SOFTWARE AG: CUSTOMER INFORMATION

CONNECTIONS

VOL. 13 No. 4 ■ DECEMBER, 1988 FALL ISSUE

Software AG's New Headquarters

The following is an excerpt of a letter that was sent to 4,700 contacts in almost 1,000 user sites in the United States.

John Norris Maguire
Chairman of the Board

I'm pleased to announce that after fifteen years headquartered in the Reston International Center, we now have our own building! During that period in the International Center, we outgrew available space necessitating locating a substantial portion of our Reston staff into numerous buildings around the Reston area.

Now, we have consolidated our 260 Reston-based employees under one roof which should improve our productivity and service to you.

Our new address is:

Software AG of North America, Inc.
11190 Sunrise Valley Drive
Reston, Virginia 22091
Main switchboard telephone:
703/860-5050
FAX: 703/391-6975
TELEX: 275301

We are very excited about our new products and especially our open Integrated Software Architecture (ISA). This architecture is available today and running (as opposed to others that won't be here until the 1990s and only on *one* hardware platform).

We would like to extend an invitation to you and your colleagues to visit our new Headquarters. Your visit would be tailored to your needs/desires be it about products or any other detail you desire. We would also be happy to discuss with you all of our product plans for the future.

Please contact your local salesperson to arrange such a visit. I would participate, if appropriate, in the visit.

I hope you accept my invitation.

NATURAL 2 Exceeds COBOL Performance Levels

Doug Henrich
Product Manager

NATURAL 2 applications can now exceed COBOL performance and rival optimized COBOL (CA Optimizer) applications. These performance levels are achieved through the use of the NATURAL 2 OPTIMIZER COMPILER. Software AG announced the NATURAL 2 OPTIMIZER COMPILER in September at the 18th International Users' Conference in Nashville.

Test results from beta site customers show that optimized NATURAL 2 applications, on average, perform up to 3 times faster (in CPU time) than standard third generation COBOL! And as fast as optimized COBOL, in its execution of statements for complex data manipulation.

Beta site customers also found that new and existing NATURAL 2 applications using the NATURAL 2 OPTIMIZER COMPILER will experience significant overall CPU savings.

How It Works

The NATURAL 2 OPTIMIZER COMPILER compiles NATURAL programs into optimized machine code, providing considerable reductions in CPU usage for programs which contain a large amount of data manipulation, such as array processing, computation, transfer and logical condition processing. The NATURAL 2 OPTIMIZER COMPILER compiles all assignments, such as arithmetic, conditional and control statements, into optimized machine code. Because of this optimization, the

continued on page 5

Inside this Issue:

President's Message

Bill Wagner
University of Texas at Austin

One of the more fascinating of the old Chinese curses reads: "May you live in interesting times." The Nashville Conference showed that times are indeed interesting for both Software AG and the user community. Challenges are ahead for both parties. As Michael Braude of the Gartner Group pointed out, the spectre of DB2 is alive and well (even if not particularly efficient) in the data processing world, and Software AG may have some interesting choices to make if their strategies are to reflect this reality. The challenge facing the user community is to assimilate the tremendous amount of new products that Software AG is offering. For one person to simply know what products are available is difficult; to understand something of the internals of all the products is virtually impossible. The result is that as Software AG products proliferate, so does the number of people responsible for their selection, installation, and maintenance. In short, the number of people in the user community is growing far faster than the number of sites.

SAGGROUP has taken a big step toward accommodating both the number of users and the variety of products with the approval of several Constitutional amendments. The four positions of "Product Representatives" have been expanded to five "Functional Area Representatives," each looking after the interests of an entire family of Software AG products. As new products are introduced, they will be added to one of the existing categories. Each of the Representatives may also, at their discretion, appoint Technical Advisors for individual products. These advisors will participate directly in SAGGROUP through the Change/Enhancement process, providing the expertise in specific products that the Functional Area Representative cannot be expected to have.

continued on page 9

Table of Contents

SOFTWARE AG NEWS

18th International Users' Conference A Success

The 18th Annual International Software AG Users' Conference, held in Nashville, Tennessee on September 25-29, 1988 at the Opryland Hotel, proved a success. This year's conference featured an expanded demo room, over 80 sessions, 15 new product annoucements and the second annual Executive Track Conference.

YEARLY CONFERENCE BRINGS USERS TOGETHER

The International Users' Conference provides a forum for users to meet and share ideas. This interaction with other users and Software AG is an invaluable part of attending the yearly users' conference. At no other time are nearly 2000 Software AG users assembled to discuss how they use Software AG products to solve their business problems.

VARIETY OF PRODUCTS MAKE DEMO ROOM A SUCCESS!

This year's demo room featured 24 booths displaying 50 products for IBM, DEC, and WANG hardware. Users enjoyed the demo room atmosphere which provided individual enclosed classrooms seating 15 people each. Product demos were held 12 hours a day to meet heavy demand. Also, each demonstrated product was graphically displayed as a part of ISA, Software AG's open Integrated Software Architecture.

OVER 80 SESSIONS PROVIDED UP-TO-DATE INFORMATION

During the 5 day conference in Nashville, over 80 sessions were held and many of them concurrently. Software AG employees, users, and industry analysts presented a vast amount of information through presentations, classes, and workshops.

Some Software AG Highlights

- ISA—Software AG described its open Integrated Software Architecture in detail
- 15 New Products Announced
- Open Forum—Users met one-on-one with Software AG executives
- Presentations were given on new products and new product versions
- Classes were taught on various products

Here's a Sampling of Users' Presentations

- Experiences with ADABAS SQL
- How to "Turn On" Data Administration
- Financial Package Built on ADABAS/NATURAL
- Hugh Volume—Creative Solution
- NATURAL Optimizer Testing and Performance
- Using ADABAS/VMS in a High-Availability VAX Cluster Environment

Industry Analysts Discussed Industry Trends

- John Logan, Vice President, Aberdeen Group, described

continued on page 26

NATURAL CONSTRUCT VMS Overview

Software AG incorporates Computer-Aided Software Engineering (CASE) technology with NATURAL, the world's most widely used and most highly acclaimed fourth generation technology. The result is NATURAL CONSTRUCT, a CASE tool that extends the portable NATURAL environment into even greater realms of productivity, functionality, and high performance. NATURAL CONSTRUCT provides a quick and easy way to generate efficient, high-quality, environment-independent NATURAL applications.

NATURAL CONSTRUCT is one of the first products to be released in the NATURAL Engineering Series, a suite of integrated CASE products that provide automated support for application development activities throughout the system development life cycle.

NATURAL CONSTRUCT is an application generator designed to support development of business applications using NATURAL. It facilitates both rapid prototyping and development of complete business applications. All NATURAL CONSTRUCT functions are interactive, including program specification, generation, and testing. NATURAL CONSTRUCT makes use of existing NATURAL facilities for painting maps and defining application views and data areas. To this base, it adds a program generator, a Help maintenance system, and a library of program models for the generation of interactive file maintenance, file browsing, and menu programs.

NATURAL CONSTRUCT's standard program models are application templates that incorporate comprehensive error-handling, context-sensitive Help, cursor-selection, omnidirectional scrolling, and other productivity-enhancing features that are typically omitted from hand-coded applications, due to time constraints. Organizations can modify the standard models to add or change application functions. Standard models can also be modified to enforce local standards for code structure, documentation, security, and user interfaces.

