CARDS
OLC = 3                                         OTHER-LIMITS-COUNT
OL  1 = 2000                                    OTHER-CARD-LIMIT
OL  2 = 1500                                    OTHER-CARD-LIMIT
OL  3 = 1800                                    OTHER-CARD-LIMIT
CO = CLERK                                      CURRENT-OCCUPATION
YI = 19500                                      YEARLY-INCOME
PH = 412-555-1677                               HOME-PHONE
BP = 412-555-3048                               BUSINESS-PHONE
YJ = 5                                          YEARS-AT-JOB
ED = 10/19/99                                   CARD-EXPIRE-DATE
DB = 01/15/66                                   DATE-OF-BIRTH
MS = S                                          MARITAL-STATUS
NC = 0                                          NUMBER-CHILDREN
DL = 288726439                                  DRIV-LIC-NUMBER
DS = GA                                         DRIV-LIC-STATE
```

Such a record appears in compressed, unreadable form on the Protection Log if any of its fields changed. Manually scanning thousands of Before and After images to determine which were updated by matching records and fields to one another would be labor-intensive. Locating changes made to "important" fields would be difficult and error-prone. For example, consider these two records:

Before		After
AN = 293874628		AN = 293874628
CL = 1500.00	**<- - - - - ->**	**CL = 9999.99**
CS = SEWICKLEY, PA 15143		CS = SEWICKLEY, PA 15143
CN = JOHN DOE		CN = JOHN DOE
CD = 700 MAIN STREET	**<- - ->**	**CD = 172 SCAIFFE ROAD**
FY = 18.00		FY = 18.00
OCC = 3		OCC = 3
OC 1 = DINERS CLUB		OC 1 = DINERS CLUB
OC 2 = AMERICAN EXPRESS		OC 2 = AMERICAN EXPRESS
OC 3 = VISA		OC 3 = VISA
OLC = 3		OLC = 3
OL 1 = 2000		OL 1 = 2000
OL 2 = 1500	**<- - - - - ->**	**OL 2 = 1600**
OL 3 = 1800		OL 3 = 1800
CO = CLERK		CO = CLERK
YI = 19500		YI = 19500
PH = 412-555-1677	**<- - ->**	**PH = 412-555-2805**
BP = 412-555-3048		BP = 412-555-3048
YJ = 5	**<- - - - - ->**	**YJ = 6**
ED = 10/19/99		ED = 10/19/99
DB = 01/15/66		DB = 01/15/66
MS = S	**<- - - - - ->**	**MS = M**
NC = 0		NC = 0
DL = 288726439	**<- - ->**	**DL = 502378091**
DS = GA	**<- - - - - ->**	**DS = PA**

For the example on the previous page, assume that as a credit card issuer, the only important field to the Auditor is CREDIT-LIMIT (CL). The Auditor wishes to investigate unapproved changes to a customer's credit limit. The fact that a certain customer received a new phone number doesn't concern the Auditor. Instead of the lengthy printout of each changed record (as seen above), it would be helpful to get a printout that showed only the following data:

```
*  AN = 293874628                                    CUST-ACCOUNT-NUMBER
*  CN = JOHN DOE                                     CUSTOMER-NAME
B: CL = 1500.00                                      CREDIT-LIMIT
A: CL = 9999.99                                      CREDIT-LIMIT
```

To get such a report using AUDITRE, the Auditor codes a simple statement like this:

```
              AUDIT          AN*,CN*,CL,FNR=12345
```

When a customer's credit limit changed, the Auditor needs the name and account number of that customer. This is an important use of "key fields". The Auditor marked CUST-ACCOUNT-NUMBER (AN) and CUSTOMER-NAME (CN) with asterisks in the above statement to indicate that AUDITRE should print the customer name and account number whenever the Protection Log shows that a customer's CREDIT-LIMIT (CL) was updated.

AUDITRE automatically performs the task of matching Before and After images of the same record and determining if important fields were updated. The Auditor determines if the changes warrant further investigation.

AUDITRE Summarizes Update Activity

With AUDITRE, it is possible to see all the updates to the important fields on the CUSTOMER-DATA file (or any other ADABAS file). Reports show who changed what, when they changed it, and what they changed it to and from. The Auditor may select data by date, time, file, user, etc. At the end of each report, AUDITRE produces a summary. This summary is useful, for example, if the Auditor knows that a change in the credit limit was only authorized for 10 people, yet 12 changed. A typical summary looks like this:

```
FILE:   12345           DELETES: 0          UPDATES: 24      ADDS:  0

FIELD LONG-NAME             OCC FROM  OCC TO    UPDATES  DELETES   ADDS
AN    CUST-ACCOUNT-NO                               0        0      0
CL    CREDIT-LIMIT                                 12        0      0
CS    CUST-CITY-ST-ZIP                              0        0      0
CN    CUSTOMER-NAME                                 0        0      0
CD    CUSTOMER-ST-ADDRESS                           2        0      0
FY    INT-RATE-YEARLY                               0        0      0
OCC   OTHER-CARDS-COUNT                             0        0      0
OC    OTHER-CARDS                 1       10        4        0      0
OLC   OTHER-LIMITS-COUNT                            0        0      0
OL    OTHER-CARD-LIMIT            1       10        4        0      0
CO    CURRENT-OCCUPATION                            0        0      0
YI    YEARLY-INCOME                                 0        0      0
PH    HOME-PHONE                                    1        0      0
BP    BUSINESS-PHONE                                0        0      0
YJ    YEARS-AT-JOB                                  2        0      0
ED    CARD-EXPIRE-DATE                              0        0      0
DB    DATE-OF-BIRTH                                 0        0      0
MS    MARITAL-STATUS                                1        0      0
NC    NUMBER-CHILDREN                               0        0      0
DL    DRIV-LIC-NUMBER                               0        0      0
DS    DRIV-LIC-STATE                                0        0      0
```

AUDITRE is not limited to reporting or selecting from just one updated file. Records and fields may be selected from across logically related updated files.

The Auditor can also get summary reports of update activity by file by coding these simple statements:

```
REPORT          TYPE=SUMMARY
CONTROL         FNR
```

These statements would produce a summary of all updates made to the database, broken down by file:

```
FNR          COUNT          %
  1           3827        16.8
  3          12103        53.0
  9            102         0.5
 12           6789        29.7

 **          22821       100.0
```

Notice that the above summary report indicates that updates were made to files 1, 3, 9, and 12. If the Auditor expected only 95 updates to the PAYROLL-MASTER file (file 9), the fact that 102 records were updated on that file could warrant an investigation. AUDITRE allows the Auditor to perform this investigation by creating additional summary reports by user, hour, etc. The Auditor might choose to code a summary report by user and hour:

```
REPORT    TYPE=SUMMARY
          HEADING='DEEPER INVESTIGATION OF FILE 9'
INCLUDE   FNR=9
CONTROL   UID,HOUR
```

This could produce the following output:

```
DEEPER INVESTIGATION OF FILE 9

HR      USER-ID     COUNT       %
10      PYR3            5      4.9
10      PYR1           12     11.8
10      ****           17     16.7

11      PYR1           26     25.5
11      PYR2           17     16.7
11      PYR3            4      3.9
11      ****           47     46.1
15      PYR1            3      2.9
15      PYR3           10      9.8
15      ****           13     12.7

17      PYR1            9      8.8
17      PYR2            9      8.8
17      ****           18     17.6

20      RECV            7      6.9
20      ****            7      6.9

**      ****          102    100.0
```

The Auditor now knows that the user "RECV" (an employee in the Receiving department) manipulated PAYROLL-MASTER file records after office hours. To determine exactly what this user changed, the Auditor would code a detail report, using AUDITRE, to analyze all the changes to PAYROLL-MASTER file records by User-Id RECV, showing only the changed fields. It might be a good idea to run a similar report on any old, archived Protection Logs, as the employee might have been changing payroll records for months or years. The Auditor would certainly want to know which records changed, when, and from what to what.

AUDITRE Provides "After the Fact" Selective Protection Logging

Another useful function of AUDITRE is the OUTPUT statement. Using this statement, the contents of Protection Logs can be "subdivided" and stored on datasets for later processing. This is of great help, for example, when the update data from a Production database normally takes up 20 or more reels of tape. If the Auditor is only concerned about archiving the data for updates occurring on a few of the files, AUDITRE will select only the Protection Log records for these files. This might reduce the tape volume to only 3 reels (or even 1!). Only a few parameter statements like these must be coded:

```
INCLUDE FNR=(7,789,21-24,45)
OUTPUT

or

INCLUDE FNR=7
OUTPUT
INCLUDE FNR=789
OUTPUT
•
•
•
```

This AUDITRE function can be viewed as "after the fact" selective Protection Logging.

AUDITRE Benefits the Entire Organization

AUDITRE benefits many people in the organization, besides auditors. A standardized auditing capability benefits system design and development efforts. Detail reports can show changes made to the database by user, hour, program, etc., aiding in the debugging process for an application. Developers can use the reports to verify both the source and the results of updates to the database. This capability also overcomes the challenges of auditing ADABAS databases.

Which Organizations Can Benefit from AUDITRE?

AUDITRE can benefit many different types of organizations. For example in the petroleum industry, AUDITRE can verify changes to ownership interests in oil leases. Magazine publishers can use AUDITRE to monitor changes to subscribers' EXPIRATION-DATEs. Stock brokerages can use AUDITRE to verify that a stock transaction took place as ordered by the client. Colleges and universities can track changes to alumni addresses for fund-raising purposes. Any organization needing a report of changes to ADABAS data can find a good use for AUDITRE.

AUDITRE's archiving capability is useful to many departments in the same organization. The Payroll Department may discover unauthorized salary increases. Meanwhile, the Marketing Department might use summary reports for several months of Protection Log data to determine how many new customers were gained during the year by tracking adds and deletes to the CUSTOMER database. The Purchasing Department could examine RECEIVING data for changes to expected RECEIVING-DATEs, etc., to determine if any suppliers frequently postpone merchandise delivery, causing supply shortages for the organization.

There are many other uses for AUDITRE. If a question can be answered by examining changes to the database, or a problem investigated by reporting on archived update data, AUDITRE can help.

User-Friendly Documentation

AUDITRE documentation is contained in a single reference manual. The clear, concise description of this powerful product allows the user to produce complete, meaningful reports within one hour of installation. For installation itself, only a few pages must be read. Error messages are fully described, and all limitations and restrictions are well documented. The reference manual is fully indexed for convenience.

Quick and Easy Installation

AUDITRE operates under OS/VS1, MVS (including XA and ESA), DOS/VSE (including SP), VM/CMS, and Siemens BS/2000. Installation should require only a few minutes, and a few cylinders of disk space for the source and load modules on the tape.

Customer Support

Treehouse Software, Inc. supports AUDITRE and its other products from its headquarters in Pennsylvania. Support representatives in several foreign countries serve international customers in their area. AUDITRE users have always had direct input to the product developers. User questions are answered quickly, problems are discussed directly, and change/enhancement requests are reviewed and implemented. AUDITRE is so easy to use that formal classroom instruction is not necessary.

AUDITRE is available under a 30-day free trial agreement. AUDITRE is attractively priced. Many customers report saving the cost of AUDITRE in a few weeks. Contact us for more information about pricing and payment plans.

Conclusion

AUDITRE offers a powerful, generalized auditing system for ADABAS. AUDITRE can benefit all levels of the organization:

- Data Processing Management
- Database Administrators
- Systems Analysts
- Application Programmers
- Data Processing Auditors
- End Users

AUDITRE provides these advantages:

- centralization of Audit data
- uniformity of Auditing techniques across applications
- cost reduction, savings in development time, DASD requirements, and CPU time
- simplicity, minimum installation and training time
- security of Audit data
- flexibility, selectivity, multiple reports on multiple files in one Protection Log pass

AUDITRE is the "ideal solution" to ADABAS auditing problems, and to some non-auditing concerns.

This page intentionally left blank.

2605 Nicholson Road, Suite 1230
Sewickley, PA 15143
Phone: 724.759.7070
Fax: 724.759.7067
E-mail: tsi@treehouse.com
http://www.treehouse.com