SOFTWARE AG

CONNECTIONS

VOL. 13 ■ NO. 2 ■ MAY, 1988

SPRING ISSUE

NATURAL CONSTRUCT Available Soon for IBM Environments

Jim O'Leary
Product Manager

The introduction of NATURAL in 1979 and NATURAL 2 in 1987 produced quantum improvements in software development productivity. The release in January of the NATURAL CONSTRUCT application generator for VAX environments, and the upcoming June release for IBM environments, represent important new milestones. With NATURAL CONSTRUCT, NATURAL developers can prototype and develop common business applications in a fraction of the time previously required. This exciting new extension of the NATURAL development environment will change the way NATURAL application systems are developed in the future.

CASE Comes to Software AG

NATURAL CONSTRUCT is the first in Software AG's new line of Computer Aided Software Engineering (CASE) products, the NATURAL Engineering Series. The goal of the NATURAL Engineering Series is to provide automated support for the entire software development life cycle, from application planning through maintenance. NATURAL CONSTRUCT brings automation to the implementation and maintenance phases by providing the ability to generate NATURAL applications.

Programs produced by NATURAL CONSTRUCT can be used in production "as is," or may be modified to accommodate the needs of a particular application. In the latter case, NATURAL CONSTRUCT serves as a potent prototype tool for generating fully functional, first-cut applications, quickly. NATURAL CONSTRUCT is far more than just a prototyping system, however, as evidenced by its past use in the development of several commercial application packages, as well as a large number of customized applications.

Generating NATURAL Applications

NATURAL CONSTRUCT is based on the observation that most application programs are variations of a small number of common program types, represented in NATURAL CONSTRUCT as program "models." Each model consists of a combination of standard NATURAL statements, which specify the basic program structure and control flow of the program, and NATURAL CONSTRUCT generator statements, which specify open characteristics, such as what data base files and maps are used, or whether audit trail logging is desired.

continued on page 10

President's Message

Carlo A. Scagnelli
Central Hudson Gas & Electric Corporation

Having just seen a video of myself in Miami, you are all to be commended for your ability to endure such torture (and I didn't have to watch a 20 foot screen as well). I vow that Nashville will be much shorter both to your benefit (and mine).

There seems to be a misunderstanding as to how to form a SIG (Special Interest Group). A SIG is a group of users with common interests who formalize their relationship as a viable entity within SAGGROUP. To initiate the process, a user communicates with the *SAGGROUP Elected Product Representative* indicating the purpose of the SIG, its aims, goals etc. The SIG is tentatively approved if a group with similar interests does not already exist. It is incumbent on the initiator to prepare a Connections article with the goal of getting other users to join the SIG. At the next International Users' Conference, the 'chairperson' can petition for permanent status again through the related *SAGGROUP Product Representative*. The petition will be approved IF the SIG proves that it is active and viable. Note that the SIG is established through SAGGROUP, not Software AG and all contact must be made through your elected representatives, not Software AG personnel.

I've received several newsletters from SIGs and regions over the past few months with an outstanding effort from the Eastern Regional VAX

continued on page 2

President's Message

continued from page 1

VMS SIG. It contained hints, suggestions and a program written in DCL called READ QUE and reflects a level of activity that should be more evident among other SIGs; a way to keep people informed, interested and motivated between Connections issues. Copies should also be forwarded to your SAGGROUP product reps for inclusion in Connections. We thought you were out there; what better way to let us all know.

The College/University BIG conference was recently held in Wilmington Delaware. All you BIG chairpersons take note; this meeting had over 200 attendees discussing *industry* issues. There's no reason that other BIG's can't get it together in the same way. It takes time and effort but if spread out among several individuals the results can be more than substantial. Get in touch with Randy Ebeling, College/University BIG Chairperson and see how it's done. I've always found that people want to get involved but need the leadership and direction to get things started. That's where the chairpersons come in. The work comes with the territory. You'll be happily surprised at the results of your efforts.

1988 Connections

Editorial deadlines for this year's Connections are as follows:

Issue	Editorial Deadline
August 1988	June 15th
December 1988 (post-conference issue)	October 21st

Please submit materials to:

Pam Ellis
Software AG
11800 Sunrise Valley Drive
Reston, VA 22091

If you have any questions regarding editorial please feel free to contact me at: 703-648-2570.

Table of Contents

Software AG News PAGE

NATURAL CONSTRUCT Available Soon for IBM Environments 1

NATURAL Elite Experiences Growth 3

Where Can You Find a Consultant When You Need One? 3

Conversion Guidelines: NATURAL V1.2 to NATURAL 2 4

New Options for Preventing Duplicate User ID with NATURAL 2 Security 5

Programmed Programming Success 5

Human Services Solutions: When the Best Meets the Best 13

Software AG Product Status Report 24

User Group News PAGE

President's Message 1

NATURAL 2 Conversion Experiences 14

ADABAS 5 Performance Update 14

ADABAS Product Representative Report 15

The 8th Annual Israel Users' Group Conference 16

NATURAL Elite Special Interest Group Started 16

How One User Increased NATURAL 1.2.7 Buffer Pools Under CICS 23

Regional Reports PAGE

New England 17

Eastern 17

Sierra Pacific 18

Southwest 19

NATURAL Elite Experiences Growth

Susan Lowy
Instructional Technologist
Customer Service Division

If the first part of 1988 is any indication, NATURAL Elite is definitely making a big impact in the Software AG product line. Not only does Elite continue to grow as a product, but it also continues to grow in user recognition. Currently, we have approximately 130 customers!

NATURAL Elite for VMS is currently in beta test and will soon be available. The following courses will be available with the initial release of this product:

Introduction to NATURAL (V1.3)
NATURAL for the Business Professional (V1.3)
SUPER NATURAL Fundamentals
Advanced SUPER NATURAL
SUPER NATURAL Administration

When VMS Elite is compatible with NATURAL 2, the corresponding courses will also be made available.

Another exciting occurrence is the release of Elite 2.0 for both the Presentation and Administration subsystems (we anticipate the release of the Authoring subsystem later this year). Elite 2.0 will run under both NATURAL 1.2 and NATURAL 2. During the install procedures, the DBA indicates which version of NATURAL will be used to run Elite. You will be able to define PF-keys to match those established by your organization and you will be able to access your own applications, as desired. Also, you'll be able to toggle between Elite and NATURAL 2 using a PF-key.

With the release of the Authoring subsystem, this version of Elite will take advantage of some of the enhanced capabilities of NATURAL 2, including the use of the NATURAL 2 Map Editor.

Amid all these software releases, we still continue to develop courses. We have concentrated our recent efforts on the development of courses geared to the NATURAL 2 user. These courses include:

NATURAL 2 Enhancements (July 1987)
NATURAL 2 Introduction (November 1987)
NATURAL 2 for the Business Professional (December 1987)
NATURAL 2 Intermediate Programming (April 1988)
NATURAL 2 Advanced Programming (Summer 1988)
NATURAL 2 Structured Design (Summer 1988)

The last three courses listed above are part of our structured mode series (i.e., these courses concentrate on NATURAL's structured mode format versus the report mode format).

In addition, we continue to work on ADABAS courses (to be released concurrently with the release of ADABAS 5). In April, we sent the NATURAL CONSTRUCT Applications course to beta testing, and we expect to release it in May.

