TREETIPS

A Publication of TREEHOUSE SOFTWARE

Issue #11- March 1992

Treehouse Survives Avalanche!

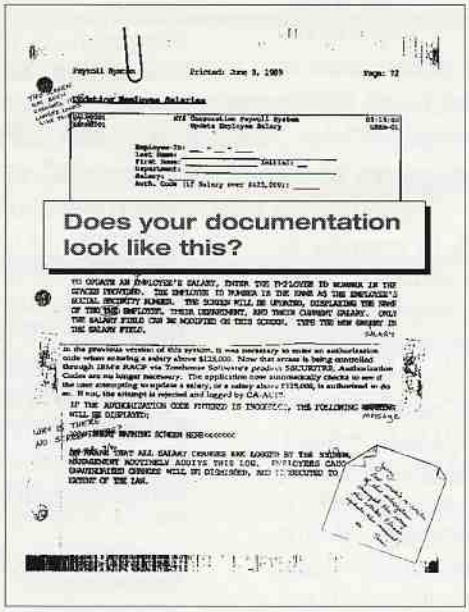
When we sent out our DynaDoc flyer (pictured on this page), offering to **document a NATURAL application free of charge**, we knew sites would respond.

One of the respondents summed up the situation at many sites, **"Our current documentation has a lot to be desired. We have always been admonished for our lack of documentation and would really like to take advantage of this offer."**

The responses rolled in like an avalanche! In the two months since our initial mailing, we have received **nearly 100 tapes** from NATURAL sites. We continue to receive tapes almost daily. Because DynaDoc is powerful and easy to use, we were able to easily dig ourselves out from under the avalanche and deliver **all** the documentation we promised **more quickly than users expected**.

When we receive the tape, we use DynaDoc to produce a **user manual**, **technical reference guide**, and other documents. These are printed on our desktop publishing system, bound, and mailed to the user.

The quantity of documentation will vary, depending upon the quality and quantity of the material sent to us. If a customer sends only a few disjointed programs, which contain no comment lines, use only a few maps, with no PREDICT objects and Help screens, we cannot provide a great deal of documentation. But we can still provide *some* documentation. **DynaDoc guarantees at least a minimum consistent set of documentation for all NATURAL applications.** On the other hand, many sites received impressive technical guides and user manuals. In all cases, the response is much the same: **"How can you produce so much high-quality documentation so quickly?"**



Some sites have invited us to **visit** them. Our representatives install DynaDoc, show the customer how it works, and answer their questions. All of this takes **less than a day**. The customer is then able to put DynaDoc to work documenting **all** their NATURAL applications.

Due to the overwhelming response, **we have decided to keep the offer of free documentation open.** If you would like to receive free documentation for your favorite (or maybe your ugliest) NATURAL application, **send us a tape**. Please, one application per customer. The DynaDoc flyer contains instructions for creating the tape. If you missed the flyer, or have questions, let us know.

Be sure to catch the DynaDoc article on page 5!

Table of Contents

Joe Gilley Joins TSI!

Joe Gilley, formerly Database Product Manager at Software AG, has joined Treehouse Software. Joe will be working for TSI out of an office in his home in Virginia.

Joe will be contacting many of you soon. In the meantime, if you would like to speak with Joe, you may call him at (703) 450-4674.

Editor's Sproutings

by Michael Salsbury

User Group Visits

TSI attended several user group meetings during the past few months, including the **Hill Country User Group** meeting. At that meeting, **John Baird** of the **Texas Attorney General's Office** presented their PC/Mainframe Transfer System. Treehouse Software's speaker, **Paul Correa**, followed John's presentation with a discussion of **DynaDoc**. Prior to the meeting, John sent us a tape with the source code for his system. **Paul distributed documentation for John's system**, and told the attendees how he **documented it using DynaDoc**.

We plan to make presentations during the next few months at several user groups, including the Midwestern Regional, the Rocky Mountain West Regional, Western Canada Regional, Northeastern US Regional, Northwestern Regional, Great Plains Local, and Delaware Valley User Groups. **We hope to be able to visit every regional user group this year.** If you want to see Treehouse at your next user group meeting, let us know and we will try to arrange it.

SAG User Group Mini-Conference?

One user group officer we've spoken to recently is hoping to organize a two-day **"mini-conference"**. The first day would be sponsored and conducted by Software AG, and would include presentations on topics currently relevant to SAG and the user group members. The second day would be a **"vendor day"**, sponsored by vendors of ADABAS/NATURAL related products and services. Vendors would be allowed to demonstrate their offerings and talk to prospective customers. TSI looks forward to participating. We are also willing to work with any other user group which would like to arrange a vendor day. Let us know if we can help you.

Software AG Symposium

We recently received the agenda for the Software AG Symposium. Our request to speak at the Symposium was rejected, but we believe it promises to be one of the best conferences ever. We hope to see you there!

PAC Education from TSI?

A PAC customer told us that SAG is offering a **"FAST-PAC"** class at $15,000 which lasts for **15 days** and helps the customer implement PAC and get started using it.

Another Software AG customer recently requested PAC education from TSI. We told them **we don't provide PAC education** at this time. TSI would certainly provide PAC education if the demand for it continues to grow, and certainly for less than $15,000. We'll provide N_2O education, too, if users request it. However, N_2O is so easy to use that we know the class won't last for 15 days!

Software AG Makes a Statement

Software AG has been getting a lot of media coverage lately. We've seen announcements for NATURAL OS/2, ENTIRE, and other SAG products. We saw that Software AG has added a relational DBMS to ENTIRE, said to be a former Siemens Nixdorf product called DDB/4.

We also noticed that Software AG now guarantees that sites who use its RAD products will develop higher quality applications in half the time, or they can **return the products to SAG for a full refund**. For more information, contact Software AG.

Partnering On the Rise

According to a *Washington Post* article, customers are beginning to chafe at the idea of being tied to one vendor, prompting vendors to form alliances. **Joe Agro**, executive vice president of Software AG, said in the article, "It means that **technology is going to get better**, and there are going to be more choices, **more options**." We hope that Software AG's new Partners in Solutions program will help eliminate any problems that ADABAS/NATURAL sites might have in choosing complementary products.

Centralized Security

Security continues to be a hot topic. The **consensus** on the issue of centralizing security administration appears to be that **all rules should be kept in one place**, the

(continued on next page)

TREETIPS

Editing, Writing, and Design
Michael Salsbury

Production, Layout, and Distribution
Nick Viola
Terri Hammerschmitt

Back issues available upon request.
Reference manuals for all products also available.

Editor's Sproutings

(continued from previous page)

System Security Facility (i.e., RACF, CA-ACF2, or CA-TOP SECRET).

