Central Support
ON–LINE

Published for System Software Users

Volume 2 *Number 4* *4th Quarter 1989*

LOGON:

In This Issue

The results of the **Reader's Questionnaire** have been tabulated and analyzed. We report the findings in the following article. And they show that we've been doing a fairly good job of providing you with technical information for operating your McDonnell Douglas computer system. To those of you who responded, thank you.

The **STAR POWER** topic is about using **Synonym Verb Definitions** to "zip" around your system by using abbreviations for commonly used TCL-verbs.

Will Edwards continues the **Performance Series** in the **ROS** column. This fourth installment takes a slightly broader and more technical look at system bottlenecks. Gary Moote shines some light on those mysterious abort messages.

A new column called PGM debuts in this issue. We will spotlight suggestions and solutions to programming issues in this column. Teri Garrett describes several helpful DATA/BASIC features to start unraveling those ancient--but still used--programs.

In **THINK TANK**, Garrett Hildebrand conducts an excursion through several Software Engineering (SWE) topics.

Niki Jhaveri's third segment in his **SNA Series** discusses the primary SNA building blocks. The benefits of using our Data Communications Group in designing and installing your data network are presented in **Network Services**.

After many requests, we are pleased to dedicate **FEEDBACK** as a **Holiday Greetings** album. During a support call, haven't you ever wondered what the analyst "on the other end of the telephone" looks like. Well, now you can see for yourself on page 15.

- L. W. Abel

Results of *ON-LINE* Readers' Questionnaire

Since initiating the first issue of *ON-LINE*, we have concentrated on publishing the best Technical Support newsletter for our customers and our Independent Sales Organizations (ISOs). Many times while talking with you during a support call, the analyst solicited user's input about *ON-LINE*. Those occasional comments have helped guide us in developing columns and specific articles that best meet your interests and requirements. Finally, we decided to ask you directly with a reader's survey.

We enclosed a Reader's Questionnaire in our second quarter 1989 *ON-LINE* issue. Of the approximately 3,200 copies that were distributed, we received 61 responses. Of these, eight were from either our employees or requests for mailing address changes. The remaining 53 returned surveys were from End-Users (51) and ISOs (2). Their replies represent nearly a 1.7 percent response to our survey.

The following material highlights the responses to specific questions and interest areas. Your response to our survey is appreciated.

High & Low Categories

We asked you to rate *ON-LINE* according to several criteria. The ratings ranged from LOW (1) to HIGH (5). The rating results for these categories follow:

• Appearance	4.29
• Technical Information	4.22
• Topic Relevance	4.16
• Readability	4.14
• Article Length	4.00
• Frequency	3.14

These results were consistent with our expectations. The lower value for the Frequency category reflects many of your comments about wanting *ON-LINE* to be published more often than just quarterly. And some of you are interested in longer articles. In fact, the two most frequent responses to Question 5: "What don't you like about *ON-LINE*?" were that the newsletter "could be issued more frequently" and "not long enough."

We are actively considering revamping our publishing schedule to provide either a more frequent distribution or a larger issue, or perhaps, both of these.

The Most & Least Useful Columns

Since our emphasis is on providing you with the best and most recent technical support information about your McDonnell Douglas computer systems, your choices of the most useful columns reflected our expectations again. You chose ROS, STAR POWER, and LOGON: as being most useful and informative.

Both ROS and STAR POWER are typically targeted for the MIS or DP staff whose responsibilities include maintaining daily systems operations and providing enhanced systems performance. Besides presenting the Current OS Releases and Patches, the ROS articles--especially the Performance Series --provide detailed technical information about the REALITY Operating System.

Complementing the ROS articles are those written for STAR POWER. These are the "hands-on" articles offering tips and suggestions about special features or problem areas that you can use immediately. Most topics are suggested (and usually written) by the Central Support analyst who has dealt with a specific question or problem during the preceding quarter. For instance, our first several issues reported how to handle GFEs, SHUTDOWN, and Emergency Flushes. Since then, the number of calls about these problems has continued to decline.

The LOGON: column presents issues related to the operation of Central Support. Articles about changes in policies and procedures, as well as announcements of new services, can be found in this column. This column keeps our customers and ISOs informed about current technical support activities.