And, finally, it's official: an Elite Special Interest Group (SIG) has been organized and currently has over 60 members! It was organized by Tony Cortese and Michael Kochcroft of EG&G Florida, Inc. (see the related article in this issue of Connections). For more information, please call Tony Cortese at (305) 867-4042, or write:

EG&G Florida, Inc.
P.O. Box 21267
Kennedy Space Center, FL 32815

Attn: Tony Cortese
BOC-29

For information regarding the purchase of Elite, please contact your Software AG Account Representative or call Software AG's Telemarketing at 1-800-843-9534.

Where Can You Find a Consultant When You Need One?

Larry Jayne
Manager, Business Relations

If you are looking for a consultant to help you develop a business solution for your organization, where can you find one?

Within the Software AG community there are several sources of consultant skills available. Two of them are located inside Software AG. The first being the Professional Services Group who performs systems analysis, data and systems conversions and tuning. The second one is the Custom Solutions Group who develops and implements customized NATURAL based applications and systems design.

One additional source of consultants is also available to all licensed Software AG customers. It is the list of members of the Consultant Cooperative Support Program. Each of these member organizations has agreed with Software AG to work cooperatively with us in assisting our customers in developing new systems and applications to meet the customers needs.

To contact either the Software AG Professional Services Group or Custom Solutions Group call Software AG in Reston at (703) 860-5050 and we will put you in direct contact with them. You can then determine which of these groups can provide the services you need.

To obtain a list of the members of the Consultants Cooperative Support Program call Software AG in Reston at (703) 648-2565 and ask for the Consultants List. We will gladly send you a copy. You can then contact any of the consultants on the list to determine for yourself if they have the skills your organization needs.

Conversion Guidelines: NATURAL V1.2 To NATURAL 2

Mark Cade
Customer Service Division

Section 1: Basic Information

There has been a great deal of confusion concerning the implementation of existing NATURAL V1.2 applications in a NATURAL 2 environment. One of the major points of confusion concerns the definition of two terms used to describe the implementation of a NATURAL V1.2 application.

The following definitions are supplied to make the implementation process easier for you.

Term	*Definition*
Migration	Movement of NATURAL V1.2 object code to a NATURAL 2 environment or execution of NATURAL V1.2 object code in a NATURAL 2 environment by use of the fuser parm.
Conversion	Creation of NATURAL 2 object code accomplished by either recataloging a NATURAL V1.2 source program, or by the "run" of a NATURAL V1.2 source program in a NATURAL 2 environment.

Section 2: How to Determine Whether to Migrate or Convert?

The key to the implementation process is to figure out if you need to convert or if you can just migrate. Most people will be able to migrate, but there are those who will have to convert.

Answer the following questions to determine your needs:

1. Are source changes to the application necessary?
2. Does the application contain programs that are invoked using the "run" command? This is generally utilized only for a program using the dynamic global variables available in NATURAL V1.2

If you answered "yes" to either one or both of the questions, then read Section 4, "The Conversion Process"; otherwise, read Section 3, "The Migration Process."

Section 3: The Migration Process

Migration of a NATURAL V1.2 application to a NATURAL 2 environment is simply a matter of making the NATURAL V1.2 object modules accessible to a NATURAL 2 nucleus. To make the object modules accessible, use one of the 3 methods listed below:

Method 1: Use SYSMAIN to move/copy the object modules to the NATURAL 2 system file.

Method 2: Use ULDOBJN to unload the object modules and then use the NATURAL 2 INPL to load them to the NATURAL 2 system file.

Method 3: Use the NATURAL parameter fuser under NATURAL 2 and point it to the NATURAL V1.2 system file.

This is the preferred approach to the implementation of NATURAL V1.2 applications to NATURAL 2, but it can't always be done this simply.

Section 4: The Conversion Process

The implementation of a NATURAL V1.2 application that meets either one or both of the criteria presented in Section 2 should proceed with the following conversion steps:

Step 1: Move/copy both source and object code of the entire application to the NATURAL 2 environment. The easiest way to do this is to use SYSMAIN.

Step 2: Answer the following questions:

a. Does the source program being modified/"run" reference external maps? If so, perform Step 3.

b. Does the source program being modified/"run" reference global variables? If so, perform Steps 4 and 5.

If you answered "no" to both of these questions, then just stow the program under NAT2.

Step 3: Converting Maps

a. Identify any other programs that use the external map(s).

b. Edit the map(s) using the NATURAL 2 map editor. Make any changes necessary as a result of the modification to the program.

c. Stow each map in the NATURAL 2 map editor.

d. Stow each program that references the map(s).

NOTE: When stowing the V1.2 programs in the NATURAL 2 environment, you may get errors. There has been some code tightening done—meaning that code which was allowed in V1.2 may not be allowed in NATURAL 2.

Step 4: Converting Programs

a. Identify the program(s) which define the global variables.

NOTE: Only those programs which define the globals need to be converted.

b. Enter the global data area editor and enter the command "CREATE GLOBALS PROGRAM", where program is the name of the program that defines the globals. The wild card notation is supported to allow a specified set of programs to be converted or to select the entire library. This function requires the program(s) in object form.

NOTE: When you perform this step of the conversion, NATURAL will automatically name the GDA COMMON.

c. Make any required changes to the GDA COMMON created in the previous step. If any of the globals are redefined in the source program(s) being converted, then the redefinition must be done in the GDA. Stow the resulting GDA.

NOTE: NATURAL has already named the GDA COMMON for you.

d. Create the following program and stow it as ACOMMON. Failure to do so will result in the GDA not getting initialized. This program will be invoked whenever you log onto the library, and it will initialize the GDA COMMON.

```
DEFINE DATA GLOBAL USING
  COMMON
END-DEFINE
END
```

NOTE: ACOMMON does not have to be included in each program.

e. If the program being converted references an external map(s), then you must also perform Step 3. After completing Step 3, stow the program(s) or save the program(s) that are invoked with the "run" command.

continued on page 8

New Options for Preventing Duplicate User ID with NATURAL 2 Security

Lisa Crispin
Customer Service Division

Users of NATURAL SECURITY V1.2 may recognize one or more of these error messages: NAT8048, NAT3048, or "NAT4000 logon open ADABAS response: 48." These messages all signify that a duplicate User ID is active in ADABAS, and they're essential for users whose applications need access to restart data in ADABAS.

For applications which do not need restart data, however, these messages are meaningless to users. The error can be trapped in batch with an error routine if restart data is not used. Many users also want a way to prevent the error message from displaying online, when restart data is not used.

NATURAL 2 SECURITY comes to the rescue with some new options which administrators can use to prevent a duplicate User ID from occurring. User confusion is minimized and restart data, if used, is protected.

Duplicate User ID in ADABAS and Restart Data

Restart or ET data is data written with the END TRANSACTION statement to the ADABAS checkpoint file. This data can be read in NATURAL with a GET TRANSACTION DATA statement, and can be used by applications to keep track of what processing has been done—especially in the case of an abend and subsequent restart of an application.

If a user wishes to write restart (ET) data to the ADABAS checkpoint file, NATURAL must do an OP (open) to ADABAS with a restart (external) User ID in the Additions 1 field of the control block (see page 227 of the ADABAS Command Reference Manual).

When an ADABAS OP command is issued with a restart User ID, ADABAS checks to see if a User Queue Element (UQE) is already present with the same restart User ID. If one is present, a new UQE is created without a restart User ID. Without a restart User ID, no restart data can be read from or written to the checkpoint file. When a duplicate User ID is detected by ADABAS, it checks an activity timer. Depending on the value of the timer, either an RSP048 or RSP009 is issued to the new session.