The next version of NATURAL CONSTRUCT Version 1.2, will be available in December, 1988. NATURAL CONSTRUCT 1.2 will include support for NATURAL ARCHITECT Workstation and several enhancements to current program models. Model enhancements include:

- multi-file support
- multi-map support, and
- an increased number of user exits.

For more information on NATURAL CONSTRUCT contact your Software AG representative or fill-out the reply card located in this issue of Connections.

Users Give NATURAL Customer Training Highest Marks

Stan Maring
Customer Service Division

According to a June, 1988 InformationWEEK survey of 4GL users, our customers like the NATURAL training we provide. In fact, when competing against nine other 4GLs, Software AG's NATURAL training was rated the *best*!

That's not really a surprise! We frequently receive letters and comments from customers praising our courses and instructors. Here are excerpts from some we've received recently:

- "The course was a resounding success, due mostly to the efforts of your instructor."
- "The instructor's knowledge of the subject matter, clear presentation, enthusiasm, and attention to detail made the class enjoyable and informative for all involved."
- "Your instructor's willingness to spend additional time covering workshop problems was appreciated by many students."
- "Every student gave your instructor outstanding reviews on her instruction and overall effectiveness. She was very well received by everyone."
- "After attending four Software AG courses, I have found all of them to be very professional, very detailed, and a pleasure to attend."
- "Software AG's instructors were excellent. All were willing and able to apply a student's application to the course of instruction. That was very valuable."
- "I have heard nothing but the highest praise for your instructor and the way she conducted the class and workshops. Even our resident grouch loved her!"
- "We expected outstanding training from Software AG when we selected your products, and we have not been disappointed."
- "Your instructor's knowledge of the course material, presentation skills, and ability to relate to the students made this the most valuable and informative Software AG course that we have attended."
- "Your instructor stands as a credit to Software AG and to your Customer Training Department."
- "I talked to many of the students after the class and they, too, were unanimous in the high marks given the course."
- "Your course was a real pleasure for me to attend. Your instructor was knowledgeable and enthusiastic about his subject matter and imparted that to his students."

For more information on Software AG's training classes please contact the Training Registrar at (703) 391-6904.

NATURAL Elite's ADABAS 5 Courseware Series

Susan Kaplar
Product Technology Division

The NATURAL Elite group is busy preparing for the release of ADABAS 5. Elite, Software AG's computer-based training, will be providing training in ADABAS 5 for a variety of audiences. The ADABAS 5 courseware series consists of four courses and is planned for release this year. The courses are described below.

ADABAS 5 Fundamentals

This course, the first in the ADABAS 5 series, is designed for those needing to access the ADABAS data base system—whether they be end users, programmers, or DBAs. The course teaches the fundamentals of ADABAS 5 concepts and structures.

Beta test: October 1988
Release: November 1988

ADABAS 5 Program Design

This course is designed for programmers using a 3GL—such as COBOL or PL/1—to access ADABAS files. Examples of the direct calls required in the program are taught. All example code uses COBOL and/or PL/1. A brief summary of the user of NATURAL with ADABAS is included as a lesson.

Beta test: October 1988
Release: November 1988

ADABAS 5 File Design

This course is designed for systems analysts, DBAs, or advanced programmers, involved in designing ADABAS files for application development. The design is not a systems design course; rather, it is instruction on the types of functionality and considerations involved in designing an ADABAS file.

Beta test: November 1988
Release: December 1988

ADABAS 5 Utilities

This course, the last in the ADABAS 5 series, is designed primarily for DBAs or other system support personnel involved in supporting the ADABAS environment. Each utility is covered in concept and examples in batch. This course reviews the utilities as they are available in the ONLINE mode through the ONLINE Utilities module of ADABAS.

Beta test: January 1989
Release: February 1989

Remember, NATURAL Elite is an economical way to provide quality training for your shop—be it large or small.

NATURAL 2 Exceeds

continued from page 1

product produces further CPU consumption improvements over NATURAL 2.

Compiled or Not Compiled—

The NATURAL 2 OPTIMIZER COMPILER compiles the following statements:

- move statements, (i.e., RESET, ASSIGN, and MOVE),
- arithmetic statements (COMPUTE, ADD, SUBTRACT, MULTIPLY, and DIVIDE),
- conditional statements (IF, DECIDE),
- control statements (FOR and REPEAT).

The NATURAL 2 OPTIMIZER COMPILER does NOT compile the following statements:

- I/O statements (data base operations, INPUT, WRITE, etc. . . .),
- complex special statements like EXAMINE or SEPARATE,
- calling statements like FETCH, PERFORM, CALLNAT, etc. . . .,
- the MOVED INDEX statement (array processing is optimized).

When To Use the NATURAL 2 OPTIMIZER COMPILER

Programs which contain string processing, command analysis, statistics, complex arithmetic, and the like, will find the NATURAL 2 OPTIMIZER COMPILER highly efficient and therefore is strongly recommended.

However, the Optimizer should not be used for programs which are only used very rarely. Also, the Optimizer will be of little benefit for programs which contain only data base operations. These programs may realize substantial performance gains with the ADABAS High Performance Environment (HPE). ADABAS HPE eliminates expensive inter-region communications.

Using the Optimizer

The NATURAL 2 profile parameter MCG (Machine Code Generation) is used to activate the Optimizer. This may be dynamically assigned for a given NATURAL 2 session. Additional options can check for arithmetic overflow, division by zero, and index boundary errors.

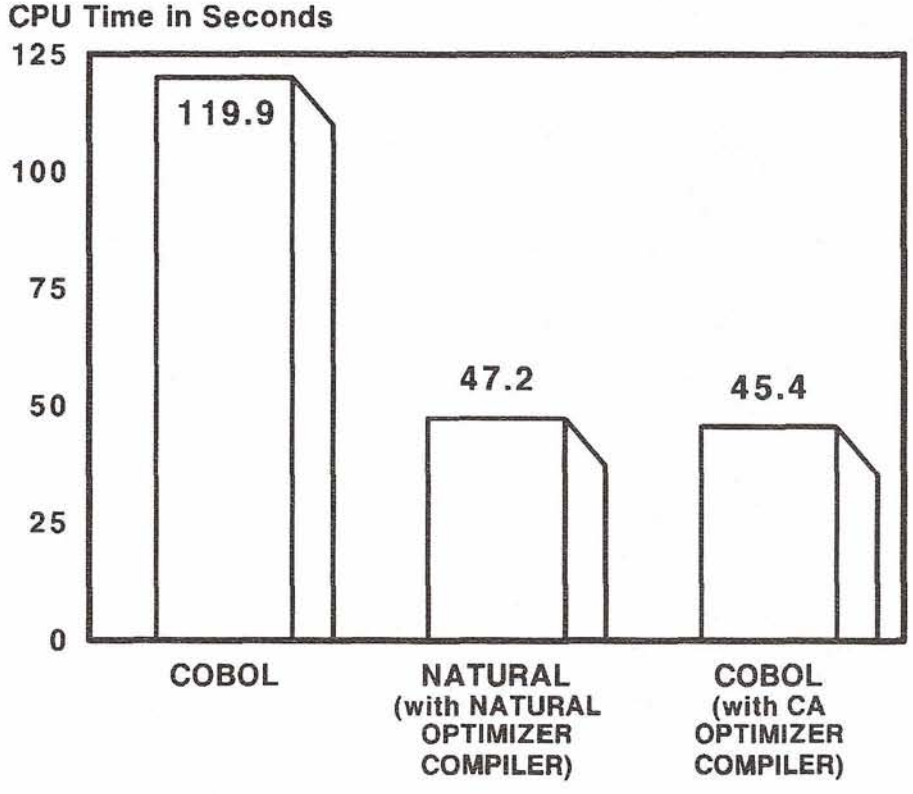
The NATURAL 2 OPTIMIZER COMPILER operates on IBM, IBM plug compatible and Siemens hardware platforms, MVS/SP, MVS/XA, VM/CMS, DOS/VSE, VSE/SP, BS2000, and MSP operating systems, and Software AG's COM-PLETE, and ADABAS TPF teleprocessing environments, as well as IBM's CICS, TSO, CMS, and IMS/DC teleprocessing environments. NATURAL 2 OPTIMIZER COMPILER compiles NATURAL 2 programs accessing data from ADABAS, VSAM, DL/1, DB2 and SQL/DS, in both on-line and batch environments. NATURAL 2 OPTIMIZER COMPILER will be available shortly after the release of NATURAL 2.1.4. Please contact your Software AG representative for product details and pricing information.