ISPNews recently stated that "the changing role of the mainframe in many enterprises requires such facilities if the organization is to have a comprehensive, cohesive security strategy." *ISPNews* also stated that **"implementing security in a consistent manner across the organization is likely to be the most productive and cost-effective approach."** They went on to write that organizing an enterprise-wide security system requires software that logically links the organization's various platforms, providing common security administration, system-wide synchronization of User-Ids, automated network-wide synchronization and propagation of security changes, and auditing capabilities. *For ADABAS/NATURAL sites, this means that a product like SECURITRE, which secures ADABAS/NATURAL with the System Security Facility, is needed.*

Password Caveats

Charles Cresson Wood, an independent information security consultant, stated in a letter to the editor of *ISPNews* that "Statistics clearly show that, in spite of management policies, employee training, and widespread appreciation of the problem, most **passwords continue to be easily guessed**." In a reply to that letter, William Murray stated that he preferred the presence of passwords to their absence, but that the growth and diversity of information systems has made it **almost impossible to manage reusable passwords effectively**. *TSI recommends that all sites consider replacing password-based security measures with User-Id based security, i.e., install SECURITRE.*

Passwords are not recommended because "most passwords continue to be easily guessed", and because it is "almost impossible to manage reusable passwords effectively."

Is RACF Gaining Market Share?

An informal *Computerworld* poll appears to suggest that **IBM's RACF is gaining in popularity**. Ken Farber, a research and development consultant at Computer Associates (CA), said in the article about the poll that the suggestion of any shift in the use of security packages is **a "total inaccuracy"**. However, according to the article, Charles Schwab & Co. in San Francisco is writing routines to convert its site's security from CA-TOP SECRET to RACF after four years of running CA's product. They estimate that the process will take **12 to 18 months** to complete. If any of you are going through this sort of conversion, we hope you will share your experiences with us. *SECURITRE customers: Remember that SECURITRE works with all of the major security packages. Only one SECURITRE parameter modification is required to change, for example, from CA-ACF2 to RACF.*

The Rumor Mill

It has been rumored that we are hiring all of the former **Legent** employees. That's not true. We are only hiring a few.

Who Invented 4GLs?

David Downing, the President of Generation Five and Services, Inc., in Hamilton, Virginia, wrote to Computerworld in 1988 about 4GLs. He stated that in 1969 he had originated an interpretive system featuring a high-level procedural language and data dictionary, known as **Dumbol** (Downing Unique Macro Cobol). The system, he stated, was still alive and well, marketed as Generation Five. He said that he didn't have the first 4GL, but that he may have possibly originated **the first interpretive system**, and possibly **the oldest continually marketed 4GL**.

George Szakach, president of Treehouse Software, wrote Mr. Downing a letter. George said that he too had written an interpretive system (not just a language) in 1969. George said people didn't believe he could:

- write a compiler which accepts lines of COBOL source code OUT OF ORDER
- generate and execute something other than machine code
- do the whole thing in COBOL (after adding a few nifty, new statements)
- make it run from multiples of these new inventions called CRTs

"The biggest problems I had involved storing **tons of source and pseudo-object program code**, accessing it in a number of logical sequential (or sorted) ways, and handling **lots of simultaneous users** pounding on terminals," said George.

When he joined Software AG, George found instant solutions to those problems. ADABAS could handle the data and access needs, and COM-PLETE could handle the terminal users. George said, "I approached management, but they wanted me to concentrate on ADAMINT development." SAG Darmstadt later came out with the primitive beginnings of a 4GL called **ADACOM**. Ideas on how to improve it abounded,

(continued on next page)

Editor's Sproutings

(continued from previous page)

until it became a viable programming language, later known as **NATURAL**. Today, NATURAL is one of the premier 4GLs in the world.

Macintosh News: What is MindLink?

In the last issue, we mentioned **Kid Pix**, a Macintosh paint program for children. Since then, we learned about another product called **MindLink**. MindLink is advertised as **"Software for Creative Thinking"** and a **"Brainstorming Tool"**. We never really thought of computers as tools for *generating* creativity, although we have often used them to *express* creativity. The MindLink people sent us a copy to review, and we have to admit that **we're impressed**!

MindLink combines **idea generation** and **problem solving** by merging a library of creative thinking tools and techniques into a simple, flexible, problem solving program. The program guides the user through **a unique solution process**. MindLink helps the user trigger ideas, examine the benefits and concerns about the ideas, imagine ways to get around problems with the ideas, and develop the ideas into solutions. If the user gets stuck at any point in the process, MindLink provides **"idea triggers"**. These take the form of mental exercises which distance the user's mind from the problem or cause the user to look at the problem in an unusual way. Some exercises are word games, others are thought puzzles, etc. **MindLink makes problem solving a (fun) game!** For more information about MindLink, contact MindLink Inc., Box 247, North Pomfret, Vermont 05053.

TREEHOUSE SOFTWARE

Changing of the Guard at UT Austin

Robby Simpson, the Director of Data Processing at the **University of Texas at Austin**, recently announced his retirement. **Randy Ebeling** has been promoted from the position of Assistant to the Director to replace Robby as the new Director. As many of you know, the University of Texas is a long-time ADABAS/NATURAL site, **always on the forefront** of technology. Randy Ebeling is a well-known **Software AG proponent and ADABAS/NATURAL expert.**

Randy spoke with us about their involvement with ADABAS and NATURAL. **"U.T. Austin is a leading edge environment**, especially related to ADABAS and NATURAL. We regularly alpha and beta test new releases of Software AG products. **We have found ADABAS and NATURAL to be solid, reliable performers.** We are very excited about products like ADABAS TRS, ADABAS/HPE, ADABAS/FASTPATH, and ENTIRE. We are gaining excitement about Software AG's UNIX products, too, because **we have to find more cost-effective machines.** We'll be sharing our experiences with all of these products at the Software AG Symposium in May."

Randy Celebrates his Birthday

Of his new role at the University, Randy says, "While my responsibilities have been widened to include much more than ADABAS and NATURAL, **I look forward to continuing my personal involvement with the SAG product line.** I hope to be able to continue speaking at meetings around the country about SAG products, and sharing our experiences with other users."

Randy also mentioned Treehouse Software. "We are very encouraged by the fact that Software AG has taken an interest in partnering. **We have always felt that vendors like Treehouse Software offer value to both SAG and its customers.** My relationship with Treehouse Software has always been a positive one, and I look forward to continuing to work with them in the future."

Congratulations, Randy! We look forward to working with you, also!

DynaDoc's ZPRINT Utility

What is DynaDoc?

DynaDoc, a member of the **DynaMacs** family of productivity tools for NATURAL 2, provides **self-generating documentation for NATURAL applications**. Along with user manuals and technical reference guides, DynaDoc can provide specification tools to aid in the design and development of new applications, as well as a variety of documents which can assist sites in analyzing, maintaining, and re-engineering existing applications. These documents are available through ZPRINT, a component of the DynaDoc package.