If an RSP048 is issued, the new session cannot access or write restart data. If an RSP009 is received, the existing user is backed out, so neither UQE has a restart User ID. If the new session reissues the OP following the RSP009, a new UQE is created with the restart User ID so the new session, but not the old session, can access/write restart data.

continued on page 6

Programmed Programming Success

The ability to effectively manage information within a company is important to any business. But it is critical to a company that continually operates on deadlines. That is why "Programming Business Success" has a different meaning for National Public Radio (NPR), a non-profit membership organization and program producer located in Washington, D.C. Its role is to provide quality news, cultural and entertainment programming and support services to a rapidly growing over 300 member association of non-commercial radio stations across the United States.

Most local radio broadcasting in the competitive American market is provided by stations affiliated with the three major commercial communications networks, ABC, CBS and NBC, or by small independent stations.

The NPR network, on the other hand, is non-commercial. Unlike commercial radio stations supported by selling advertising, local public radio stations are funded primarily by their audiences, with additional support from government, private industry, educational institutions, and businesses.

To qualify for NPR membership, non-commercial stations must meet technical, financial and staffing criteria. Member stations pay dues which entitle them to use all NPR programming and related services.

In addition, NPR, the first full-time multi-channel radio network to transmit programming via satellite, distributes thousands of hours of programming from a variety of sources, including international broadcasting organizations such as the British Broadcasting Corporation.

While National Public Radio had no trouble managing its information, a key ingredient—effectiveness—was missing. NPR's Computer Information Services (CIS) department recognized this fact and, two years ago, began a hardware and software overhaul that is now paying off.

This overhaul meant implementing new items such as Digital's VAX computer series and, on the software side, Software AG's ADABAS and NATURAL.

John Frederickson, Data Base Administrator with NPR's CIS department, described the system's organization prior to its renovation: "Information management within the company used to be a laissez-faire situation. Each department was responsible for its own equipment needs. Over the years a hodge podge of equipment and software applications evolved." As a result, the equipment used in NPR's five divisions ranged from DEC PDP-11's to Apple II's.

This computer configuration was not the ideal starting point for the

continued on page 12

User ID

continued from page 6

The following chart illustrates the possible scenarios when an ADABAS OP with restart User ID is issued by NATURAL, and a UQE with the same restart User ID already exists:

Time Since Last Activity (existing UQE)	Response Code to New Session	Existing UQE	New UQE	Read/Write Restart Data?
< 1 minute	RSP048	Still Active Has Restart User ID	No Restart User ID	Existing Session
> 1 minute	RSP009	Backed Out: No Restart User ID	No Restart User ID	Neither Session

NATURAL's Action based on the response code received is as follows:

ADABAS RSP Code	NATURAL Error Message to NAT user	Action by NATURAL	Access/Write Restart Data?
RSP048	NAT3048/NAT8048/ duplicate user msg	Continue if User Presses ENTER	No
RSP009	None—NATURAL traps the RSP009	Reissue the OP w/restart User ID	Yes

For details on timeout and duplicate User ID processing, please refer to:

- Appendix B of ADABAS Messages and Codes manual
- ADABAS Internals manual, pg. 41
- ADABAS Commands Reference manual, pg. 221
- ADABAS TP Programmer's Guide, pgs. 14.1-19
- Timeout Frequently Asked Questions
- NATURAL SM06 release documentation

NATURAL SECURITY V1.2 and Duplicate User ID

In V1.2, NATURAL SECURITY always uses a restart User ID on the OP. The restart User ID is the same as the security logon User ID (refer to *"The ADABAS User ID and NATURAL SECURITY"* by Ed Barrow in the Spring 1987 issue of Connections for further details).

If a second session is started with the same security User ID, an RSP048 or RSP009 (see chart) results. An RSP048 may be trapped with an error routine in batch. Online, it will cause a warning message which can be ignored if the user is not using restart data. An RSP009 is intercepted by NATURAL, which will reissue the OP command.

NATURAL SECURITY V2.1 and Duplicate User ID

Version 2 of NATURAL SECURITY provides some new ways to prevent a duplicate User ID condition from occurring. The administrator can now specify whether or not restart data will be used in a particular application. The application profiles determine whether or not NATURAL issues an OP with a restart User ID. If restart data is not used, a duplicate User ID condition will not occur, so an RSP048 will not be returned. If restart data is used, the administrator can provide for unique restart User IDs. An RSP048 will then occur only if a user is logged on to the same application and the same device type (video, batch, PC, color, TTY) in more than one concurrent session.

In V2.1, the NATURAL system variable *ETID is used for the ADABAS restart User ID. The value of *ETID is determined by the security definitions for the user and the application. Also, there is a RESTART parameter on the application profile (or user profile for private libraries) that determines whether or not restart data can be written in the application.

To Allow Writing Restart Data in a V2 Application

In order to issue an OP with a restart User ID, the security profile for the application and user must be set up as follows:

1. The user definition must contain a non-blank ETID parameter. The value of ETID can be specified in several ways:
   a. Default (logon User ID)
   b. Hard coded value other than default
   c. Generated (see next section for how to generate ETID)
   d. NATPARM ETID passed dynamically at logon time; this will override ETID in the security profile.
2. The RESTART parameter on the application profile (or user profile for private libraries) must be "Y."

The value of *ETID comes from the ETID parameter in the security profile for the User ID, NATPARM, or dynamic parm override. If the RESTART parameter in the application profile (or user profile for private libraries) is "N," *ETID is set to blank during logon, and NATURAL will not issue an OP with a restart User ID. If RESTART is "Y" and *ETID is non-blank, NATURAL will issue an OP with a restart User ID equal to the value in *ETID. If the session has a restart User ID, the user can write and read restart data to and from the checkpoint file. *ETID can be displayed in a NATURAL program to check what value is being used for the application.

NOTE: SM02 of NSC V2.1 does not allow a blank ETID on a user record; however, SM03 allows blanks. Make sure ETID is provided either from the security profile or NATPARM if restart data will be used.

Preventing Duplicate User ID if Restart Data is Used

The new capability to generate unique ETIDs for each application/user/device type combination in

NATURAL 2 SECURITY can be used to prevent duplicate User ID problems.

To make NATURAL SECURITY generate the ETID, enter "?" in the ETID field on the user profile, or put a "Y" for "Generate ETID" on the Modify General Options screen of Security Administrator Services. When the user logs on to an application, NATURAL SECURITY will use the following algorithm to determine ETID:

> ETID is DUUUUAAA where: D is device type (i.e., B for Batch), UUUU is a unique identifier for the User ID, and AAA is a unique identifier for the application. For more details, see the NATURAL 2 SECURITY Manual, pgs. 6-18.

A generated ETID is unique to the application, User ID, and device. If the same User ID logs on to two separate applications or two different device types concurrently, each session will have a unique restart User ID allowing each session to read and to write restart data. Logging on to the same application on the same device type concurrently will result in a duplicate User ID condition (RSP048 or RSP009).

NOTE: For private libraries, in SM02 of NSC 2.1, ETID is the same as the User ID; in SM03, this is changed so that ETID is DUUUU (the last three characters are blank).

The security administrator has other options to prevent potential loss of restart data due to a duplicate User ID condition. For example, batch jobs can be set up to use auto logon, with a different job name for each job. All batch jobs can be run in the same initiator so that two jobs cannot run concurrently.

To make sure that restart data is always available, define the application with RESTART = Y and a unique ETID, and make sure online users do not ignore duplicate User ID warnings.