The other columns which were ranked (by number of responses) were: PROC DOC, MARKETING, FEEDBACK, COMMS, APPS, 14-100, and 7000. By contrast to those columns ranked most useful, those of least interest included: 7000, 14-100, APPS, COMMS, MARKETING, and FEEDBACK. Analysis of these responses reveal that you generally read only those articles that directly affect your system's operations or relate to a specific product.

What You Like BEST about *ON-LINE*

We received 49 responses to this question. These were summarized into the following categories:

- Good and current technical information
- System Performance issues and suggestions
- Examples of How-to, Fixes, and Workarounds
- Good Customer/Company exchange
- Training for operations staff
- Short, technical articles
- Specific to McDonnell Douglas systems

Your primary interest was in obtaining information to make your computer operations faster and simpler. This, too, is one of our primary objectives.

The Performance Series (ROS) received many positive responses. We are continuing this Series with a little more technical emphasis.

What You DON'T Like about *ON-LINE*

Of the 23 responses to this question, nearly 60 percent replied that *ON-LINE* should be more frequent or longer (more pages). The next category simply reported: "Nothing!" Other concerns, which we are reviewing, include wanting new columns (see the new PGM column in this issue), presenting a better balance, and handling mailing and distribution problems.

Other Comments, Etc.

The majority of respondents operate either Series 6000 (2.3 OS) or Series 9000 (5.3 OS) systems. There were only four Series 18 (6.0 OS) and only two Series 4700 (4.3 OS). There were no Series 7000 respondents.

Over half of you pass your copy of *ON-LINE* on to other readers in your company. Nearly a quarter of the replies requested that we provide additional copies for your staff.

The most frequent response to the "Other comments" question related to mailing and distribution requests. Rose Myszka manages distribution and you should contact her at (714) 566-4802 about any mailing changes or problems.

Other comments centered around wanting more information and examples, adding columns, being easier (less technical) to read, providing problem-solving techniques, and a Question and Answer column (we will have a new Q&A column in the next issue).

Summary

The *ON-LINE* staff sincerely appreciates the time you spent in responding to our Reader's Questionnaire. Your efforts will help us to provide you with an ever improving quality technical newsletter. Please write us any time with your suggestions and comments!

- L.W. Abel, Editor

REALISM Support Package Announced

REALISM, REALITY Integrated System Management, is a new user interface for the REALITY operating system providing sophisticated and practical user functions. The **REALISM SHELL** gives you effective security against unauthorized access at both a software and hardware level. It integrates security as well as menu-driven access, directory assistance, on-line help, and user profiles into the ROS environment. REALISM includes office automation facilities including electronic mail, a personal diary and notepad, address book, calculator and front desk system. A Toolkit allows ENGLISH sentences to be built with prompts, quick data entry and a full screen editor.

REALISM DEVELOPER is another module that can be purchased in addition to the SHELL. The DEVELOPER is a Data/Basic applications program and report generator. On-line help and documentation are included, as well as documentation facilities for programs written by DEVELOPER. A screen painter, input validation, pre and post-input processing, audit trails, transaction logging and error messages are just some of DEVELOPER's many features.

McDonnell Douglas Field Service Company has designed the support package to be flexible so that it can meet the needs of all REALISM users. Support can be purchased as a contract for either a specific number or an unlimited number of calls.

A **12 Call-Pak** and **24 Call-Pak** contract is available, as well as a contract with unlimited calls. The unlimited call contract is an end user-only option. Whether you have purchased the REALISM SHELL or both the SHELL and DEVELOPER, your support options will be the same.

REALISM support will be provided from 8:00 a.m. to 5:00 p.m. in your local time zone through our centralized software support organization in Santa Ana, California. In addition to telephone support, users will receive *ON-LINE*, which will have articles on REALISM, and access to the REALISM module on our new Bulletin Board System. Patches will also be made available on a timely basis.

New releases will be announced in this newsletter as well as by letter to our REALISM support contract customers. New releases and documentation may be purchased separately. Installation services can be obtained from your Independent Sales Organization (ISO) or contracted for separately from McDonnell Douglas Field Service Company. A data sheet defining the REALISM support options more specifically is available through your ISO or McDonnell Douglas representative.