Benchmark - Frankfurt, West Germany

Command Profile

Move	**58.6%**
If	**19.5%**
Perform	**11.4%**
Add	**7.8%**
Compute	**1.3%**
Call	**1.1%**
Subtract	**0.2%**
Multiply	**0.1%**
Total	**100.0%**

Total Commands
10,000,000



SUPER NATURAL and Batch Transactions

Lisa Crispin
Customer Service Division

One of the most frequently-asked questions about SUPER NATURAL is how to run SUPER NATURAL transactions in batch mode. Creating and running batch SUPER NATURAL transactions isn't difficult, but several factors are involved. The SUPER NATURAL administrator has to decide who will be allowed to create and run batch transactions; decide how batch transactions will be submitted; set up the correct SUPER NATURAL options; code the JCL/JCS; and take care of security considerations and various other issues.

The SUPER NATURAL Installation and Maintenance Manual documents batch run mode (pg. 6-28 and 6-33 of the V2.1 manual; pg. 5-15 of the V2.2 manual) and JCL/JCS for batch transactions (pg. 6-38 of the V2.1 manual; pg. 5-63 of the V2.2 manual).

For a detailed discussion of how to run batch SUPER NATURAL transactions, read on!

Why Batch Mode?

Some SUPER NATURAL users may want to run a transaction that reads many records and/or prints a long report. For reasons of efficiency, these transactions may be better run in batch mode. Another reason for running in batch mode is that the administrator may have defined some user options to be available only in batch mode. For example, non-descriptor criteria may be allowed only in batch mode.

RJE Submission

Standard batch NATURAL jobs can be used to run SUPER NATURAL batch transactions. Since SUPER NATURAL users are usually end users who would not normally code JCL and submit batch jobs, SUPER NATURAL provides for submission of batch jobs through SUPER NATURAL using RJE (Remote Job Entry) facilities. The SUPER NATURAL administrator sets up the JCL and the user options appropriately; the user than can run the transaction and the batch job is automatically submitted. RJE submission is available only for COM-PLETE and NATURAL Advanced Facilities users. Other users are limited to running batch transactions using the same methods they would use to run regular NATURAL programs in batch.

NOTE: To use RJE in SUPER NATURAL under COM-PLETE, the module NATRJE has to be included in the linkedit of the NATURAL nucleus, and must be a CSTATIC entry in NATPARMS. NATRJE is the same module as COM-PLETE's RJE. NATURAL/AF users already have NATRJE in place, so no extra steps are required.

Setting Up For Batch Transactions

To create a batch mode transaction in SUPER NATURAL, the user option for Batch (B) Run Mode must be set to "Y." The appropriate report destination must also be available. For example, if output is to be routed to a printer, Printer (P) must be set to "Y" and the desired Logical Printer Number must be assigned.

A user who has Run Mode Batch = Y in the user options can save batch mode transactions. In order to run batch jobs from SUPER NATURAL, RJE must be available and additional steps must be taken.

If RJE is to be used, the SUPER NATURAL administrator must code JCL/JCS for a batch NATURAL job. A unique JCL member can be created for each user, or more than one user can use the same JCL member, as desired. The JCL members are stored as source members in NATURAL library ZJCL.

Administrators can add, copy, or modify JCL members from the SUPER NATURAL maintenance menu. The ADD and MODIFY options invoke the NATURAL editor, where JCL can be coded and SAVEd, then a "." used to return to the maintenance menu. It's also possible to logon to the ZJCL library and edit the JCL members in NATURAL without going through the SUPER NATURAL maintenance menu.

Once one or more JCL members have been created, the administrator needs to modify the user options for each user who will be allowed to submit batch transactions through RJE. Three options need to be updated:

1. JCLID: This is the name of the JCL member the user will use. The default is "JCL01"
2. Run Mode Option for Batch (B) must be set to "Y"
3. Batch Job Entry/(RJE available) must be set to "Y", with default "Y"

Now the user can run batch transactions and SUPER NATURAL will automatically submit the batch jobs.

JCL/JCS

JCL/JCS may be coded the same way as any batch NATURAL job for the site. The JCL/JCS must execute batch NATURAL with appropriate NATPARMs. If JCL is to be submitted outside of SUPER NATURAL (in other words, without using the SUPER NATURAL RJE facility), the NATURAL control statements can be coded to logon to SYSSN and execute the transaction from the user's private library, or logon directly to the private library and execute the transaction. For example:

```
LOGON SYSSN
EX IPGMNAME YLIBRARY
```

or

```
LOGON YLIBRARY
EX IPGMNAME
```

NOTE: NATURAL SECURITY users have special requirements; see the section on NATURAL SECURITY considerations below.

A JCL member for use with SUPER NATURAL RJE is coded in the same way except for the NATURAL control statements: the //CMSYNIN DD * data for OS, or the instream data following the ADARUN card and/* for DOS. If a ">" is coded in column 2 on a line by itself, SUPER NATURAL will inset the correct LOGON and EX commands when the transaction is submitted to RJE. Using this recommended method for a transaction named "REPORT1" for the user SALZC, the administrator would code the NATURAL control statements in the JCL as this:

```
>
FIN
```

When the job is submitted, SUPER NATURAL will insert commands to change the instream data to this:

```
LOGON SYSSN
EX IREPORT1 YSALZC
FIN
```

("I" is the program prefix for transactions, and "Y" is the private library prefix.)

The ">" insertion character allows a JCL member to be used over and over by the same or multiple users without having to modify the JCL member. Another feature SUPER NATURAL provides for flexibility in JCL is the use of the character string "*******", which is replaced by the User ID of the user running the transaction. This can be used in the job card (and POWER job card in DOS):

Example for OS:

```
//******* JOB BATCHNSN,
CLASS=A,
MSGCLASS=X
```

Example for DOS:

```
* $$ JOB JNM=*******,
CLASS=A,
DISP=D // JOB *******
```

NATURAL SECURITY Considerations

If SUPER NATURAL is installed in NATURAL SECURITY, the NATPARM AUTO=ON must be used, as long as the user submitting the batch job is defined to NATURAL SECURITY as a person or an administrator. If the user is defined only as a member, AUTO=ON can still be used if SYSSN is the default application for that User ID. If the member does not have SYSSN for a default application, the NATURAL SECURITY logon commands must be coded in the JCL. For example:

```
SYSSN,USERID
%*
PASSWORD
```

If batch transactions are run in SUPER NATURAL using RJE to submit the jobs while the user is still logged on to SUPER NATURAL, steps must be taken to prevent an ADABAS RSP048 (Duplicate User ID in ADABAS) from occurring.