Preventing Duplicate User ID if No Restart Data Is Used

If restart data is not being used, an RSP048 can be avoided by setting the RESTART parm to "N" in the application profile (or user profile, for private libraries). ETID will be set to blanks at logon time and NATURAL will not issue an OP with a restart User ID. Setting ETID to blanks will accomplish the same result.

Duplicate User ID Scenarios

When a NATURAL session is in progress, there is no way to tell whether ET (restart) data is being written to the checkpoint file. A NATURAL program may appear to write restart data and read it back in immediately even if ETID is blank and RESTART = N. However, the program is actually writing restart data to and reading it from the ADABAS buffer, and the data will not be stored on the ADABAS checkpoint file. Administrators and application developers have to set up the application and the production environment to ensure that restart data, when used, is always available.

When planning to implement a new application, consider all the possible scenarios where a duplicate User ID situation could occur. Define the application profile with appropriate values for the RESTART and ETID parms. Evaluate the TP and batch environment to identify and prevent possible problems. Consider the following scenarios where a duplicate restart ID in ADABAS can occur (assume all applications are defined with RESTART = Y and that ETID is non-blank):

1. A batch application abends, leaving a UQE with a restart User ID; the application is resubmitted after more than one minute.
2. The same NATURAL SECURITY User ID logs onto the same application from two separate video terminals concurrently.
3. A NATURAL SECURITY user abandons an online session without ending it normally with a FIN or TERMINATE, and logs onto the same application in a new session after more than one minute.
4. An application runs in batch and online concurrently.
5. Two applications using the same NATURAL SECURITY User ID run in batch concurrently.

Using the charts and information on NATURAL 2 SECURITY's RESTART and ETID parms above, evaluate each scenario. Different results can occur, depending on the value of ETID.

In *Scenario 4*, for example, if NATURAL 2 SECURITY generates the ETID, the ETID will be unique for online and batch, no duplicate User ID will be found, and restart data will be accessible to both applications. In *Scenario 2*, however, even if the ETID is generated, it will be the same for both sessions; the second User ID to logon will receive an RSP048 and will not have access to restart data.

In *Scenario 3*, the original session will no longer have access to restart data even if a generated ETID is used, since the same User ID is logged on to the same device type in the same application.

In *Scenario 1*, where the first batch job abended, the second job will have access to the restart data. *Scenario 5* is not as clear-cut. If the jobs use unique ETIDs, each will have access to its own restart data.

There are too many possibilities to cover here; however, the moral of these scenarios is that administrators must carefully plan applications that use restart data. For example, each user of an application that uses restart data must have a unique NATURAL SECURITY User ID, or the ETID (thus the restart User ID) cannot be unique for each user.

NATURAL 2 SECURITY provides new tools to manage the use of restart User ID in applications. Use these tools to minimize user confusion by eliminating inappropriate duplicate User ID error messages, and to maximize the benefits of restart data by ensuring access to it when needed.

Conversion

continued from page 4

NOTE: When stowing the program(s), you may get errors. There has been code tightening—meaning that certain code which was allowed in NATURAL V1.2 may not be allowed in the NATURAL 2 environment.

Step 5: Completion of the Conversion Process

a. Logon to the application in NATURAL 2. This will cause the GDA COMMON to be initialized. Any NATURAL V1.2 program that is executed will access the global variables in the same method as used in NATURAL V1.2, and any program that is "run" or has been restowed in NATURAL 2 will access the global variables using the GDA COMMON. NATURAL 2 will automatically synchronize the V1.2 global variables and the GDA COMMON, so that no data is lost when switching between a NATURAL V1.2 program that hasn't been converted and a NATURAL V1.2 program that has been converted to NATURAL 2.

b. Begin the execution of the application.

The following is a NATURAL V1.2 application which contains five programs and two maps. This application will be used as an example to walk you through the conversion process.

MAINMENU PROGRAM:
This program defines all the global variables and fetches and runs programs.

ADDPGM PROGRAM:
This program is invoked by a FETCH command from the MAINMENU. It does not use global variables.

DELPGM PROGRAM:
Same as the ADDPGM program.

UPDPGM PROGRAM:
Same as the ADDPGM program.

REPORT PROGRAM:
This program uses dynamic global variables and is invoked by a "run" command from the MAINMENU.

MAINMAP MAP:
This map is used by the MAINMENU program.

THEMAP MAP:
This map is used by the ADDPGM, DELPGM, and UPDPGM programs.

```
************************************  MAINMENU  ************************************

0010 RESET #SELECTION (A1) +NAME (A20) #NUMBER (N8)
0020       +VAR (A22)
0030 INPUT USING MAP 'MAINMAP' #SELECTION +NAME #NUMBER
0040 IF #SELECTION = '.'
0050    STOP
0060 FIND NUMBER CONVERT WITH PERSONNEL-NUMBER = #NUMBER
0070 IF *NUMBER = 0
0080 DO
0090 IF #SELECTION = 'A'
0100  FETCH 'ADDPGM' #NUMBER
0110 ELSE
0120 IF #SELECTION = 'D'
0130  REINPUT 'THAT EMPLOYEE DOES NOT EXIST' MARK 1
0140 ELSE
0150 IF #SELECTION = 'U'
0160  REINPUT 'THAT EMPLOYEE DOES NOT EXIST' MARK 1
0170 ELSE
0180 IF #SELECTION = 'R'
0190  REINPUT 'THAT EMPLOYEE DOES NOT EXIST' MARK 1
0200 DOEND
0210 ELSE
0220 DO
0230 IF #SELECTION = 'A'
0240    REINPUT 'THAT EMPLOYEE ALREADY EXISTS' MARK 1
0250 ELSE
0260 IF #SELECTION = 'D'
0270    FETCH 'DELPGM' #NUMBER
0280 ELSE
0290 IF #SELECTION = 'U'
0300    FETCH 'UPDPGM' #NUMBER
0310 ELSE
0320 IF #SELECTION = 'R'
0330  DO COMPRESS '''' +NAME '''' INTO +VAR LEAVING NO SPACE
0340   RUN 'REPORT'
0350  DOEND
0360 DOEND
0370 END

************************************  ADDPGM  ************************************

0010 INPUT #NUMBER (N8)
0020 MOVE #NUMBER TO #PERSON (N8)
0030 INPUT USING MAP 'THEMAP' #PERSON #FIRST-NAME (A15) #NAME (A20)
0040    #CITY (A15) #ZIP (N5) #JOB (A20)
0050 STORE CONVERT PERSONNEL-NUMBER = #PERSON NAME = #NAME
0060  FIRST-NAME = #FIRST-NAME CITY = #CITY ZIP = #ZIP JOB = #JOB
0070 END TRANSACTION
0080 FETCH 'MAINMENU'
0090 END

************************************  DELPGM  ************************************

0010 INPUT #NUMBER (N8)
0020 FIND CONVERT WITH PERSONNEL-NUMBER = #NUMBER
0030 INPUT USING MAP 'THEMAP' PERSONNEL-NUMBER FIRST-NAME NAME
0040    CITY ZIP JOB
0050 DELETE
0060 END TRANSACTION
0070 FETCH 'MAINMENU'
0080 END

************************************  UPDPGM  ************************************

0010 INPUT #NUMBER (N8)
0020 FIND CONVERT WITH PERSONNEL-NUMBER = #NUMBER
0030 INPUT USING MAP 'THEMAP' PERSONNEL-NUMBER FIRST-NAME NAME CITY
0040    ZIP JOB
0050  UPDATE SAME RECORD
0060  END TRANSACTION
0070 FETCH 'MAINMENU'
0080 END

************************************  REPORT  ************************************

0010 FIND CONVERT WITH NAME = &VAR
0020  DISPLAY PERSONNEL-NUMBER NAME JOB SALARY
0030 FETCH 'MAINMENU'
0040 END
```

This figure shows the source code for the programs.