We have designed our new REALISM service offerings with your business needs in mind. We hope you will find them both cost-effective and versatile.

STAR POWER

This column provides information you can use immediately to improve the utilization of your system or to avoid potential problems. Usually, the topics discussed are derived from customers' questions or problems which have been frequently reported to Central Support during the past quarter.

Using Synonym Verb Definitions

Synonym Verb Definitions (SVDs) can give you very useful shortcuts in your daily work. You can easily define often or routinely used TCL-verbs* with fewer keystrokes by replacing "long" TCL-verbs like DELETE-CATALOG or SAVE-LIST with abbreviations, such as DC or SL.

You can even do the same with a whole family of TCL-verbs. SPOOLER Synonyms, for example, might be: SJ = SP-JOBS, SC = SP-COPY, SD = SP-DELETE, SPL = SP-LOOK, SST = SP-STATUS, etc.

This feature is available only on the 2.3, 4.3, 5.3 and 6.0 Operating System (OS) releases.

On earlier releases, many programmers and system administrators simply copied the Master Dictionary (MD) entry of a particular verb to another shorter-named MD entry. Thus, each entry "points" to the same System Mode. (A Mode is the Assembly language program used by the Operating System.)

For example, the TCL-verb to save a list of selected items is called SAVE-LIST. The MD entry for SAVE-LIST is the following:

```
     CT MD SAVE-LIST
001  PL
002  2064
```

In order for the system to execute a TCL verb, it must determine from the account's MD where to look in absolute memory (ABS) for the specific modal entry point. This location in ABS must match the location indicated by the MD. If not, a "mismatch" occurs, and the TCL-verb will not work correctly.

Usually, whenever a mismatch occurs, the process will abort. Often, your Central Support Analyst will ask you to perform a VERIFY-SYSTEM or similar process to verify that the System Modes do not contain mismatches. It is critical that the Mode verifies and that the MD points to that Mode.

The current releases (see "Current OS Release and Patches" article on page 5) contain significant enhancements over previous releases. In some cases, the modal addresses of TCL-verbs have been replaced with new ones. Using an "old" address can

cause an abort (e.g., "Illegal Opcode"), something unexpected to happen, or even nothing to happen.

It is crucial that the MDs for the TCL-verbs are correct. This is insured by using the authorized tape and following the Implementation Advisory (IA) whenever upgrading or patching your system.

Why Have Synonym Verbs?

Since we recognize that TCL-verbs may change from release to release, we created the Synonym Verb Definition feature. This feature lets you create your own "shortcut" verbs which are synonymous, yet easier to remember and use.

And we have designed it to allow the TCL-verb definition to change without your having to change the Synonym. For example, if the TCL-verb SAVE-LIST's modal definition is redefined on a new release or patch, the Synonym Verb is still "pointing" to the TCL-verb named "SAVE-LIST" in the account's MD file, which now contains "new" modal information.

Creating Synonym Verbs Definitions

You create SVDs by entering the synonym-name in the MD file. For example, a commonly used TCL-verb is SAVE-LIST. To make a Synonym Verb Definition for SAVE-LIST, you first select a name for it that is not already an item in the MD file. In this case, let's use "SL" as our Synonym for SAVE-LIST. We edit the item SL in the MD file. We enter the following definition:

```
    ED MD SL
001 V
002 SAVE-LIST
```

The system recognizes that the example above is a Synonym Verb Definition because the letter "V" is in attribute 001. Attribute 002 contains the name of the TCL-verb that this synonym represents (or "points" to). Either SAVE-LIST or SL can now be used to execute the verb.

After you have created your Synonym Verb library, you can generate a printout of your Synonyms by using the following ENGLISH statement:

```
SORT MD WITH *A1 = "V" *A1 *A2 (P)
```

I recommend that you generate a new listing whenever you change or add new Synonym Verb Definitions. In this way, you'll have at a glance all your Synonym Verbs which will let you "zip" around your system easier and faster.

---

* Note: TCL = Terminal Control Language. This is the primary interface between you and your system. Many analysts refer to TCL as "Tickle" or by saying "...at the colon (:)..."

TCL-verbs are those "verbs" or processes that are accessed or initiated from TCL. Please refer to your *Programmer's Reference Manual* for more details.