What is ZPRINT?

ZPRINT is a generalized print utility. The ZPRINT listings are kept up-to-date by scanning actual source code to determine map, subroutine, and include code usage. DynaDoc also scans source program comment blocks to retrieve the **current overview text** and **technical documentation** which is contained in those comment blocks.

Exploded Program Listings

NATURAL programs often make use of copycode. Unfortunately, NATURAL does not expand copycode in the listings it produces. This is where DynaDoc's ZPRINT Utility can be of benefit. For example, consider the following NATURAL program listing, as printed by NATURAL:

```
0010 ********************************
0020 * PROGRAM NAME: ABCTEST        *
0030 ********************************
0040 DEFINE DATA
.
.
.
0180 INCLUDE SPEC123A
0190 ADD 1 TO #COUNTER
.
.
.
0900 INCLUDE . . .
.
.
.
1130 INCLUDE . . .
.
.
.
1630 ADD 1 TO #COUNTER
1640 DISPLAY #COUNTER
1650 END
```

Note that the INCLUDE code referenced in lines 180, 900, and 1130, does not appear in the listing. The reader might have to look at several printouts to follow program logic. However, with ZPRINT, the listing would be printed as:

```
                0010 ********************************
                0020 * PROGRAM NAME: ABCTEST        *
                0030 ********************************
  +------DEFINE 0040 DEFINE DATA
  !             0050   PARAMETER
  !             .
  !             .
  !             .
  +--END-DEFINE 0170 END-DEFINE
                0180 INCLUDE SPEC123A
                  0010 * TITLE: COPYCODE TO SEND PARMCARD
                  0020 *          TO INTERNAL READER
                  0030 * DESC: USED BY PROGRAMS THAT SEND
                  0040 *       JCL THROUGH INTERNAL READER
                  0050 *       MOVE PARM CARD TO GDA BEFORE
                  0060 *       USING THIS COPYCODE
                  0070 *
                  0080 *NOTE*********************************
                  0090 * COPYCODE NAME: SPEC123A
                  0100 * AUTHOR: JOHN DOE
                  0110 * DATE WRITTEN:  1/10/92
                  0120 ASSIGN #ERROR133A = '133 - JOB FAILED'
        +------IF 0130 IF *INIT-PROGRAM = 'NATT'
        !         0140   WRITE #JCL-REC
        !         0150 ELSE
        !         0160 CALL 'P12345' #CALL-JOB-REC
        +--END-IF 0170 END-IF
                  .
                  .
                  .
                  0490 MOVE 'E' TO #JOB-CARDS-SENT
                0190 ADD 1 TO #COUNTER
                .
                .
                .
+------------DECIDE 0300 DECIDE ON FIRST VALUE OF #PAYCODE
!               0310   VALUE 'A'
!               0320       MOVE 1000 TO #SALARY
!       +-------IF 0330       IF #J = 0 AND #PR-FIELD NE ' '
!       !       0340         ADD 1 TO #K
!       !       0350         MOVE #OLD-PR TO #PR-FIELD
!       +---END-IF 0360      END-IF
!               0370   VALUE 'B'
!               0380       MOVE 1500 TO #SALARY
! <---ESCAPE BOTTOM 0390     ESCAPE BOTTOM
!               .
!               .
!               .
+--------END-DECIDE 0510 END-DECIDE
                .
                .
                .
                0900 INCLUDE . . .
                .   (INCLUDE code
                .   would be exploded here.)
                .
                1130 INCLUDE . . .
                .   (INCLUDE code
                .   would be exploded here.)
                .
                1630 ADD 1 TO #COUNTER
                1640 DISPLAY #COUNTER
                1650 END
```

This listing is easier to follow. **Copycode** (highlighted in the above listing) has been exploded, and (optionally) **processing loops** have been delineated or "blocked". ZPRINT will block DEFINE, LOCAL, PERFORM, SET, IF, USING, FOR, DEFINE SUBROUTINE, ESCAPE BOTTOM, and other statements. This can be very helpful in analysis, maintenance, and debugging of the code.

(continued on next page)

DynaDoc's ZPRINT Utility

(continued from previous page)

Maps in Technical Format

NATURAL programmers make constant use of maps. Often, it is helpful to have a hardcopy version of maps. The following listing shows how a map is normally printed by NATURAL:

```
18:47:05           Map          :  DMM03002 Lib:  ZLIB
0001 * MAP2: PROTOTYPE
0002 * INPUT USING MAP 'XXXXXXXX'
0003 *      #COMPANY-HEADER #DATE-TIME #OPTION-DESC #PROGRAM #PROGRAM-DESC
0004 *      #SALES-ASSOC #SALES-DATA(*) #SALES-GROUP #USER-ID
0005 DEFINE DATA PARAMETER
0006 1 #COMPANY-HEADER (A050)
0007 1 #DATE-TIME (A011)
0008 1 #OPTION-DESC (A020)
0009 1 #PROGRAM (A008)
0010 1 #PROGRAM-DESC (A050)
0011 1 #SALES-ASSOC (A020)
0012 1 #SALES-DATA (A030/00001:00005)
0013 1 #SALES-GROUP (A008)
0014 1 #USER-ID (A008)
0015 END-DEFINE
0016 FORMAT PS=021 LS=080 ZP=OFF SG=OFF KD=ON  IP=OFF
0017 * MAP2: MAP PROFILES ****************************        200***********
0018 * .TTAAAMMOO   D I D I N D I D I        ?_)-:&+(   'ZAS01002',=         *
0019 * 023079        NO NUCN_              X        01 SYSPROP YL          1 *
0020 ************************************************************************
0021 INPUT          (     IP=OFF      HE='ZAS01002',=                       /*
0022                                                                        )
0023  002T #PROGRAM  (AD=OI ) /*.01U008 A008 .
0024  015T #COMPANY-HEADER  (AD=OI ) /*.01U050 A050 .
0025  069T #DATE-TIME  (AD=OI ) /*.01U011 A011 .
0026 /
0027  002T #OPTION-DESC  (AD=OI AL=010  ) /*.01U010 A020
0028  015T #PROGRAM-DESC  (AD=OI ) /*.01U050 A050 .
0029  072T #USER-ID  (AD=OIR ) /*.01U008 A008 .
0030 /
0031 /
0032 001T 'Sales Group.....:'
0033 022T #SALES-GROUP  (AD=OI ) /*.01U008 A008 .
0034 /
0035 /
0036 /
0037 /
0038 /
0039  004T 'Sales Data..:'
0040  018T #SALES-DATA (001)  (AD=MDT'_' ) /*.99U030 A030 .
0041    /* A10000500001000010000100001000011000050010000001H
0042 /
0043  018T #SALES-DATA (002)  (AD=MDT'_' ) /*.99U030 A030 .
0044 /
0045  018T #SALES-DATA (003)  (AD=MDT'_' ) /*.99U030 A030 .
0046 /
0047  018T #SALES-DATA (004)  (AD=MDT'_' ) /*.99U030 A030 .
0048 /
0049  018T #SALES-DATA (005)  (AD=MDT'_' ) /*.99U030 A030 .
0050 /
0051 /
0052 /
0053 /
0054  001T 'Sales Associate...:'
0055  022T #SALES-ASSOC  (AD=MDT'_' ) /*.01U020 A020 .
0056 /
0057 /
0058 /
0059 * MAP2: VALIDATION ***************************************
```

This listing is not easy to understand. For example, what do the strange-looking comments mean? It is also difficult to determine from this printout what the screen will look like when displayed by the application.