SOFTWARE AG

INFORMATION REQUEST CARD

CO-0588

YES, Please send me additional information on:

- ☐ NATURAL CONSTRUCT
- ☐ VAX Products
- ☐ NET-PASS Version 2
- ☐ NATURAL ELITE Computer Based Training
- ☐ Please Include me on the Customer Training Mailing List
- ☐ Other _____

NAME _____

TITLE _____

COMPANY NAME _____ CUSTOMER # _____

ADDRESS _____

CITY _____ STATE _____ ZIP _____

TELEPHONE _____

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 3115 RESTON, VA

Postage Will Be Paid by Addressee

Software AG
11800 Sunrise Valley Drive
Reston, VA 22091
Attn: Tim Fields

We will present 2 scenarios as examples of the conversion process.

Scenario 1

If there are no source changes necessary to any of the programs or maps, then use the following steps:

1. Copy/move the application to the NATURAL 2 ENVIRONMENT.
2. Using Step 2, we answer "no" to question 2a since the program REPORT, which is invoked with the RUN command, does not reference an external map. We answer "yes" to question 2b since the program REPORT does reference global variables. We now proceed to Step 4.
3. Using Step 4a, we identify that the MAINMENU defines the global variables.
4. Using Step 4b, we go into the global data area editor and enter "CREATE GLOBALS MAINMENU". Refer to Figures 1A and 1B.
5. Using Step 4c, we see that there are no changes necessary to the GDA COMMON, so we stow the GDA COMMON.
6. Using Step 4d, we create and stow the program ACOMMON.
7. Using Step 4e, we see that MAINMENU uses an external map MAINMAP, so we back up to Step 3 to stow the map and check for other programs that may reference the map MAINMAP. Now we may stow the program MAINMENU.
8. At this point, we log off and then log back on so that the GDA COMMON can be initialized by the program ACOMMON.
9. We can now proceed to execute the application.

Scenario 2

We have decided that we want to make a source change to the program ADDPGM. The following steps are for the conversion of the program ADDPGM.

1. Using Step 2, we answer "yes" to question 2a since the program ADDPGM does reference the external map THEMAP. We answer "no" to question 2b since the program ADDPGM does not reference global variables. We now proceed to Step 3.
2. Using Step 3a, we identify that programs DELPGM and UPDPGM also use the map THEMAP.
3. Using Step 3b, we edit the map THEMAP and make any necessary changes.
4. Using Step 3c, we stow the map in the NATURAL 2 map editor.
5. Using Step 3d, we stow the programs ADDPGM, DELPGM, and UPDPGM.
6. We can now proceed with the rest of the conversion as laid out in Scenario 1.

```
************************************  THEMAP  ************************************

                                    ?THEMAP

                        PERSONNEL ID    :99999999
                          FIRST NAME    >XXXXXXXXXXXXXXX
                           LAST NAME    >XXXXXXXXXXXXXXXXXXXX
                                CITY    >XXXXXXXXXXXXXXX
                                 ZIP    >99999
                           JOB TITLE    >XXXXXXXXXXXXXXXXXXXX

************************************  MAINMAP  ************************************

                                  ?MAIN?MENU

                                                    OPTIONAL = ?O
                                                    REQUIRED = ?R

SELECT THE FUNCTION THAT YOU WISH TO PERFORM.

    CODE        FUNCTION                                NAME          NUMBER
-----------  ----------------------------------  ----------------  ----------
      ?A     ADD EMPLOYEES                                                R
      ?D     DELETE EMPLOYEES                                             R
      ?U     UPDATE EMPLOYEE INFO                                         R
      ?R     GENERATE AN EMPLOYEE REPORT                                  R
      ?.     TERMINATE
-----------  ----------------------------------  ----------------  ----------

          SELECTION>X
              NAME>XXXXXXXXXXXXXXXXXXXX
            NUMBER>99999999
```

This figure shows the layouts of the maps.

Figure 1A:

```
LIBRARY: CONVERT       NAME: GLOBAL                              DBID: 200 FNR:  1
COMMAND: CREATE GLOBALS MAINMENU                                          > +
I  T  L  NAME                          F     LENG       INDEX/INIT/EM/NAME/COMMENT
-  -  -  ---------- ALL ----------     -     ----    ------------------------------
```

Figure 1B:

```
SYSGDA 0055: GLOBAL DATA AREA CREATED.
17:08:37                        ***** E D I T DATA *****                  88-04-07
LIBRARY: CONVERT       NAME: COMMON      GLOBAL                   DBID: 200 FNR:  1
COMMAND:                                                                  > +
I  T  L  NAME                          F     LENG       INDEX/INIT/EM/NAME/COMMENT
-  -  -  ---------- ALL ----------     -     ----    ------------------------------
      1  +VAR                          A       22
      1  +NAME                         A       20
```

Figures 1A and 1B.

NATURAL CONSTRUCT

continued from page 1

Developing a program using NATURAL CONSTRUCT is a simple process:

1. The application developer uses standard NATURAL facilities to define views, paint maps, and create data areas as needed in the application.
2. The developer invokes NATURAL CONSTRUCT and selects the model that most closely resembles the program he/she wants to build.
3. NATURAL CONSTRUCT prompts the developer for values that resolve the open characteristics defined in the program model, then generates the program.

Each time you generate a program, NATURAL CONSTRUCT saves the parameter values you specify, so you can regenerate without having to respecify values for parameters you wish to leave unchanged. By supplying NATURAL CONSTRUCT with different values at generation time, an unlimited variety of programs can be generated from each model.

Standard Models

The NATURAL CONSTRUCT package includes models that are used as the basis for generating three kinds of interactive programs: file maintenance, browse, and menu.

Each model incorporates the design expertise of experienced NATURAL system designers. The programs generated from these models are highly functional and efficient, and encourage effective coding practices for developers who are new to NATURAL. Since all generated programs observe NATURAL 2 structured mode conventions, NATURAL CONSTRUCT offers a way to get up to speed quickly with new NATURAL 2 features.

The standard models provided with NATURAL CONSTRUCT include an impressive array of features:

- The file maintenance model supports updates to JOINed files, use of external maps, use of global data, incorporation of validation and referential integrity rules, use of ADABAS MU fields and PE groups, and optional audit trail generation.
- The browse model offers scrollng up, down, left, and right, under the control of both commands and function keys.
- The menu model offers either external or generated maps, plus selection using either function keys or direct commands.

Adapting to Specific Needs

The design of NATURAL CONSTRUCT takes into account the fact that no finite set of application models can accommodate the development requirements of every situation. Sometimes application features are needed that are not supported by any NATURAL CONSTRUCT model. In such situations, the developer has three choices: modify an existing model, create a new model, or modify the generated output.

Modifying the generated output is appropriate when changes are desired for a small percentage of programs generated from a given model. In these cases, the developer simply generates a program using NATURAL CONSTRUCT, then invokes the NATURAL program editor to make the desired modifications. This process is made simpler by the fact that NATURAL CONSTRUCT generates structured, commented source code with a reproducible structure that is easy to understand and modify. And because the generated code is NATURAL, the amount of source code one must deal with is a small fraction of functionally equivalent output from a COBOL generator.