In NATURAL SECURITY V2, define application SYSSN with RESTART = N on the application profile. Restart data is not used in SUPER NATURAL, and turning RESTART off will prevent a duplicate User ID from occurring.

In NATURAL SECURITY V1.2, an error routine to trap NAT3048/NAT8048 at logon time must be used. (Please see NATURAL V1.2 SM06 Release Notes for details on coding this error routine.) The error routine must reside in SYSLIB, and must be defined as the error routine on the user's default application of AUTO=ON is used. If AUTO=ON cannot be used because the User ID is defined as a member with no default application, the JCL to run the SUPER NATURAL transaction must contain control statements to do the initial logon to an application that has the NAT3048/NAT8048 trap defined as its error routine. At logon time, the error routine should ignore the NAT3048 or NAT8048 and allow the job to continue with the "LOGON SYSSN".

As with vanilla NATURAL, there are two ways to execute SUPER NATURAL transactions in batch. The control statements can be either:

```
LOGON SYSSN
EX IPGMNAME YLIBRARY
```

or

```
LOGON YLIBRARY
EX IPGMNAME
```

Since SUPER NATURAL private libraries are not defined as applications to Security, the administrator would have to define each private library to Security before using the second method. Therefore, the first method, LOGON SYSSN, is recommended. Again, if the RJE facility is being used to submit batch jobs from SUPER NATURAL, use of the insertion character ">" is recommended to allow SUPER NATURAL to insert the LOGON SYSSN and EX IPGMNAME LIBRARY commands.

MENU must be defined as the STARTUP program for SYSSN in order to execute batch transactions from SYSSN. MENU accepts the command EX IPGMNAME YLIBRARY where IPGMNAME is the transaction name and YLIBRARY is the private library as input to a variable called #BATCH-LINE.

If the control statements are hard-coded rather than using the ">" insertion character, the transaction name must consist of eight positions. If the name is less than eight characters, trailing spaces must be used to pad the name to eight positions. For example, if transaction ITEST in library YSALZC is to be run:

```
LOGON SYSSN
EX ITEST YSALZC
```

will result in an error under NATURAL SECURITY.

```
LOGON SYSSN
EX ITEST      YSALZC
```

will work correctly. SUPER NATURAL generates the commands correctly when ">" is used.

Caveats

Once a job is submitted to run a SUPER NATURAL batch transaction, the transaction should not be modified until after the job has run. If changes are made before the batch transaction is run, it may get an error, or fail to produce the expected output. For example, say a user runs a batch transaction, but the job does not run right away. Then the user changes the transaction defaults to online run mode. When the batch job finally runs, it will try to run the transaction which is now online mode, and terminate with an error.

If report destination PRINTER is used, the online NATURAL session where SUPER NATURAL is running must have the PRINTER parameter set high enough in the NATPARM or dynamic override so that the code generated by SUPER NATURAL will pass the syntax checker. For example, if report destination is logical printer 8, PRINTER=8 (or higher) must be used. The number of printers allocated can be checked by typing in the command GLOBALS at a NEXT prompt (or NATURAL 2 command line).

Note also that batch JCL to execute a batch transaction must contain a //CMPRTXX DD card (for OS) or BPRINTD parameter (for DOS) that matches the logical printer being used. For example, in OS, if logical printer 2 is the destination, the JCL must contain a //CMPRT02 DD card. The same idea also applies to workfiles as a report destination—BWORKD or //CMWKFXX DD must be present for the workfile number being used.

I hope you find these tips are helpful as you run SUPER NATURAL transactions in batch mode.

Processing Rules in PREDICT 2 and NATURAL 2

Wendy Crist
Customer Service Division

General Information

NATURAL 2 uses processing rules to check the validity of input data for specified fields. The use of processing rules moves the validity check of input data from the program to the input map where the data can be checked as it is entered and before program processing begins.

PREDICT 2 provides a tool for centralizing the definition, maintenance, and storage of these rules. The use of centralized rules can allow for a consistent implementation of the rules.

Processing rules can be defined through PREDICT or through the NATURAL map editor. There are four types of processing rules: *inline, conceptual, free,* and *automatic.* Inline and free rules can be defined in the NATURAL map editor. Conceptual, free, and automatic rules can be defined in PREDICT. The status of the rule (conceptual, free, or automatic) is assigned in PREDICT depending on how the rule is to be used.

Inline rules are created using the NATURAL 2 map editor. The rule is added or updated by editing the desired field on the map and specifying processing rule editing. Inline rules are stored with the map in which they are created.

Conceptual rules are in the design stage and cannot be used in a map until their status is changed to free or automatic. Conceptual rules are created and maintained through PREDICT.

Free rules may be used on any field (data base or user-defined) in a map at the programmer's discretion. These rules are created and maintained through PREDICT or through the NATURAL 2 map editor.

Automatic rules are generated in the same manner as free and conceptual rules. The rule becomes automatic when it is connected to a field. The connection is set up when the rule name is added to the field entry in PREDICT and the DDM for the file or user view containing the field is generated. The rule becomes part of the DDM and PREDICT changes the status of the rule from free or conceptual to automatic. These rules are created and maintained through PREDICT.

CREATING VERIFICATION RULES

In PREDICT

The verification TYPE and VALUE(s) are used to generate the NATURAL code for the rule. These are supplied by the user when a verification is added or modified in PREDICT. A verification can be of the following types:

- range, equal, less, greater, user, or table

Range is used with two values to indicate a consecutive set of values. Equal, less, and greater are used with one value. For a table type rule, several values may be specified. For all types, except user, PREDICT uses the type and value(s) to create NATURAL code. Code for verifications of user type must be supplied by the user.

To create a rule under PREDICT, add a verification with a TYPE and allowed VALUE(s). From the add screen on the verification menu, choose the option to EDIT the verification. On a new verification, this will display a blank edit screen. PREDICT will generate NATURAL code for the rule if the command GEN is entered on the command line. The code may be modified by the user.

Structured code can be generated by issuing the command GLOBALS SM=ON from the verification rule editor. The CHECK command can be used to check the syntax of the rule. The CAT command does not perform a syntax check. In PREDICT, the CAT and SAVE commands perform the same function: to save the entered code.

To save generated code as a conceptual rule, issue the SAVE or CAT command. To save it as a free rule, enter the command SAVE FREE or CAT FREE.

To create an automatic rule, the verification is saved as a free or conceptual rule. The verification name must be added to the desired field, and the DDM for that file or user view must be generated. At this time, PREDICT changes the status of the rule from free or conceptual to automatic. The rule becomes a part of the DDM, and the code for the rule is stored in the DDM, as well as on the FDIC file. When the DDM is used in a NATURAL map, the rule is included automatically.

In NATURAL

Inline rules are created in the NATURAL 2 map editor. Select option "D" for DISPLAY FIELD DEFINITIONS from the map editor menu. A "P" should be entered next to the field for which a processing rule is being added. It is possible to enter a rank on the rule to set a priority for the rule. The rank is entered along with the "P" in the format "Prr" where "rr" is a value from 0 to 99 (refer to the NATURAL Reference Manual section on the map editor for suggested rank settings based upon the rule type and usage). If this is a new rule, a blank edit screen will be displayed. The user must then enter the desired code. An END statement *must not* be entered in this code. When finished, enter a "." on the command line.