In addition to viewing **map screens as displayed by the applications**, it might also be helpful for programmers to view the fields referenced in a map and examine the processing rules. ZPRINT provides this information, as illustrated below:

```
+-----------------------------------------------------------------------------+
!  DMP03001                  BIG SALES COMPANY, INC.                 1/31  12:30 !
!  DMM03001                  SALES DATA ENTRY SCREEN                   BIGUSER !
!                                                                             !
! Sales Group.....: EASTERN                                                   !
!                                                                             !
!                                                                             !
!                                                                             !
!    Sales Data..: CUSTOMER IS EVALUATING ANOTHER                             !
!                  PRODUCT AT THIS TIME AND IS                                !
!                  NOT READY TO PURCHASE ANYTHING                             !
!                  FROM US.  I PLAN TO CONTACT                                !
!                  HER AGAIN IN 6 MONTHS.                                     !
!                                                                             !
!                                                                             !
!                                                                             !
! Sales Associate...:  JOE SMITH                                              !
!                                                                             !
!                                                                             !
!                                                                             !
!  ENTER-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---  .!
+-----------------------------------------------------------------------------+

+-------------------------------------------------------------+
Field                    Format Dimension         From
+-------------------------------------------------------------+

#DATE-TIME               A011
#COMPANY-HEADER          A050
#OPTION-DESC             A020
#PROGRAM                 A008
#PROGRAM-DESC            A050
#SALES-ASSOC             A020
#SALES-DATA              A030   001:005
#SALES-GROUP             A008
#USER-ID                 A008

+-------------------------------------------------------------+
Processing Rules
+-------------------------------------------------------------+

Field : #SALES-GROUP             Rank : 00
-----------------------------------------

Type.........: PREDICT
Ver. Type....: User-defined
Code.........: IF
                 .
                 .
                 .
               END-IF
```

ZPRINT prints the map generated screen as it would be presented by the application. Example values are automatically placed in the fields on the screen to improve clarity. The **example values for the fields** on the map may be defined to DynaDoc, or entered in structured comments in the processing rules for the fields.

The printout of the screen is followed by a listing of the **fields referenced in the map**, and any **processing rules**. This information can be very useful for debugging and documenting applications.

(continued on page 11)

Implementing SECURITRE

SECURITRE is Treehouse Software's interface between the ADABAS/NATURAL environment and a System Security Facility (SSF), e.g., RACF, CA-ACF2, or CA-TOP SECRET. Many sites have purchased SECURITRE, and several are currently evaluating it. One question we are often asked by prospective customers is, **"Just what is involved in implementing SECURITRE?"**

SECURITRE consists of **three basic components**: Security for ADABAS, Utility Security, and Security for NATURAL. Each of these components may be used separately, or any combination of the three components may be used together. In our experience, if a site plans to install all three components, it will usually implement them in the following order:

1. Security for ADABAS
2. Utility Security
3. Security for NATURAL

Each of these components communicates authorization requests to the SSF through use of a **"pseudo-dataset name"**, to which a user may be granted access by the SSF. Note that the pseudo-dataset name does not refer to a physical dataset, but is merely a label used to identify a resource to be secured (e.g., a NATURAL DDM or ADABAS file). SECURITRE determines how to generate the pseudo-dataset name based on the parameter values specified by the site.

"Just what is involved in implementing SECURITRE?"

SECURITRE for ADABAS

To implement SECURITRE for ADABAS, the site begins by creating a set of parameters for the SECURITRE User-Exit-1 to ADABAS. There are several parameters available for tailoring SECURITRE for ADABAS. The most important of these are the MODE, CLASS, PREFIX, QUALIFY, DELIM, USERID, USERID2, NOIDRED, and NOIDUPD parameters.

The value of the **MODE** parameter determines what action SECURITRE will take when it receives an ADABAS command from a user. The default value, **FAIL**, will cause SECURITRE to prevent access and return an ADABAS response code 200 ("SECURITY VIOLATION") if the user is not authorized to access the desired ADABAS data. Therefore, during the initial implementation, it is suggested that the MODE parameter be set to **DORMANT** (security checking will not be done) or **WARN** (security checking is done, unauthorized access attempts are logged by the SSF, but SECURITRE will not prevent unauthorized commands from executing). The use of the DORMANT or WARN mode allows the site to **phase in the implementation of SECURITRE** while avoiding the possibility that access to a Production application would be prevented because of an oversight in defining SSF rules. The MODE parameter **may be overridden at the file level** if some files require more or less security.

The **CLASS** parameter specifies the resource class to be used by SECURITRE when requesting authorization information from the SSF. The default value, **DATASET**, is useful during testing, or at sites where SECURITRE will not secure many different dataset names. When used in a Production environment, CLASS=DATASET has limitations in that the SSFs may take additional memory for each user when generic security rules are used. Therefore, sites may decide to set up a special resource class for ADABAS/NATURAL rules. A drawback to using a class other than DATASET is that the SSF may not allow "masking" for different classes (e.g., a rule may be established for "ADABAS.PRD.F123.*" but not "ADABAS.PRD.*.F123"). Sites will need to give careful consideration to the CLASS setting when implementing SECURITRE in a Production environment.

The PREFIX, QUALIFY, and DELIM parameters specify the initial fixed portion of all dataset names generated for ADABAS file security in SECURITRE User-Exit-1. In the dataset name "ADABAS.PROD.PAYROLL", the **PREFIX** is "ADABAS", the **QUALIFY** is "PROD", and the **DELIM** is ".". The "PAYROLL" portion of the dataset name is a site-specified label for a particular file or group of files.

The USERID and USERID2 parameters specify where SECURITRE can find the SSF User-Id of the user issuing the ADABAS command. **USERID** identifies the first location SECURITRE should look, and **USERID2** specifies the second possible location. SECURITRE can locate the User-Id in the ADABAS 4 Control Block, the ADABAS 5 USERINFO Area, or in another location. It is not necessary to specify a value for USERID2 unless it is likely that the User-Id will not be found in the location specified in USERID.