When a variation of an existing program model is needed for a large number of required programs, changing the model becomes a more effective choice than changing the generated output. Among the reasons for changing a model are:

- to enforce design and coding standards
- to enforce user interface standards
- to add calls to shop-standard routines (e.g., for security)

Changing a model can also be an effective way of leveraging development effort when applying mass changes, since changes are automatically incorporated in programs generated from the changed model. Because NATURAL CONSTRUCT models are predominately NATURAL source code, most changes can be made by NATURAL developers with little additional training.

For situations in which no existing model adequately represents the function of a large class of similar programs, NATURAL CONSTRUCT gives you the tools to develop your own models. The mechanisms available for defining the open characteristics of user-defined models include simple string substitution, conditional inclusion of code, and exits to retrieve NATURAL code from external sources via user-written subprograms. By combining these mechanisms with standard NATURAL statements, you can define models for new classes of programs that can be generated by NATURAL CONSTRUCT.

HELP Text Management

Prototyping is just one of the ways NATURAL CONSTRUCT facilitates end user involvement in development. It also offers a comprehensive, yet easy-to-use system for creating and maintaining HELP documents that is suitable for use by end users. HELP text documents can be defined for systems, programs within systems, and even fields within program maps. The HELP text maintenance system uses the NATURAL 2 editor for text entry and editing, specifying page headings, and defining page breaks. During program execution, HELP text is presented in a window, and can be viewed sequentially, or randomly by page.

Advantages of NATURAL Foundation

NATURAL CONSTRUCT is itself written in NATURAL, and runs in all supported NATURAL environments without special prerequisites. NATURAL CONSTRUCT makes use of the NATURAL map editor, program editor, and data area editors, creating an integrated extension to NATURAL's interactive development environment. By generating NATURAL source code, NATURAL CONSTRUCT enables you to make use of your existing NATURAL test environment. And because the NATURAL code generated is environment-independent, NATURAL CONSTRUCT does not concern itself with environment-

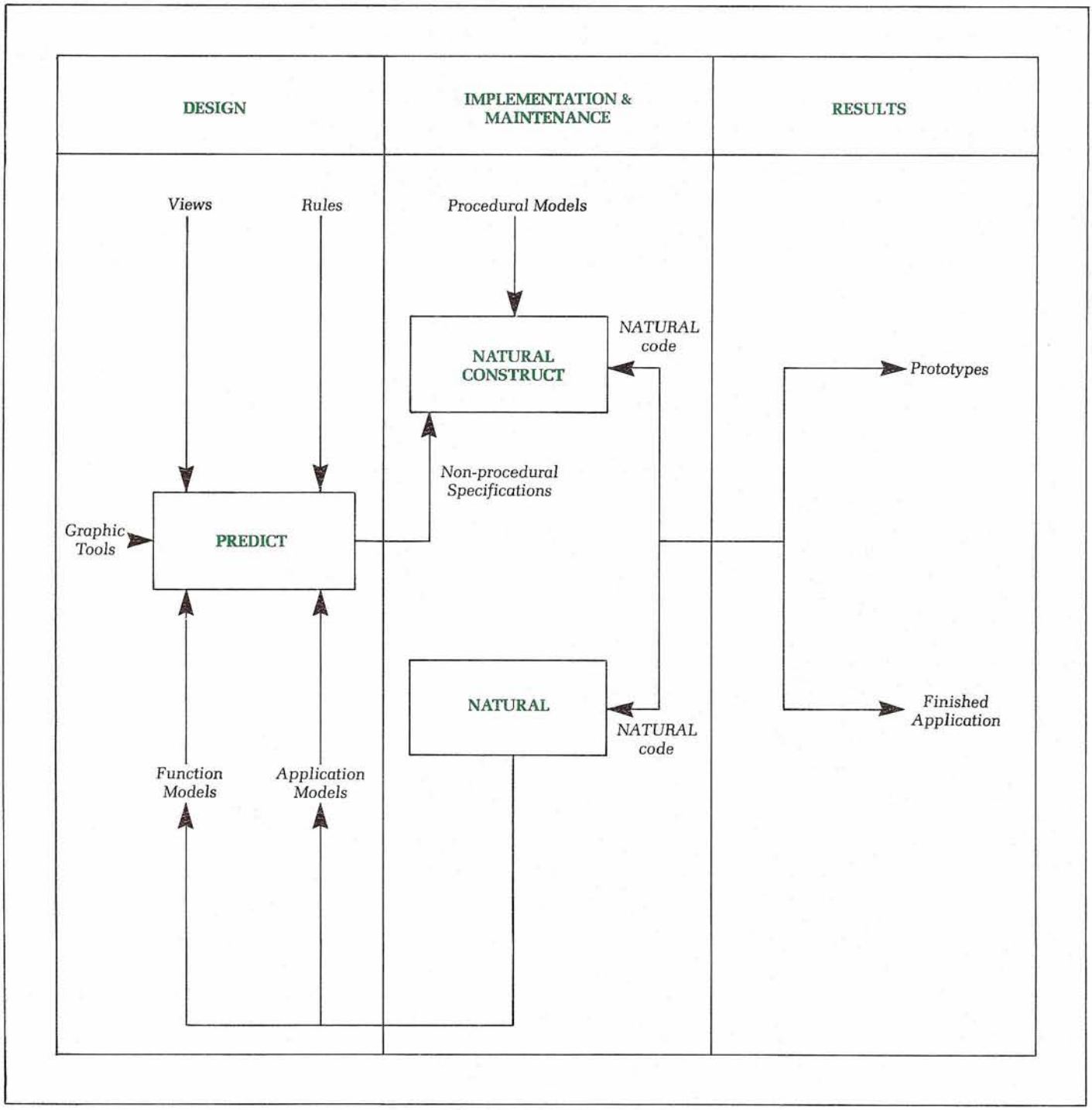
specific details. Generation becomes faster, easier, more flexible, and more reliable than with COBOL-based alternatives.

Foundation for the Future

NATURAL CONSTRUCT Version 2.1 represents the first of several stages planned for the NATURAL Engineering Series. Plans for upcoming releases of NATURAL CONSTRUCT include providing access to generator facilities from the NATURAL editor, offering new and extended application models, and extending PREDICT support to store generation parameters. Other components planned for the NATURAL Engineering Series include PREDICT CASE, a mainframe CASE repository and specification tool for planning, analysis, and design activities.

NATURAL CONSTRUCT Version 1.1 runs in VAX environments under NATURAL Version 1.3, with prices ranging from $3,130 to $25,000, depending on processor size. NATURAL CONSTRUCT Version 2.1 for IBM environments using NATURAL 2 will be available in June, with prices ranging from $7,000 for group 10 processors running VM and VSE, to $45,000 for group 40 processors running MVS.

A 10% discount on IBM prices will be offered for orders placed within the first 90 days of release. For more information, please contact your local Software AG representative or call 1-800-843-9534. Also, you can refer to the enclosed information card to send for more information.



NATURAL CONSTRUCT
IN THE SOFTWARE DEVELOPMENT LIFE CYCLE

NPR

continued from page 5

CIS department, which had the job of infusing a smooth, readily-tapped computer information system into the company. So, much of the (outdated) computer equipment was sold off and the company has now moved to DEC VAXes, with a few exceptions. Frederickson said that ". . . in choosing a new system, management wanted state-of-the-art equipment. As a result, Digital Equipment Corporation's VAX processors were selected."