To use a free rule from the NATURAL 2 map editor, select option "D" for DISPLAY FIELD DEFINITIONS from the map editor menu. A "P" and optional rank should be entered next to the field for which a rule is being pulled in. A blank split edit screen will be displayed. The user must then enter the name of the free rule in PREDICT on the underscores above the ">" symbol. After pressing ENTER, the code for the rule will be displayed. Enter a "." to save the rule for that field.

To create a free rule from the NATURAL 2 map editor, select option "D" for DISPLAY FIELD DEFINITIONS from the map editor menu. A "P" and optional rank should be entered next to the field for which a rule is being defined. A blank split edit screen will be displayed. The user must then enter the desired code. An END statement *must not* be entered in this code. To save the rule as a free rule in PREDICT, enter the verification rule name on the underscores on the line above the ">" symbol and enter a "." on the command line. The PREDICT ADD VERIFICATION screen will be displayed. The status

and format of the rule will have been set by PREDICT. The user needs to enter the verification type and value(s). The rule is stored on the dictionary.

MAINTAINING VERIFICATION RULES

In PREDICT

User, automatic, free, and conceptual rules are maintained through the PREDICT maintenance subsystem.

If an automatic rule changes after the DDM is generated, there are two ways to get the updated rule into the DDM. The first way is to re-generate the DDM specifying "Y" on the generate and replace verification rule(s) options. The second way to update the rule(s) in the DDM is to issue the GEN VER command. This command will only update the rules in the DDM. It will not process any other changes made to the file or userview (for example, new fields added or formats changed). If the file or userview fields entries were changed in addition to the rule changing, the DDM must be re-generated. The GEN VER command will not update the DDM unless the verification was included when the DDM was first generated.

In NATURAL

Free and inline rules can be maintained through the NATURAL 2 map editor. They are maintained in the same manner as when creating free and inline rules. When free rules are maintained through the map editor, the PREDICT verification rule maintenance system is invoked so that the rule definition in PREDICT remains consistent with the code in the map. The programmer may choose to save the modification as an inline rule for that map, rather than updating the free rule on PREDICT. If the user updates the free rule on PREDICT, all other maps using that free rule should be re-stowed in order for the changes to be in effect.

Automatic rules cannot be modified in the map editor, but they can be copied to a new rule name which would then become either free or inline and can be modified.

Deleting a Rule From a Field

An inline processing rule can be deleted from a field by using option "D", DISPLAY FIELD DEFINITIONS, for the field from which the rule is to be deleted. The code for the rule should be deleted or blanked out. The map must then be re-stowed.

A free processing rule is deleted in NATURAL 2 in the same manner as an inline rule, except for entering UNLINK or "U" on the command line after deleting the code. The map must then be re-stowed.

An automatic processing rule is deleted by editing the field in PREDICT and removing the verification name. The DDM should be re-generated and the maps re-stowed.

Considerations

1. All processing rules in maps are evaluated every time ENTER is pressed. This includes maps that are input-only maps.
2. Processing rules are executed in ascending order by rank and then by screen position of the field. PF-key processing rules have first screen position.
3. If more than one automatic rule is connected to a field, the rules are executed in the same order as they were assigned in PREDICT in the verification list for the field. All automatic rules for a field have the same rank.
4. Automatic rules (rules included in DDMs) are not available to NATURAL programs. Automatic rules are only available to NATURAL maps.
5. Processing rules only have access to fields that are defined in the map using the rule and to any field that is defined locally to the rule.
6. The INCLUDE statement may not be used within a processing rule.
7. The source for the map shows the name of the rule(s) included, but not the NATURAL code for the rule. The object code for the rule is included in the object code for the map.
8. A field on a map may have up to 100 rules and up to 50 of these can be automatic rules.
9. When a rule is used for a field in a map, NATURAL 2 will substitute the name of the field for the ampersand (&). This allows the same rule to be used by many different fields.

Additional information on processing rules can be found in the NATURAL 2 Reference Manual and the PREDICT 2 Reference Manual. If you have questions regarding the use of processing rules in NATURAL or PREDICT, please contact Software AG's Customer Support 1-800-525-7859.

President's Message

continued from page 2

These positions also provide an avenue for more people to become involved in the Users' Group, especially new users who may have experience in some of the new products.

As has always been the case, the key word is "involvement." At the Nashville Conference, I talked with more people than I can remember, and the overwhelming impression I got was of enthusiasm (approaching sensory overload) for the product and direction announcements. Enthusiasm is easy to come by at a conference. The challenge is to maintain it when you get home. Becoming actively involved with SAGGROUP gives you the opportunity to do just that. As you read the rest of this issue of Connections, you will see articles from the various Functional Area Representatives asking for volunteers for both user panels and Technical Advisory positions. I urge you to take advantage of these opportunities. Experience isn't required; just enthusiasm. "Interesting times" deserve interested people.

Frequently Encountered INPL Errors in the NATURAL Applications World

Lisa Theard
Customer Service Division

In order to assist in the INPL process, this list of common and not-so-common errors encountered during INPL processing has been compiled. The errors are accompanied by possible solutions to and/or causes of the problems.

This list of errors applies to the NATURAL application or add-on products only. It is also geared toward installs under NATURAL 2, although some sections contain problems that address the INPL process under both NATURAL Versions 1 and 2. NATURAL must first be installed before the INPL for any of the NATURAL application products can be performed.

The appropriate error messages and codes manuals should also be used to assist in solving these problems.

Logon Format

Some examples of the INPL syntax structure have been included in this document. If NATURAL SECURITY is installed, the logon will have to be changed to a NATURAL SECURITY logon. Refer to the "NATURAL SECURITY in Batch Mode" chapter in the NATURAL SECURITY Manual.

NATDEMO

Many times errors that occur during the INPL process are directly related to problems with the user's batch NATURAL. Try using NATDEMO from the NATURAL load library to run the INPL. If the error does not occur when using NATDEMO, then compare the NATPARM and link edit of the batch nucleus to Software AG's. Relink the batch NATURAL nucleus, if necessary.

Dynamic Parameter Errors

Incorrect specification of the parameter stream can cause the parameters to be ignored, which can cause other errors, and generate any of the NAT7000 dynamic parameter errors.

Parameter statements and streams should look similar to the following:

1. OS

   a. NATURAL 2

```
//STEP1 EXEC
   PGM=NATDEMO,
   REGION=1024K,
//    PARM=('STACK=INPL,
   IM=D,DBID=2,FNR=4,
   FSEC=(2,5),
   FUSER=(2,6),',
//    'FDIC=(2,7),MADIO=0,
   MAXCL=0,MT=0,
   ADASVC=21,INTENS=1')
```

   NOTE: When continuing dynamic parameters, the last parameter on the line to be continued must be followed by a comma.

   b. NATURAL Version 1 (NATMAINT)

```
//SYSIN DD *
FNR=4
DBID=2
IM=D
INPL
MT=0
CATLG=NATFILE,
NATFILES
NATURAL=NATDEMO
```

2. DOS

   a. NATURAL 2

```
// EXEC
NATDEMO,SIZE=550K
OBJIN=R
FNR=4,DBID=2
FSEC=(2,5)
FUSER=(2,6)
FDIC=(2,7)
MADIO=0
IM=D
STACK=INPL
MAXCL=0
MT=0
ADASVC=21
INTENS=1
```

   b. NATURAL Version 1 (NATMAINT)

```
// EXEC
NATMAINT,SIZE=768k
FNR=4
DBID=2
IM=D
INPL
MT=0
CATLG=NATFILE,
NATFILES
NATURAL=NATDEMO
```

3. CMS

   "EXECOS" MODULE "BATCH STACK=(INPL) DBID=2 FNR=4 . . . . .(etc.)"