Please Note: The TCL-verb definitions used in these examples are from the Series 9000, using release 5.3 Operating System.

• L.W. Abel

MARKETING

New ASIC-Based Computer Systems Announced

McDonnell Douglas Computer Systems Company (MDCSC) has announced new Series 6000 system models that offer significantly enhanced CPU power, reliability and expandability. The models incorporate new processor and memory architecture based on leading **CMOS Application Specific Integrated Circuits (ASIC)** technology designed by McDonnell Douglas and implemented by LSI Logic Corp. Each ASIC chip supports 100,000 logic gates using 1.5 micron technology. In new Series 6000 models, a single ASIC based board replaces up to four CPU and memory boards.

The new technology **increases Series 6000 CPU performance by 70%** and significantly enhances system reliability, expandability and maintainability. System reliability has been achieved by a significant reduction in component count and, in part, by the extensive level of simulation testing involved in the ASIC chip development process.

Introducing the new U.S. Series 6000 models, the company also announced new tape subsystems as well as upgrade kits which allow most currently installed Series 6000 systems to be easily upgraded. Concurrently, the company indicated it has repositioned existing Series 6000 systems to enhance price/performance value.

The following are new U.S. base model configurations:

6404	6604
10.0 MIC CPU	10.0 MIC CPU
2 MB Memory	2 MB Memory
75 MB Disk ESDI	2 MB Cache
120 MB Tape Cart. Drive	140 MB ESDI
8 Ports	16 Ports
Tower Cabinet	Low-Boy Cabinet
Plus REALITY OS 2.3	Plus REALITY OS 2.3

• Stan Goodrich

ROS

Current OS Releases And Patches

The following table contains the most current Operating System (OS) revisions and patch levels for each supported system. McDonnell Douglas Field Service Company has recently assumed responsibility of installing all patch tapes for Independent Sales Organizations (ISOs) and branch customers.

If you do not have the current patch tape installed for your particular Operating System, please contact your Field Engineer (FE) through Central Dispatch to schedule a time for installation.

Series	Release	Patches (PP = Paper Patches)
4700	4.3RevD	PP1-2
6000	2.3RevD	RevC Tape (Includes PP1-175)
	1.1RevD	RevB Tape
9000	5.3RevD	RevC Tape (Includes PP1-121)
	1.3RevC	RevA Tape
18	6.0RevF	RevB Tape (Includes PP1-114)

Please note: New Patch Tapes for Series 6000 (2.3), Series 9000 (5.3), and Series 18 (6.0) are now available.

- Mike Bingman

Performance (Part 4)

For the next several issues this column will discuss the analysis that Central Support would perform when evaluating a system's performance. This article will introduce some of the terms that are used and a brief overview of the methodology.

Performance degradation of a computer system is primarily a result of two factors: **bottlenecks** and **queues**. Therefore, identifying the bottlenecks and queues within the system will lead to pinpointing those processes which are most likely to produce an effective performance enhancement.

Bottlenecks

To put it simply, a bottleneck is any common part of the system that does not have infinite capacity. The part of the system with the lowest capacity will eventually produce a bottleneck.

For example, a common bottleneck is the disc sub-system, where frequently accessed data resides on a few drives. This bottleneck can be overcome, however, by having enough disc drives and controllers over which to spread the data.

Other possible bottlenecks are main memory size, network capacities, or the application process itself where single records are locked for long periods of time.

Queues

A queue is a line of requests for a common system resource. The longer a queue becomes the more chance it has of becoming a bottleneck. If there were no queues, there would be no bottlenecks, and vice versa. There are some system resources that do not allow queueing. An example of this is system locks: after a lock is released, the first process to obtain the lock gets it. Other processes that need the same lock must try to obtain the lock later.

The time required to perform any transaction being queued is a function of two factors: the time spent waiting for the resource and the time spent doing the work after having obtained it. Therefore, the time that the process will wait is:

Delay = Wait time + Process time **[Eq. 1]**

When both the number of other jobs waiting in the queue and the process time are random, the delay is then represented by the following equation:

Delay = Process time /(1-Load) **[Eq. 2]**

This expression will produce the delay time as the load on the resource increases, where Load = 1 for a 100% loaded resource. Some references refer to this equation as the "stretching factor"; i.e., the service time stretches as the load increases.