Today, NPR is operating four VAX 8250 configured as a VAX cluster. Data is stored on eight RA81 disk drives and this configuration supports a network of 300 terminals.

As for software, Frederickson said, "We needed something that was fast, that didn't eat the resources of the computer. We needed to implement elaborate data base management functions and we needed to be able to quickly store and retrieve information." After investigating numerous products, Frederickson chose ADABAS and NATURAL. Both products are marketed locally by Software AG of North America, based in Reston, Virginia.

As Frederickson said, "This software stood out for several reasons. The products permitted the creation of a large data base which did not hog the VAX's memory; they allowed the CIS department to tailor programs to specific departmental needs; and they proved easy to use." Elaborating on the last points, he went on: "We needed to work with a language we had or have a language that would be easy to use and learn. In fact, NATURAL, being a fourth-generation language, allowed us to produce the applications we wanted."

NPR wanted to have all of these features in its computer systems because as a non-profit business it had special requirements. To meet its unique needs, NPR had to create a large data base containing the informational make-up of all of its customers, along with more esoteric systems such as one which predicts when the sun will interfere with satellite transmissions.

Through its five divisions—Distribution, Representation, News and Information Programming, Cultural Programming and Program Services, and Administration—NPR offers an extensive array of services, ranging from managing an innovative satellite system to furnishing training and promotional assistance to its member stations.

John Frederickson, Data Base Administrator for NPR

The Distribution Division manages all technical and operational aspects of the satellite interconnection system, and administers related services including scheduling, facilities coordination and billing.

The News and Information Division produces daily news magazines such as "All Things Considered" and "Morning Edition," which examine a wide range of important national and international issues beyond the headlines. National Public Radio's arts and performance programming, provided by the Cultural Programming and Program Services Division, takes listeners to the great concert halls of the world, to crowded jazz clubs and lively folk festivals. It also features innovative radio drama and cultural documentaries, including the daily arts program, "Performance Today," which uses ADABAS and NATURAL applications as production tools.

Representation is responsible for the development and implementation of policies which support public radio. And the Administrative Division offers institutional support such as CIS, fund raising and financial management.

Obviously, the five divisions have different automation needs which, according to Frederickson, are handled by special applications created with NATURAL. However, their common use of customer information was the thread he used to tie them all together.

A customer information data base was created using ADABAS. The data base contains customer addresses, contact names for varied purposes and general information on the station, such as its signal coverage and audience. The status of member stations and other customers, including independent radio show producers and radio networks, is also contained in the data base.

This information can be tapped by the various departments. For instance, when the Distribution Division's equipment department or satellite depot has to send a repaired piece of equipment to a customer, the name and shipping address of that customer can be pulled up as well as the technical contact's name. Similarly, when the Distribution Division's marketing department receives a request for a program transmission, marketing can check the customer data base to ensure an "authorized requestor" is making the request.

NPR's special applications also tie into this data base for customer information.

SOAP is one of these applications. "It's our sun outage prediction system," explained Frederickson. "Twice a year the sun disrupts our member stations' satellite reception. Although it only lasts a few minutes, it results in a broadcast blackout."

As a result, NPR needs to know when these outages will occur so a transmission is not scheduled during that time period. The geographic coordinates of the stations' satellite ground stations or antennas are stored in the customer information data base. This information is input into the SOAP application, which then uses the geographic coordinates to compute sun spot outages for each station.

Another specialized application was built for the Representation Division's national affairs department. In effect, the program helps speed up the process of tracking support for public broadcasting.

Frederickson noted that NPR keeps track of legislation pending in Congress. When legislation comes up in Congress that affects public radio, NPR provides reports containing this information to its member radio stations. These reports enable members to focus their lobbying efforts. The system provides users with a list of committees that control or affect the legislation, and the names of committee members as well as their districts.

Another system was built for NPR's library. "Whenever a show like 'All Things Considered' goes on the air, our program librarians listen. They itemize and classify each segment of material that went on the air, making note of the reporter's name, people involved, the 'slug' (a one- or two-word story title), and the nature of the piece. For instance, you might look up each story about science fiction author Ray Bradbury or look through all the pieces with Nicaragua and Ronald Reagan in them," Frederickson said.

Other applications which NPR has developed with ADABAS and NATURAL include:

- Inventory Control/Fixed Asset Accounting.
- Purchasing
- Contracts
- Radio Program Production/Distribution

NPR has developed nearly twenty-five systems with ADABAS and NATURAL and adds, on average, one completely new system each month—with a total staff of only eleven in the CIS department. These applications access a total of 115 files in four data bases.

As Frederickson stated: "We have been able to manage information more efficiently through the use of good software tools." He defined a good tool as one which "increases productivity, increases the quality of the product you deliver, increases accountability, and reduces operating costs."

"We've achieved this. We've increased accountability because we can track costs accurately. We've increased productivity because of the rate at which we can generate reports and build systems. And ultimately, we should reduce operating costs because we've created global resources instead of having each department doing their own data processing on different equipment." "Over the last two years," Frederickson continued, "we have produced one new application system a month with ADABAS and NATURAL. Based on new Software AG product announcements, we can foresee accelerating that rate of development even further."

"I'm fortunate to be working within a management structure which allows the implementation of state-of-the-art technologies," Frederickson added. "I can say unequivocally that we would not have been able to deliver twenty-two systems in two years without a fourth-generation technology. During the last five years, the computer department has only added one new slot to its staff."

To ensure its continuing success in radio program distribution, National Public Radio will continue to search for the most cost-effective technical solutions. An integral part of this approach will be the continued use of innovative information management tools from Software AG.

Human Services Solutions: When the Best Meets the Best

Paul F. Peterson
Customer Service Division

What happens when you combine the best system support software on the market with the best social services system in existence?

The answer is Human Services Solutions, a new department within Software AG's Customer Service Division. The purpose of HSS is to provide state governments with optimum human service system solutions.

The first product HSS will take on is the ACCESS system, a social welfare administration system developed by the State of Vermont. Many people think this is the finest welfare administration system in the country, and we agree. The first task of HSS will be to review ACCESS, convert it to NATURAL 2, and make it even better.

Working for me are Jennifer Trombley and Jill Paul. Jennifer is the former Director of Computer Services with the Vermont Department of Social Welfare. Jill worked with Mathematica Policy Research early in the development stages of the project and later worked for the Department of Social Welfare. She also worked with the Income Maintenance Project at Stanford Research Institute.

In the future, Human Services Solutions will use other public domain human service systems. Our intent always will be to identify the best of the existing systems and then make them even better. For further information, please call HSS at 1-800-843-9534.

USER'S GROUP NEWS

NATURAL 2 Conversion Experiences

Darrell Davenport
Co-chairman of the NATURAL tips and techniques SIG

Now that NATURAL 2 has been released for a while many sites have made plans for its place in their production environment. This means there is an enormous amount of attention on conversion at the moment. As most are aware by now, Software AG provides a lot of invaluable documentation on this topic. They have gone to great effort to make this as painless as possible. As a result, there are many ways to do the conversion and each has its own pros and cons. I have been asked to share my views and experiences on this subject. We have been working with NATURAL 2 as a beta site for over a year and we have consistently tried to make the best possible plan of attack on its implementation. We are currently converting our batch programs as we do maintenance on them, developing our new applications in NATURAL 2, and have converted a few online systems. The rest of this document outlines our plan of attack which can be used as a guideline.