   NOTE: Disregarding the NAT7000 error codes can cause unpredictable or invalid results.

NATURAL CONSTRUCT Version 2

Looping INPL Job

Try removing STACK=INPL from the parameter stream, initiate a logon to library SYSINPL and issue the INPL command in the NATURAL input cards.

Example:

```
LOGON SYSINPL
INPL
B
FIN
```

NATURAL Elite Version 1

This section applies to NATURAL Elite Version 1.3.2 and subsequent releases.

NAT1106—Data String For "Input" Field Too Long

Specifying the INPL in both the parameter stream and the NATURAL input cards. The INPL command should only appear once in the job. Input cards should be:

```
LOGON SYSINPL
INPL
1
FIN
```

or, if using STACK=INPL:

```
1
FIN
```

NATURAL SECURITY Version 2

Refer to the NATURAL SECURITY Installation Notes for the parameters required to run the INPL.

CPU Time Out Before INPL Data Set Is Read From Tape

Try removing STACK=INPL from the parameter stream, initiate a logon to library SYSINPL and issue the INPL command in the NATURAL input cards.

Example:

```
LOGON SYSINPL
INPL
B
FIN
```

SOFTWARE AG

INFORMATION REQUEST CARD

CO-1288

YES, Please send me additional information on:

- ☐ NATURAL 2 OPTIMIZER
- ☐ NATURAL CONSTRUCT
- ☐ 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
11190 Sunrise Valley Drive
Reston, VA 22091
Attn: Tim Fields

NOTE: If error code NAT0082 results, then insert a "." in the input cards. Refer to error code NAT0082, in this section.

Looping INPL Job

Run an ADAREP and compare the listed FSEC FDT with NATFDT on the current NATURAL install tape. If the FSEC FDT does not match, load a new FSEC file using NATFDT from the current NATURAL install tape.

NAT0082—Invalid Command, Or Object Does Not Exist In Library

Specify STACK=INPL in the parameter stream. The INPL command should be passed in the parameter stream, not in the NATURAL input cards.

If no room for STACK=INPL in the parameter stream, then specify the input card as follows:

```
LOGON SYSINPL
INPL
B
.
FIN
```

NAT0818—User ID Is Missing or This Terminal Is Not Defined

NOTE: May be accompanied by NAT1520 "Device Or Logical Printer Not Available".

Specify CMWKF01.

NAT0819—Logon Unsuccessful (Batch Mode Execution)

NOTE: May be accompanied by NAT9990 "NATURAL Initialization Failed."

1. Specify STACK=INPL and AUTO=OFF in the parameter stream
2. Check for parameter errors in specifying the system files. For instance, when specifying the FSEC file, it should be FSEC=(DBID,FILE-NUMBER) not FSEC=FILE-NUMBER.

   Example:

   If the FSEC file is file number 5 in data base 2:

   Correct: FSEC=(2,5)
   Incorrect: FSEC=5 or FSEC=(5)

   The above system file specification applies to FSEC, FDIC, FUSER and FNAT. FNR is still specified as FNR=FILE-NUMBER (FNR=4).

NAT0954—Abnormal Termination During Program Execution

NOTE: May also occur as ERROR 0954 and occurs in module INPL.

Specify CMWKF01.

NAT1009—Program Interrupted After Too Many ADABAS Calls

1. Set MADIO to zero in the parameter stream.
2. Check the parameter stream to ensure that the parameters are being specified and interpreted properly.

NAT1029—Interrupt After Too Many NATURAL Program Calls

1. Set MAXCL to zero in the parameter stream.
2. Check the parameter stream to ensure that the parameters are being specified and interpreted properly.

NAT1106—Data String For "Input" Field Too Long

NOTE: May also occur as ERROR 1106.

1. The INPL command should only appear once in the job. Make sure that the INPL command does not appear in the NATURAL input cards. For instance, using STACK=INPL in the parameter stream, the input cards should be as follows:

   ```
   B
   FIN
   ```

2. Refer to error code NAT0819, Item 2, in this section, to determine if it may be applicable.

NAT1500—Open For "Work" File Failed

NOTE: May also occur as ERROR 1500.

1. Specify STACK=INPL in the parameter stream.
2. Incorrect specification of work file or not using work file (CMWKF01).

NAT1505—Invalid Data For Numeric Input Field

NOTE: May also occur as ERROR 1505.

1. Incorrect input. Using incorrect function code for the INPL. Passing STACK=INPL in the parameter stream, the input cards should be:

   ```
   B
   FIN
   ```

2. Using incorrect NATURAL.
   a. OS/DOS: Using incorrect NATDEMO or batch NATURAL. For instance, using a previous or back-leveled NATDEMO or batch NATURAL to run an INPL. Use NATDEMO or batch NATURAL for the SM being installed.
   b. CMS: Using incorrect gxec. For instance, using a previous or back-leveled EXEC to run an INPL. Use EXEC for the SM being installed.

NAT1520—Device Or Logical Printer Not Available

Refer to error code NAT0818, in this section, to determine if it may be applicable.

NAT3017—Invalid File Number

NOTE: Error occurs in module INTSEC and may be accompanied by NAT9988 "Invalid System File Information" and/or NAT9987 "NATURAL Session Terminated Abnormally."

Specifying a file number for the FSEC file that was used by the ADABAS SYSFILES parameter to establish the CHECKPOINT or ADABAS Security file.

The file number for each ADABAS system file is determined by the file numbers which are specified in the SYSFILES parameter when the data base is established (see "Establishing A Data Base" in the ADABAS Installation Manual). Even if the file is deleted and reloaded as the FSEC file, ADABAS will still treat it as whatever file the SYSFILES parameter established it as. Create a new FSEC file.

NAT3061—An Error Was Detected In The Search Buffer

NOTE: Error occurs in module INTSEC.

1. The system file does not match the field definition table on the NATURAL 2 install tape. Run an ADAREP and compare the listed FSEC FDT with NATFDT on the tape. If the FSEC FDT does not match, load a new Version 2 FSEC file using NATFDT from the NATURAL 2 tape.
2. Using a NATURAL Version 1 system file: Run an ADAREP and compare the listed FSEC FDT with NATFDT on the NATURAL 2 install tape. If the FSEC FDT does not match, load a new Version 2 FSEC file using NATFDT from the NATURAL 2 tape.
3. System files are all combined on FNAT and the format is from the dictionary FDT (DICFDT or PRD222.SYSF). Perform FILEMOD on the system file and add or modify the following fields:

   AF should have a length of 64
   BX should have a length of 35
   LX should be added:
   LX=LO(1,33),LP(1,33)

NAT9987—NATURAL Session Terminated Abnormally

Refer to error code NAT3017, in this section, to determine if it may be applicable.