Relationship between Queues and Bottlenecks

Once the bottlenecks and queues have been identified, a list of strained system resources can be made. The relationship between these bottlenecks and queues can be illustrated by the following example.

If a process requires an average of 35 milliseconds of CPU time and two disc reads in order to complete each input cycle, the CPU needs to be queued seven times and the disc twice for this process.

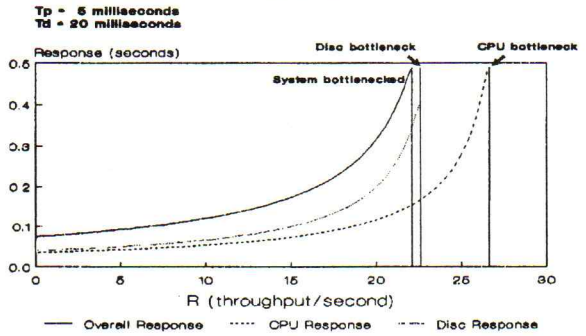
Using Equation 2 above we can obtain certain estimates of response time. Let Tp be the average process time for a given task, Td the average disc access time, and R the system throughput per second. Thus, the load on the CPU is 7RTp and the load on the disc is 2RTd. Substituting in Equation 2 the following equation is derived:

Response time = (7Tp/(1-7RTp)) + (2Td/(1-2RTd) **[Eq. 3]**

This illustrates a process waiting in queue and being serviced seven times by the CPU and twice for the disc.

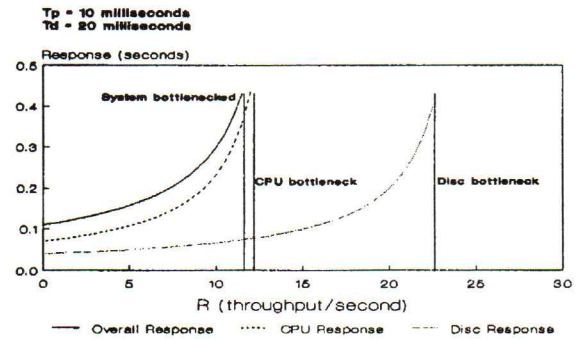
For example, in Figure 1 below, the CPU average process time, Tp, is 5 milliseconds and the disc access time, Td, is 20 milliseconds. Plotted are the disc, CPU, and overall response time curves which show that the disc capacity is reached before that of the CPU. In other words, the disc is the bottleneck.

Figure 1



By contrast, as Figure 2 shows, the average CPU process time has been doubled, to twenty milliseconds, the CPU capacity is reached before that of the disc. In other words, the CPU is the bottleneck.

Figure 2



As the system load starts to bottleneck, the degradation of response time will begin to increase dramatically. In the examples given above, this begins to happen between 50% to 70% of full capacity.

An important step in this kind of analysis is an agreement on what constitutes a slow, fast or acceptable system. For instance, when asked about response time, many users might not expect that "all requests to complete within several seconds" is a reasonable request. However, this is an unreasonable request because the types of jobs being run are so varied. Even always running a certain benchmark or task at a set time would not serve as a good working statement. If the system in the example above had 60 users all submitting a request at the same time, then the system response time would be very high.

Although rare, worst case examples must be taken into account in defining the working statment we should be making, such as "a given task will complete in a given time 99% of the time." In order to be able to fully define the statement, all the parameters in this analysis of response time, workload and configuration are required. It is this analysis that we will be covering in future articles.

Sources:

Date C. J., An Introduction to Database Systems, Addison-Wesley. 1981.

Highleyman W. H., Performance Analysis of Transaction Processing Systems, Prentice-Hall. 1989.

Lavenberg S. S., Computer Performance Modeling Handbook, Academic Press. 1983.

Tanenbaum A. S., Structured Computer Organization, Prentice-Hall. 1984

Handling Aborts

The REALITY Operating System executes in two modes: Monitor or Virtual. When executing in Monitor mode, the system is responsible for Input/Output (I/O), memory management, process scheduling, servicing faults, and various interrupts. When a user process gains control, the system is executing in Virtual mode.