First, stay with the recommendations of Software AG as much as possible. In other words, convert whole applications at once, build the GDA, and code the DEFINE DATA GLOBAL USING statement in each program as it is converted (if it uses global variables). We decided to leave the programs as close to the original as possible (not to use the arrays, subroutines etc.) until conversion is over. Then as maintenance is necessary, enhancements can occur. This speeds up the conversion and insures fewer problems during implementation. It also allows more time for the programming staff to absorb the new capabilities of NATURAL 2. As it turns out, the effort of building GDA's and tying them to the programs will have to be done eventually and since it is easy, why not get it over with up front?

Since many of our applications use NATURAL CONNECTION (the PC/mainframe link) we are not able to use the migration option (since NATURAL CONNECTION cannot yet run on NATURAL 2). Therefore, we are running NATURAL 1.2 SM07 and NATURAL 2.2 in parallel. Our single system image software is automatically able to direct each application to the proper environment without the user knowing it.

Conversion is fairly simple and can be done using a distributed, but unsupported Software AG utility: NATCONV. This utility can add a DEFINE DATA section to a program(s) and take out the redundant global variable definitions. Then it re-stows the program. If you encounter difficulties or you don't use global variables, you may choose to re-stow the programs by hand. Either way is easy. Maps are simply stowed to be converted.

Some additional things we found to watch for when converting programs are listed below. These are as of SM02 and may be resolved in SM03. This is not a catch all list:

continued on page 15

ADABAS 5 Performance Update

Bob Becker
Manager, Data Base Administration
Foremost Insurance

At the Software AG International Conference in Miami, we spoke about the performance differences between ADABAS V5 and ADABAS V4. Since then, we have continued our benchmark executions and have determined a rather startling piece of information.

ADABAS 5 has a significant decrease in CPU utilization as the number of concurrent users increases. For example, running 1 user through ADABAS V5 and ADABAS V4 performing 5,000 read random commands, we find that there is a 22% increase in CPU utilization from ADABAS V4 to ADABAS V5. However, running 50 users with the same test criteria and same software parameter settings, we find a 37% decrease in CPU utilization from ADABAS V4 to ADABAS V5. It appears as though Version 5 has optimized its code for the environment which contains a higher number of concurrent users.

Even with 10 users concurrently active within ADABAS 5, we find a significant decrease in CPU utilization over Version 4. Therefore, it would appear that on-line environments will be benefiting most from the Version 5 performance improvements, rather than the single user environments.

The ADABAS dump/restore facility has also been significantly improved with allowing multiple tape or cartridge drives to be utilized while dumping the entire database. We are capable of dumping an entire 25 single density 3380 database in 30 minutes of elapse time using five tape or cartridge drives. This is a very improved facility within the utility. It should allow many more installations the ability to dump an entire database on a daily basis. Of course, the dumping can occur while the database is being updated, which is the same facility that is presently available in ADABAS V4.

If you wish to discuss these performance improvements with me, please feel free to call me at Foremost Insurance Company—(616) 956-2664.

ADABAS Product Representative Report

Bob Becker
Manager, Data Base Administration
Foremost Insurance

This is my first article in Software AG's Connections as the ADABAS Product Representative. My name is Bob Becker and I currently hold the position of Manager of Data Base Administration at Foremost Insurance in Grand Rapids, Michigan. We at Foremost are long-time users of Software AG products, having used ADABAS for the past 14 years, and NATURAL for the past 9 years.

This is a very interesting time for all ADABAS users because ADABAS 5 is close to a general distribution release. As many of you know, I have been testing ADABAS 5 since October, 1986. I think you will be very impressed with ADABAS 5, particularly with the on-line services. ADABAS on-line services provides a high degree of interaction with ADABAS from the Data Base Administrator's perspective. You may be interested in reading about some of the performance highlights in the article entitled *ADABAS Version 5 Performance Update* which appears in this edition of Connections.

Since the International Conference in Miami, several personnel changes have occurred within the user community. Kerrie Meyler has taken the position as Southwest Regional Representative, replacing Larry Godec. Larry has become the Executive Committee Secretary. As a result of Kerrie's transfer, the DOS Special Interest Group position is vacant. If you are a DOS user and would like additional information about becoming this very important Special Interest Group Chairperson, please contact either me at Foremost Insurance or Larry Jayne at Software AG.

Over the past few months, I have spoken with several of you about ADABAS issues, including performance topics and ADABAS change enhancements requests. If you have any ADABAS related issues, questions, or suggestions to improve the product, I would enjoy discussing them with you. It is only through our input to Software AG that we can improve upon the ADABAS features and product performance. Please contact me at Foremost Insurance Company—(616) 956-2664.

NATURAL 2

continued from page 14

1. NATURAL 2 will print in column 1 (NATURAL 1.2 always started in column 2). This may impact your microfiche jobs or if you have special characters in column 1 and your printer is looking for carriage control, your page spacing may be off.
2. AT TOP OF PAGE cannot be in REPEAT UNTIL loops (implied DO . . . DOEND). The same goes for WRITE TITLE and AT END OF PAGE etc.
3. IF . . . DO . . . ELSE must have the closing DOEND before the ELSE.
4. OBTAIN GROUP-WITH-MU-WITHIN-PE(1-n,1-n) will cause syntax error during compile if any of the elementary fields within the PE are not MU.
5. TERMINATE #COND-CODE, if inside of a READ WORK loop, or after a WRITE WORK FILE will not close the file, which causes a C03 abend. It also now reserves condition codes 1-31 for NATURAL messages. If you use those values, expect strange messages on otherwise normal looking output.
6. RESET *PAGE-NUMBER does not syntax check—use MOVE 0 TO *PAGE-NUMBER as a solution.
7. If your maps from NATURAL 1.2 have modified names for the fields (not IFIELD01 etc.) read the documentation on passing parms to maps for NATURAL 2. You may need to do some extra conversion effort.
8. At break with work files may work differently than you expect, especially if you are using system functions like "OLD(#VAR); (you will get an abend during compilation message). And PERFORM BREAK PROCESSING may cause the last element to print twice. (Put a STOP just before the END to fix this.) If you use AT BREAK with workfiles a lot, anticipate some extra time for conversion.
9. WRITE statements that use "/" for line feeds: if the "/" starts a new page you may or may not get correct title and header produced. And if the page size is hit exactly at the same time as the logical page ends, NATURAL may try to "window" your output and echo the page without the top header line.
10. Disallowed modules with SYSSEC (SM02) works differently and once a fetch is attempted, an error will not recover from NAT0963. You will need an error handler for the application to catch the messages now.
11. If you use startup modules (defined to security) be sure they transfer control to another program before they quit. If not, then you must add a statement to blank the startup program name. The reason is that the startup module replaces the NEXT prompt and unless control is transferred or the startup is blanked, you will be in an endless loop. The way to blank the startup is: "MOVE" TO *STARTUP".
12. *DATJ; the format has changed slightly.

For those programmers used to the NATURAL 1.2 editor some commands have been changed:

The "P" command that used to split the editor screen and put a programs' source on the bottom half now purges the program's source from your library *without warning.*

If you are used to using "S" for scanning source, it now is the split screen command—"SC" is scan. SCAN will now search from the current line to the bottom, not from the top each time. However, you can stack editor commands to scan from the top if you wish (i.e. T,SC #VAR1). To continue the scan you must enter "SC=" (or push PF10 that is defined as "SC=")

You can now EXECUTE programs from the editor, but don't use the "EX" command to do it. "EX" means Erase from the top line to the line marked "X" in the source work area.

I hope these suggestions help you with your conversion and save you sometime.