The **NOIDRED** and **NOIDUPD** parameters specify the action SECURITRE should take when the SSF User-Id cannot be found for the user issuing the ADABAS "read" or "update" command. An example of an instance in which SECURITRE may be unable to locate

(continued on next page)

Implementing SECURITRE

(continued from previous page)

a User-Id for a command is an asynchronous task under CICS, which has no User-Id. Depending upon the value of the NOIDRED and NOIDUPD parameters, SECURITRE may "**ACCEPT**" the command and allow it to process normally, or it may "**REJECT**" the command, treating it as if SECURITRE is in FAIL mode. When this happens, a message is written to the operator's console, reading "NO USERID IN CMD FROM JOB [jobname]".

Utility Security

Utility Security is used to determine if a user has the authority to run an ADABAS utility and utility function against a given file. Utility Security checking takes place **before ADARUN is invoked**. Once access has been approved, control is passed to ADARUN. Utility Security includes parameters to allow the site to specify the dataset names generated by SECURITRE for utility runs. The most important Utility Security parameters are UTMODE, UTPREF, and UTORDER.

The **UTMODE** parameter is comparable to the MODE parameter in SECURITRE for ADABAS. The mode options (DORMANT, WARN, and FAIL) facilitate the phasing in of Utility Security.

The **UTPREF** parameter is used to specify a prefix to be placed in the psuedo dataset name generated for utility runs. The value of UTPREF will become the first characters of all pseudo-dataset names generated by Utility Security. In the example below, UTPREF was specified as "ADABAS.UTL".

The **UTORDER** parameter specifies which potential pseudo-dataset name components (i.e., UTILity, FUNCtion, and FILE number) should be included, and the order in which they should be combined. The flexibility provided by UTORDER allows Security Administrators to determine the best way to organize the pseudo-dataset names to make the best use of "generics". Depending upon how security is implemented, **generics can drastically reduce the number of rules which need to be defined in the SSF**.

For example, execution of the ADASAV utility, RESTONL function, against file 123 could generate a different pseudo-dataset name, depending upon the values specified for the UTORDER parameter, as shown below:

UTORDER	Pseudo-dataset Name Used
UTIL,FUNC,FILE	ADABAS.UTL.SAV.RESTONL.F123
FILE,UTIL,FUNC	ADABAS.UTL.F123.SAV.RESTONL
FUNC,UTIL	ADABAS.UTL.RESTONL.SAV
UTIL	ADABAS.UTL.SAV

SECURITRE for NATURAL

SECURITRE for NATURAL has many parameters which make it **very flexible**. During the initial installation, special attention should be paid to the SERVER, PREFIX, QUALIFY, DELIM, and MODE parameters.

SSF calls generated by SECURITRE for NATURAL are directed first to an APF-authorized database (the **SERVER**) which is running SECURITRE for ADABAS. This is done so that the NATURAL module does not have to be executed from **an APF-authorized library**. SSF calls must originate from an APF-authorized library, so SECURITRE can make its SSF calls from its ADABAS User-Exit-1.

The **PREFIX**, **QUALIFY**, and **DELIM** parameters specify how the beginning of the pseudo-dataset name should be created. These parameters are used in all calls made by SECURITRE for NATURAL.

The different MODE parameters (NSIMODE, LGNMODE, DDMMODE, PGMCHK, and RUNCHK) specify the type of security required for each component of SECURITRE for NATURAL. **NSIMODE** specifies the mode for NATURAL Session Initialization security. **LGNMODE** specifies the mode for logon security. **DDMMODE** specifies the mode for DDM level security. **PGMCHK** determines whether program level access (EDIT, SAVE, CAT, and STOW) controls are to be used. **RUNCHK** specifies the mode for program execution security, which limits whether or not a user can RUN any NATURAL program.

After Implementation

Once the site has had the opportunity to work with SECURITRE, it may decide to "fine tune" some of the parameter settings. The SECURITRE Reference Manual provides information about all of SECURITRE's parameters and the effects of changing the values of the parameters. Further, Treehouse Software's technical support staff is available to discuss SECURITRE parameter settings and their effects.

TREEHOUSE
SOFTWARE

N_2O's Checkout/Checkin Feature

N_2O's Checkout/Checkin feature is a method of controlling and monitoring changes in the development life cycle of NATURAL applications. This feature is designed to protect the integrity of a program throughout its life cycle by controlling who is allowed to make changes to, or migrate, the program. **Checkout/Checkin was first available in N_2O version 2.2, released in April 1991.**

Checkout/Checkin works transparently during a migration. The normal development cycle of existing programs is controlled and protected automatically.

The "BASE" Environment

An administrator implements Checkout/Checkin by designating an existing FUSER to N_2O as the "BASE" environment. **The "BASE" environment** is treated as a source repository. At most sites, the BASE is the Production environment. When migrating from a BASE environment, the programs which are selected for migration are automatically checked out by N_2O. Further migration of these programs by other users is prohibited. When these same programs are migrated back to the BASE environment, N_2O automatically marks them as checked in, and the programs then become accessible to other users.

Checkout/Checkin is designed to protect the integrity of a program throughout its life cycle by controlling who is allowed to make changes to, or migrate, the program.

Checkout/Checkin Modes

N_2O's Checkout/Checkin feature offers three modes: NONE, WARN, and FAIL. When the mode is set to **NONE**, Checkout/Checkin protection is disabled.

In **WARN** mode, Checkout/Checkin protection is provided. Multiple users may check out the same program, although a warning is issued any time this situation occurs. N_2O will track which users have programs checked out. N_2O prevents migrations of programs which are not checked out.

In **FAIL** mode, further Checkout/Checkin protection is provided, in that only one user may have a program checked out at any time. FAIL mode provides the most comprehensive protection. The rest of this discussion assumes FAIL mode is in effect.

Enforcement of Rules

Checkout/Checkin is important because it **prevents migrations that may corrupt programs** during their development cycle. When a migration from the BASE environment is attempted, Checkout/Checkin prevents an already "checked out" program from being migrated by another user. Migrations occurring between two Development environments are prohibited if the programs being migrated are not checked out to the requestor. When a user is prevented from migrating a program, an **on-line report** can be used to view who checked out the program, and where the checked out program currently resides.

N2OEDIT

An optional N_2O feature which further enhances the Checkout/Checkin process is N2OEDIT. N2OEDIT is used in conjunction with Checkout/Checkin to ensure that only the user who has checked out a program can edit that program. Additional uses for N2OEDIT are discussed in the N_2O Reference Manual.

Flexibility

While Checkout/Checkin places controls on an environment, it is flexible and **easily adapts to the site's established policies and procedures**. As previously mentioned, the implementation of Checkout/Checkin starts with the specification of a BASE environment to N_2O. This BASE environment is where the source code is stored and does not necessarily have to be where Production object code executes. Checkout/Checkin supports **any number of environments** in a program's development cycle. Checkout/Checkin tracks the program through each environment, and **there is no limit to the number of times a program can be migrated** before being checked in to the BASE environment.