NAT9988—Invalid System File Information

1. Trying to INPL the NATURAL SECURITY data sets without having first performed the INPL for base NATURAL. NATURAL must be installed before the INPL for any of the NATURAL Application products can be performed.
2. Refer to error code NAT3017, in this section, to determine if it may be applicable.

NAT9990—NATURAL Initialization Failed

Refer to error code NAT0819, in this section, to determine if it may be applicable.

NAT9992—Storage Required Exceeds 'MSIZE' Value

Increase the region size.

PREDICT Version 2

This section applies to PREDICT Version 2.2.2 and subsequent releases.

Tried To Read Incompatible Format Data Set, INPL Stopped

Incorrect input. Using incorrect function code for the INPL. Input cards should be:

PREDICT V2.2	PREDICT V2.3
LOGON SYSINPL	LOGON SYSINPL
INPL	INPL
1	B
FIN	FIN

or, if using STACK=INPL:

1	B
FIN	FIN

ABENDSOC4

NOTE: Error is specific to install under NATURAL Version 1.

1. The NATURAL load module is not specified. Specify the load module in the input cards.

Example:

NATURAL=NATDEMO

2. Specifying a load module for DDMs. CATLG should be specified.

ABENDU0007

Incorrect ADARUN card.

NAT0082—Invalid Command, Or Object Does Not Exist In Library

NOTE: Error can occur on the INPL or the error text load step and may be accompanied by NAT9987 - "NATURAL Session Terminated Abnormally."

1. On the INPL step:
   a. Find and copy module LODSRCN to the SYSTEM library on the FNAT file. Rerun the INPL job.
2. On the error text load step:
   a. NATURAL Version 1.2.7: Find and copy modules ERRORTXT, ERROREXT and HELPTBLE to the SYSTEM library on the FNAT file. Rerun the error text load job.
   b. NATURAL 2: Find and copy modules ERRORTXT and ERROREXT to the SYSTEM library on the FNAT file. Rerun the error text load job.
3. NATURAL 2 only:
   a. Using NATDEMO from the NATURAL install tape and not specifying the ADASVC parameter. Specify ADASVC.
   b. Using linked up batch NATURAL and the ADASVC parameter is not contained in the NATPARM or parameter cards. Specify ADASVC.

NAT0818—User ID Is Missing Or This Terminal Is Not Defined

NOTE: May be accompanied by NAT9990 'NATURAL Initialization Failed.

Specify FIN at the end of the input cards. Refer to error code NAT0954, in this section, for examples of correct input data.

NAT0953—Time Limit Exceeded

1. Set MT to zero in the parameter stream.
2. Check the parameter stream to ensure that the parameters are being specified and interpreted properly.

NAT0954—Abnormal Termination During Program Compilation

NOTE: Error occurs in module LOGONM1 and may be accompanied by NAT6768 in module LODNM1. NAT6768 is currently undocumented.

Specify FIN at the end of the input cards:

Example:

PREDICT V2.2	PREDICT V2.3
LOGON SYSINPL	LOGON SYSINPL
INPL	INPL
1	B
FIN	FIN

or, if using STACK=INPL

1	B
FIN	FIN

NAT1106—Data String For "Input" Field Too Long

NOTE: May also occur as ERROR 1106.

1. Incorrect input data. Refer to error code NAT0954, in this section, for examples of correct input data.
2. Refer to error code NAT7300, in this section, to determine if it may be applicable.

NAT6768

NOTE: Error occurs in module LODNM1 and is undocumented at this time.

Refer to error code NAT0954, in this section, to determine if it may be applicable.

NAT7300—Dynamic Parm Error: Parameter Not Found

NOTE: Error occurs in module INPL and may be accompanied by NAT1106/ERROR 1106 "Data String for 'Input' Field Too Long."

Using incorrect INPL utility.

Example:

Using NATMAINT to perform the PREDICT INPL under NATURAL 2.

NATMAINT does not exist and is not valid for NATURAL 2. Use OJINPL (OS), DJINPL/XJINPL (DOS) or INPL EXEC (CMS) from NATURAL 2.

NAT9987—NATURAL Session Terminated Abnormally

NOTE: Error is specific to install under NATURAL 2 and the job will usually show a condition code of 8.

1. Check INPL output carefully for error codes and messages. If no error codes or messages are found in the output, this error message can be ignored as long as the INPL output shows that a total number of modules and sources were loaded.
2. Refer to error code NAT0082 in this section to determine if it may be applicable.

NAT9990—NATURAL Initialization Field

Refer to error code NAT0818 in this section to determine if it may be applicable.

NAT9992—Storage Required Exceeds 'MSIZE' Value

1. OS/DOS: Using incorrect NATDEMO or batch NATURAL. For instance, using a previous or back-leveled NATDEMO or batch NATURAL to run an INPL. Use NATDEMO or batch NATURAL for the NATURAL SM under which the product is being installed.
2. CMS: Using incorrect EXEC. For instance, using a previous or back-leveled EXEC to run an INPL. Use EXEC for the NATURAL SM under which the product is being installed.

REVIEW Version 2

NAT0082—Invalid Command, Or Object Does Not Exist In Library

Find and copy module LODSCRN to the SYSTEM library on the FNAT file. Rerun the INPL job.

SUPER NATURAL Version 2

This section applies to SUPER NATURAL Version 2.1.2 and subsequent releases.

Inactive INPL Job

Incorrect specification of the PROGRAM parameter on the ADARUN card. PROGRAM should be USER.

Example: ADARUN PR=USER, MODE=MULTI,SVC=21,DA=2

ABENDSOC4

If work files which are not being used are included in the JCL/JCS, they should be dummied out instead of commented out.

NAT0082—Invalid Command, Or Object Does Not Exist In Library

Refer to error code NAT0082 in the REVIEW Version 2 section for possible explanation of the problem.

NAT1009—Program Interrupted After Too Many ADABAS Calls

NOTE: May also occur as ERROR 1009.

Refer to error code NAT1009 in the NATURAL SECURITY Version 2 section for possible explanations of the problem.

NAT1106—Data String For "Input" Field Too Long

Specifying INPL in both the parameter stream and the input cards. Specify either STACK=INPL in the parameter stream or INPL in the input cards, not both.

NAT1311—Index In "Move Indexed" Statement Is Zero Or Negative

Incorrect work file allocation. Work files should be:

CMWKF01—Input Data Set
CMWKF02—Error Text Data Set

NAT1500—Open for "Work" File Failed

Incorrect specification of work file or not using work file (CMWKF01).

NAT1502—Permanent Physical I/O Error

Check for hardware errors/problems. Try running the INPL job on different hardware, if possible (i.e., load tape on a different tape drive).

NAT3061—An Error Was Detected In The Search Buffer

Refer to error code NAT3061 in the NATURAL SECURITY Version 2 section for possible explanations of the problem.

NAT9992—Storage Required Exceeds 'MSIZE' Value

Refer to error code NAT1505 in the NATURAL SECURITY Version 2 section for possible explanations of the problem.

NAT9997—Not Enough Memory To Start NATURAL

Increase the region size.

USER'S GROUP NEWS

September SAGGROUP Executive Committee Meeting Overview

Steven L. Baker
SAGGROUP Secretary

The SAGGROUP Executive Committee meeting was held at the Opryland Hotel in Nashville, Tennessee in conjunction with the International Users' Conference.