The system can abort in either mode of execution. When in **Monitor mode**, any aborts will be trapped by the Monitor debugger. Monitor aborts will only appear on port 0 and will cause the whole system to halt. An example of a Monitor abort is:

```
MONITOR ERROR 00
E 0073E 08
>
```

You should get assistance from Central Support in troubleshooting these aborts before trying to recover the system.

When in **Virtual mode**, any aborts will be trapped by the software debugger. Any indication of the abort will only be displayed by the aborted process. An example of a Virtual abort is the following:

```
CROSSING FRAME LIMIT; REG = 11.
ABORT @ 242.07F
!
```

Since Spooler and TIPH processes have no connected terminals, no abort message will be output. As a general rule for Spooler and TIPH processes, when the first entry under the return stack heading is "1.xxx" or "21.xxx"; then the process has aborted. This can be determined by displaying WHERE for the process in question.

Error Codes & Messages

It should be noted that Monitor aborts display the error codes, whereas a Virtual abort displays the cause. Below is a table showing the error code and the associated message.

Error Code	Error Message
0	Illegal Opcode
1	Return Stack Empty
2	Return Stack Full
3	Referencing Frame Zero
4	Crossing Frame Limit
5	Forward Link Zero
6	Backward Link Zero
7	Privileged Opcode
8	Referencing Illegal Frame
20	I/O Handling Error
23	LAD error

An Illegal Opcode (error code 0) is caused by one of two possibilities: a HALT instruction or object code error. A HALT instruction is executed in order to prevent damage from occurring or to aid in system debugging. If any patches have been recently applied to the system, double check their accuracy and also verify the system's ABS (use VERIFY-SYSTEM) for possible object code corruption.

Since each process has a maximum of 11 Return Stack entries, should a process attempt to stack a 12th entry, the process will abort with a Return Stack Full (error code 2). A Return Stack Empty (error code 1) abort occurs when a process executes an Assembler RETURN when the stack is empty.

Forward Link Zero (error code 5) and Backward Link Zero (error code 6) aborts occur when a referenced frame is supposed to link forward or backward to another frame but does not. If this occurs when referencing a file, it is indicative of a GFE within the referenced data. Also, system functions such as SP-JOBS or WORKSPACE may abort with these types of aborts if a system table is corrupted.

Crossing Frame Limit (error code 4), Referencing Illegal Frame (error code 8), or Referencing Frame Zero (error code 3) aborts are frequently due to GFEs in the files that are being accessed. Checking the files and correcting the data will fix these problems.

A Privileged Opcode abort (error code 7) is due to a Monitor operation being executed in Virtual mode. If this abort occurs, be sure the system verifies and all patches are installed correctly.

I/O Handling errors (error code 20) occur when either a disc, tape, printer, or terminal I/O operation is invalid. These errors usually indicate system table corruption or a hardware problem.

As a final note, whenever an abort occurs, first check to see if the system verifies by doing VERIFY-SYSTEM from SYSPROG. This is particularly true if several processes abort simultaneously. In the case of a cataloged program, recataloging and rerunning the program may fix the problem.

If you encounter an abort that you want investigated, then please do not "END" out of the abort before contacting Central Support. By "END"ing out of the abort, you will initialize the process, which clears important process information about the cause of the abort.

- Gary Moote

PGM

For this new column, Central Support analyst Teri Garrett discusses several features you can use when "debugging" DATA/BASIC programs.

In the future, this column will provide you with more useful hints and tips about using DATA/BASIC and PROC features and techniques. If you have a question or a favorite feature about programming you would like discussed, please write to us.

Programming Tips

It's finally happened. The program everyone hates...the program that has been changed and added to by nearly every programmer within a 50 mile radius needs to be modified and you have been given the task.

What can you do? You copy the program to the printer hoping it will somehow all make sense if you just stare at it. As the printer spits out page after page after miserable page, you realize it will take hours just to get the variables used in the program into some sort of order.

What should you do? Peruse the want ads? Run screaming into the night? Maybe; but first, try these two steps.

1. **BASIC file-name item-id (X**

   Compiling a program with the "X" option builds the cross-reference table of variables and labels and puts the items in the CSYM file. Of course, you must have CSYM defined in your account.*

2. **BREF file-name item-id**

   The BREF verb will display the cross-reference table listing all variables used in the program along with the program line numbers where each variable is accessed. It will even denote (with "*")

those lines where the variable may change value. Using the "P" option will send this listing to the printer.