Utilities

As stated earlier, Checkout/Checkin operates transparently in a normal migration cycle, but situations arise which require special attention. For this reason, N_2O provides a set of Checkout/Checkin utilities.

Security exists to control access to the various utilities. For example, some utilities only allow individual users to make changes to the Checkout/Checkin status of the programs under their control. Other utilities allow project leaders or other managers to make changes to the Checkout/Checkin status of any program.

(continued on next page)

Checkout/Checkin

(continued from previous page)

Checkout Utility

Migrations of existing programs from a BASE environment will cause N_2O to check out the programs for the user requesting the migration. However, this cannot be done with **new programs**, as they do not (yet) exist in the BASE environment. The Checkout Utility is used to mark a new program as checked out. The utility allows the site to specify who has responsibility for the program, the environment where it is being developed, and the BASE environment that represents Production (the final location to which the program will be migrated once development has finished). Since a program cannot be migrated unless it has been checked out, a new program would not normally be able to migrate to Production. In this instance, the Checkout Utility can be **used to indicate to N_2O that the program is checked out**, and may be migrated to Production. Thus, the Checkout Utility enables the Project Leader to retain control over new programs being added to the application.

The Checkout Utility can also be usedwhen changes are made to a program in Development which was not previously checked out. This can occur when Checkout/Checkin is first implemented and programs already reside in the Development environment.

Cancel Utility

The Cancel Utility **cancels the checkout status** of a program. It is used when changes are no longer needed to a program.

Transfer Utility

The Transfer Utility **transfers checkout responsibility** from one user to another. It is often used when one user checks out a program, makes changes to it, but cannot check in the program until another user makes changes to that program. In this case, the utility is used to transfer responsibility to the second user.

Extract Utility

The Extract Utility is used when a copy of a program is needed. For example, the Extract Utility would be **used when a program is being used as skeleton code for a new program**. The Extract Utility essentially allows a migration that does not follow the same Checkout/Checkin rules as normal migrations. The Extract Utility allows the user to migrate a program as long as it does not overwrite a program which is checked out, and as long as it does not migrate into a BASE environment. If it is necessary to migrate an extracted program to a BASE environment, the program may be first checked out by the Checkout Utility.

Reject Utility

The Reject Utility **reverts a program's current "location" back to its previous Development environment**. For instance, assume that a program is migrated from a Development environment to a Quality Assurance environment. Quality Assurance determines that the program does not meet its standards. Instead of migrating the program back to the Development environment, the Reject Utility is used to change the current location of the Checkout program back to the Development environment.

Reporting Subsystem

N_2O's reporting subsystem provides a complete audit trail of ***all*** migration activity. The reporting subsystem includes information on checkout/checkin activities which is useful for **all personnel levels**:

- A **Programmer** may determine what programs they have checked out
- A **Project Leader** may determine who is working on a program
- A **Manager** may determine what areas of an application are under development
- An **Auditor** may determine which programs were checked in since a specific date

Version 3.0

The Checkout/Checkin feature will be enhanced in N_2O Version 3.0. The first enhancement is a **simplified selection process**. After programs are checked out, subsequent migrations will only display a list of programs that are checked out. Also, the utilities will have a specific audit trail which shows who used the Checkout Utilities, and how the utilities were used to change the Checkout/Checkin process.

Summary

Checkout/Checkin is a powerful feature which manages the massive amount of changes that occur during a development project. Checkout/Checkin controls the movement of programs to ensure the integrity of the development process, while retaining the **flexibility to fit every NATURAL site's environment**. Checkout/Checkin also provides utilities to deal with unusual situations. In addition, it provides the information to **show the current state of development** and the **audit trail of all previous development**.

If you have specific questions or comments about Checkout/Checkin or any other N_2O feature, let us know. We will be happy to answer them.

TREEHOUSE
SOFTWARE

Grace Hopper
In Memoriam

We recently learned of the death of **Grace Murray Hopper** on New Year's Day. Dr Hopper was one of the first computer programmers, and the **mother of the COBOL programming language**. According to an article in the January 6 issue of *Computerworld*, she was called "Amazing Grace" by subordinates because of her success in academia, business, and the Navy. She was the **oldest officer on active duty** in the Navy until she retired in 1986 with the rank of Rear Admiral.

George Szakach and Rich Jacobson of Treehouse had the pleasure of working with Dr. Hopper at UNIVAC in the mid-60's. George says, "When I arrived at UNIVAC in 1965, it took a few months before I realized that a 'Dr. Grace Hopper' never showed up for work in her private office. I found out she was an officer in the Navy and did most of her work in Washington. **I knew she must be important** to have a nice big room reserved in her name, just in case she showed up."

George and Rich worked on COBOL compilers. "After the first one, done in assembler (that's all we had)," George continued, "I wanted to do the second one in COBOL (because now we had two programming languages). **The idea of writing a COBOL compiler in COBOL elicited comments like 'wild, weird, impossible'**. I wrote a memo on the idea and Dr. Hopper responded. (I didn't know the memo was routed to her, and I didn't know she knew anything about COBOL.) Her response was quite positive and encouraging. **She said, in effect, 'it's about time'."**

George and Rich only saw Dr. Hopper appear once or twice in 4 years. George finally saw her in Washington in 1970. "I needed a new job, and I felt I was bursting with **'COBOL devilishness'**. I called Dr. Hopper and actually reached her. And, she remembered me from 5 years earlier! She invited me for a visit, and told me that Ocean Data Systems, Inc., was looking for someone like me. It was a perfect match. ODSI wanted to do a Database Management System, written in COBOL! This of course, had to be done in our spare time, while **our main effort was blasting out COBOL applications for ocean studies and other things**. The DBMS never got off the ground, but it was a good career move, and I thank Dr. Hopper."

"**Dr. Hopper spoke at one of the SAG conferences**, Las Vegas in 1982, I think. The speech was one of the best I've ever heard. She was enthusiastic, informative, and humorous, with her handful of **'nanosecond wires'**. And, she still remembered me, 12 years later.", George recalls. **"We won't forget her."**

TREEHOUSE
SOFTWARE

ZPRINT Utility

(continued from page 6)

File Layouts

ZPRINT can also retrieve **file layout information** from PREDICT and display it in this easily readable format:

```
                    File 5 - SALES-MASTER
                    =====================

Fields
-------

------------------------------------------------------------------------------
!    !   !    !                       !        !   !       !                 !
! Ty ! L ! DB ! Name                  ! Format ! S !   D   ! Remarks         !
!    !   !    !                       !        !   !       !                 !
------------------------------------------------------------------------------
!    ! 1 ! AA ! SALES-GROUP           ! A8     ! N ! D     !                 !
------------------------------------------------------------------------------
!    ! 1 ! AB ! SALES-ASSOC           ! A20    ! N !       !                 !
------------------------------------------------------------------------------
!    ! 1 ! AC ! SAMES-AMT             ! N7.0   ! N !       !                 !
------------------------------------------------------------------------------
.
.
.
------------------------------------------------------------------------------
! SP ! 1 ! SA ! SALES-KEY             ! A28    ! N ! D     !                 !
------------------------------------------------------------------------------
!    !   !    ! SALES-GROUP (1,8)     !        !   !       !                 !
------------------------------------------------------------------------------
!    !   !    ! SALES-ASSOC (1,20)    !        !   !       !                 !
------------------------------------------------------------------------------
.
.
.
```

There's More!