Annual International Users' Conference

The 1988 Conference was attended by over 2,000 users from regions all over the world. The Executive Track has mushroomed to over 250 people which is up over 40% from last year. New features of this year's conference included the Sunday start date, more Reach Out and CON-NECT terminals, more demonstrations, and the expanded Executive Track program.

Product demonstrations were greatly expanded to include 26 classroom type booths demonstrating over 50 different products. New features this year included a WANG VS machine running SAG software.

Next year's Conference will be held October 22-26, 1989 at the Anaheim Hilton in Anaheim, California.

New Product Representation

With the large number of products being released by Software AG, the Executive Committee felt that the existing Product Representation of ADABAS, NATURAL, COMPLETE, and PREDICT was outdated. The committee replaced that structure with four new functional areas: ADABAS, ADMINISTRATION, END-USER COMPUTING, and COMMUNICATIONS. All of Software AG's products are now represented within these groups and each group has a Functional Area Representative (FAR) assigned (see the breakdown in this issue of Connections.)

New Executive Committee Members

Elections held at the Conference resulted in the current Executive Committee representation:

- Bill Wagner, President
- Colette Farabaugh, Vice President (appointed)
- Steve Baker, Secretary
- Kelly Jones, Change/Enhancement Coordinator
- Martin Henderson, Technical Support Evaluation
- Mary Ellen Woods, Nominations/Elections
- Patti Piccola, Administration FAR
- Jim Wisdom, Application Development FAR
- Ellen Birch, Communications FAR
- Bob Becker, Data Base FAR
- Laura Jacobs, End User Computing FAR

Service/Connection

The exciting news was an update on the status of Service/Connection which provides the capability for each Software AG user to communicate with a central information depository. Gordon Perrins is the Service/Connection Coordinator and there is now worldwide interest in the product. Target date for release of phase I is February, 1989.

NATURAL CONNECTION will be required on both the PC and the mainframe, however, there will be subcomponents of this product

continued on page 17

New Change/Enhancement System Incorporated

Kelly Jones
Change/Enhancement Coordinator

During the 18th International Software AG Users' Conference, the many benefits to Software AG and users of Software AG products of maintaining open dialogue regarding the users' software product needs and expectations were stated several times. As one part of this dialogue, the Change/Enhancement System has been identified as the official means by which a user can communicate needs for increased functionality, and/or desires for future direction of existing products.

In order to increase the value of the Change/Enhancement System to SAGGROUP and Software AG, the Executive Committee and Peter Schnell, President, Software AG, have agreed to enhancements to the Change/Enhancement System itself. The major thrust of changes agreed to is to shorten the Change/Enhancement cycle from a period of 6 to 8 months to 1 to 3 months. Implementation of these enhancements is expected to benefit Software AG by bringing to their attention means by which they can increase the value of their products in a dynamic and demanding market. The users of Software AG products are expected to benefit by realizing the increased value of these products in less time than may have been seen in the past.

Effective immediately, the procedure for processing Change/Enhancements will consist of the following steps:

1. A user will prepare a Change/Enhancement Request form and forward the request to the SAGGROUP Change/Enhancement Coordinator. A Change/Enhancement Request form may be found near the back of the Connections magazine.
2. The Change/Enhancement Coordinator will enter the request into the Change/Enhancement System.
3. On a monthly basis, the Change/Enhancement Coordinator will send to appropriate SAGGROUP Functional Area Representative:

continued on page 16

A Stitch in Time Saves Nine

Dave Williamson
Utah State Office of Education

If you have installed NATURAL 2 under CICS you have probably looked for some kind of information on how to set the thread size. You need to have threads large enough to handle the requirements of your site but in CICS memory can be precious and you definitely don't want to waste it. Since NATURAL 2 is still relatively new this type of operational information has not made it into the documentation. Software AG suggests that you start with 180k threads. Is this too much or too little for your shop? I believe that the following information may be helpful in determining what a reasonable thread size should be for your shop.

First we need to define a couple of new buffers used in NATURAL 2 and explain how their size is determined. The Screen Buffer holds the image of what is on the physical screen and is determined by the physical screen size. Its use includes the optimization of data streams on communications lines and windowing. To determine the size of this buffer use the following formula.

Screen Buffer = line size (of physical device) × page size (of physical device)

The Page Buffer holds the image of the data on the logical screen or page, it is determined by the LS and PS parameters in the NATPARM module or dynamic overrides. Its use includes windowing around logical screens that are larger than the physical screens. To determine the size of this buffer use the following formula.

Page Buffer = LS (from NATPARM) × PS (from NATPARM)

Both the Page Buffer and the Screen Buffer have an associated Attribute Buffer to hold the attribute information for the fields in the respective buffer. The attribute buffers size is influenced inversely by the AVERIO parameter in the NATPARM module or dynamic overrides. To determine the size of these buffers use the following formulas.

Screen Attribute Buffer = Screen Buffer × 10 / AVERIO

Page Attribute Buffer = Page Buffer × 10 / AVERIO

To determine the approximate required thread size simply add all of the NATURAL buffer sizes together and throw in an extra 5k for miscellaneous use. These buffers are the Screen Buffer, Page Buffer, Screen Attribute Buffer, Page Attribute Buffer, ESIZE, USIZE, FSIZE, DSIZE, CSIZE. Figure 1 is an example of this calculation.

USIZE	32k	user buffer
ESIZE	28k	user buffer extension
FSIZE	15k	NATURAL DDM/symbol tables
DSIZE	1k	debug buffer
CSIZE	0k	CON-NECT buffer area
	5k	miscellaneous
Screen Buffer	2k	(24 × 80 = 1920)
Page Buffer	8k	(60 × 132 = 7920)
Screen Attribute Buffer	4k	(1920 × 10 / 5 = 3840)
Page Attribute Buffer	16k	(7920 × 10 / 5 = 15840)
	111k	required thread size

Figure 1

Just to make life interesting we should look at a couple of things that will affect what thread size you may really need. If you allow the use of dynamic parameters at natural start up, you will need to take into account the effects of dynamic parameters by either adding a cushion of a several k or use the maximum buffer settings you will allow in the calculations. The later could have several permutations. The other thing is that some of the buffers are being used slightly different than in the past. The USIZE for example is not used for the object code when running NATURAL 2 programs, as they are executed out of the buffer pool. A good description of how the USIZE is used appears in the NATURAL 2 Administrators Manual. The point is that you may need to reevaluate the settings of these buffers.

What is really needed is a tool that can be used to monitor the thread usage just as SYSTAT is used to monitor the buffer pool. Until we have that kind of tool this information may help save some memory that would otherwise be wasted.

New Executive Committee Appointees

Bill Wagner
The University of Texas at Austin

Because of the Constitutional changes and the elections, there are now several new faces on the Executive Committee. As was announced at the Conference, Laura Jacobs of the Rochester Institute of Technology has agreed to be the first End User Computing Functional Area Representative. Colette Farabaugh of Dole Processed Foods has accepted appointment to fill the vacant position of Vice President, and Mary Ellen Woods of Inland Steel will take over Colette's former duties as Nominations and Elections Chairperson.

Addresses and phone numbers for these officers can be found at the back of this newsletter. Like the rest of the Executive Committee, their effectiveness in representing you is directly related to the input they receive from you. Suggestions, comments, questions, and criticisms are always welcome. And of course, the communication lines flow both ways, as many of you will find out when you hear from Mary Ellen in the months ahead.