Great! Now you've got the variables under control. Right? Right. But...wait just a minute. What END goes with what IF? What NEXT goes with what FOR? Try the following feature to help resolve this dilemma.

3. BLIST file-name item-id (L,U,P

The BLIST verb will list DATA/BASIC source code in a more readable format. The options used above are just a few of the many options outlined in your DATA/BASIC manual.

The "L" option prints a period (.) at each level of indentation. This gives a more logical structure.

The "U" option updates the source code with the logical indenting.

Of course, the "P" option sends the listing to the printer.

Terrific. You are now much more prepared to start making changes to the program. The rest is up to you. Good Luck!

*A more complete description of these commands, including numerous helpful options, can be found in your *REALITY DATA/BASIC Programming Reference Manual.*

• Teri Garrett

THINK TANK

This segment of Think Tank takes a slow, winding track through several loosely-related topics, most of which have a common theme. If you share an interest in one of these, please write to us, sharing your thoughts or requesting elaboration in future columns.

Development Engineers

What do development engineers think about? It depends a great deal upon which point in a project they have reached. If, for example, the engineer is approaching the tail-end of a fairly major development effort, s/he spends a lot of time thinking about the trade offs that have been encountered in various aspects of the new design. Hopefully, the design is robust enough that there is room to augment it in areas which are found lacking. Where this is not the case, methods for working around any shortcomings are considered, and the engineer adds a lesson or idea to the list of things to be contemplated next time.

The next time might be a new release of the same application or operating system. Then, the thinking will change. It is a different kind of thinking, and it allows the engineer freedom to conceptualize, and to develop new architectures to support the concepts thus seen. This is a thinking process of greater scope, allowing solutions to be specified without regard to old restrictions, since these can probably just be designed away.

REALITY Operating System (ROS) 7.0

The group I work with is in the latter stages of a project called ROS 7.0. This is the newest generation of the REALITY environment, and will soon slide down a figurative code ramp to the great seas of computer users around the world. As such, engineers in our group are beginning to think about something next, such as a 7.1. For this, our thinking telescopes and becomes less restricted, even as we complete and polish ROS 7.0 during its beta trials.

When actually in the process of developing something, engineers think about such things as parsimony, execution speed, reliability, feature content and error checking, and the mutually exclusive tradeoffs between some of these. Of late, I have been thinking about parsimony versus error-checking.

Webster's defines parsimony as 1) a carefulness with (money or) resources, and 2) economy in the use of a means to an end. Any engineer knows that the more error checking a program has, the less parsimonious it is and, therefore, the less economic of system resources. In short, the contrast is slow and careful or fast and reckless.

Some developers claim that there are no data integrity errors on their system. More likely, the retrieval and update programs are so parsimonious that data discontinuities are simply not noticed. Systems like this are either entirely bug-free and running on error-free, no-fault hardware, or are programmed by fools.

Group Format Errors

Since Group Format Errors (GFEs) are simply data discontinuities which occur within a file group, and since all data on all machines may become discontinuitous due to corruption (even if only through malicious intent) then GFE's are unavoidable. Wouldn't you feel better knowing about them when they occur?

On ROS 7.0 we have made a special effort to balance parsimony with error checking. Any errors in data integrity that are noted by the OS are logged in a system log. All of the retrieval and update code is designed to be non-stop, and fast while missing as little as possible about the state of the data being retrieved or updated.

To keep the retrieval and update times down, the item and file structures have been redesigned to allow reliable checking, while helping to ensure a more robust data storage mechanism, thus compartmentalizing problems better. Naturally, the data must look the same from the application standpoint. Even more naturally, such redesign includes the popular requests for large items and items containing unrestricted binary data.

Hacking and Viruses

Speaking of data integrity and malicious intent, I have been thinking about the operating environment in terms of computer viruses. Most of us have read articles about all of the computer viruses being placed on various computers. Not too surprisingly, it turns out to be difficult--if not impossible--to introduce viruses on firmware-based machines running REALITY architecture. This leaves the aspect of system security, a traditionally weak point in many systems, as the next concern, so as to avoid system hackers.