ZPRINT is capable of producing many documents which comprise a subset of the documentation available through DynaDoc. DynaDoc provides **complete documentation** for the end user, developer, and analyst.

TREEHOUSE
SOFTWARE

Please Keep Up With Us!

Since each release of a product contains fixes as well as enhancements, it is important for customers to **upgrade to the current version as soon as possible**. This will often eliminate a problem you may be experiencing. Upgrading is also important because we do not support the previous release of a product as easily, once newer releases are available.

The following chart displays the product versions you should be running.

Treehouse Product	Version(s) Supported
AUDITRE	1.2.0
AUTOLOADER	1.2.0
DynaGen	3.30
DynaMacs	3.24
N_2O	2.3.1
SECURITRE	2.1.0
TRIM	4.0.0 / 5.0.3

ADABAS Exploits ESA

About the author: Greg Dzieweczynski is a Database Administrator for the State of Minnesota Department of Human Services in St. Paul, Minnesota. This article is based on Greg's presentation for the upcoming Software AG Symposium in New Orleans.

Introduction

Software AG's introduction of the **ADABAS Dynamic Caching** product exploits the MVS/ESA feature of **dataspaces** and **hiperspaces**. ADABAS Dynamic Caching improves the performance of ADABAS by augmenting the ADABAS buffer manager and **reducing the number of physical read EXCPs** to the database. The advantages of ADABAS Dynamic Caching are: improved performance, reduced number of physical reads to DASD, and extreme flexibility and adaptability because the caching is demand driven to allocate system resources as needed and release resources when not needed.

Identify the Working Set

To obtain effective and efficient usage of these resources some methodology must be in place to identify and control the "working set" that derives the most benefit. The database "working set" is the set of ASSOCIATOR and DATA blocks accessed during the ADABAS session. A key factor in defining the "working set" is the identification of the ASSOCIATOR and DATA blocks that can be categorized as **VERY ACTIVE** and **HEAVY READ** access. Dynamic Caching reduces the read EXCPs to the database. The write EXCPs are always issued to maintain the integrity of the database. To obtain optimum resource utilization and performance benefits, focus on this key premise of frequently read blocks.

"ADABAS Dynamic Caching improves the performance of ADABAS ..."

Speed Disparity

The disparity in speed between the central processor and DASD is well-known. The processor speed is measured in nanoseconds and DASD in milliseconds. The I/O subsystem has traditionally been a major bottleneck for many applications, especially those requiring on-line access to large databases stored on DASD. To compensate for this disparity and attempt to reduce the performance impact of DASD, new technologies such as **expanded storage (ES)** have been developed.

ES Enters the Scene

ES was originally viewed as IBM's answer to solid state devices; a mechanism used for faster paging and swapping. ES is hardware storage that is bus-attached to central storage. The access time to ES is noticeably faster than to conventional DASD. A reasonable expectation for disk I/O for a random read of a 4K page from a 3380/3390 disk drive is approximately 15 to 25 milliseconds. The movement of a page from ES takes .030 to .075 milliseconds. Thus, the use of **ES can reduce the wait time** for a page of data resulting in **shorter response times** or **higher transaction volumes** while maintaining the same response time.

Dataspaces and Hiperspaces

Dataspaces and hiperspaces were introduced with MVS/ESA to provide a performance boost to database and on-line transaction systems. **Dataspaces** are virtual storage or real memory areas that are created as holding areas for whatever data the user decides to store there.

Hiperspaces are similar to dataspaces. A hiperspace is really a different type of dataspace placed in expanded storage (ES) instead of real memory or virtual storage. The primary idea is to get the data off of DASD and into virtual storage utilizing the facilities available in the MVS architecture.

The avoidance of this physical I/O to DASD helps to minimize the adverse impact of the system performance phenomenon known as DASD drag. Records from the database reside in a dataspace or hiperspace eliminating the multiple I/O accesses to the database and the delay or associated wait time for that data from the DASD. The effect is a reduction of the number of I/Os to the database, reduced CPU time to process the remaining I/Os, improved response time and increased throughput by the database.

Again, the primary benefit of dataspaces and hiperspaces is the elimination of the amount of conventional I/O to DASD. Enabling the data to reside in a virtual storage dataspace in real memory or a hiperspace in expanded memory when it is needed, avoids the conventional I/O to DASD and the associated wait time.

(continued on next page)

ADABAS Exploits ESA

(continued from previous page)

Selection

Identification and specification of which Relative Address Block Numbers (RABNs) of the database obtain the greatest benefit from Dynamic Caching is a major issue. The ADABAS nucleus session shutdown statistics provide a basis to determine the activity level to the files in the **"Distribution of commands by file report"**. This report indicates how busy the file is or how frequently the file is being accessed.

The **"Distribution of commands by type report"** reflects all command types including the read type commands to the database files. The critical missing piece is the connection between these two reports enabling analysis of the breakout of command types for each physical file.

Performance Monitor

A good performance monitoring tool can provide this capability. It is difficult to have a high performance ADABAS transaction system without a quality performance monitoring tool(s). The performance monitor can report exactly which pertinent information is necessary to identify what physical files are good candidates for Dynamic Caching. The State of Minnesota uses the **APAS/INSIGHT** product marketed by **Goal Systems** to obtain the required information.

"It is difficult to have a high performance ADABAS transaction system without a quality performance monitoring tool(s)."

The report we use summarizes the number of times each command type was executed by that file, the percentage of total I/Os, the summation of total I/Os, the percentage of ASSOCIATOR I/Os, the summation of ASSOCIATOR I/Os, the percentage of DATA I/Os, the summation of DATA I/Os and the mean duration in seconds for each of the possible command types for every file in the database. Based on the information of this report it is a very easy process to identify the files containing a large number of read commands which generate a high percentage of the I/Os to the database.

Employing this selection technique will optimize the usage of data/hiperspaces and provide the greatest "hit ratio" and benefit to the performance of the database. These data-only address spaces are allocated, populated, and managed by the DBA staff at the State of Minnesota. The DBA staff has responsibility for the review, control, and management of these resources and the optimal utilization by the database.

Fencing

Optimization is ensured through the selection of files meeting the criteria of heavy read activity generating a large percentage of I/Os. This optimization technique is known as **"fencing"**. From a performance and tuning perspective the "fencing of RABNs" provides the optimal environment per allocated data/hiperspace resource.

The DBA staff has also developed the software to automatically regenerate the RABN ranges at initialization time of ADABAS. This capability is extremely useful in handling the changes in RABN ranges due to database reorganizations and file expansions.

Considerations

One needs to be aware of the interesting tuning considerations created by the utilization of Dynamic Caching for the "working set" of RABNs in the database. First, the dynamics of the increased real and expanded memory requirements used to accommodate the "working set" may influence the amount of paging in the system. A level of awareness should exist to avoid the size threshold at which paging could negatively impact the response time.

Second, the downsizing of the LBP (Length of Buffer Pool) may be necessary. The residency of read type RABNs in data/hiperspaces instead of the LBP means the possibility exists for the LBP to contain a larger percentage of write type RABNs in the Buffer pool. When a buffer flush occurs, the required time to write the I/O to disk may exceed the ADABAS buffer flush timer. A buffer flush is the terminology used when the database empties its buffer containing updated records that must be written to the database's direct access devices.

The current production version of ADABAS (ADABAS 5.1.8) halts processing all commands to the database during a buffer flush. At high on-line transaction rates, the queuing caused by a buffer flush can impact performance and creates fluctuations in ADABAS throughput.

ADABAS will self-adjust the interval of time between buffer flushes to compensate and meet the buffer flush timer parameter. This adjustment may cause a decrease in the time interval between buffer flushes and an increase in the number of buffer flushes. The

(continued on next page)

(continued from previous page)

downsizing of the LBP may stabilize the fluctuations in throughput by decreasing the frequency of buffer flushes.

The buffer management concepts in the ADABAS 5.2 version of the database management system have been redesigned. The new version allows larger buffer pools and contains new algorithms for asynchronous buffer flushes while processing database commands. These improvements eliminate the impact attributed to the buffer flush.

Outcome

The State of Minnesota conducted **extensive benchmark activities** at the IBM Benchmark facility in Gaithersburg, Maryland and in Minnesota to assess the ADABAS Dynamic Caching product. The measured advantages of using ADABAS Dynamic Caching were: **reduced number of CPU cycles or service units used, increased database throughput, and improved or decreased response time**. The **financial savings** from a purely charge-back perspective in using this technological advantage are significant. The savings are realized from the reduced number of physical I/Os and service units. An additional intangible benefit is the increased productivity of the workers due to the improved response time of the system.

Conclusion

DASD speed can still be the nemesis in the MVS/ESA world, even with the new structures for managing data in the ESA architecture. DASD drag will always have some negative impact on system performance. The usage of dataspaces and hiperspaces in the MVS/ESA environment has not totally eliminated the need to perform physical I/O to DASD. However, the proper use of dataspaces and hiperspaces can considerably reduce the frequency and amount of physical accesses required to DASD and enhance the performance of the database. ADABAS Dynamic Caching plays an integral role in minimizing the impact of DASD drag and improving the performance of the ADABAS database management system.

ESA Exploitation
Using **DATA** and **HIPERSPACES**
IBM 3090-400J
512M Real 2048M Expanded
1 TOR 3 AORs CICS 1.7
2400 concurrent users
30 CICS transactions/second
Data at 15 minute intervals from RMF/EPILOG/TPNS

ADABAS	BASELINE	DATASPACES	HIPERSPACES	CHANGE
TCB + IOC services	5,054,787	4,801,952	4,715,052	6%
EXCPs/second	197.1	114.1	114.3	40%
Buffer Efficiency	12.1	42.7	42.3	216%
Commands per second	291	334	317	10%
Int. response time (secs)				
mean	1.20	0.52	0.92	
95%	4.20	2.20	3.30	

Treehouse Data Processing Glossary

As part of our on-going effort to promote education, we provide the following glossary of data processing terms:

ANSI: A nervous condition, as in "He's getting really ANSI.", or an insect, as in "He's got ANSI in his pantsy".

CICS: What you get on Route 66.

CSECT: How some babies are born.

DOS: First name of a good Mexican beer.

EXPLAIN: What IBM has to do when it talks about tables when it means files, rows when it means records, and columns when it means fields.

Key Field: Where farmers raise keys.

Megabyte: What you teach your child about sharing with its siblings, as in "Give megabyte, and give milliseconds."

Object: What you do if you don't like source.

QSAM: What you tell Sam to make him wait in line.

RABN: A bird, as in "Rockin' RABN", an action, as in "RABN Peter to pay Paul", or a crime fighter, as in "Batman and RABN".

RACF: The rack located next to RACG.

Relative Address: Where your parents live.

If you have definitions to add to our list, pass them along. If we print your definition in TREETIPS, you will receive a Treehouse coffee mug, or equivalent.

TREEHOUSE SOFTWARE

Treehouse Software's Affiliate List

Area Served	Affiliate	Phone Number
Africa	Syntactica	27-11-880-4976
Australia	Computer Consultants Australia	61-3-416-3377
Brazil	MSS do Brasil Ltda.	55-61-349-2100
France	Fairware	331-39-65-0688
Germany	MaK DATA SYSTEM	49-431-3995-134
United Kingdom & Eire	OCS Software Limited	44-81-207-5434
Venezuela & Caribbean	Mega Soft Computacíon C.A.	58-2-483-53-70

Treehouse Software products and services include:

DynaMacs - a family of productivity tools for NATURAL 2
N_2O - a NATURAL application Change Management System
SECURITRE - a Centralized Security Administration package for ADABAS/NATURAL
TRIM® - an ADABAS/NATURAL performance monitor
AUDITRE - an ADABAS auditing tool
AUTOLOADER - an ADABAS file automatic unload/reload/dump utility

Consulting on ADABAS, NATURAL, and associated topics

Customized on-site classes on a variety of ADABAS/NATURAL related subjects, including:
- ADABAS (Concepts and Facilities through Advanced Topics)
- ADABAS Direct Calls
- NATURAL 2 (Beginning, Intermediate, Advanced, and End User)
- PREDICT
- COM-PLETE
- TSI Products (DynaMacs, N_2O, SECURITRE, TRIM, AUDITRE, AUTOLOADER)
- Custom Classes on ADABAS/NATURAL Designed to Meet Unique Needs

Phone:
(412) 741-1677

Fax:
(412) 741-7245

Treehouse Software, Inc.
400 Broad St., Suite 206
Sewickley, PA 15143

FIRST-CLASS MAIL
U.S. POSTAGE
PAID
Sewickley, PA 15143
Permit No. 89