Security

Generally, on systems of this type, there are three basic types of security which are feasible: account-based, user-based and location-based. When all users of a system use only one account and that account is unique (and passwords are not shared), then a type of user-based security is an apparent possibility through account-based security. Yet this scenario is not the usual case, so a higher level of security (than account-based) may be required. Location-based? Well, because a user may nowadays come into the system over a number of possible paths, it is not a total solution of itself either.

It may be desirable to provide each user of the system a unique user-id, which is applied at a level higher than the account level. With this information available, one would be able to identify any user of a system regardless of what account that user is logged onto. Of course, the other thought is that a system administrator should be able to secure certain physical locations from certain types of access, regardless of the user-id. This adds the dimension of location-based security.

It is interesting to note the sorts of programs that people write to secure their systems, given the existing limitations of the environment prior to ROS 7.0. Recently, a systems application programmer related to me his system's custom security for sensitive verbs.

Basically, any verb which would allow update or retrieval operations outside of the standard day-to-day operations is restricted. The restriction is universal, and applies to verbs such as EDIT, COPY, DELETE, etc.

When a user executes, say, the EDIT verb for the first time since his or her logon session, the verb--which is actually a replacement program--looks up the port number in a security file. If the location of that port allows it, the program passes all parameters to the previously renamed EDIT verb, and everything continues. Here, EDIT might actually be three backspace characters, yet the user sees only the Editor working as usual. This is a user-implemented form of location-based security.

Again, on this same system, if the location-based check is negative, the user is prompted for a special password. The user enters the password and is either granted or denied access to the verb. In this type of system, the password could either be a global password or any of several valid user identification codes. In the later case, this would be an example of a user-implemented form of user-based security.

The final phase of this same scheme is to allow continued access to all verbs of this class once the user has been verified, or until the user logs off.

- Garrett Hildebrand

14/100

14/100 Capabilities

One of the capabilities which the Series 14/100 provides, is for the operator at the PC keyboard to be running in the DOS environment at the same time the serial devices running under REALITY are operating in their environment.

While the PC operator is printing to the parallel printer from a DOS application, the operators at terminals on the Series 14/100 can send print jobs to the REALITY spooler; the REALITY spooler will hold the jobs until the PC operator assigns the parallel printer to REALITY.

One way to reassign the printer from DOS to REALITY, or REALITY to DOS, is to use the ROS ASSIGN command at the DOS prompt; wait for the screen display: "Configure the Parallel Printer"; and set the printer for use by REALITY.

A quicker way is to use the REAL_PRN command at the DOS prompt, which assigns the parallel printer to REALITY; and, DOS_PRN at the DOS prompt, which assigns the parallel printer to DOS:

```
C:\REALITY>REAL_PRN
C:\REALITY>DOS_PRN
```

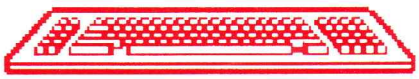
- Sam Craghead

APPS

Current Application Overlays

The following matrix provides you with the release level of Application Overlays required by each supported Series and OS. It is important that you know which Overlay you should obtain prior to a planned upgrade. For example, if you are upgrading a Series 9000 from 5.1 to 5.3 which uses REALCALC, then you will need to obtain the corresponding Overlay release (REALCALC 2.1C) before upgrading.

Application Overlay	Series 4700	Series 6000	Series 9000	Series 18
A*L*L 1.1	4.3	1.1	1.3	N/A
A*L*L 1.2 (Paper Patches 1-46)	N/A	2.3	5.3	6.0
PCmicroREALITY 2.0B	N/A	1.1	N/A	N/A
PCmicroREALITY 2.1	4.3	2.3	1.3,5.3	6.0
REALCALC 2.1C	4.3	1.1,2.3	1.3,5.3	6.0
REALGRAPH 1.0C	4.3	1.1,2.3	1.3,5.3	6.0
REALLINK 2.0	4.3	1.1,2.3	1.3,5.3	6.0
REALISM/DEVELOPER 1.0A	N/A	2.3	5.3	6.0
REALITY Integrated Office 2.3	4.3	1.1,2.3	1.3,5.3	6.0
WORDLINK 1.4	N/A	1.1	N/A	N/A
WORDLINK 1.4C	N/A	2.3	1.3,5.3	6.0
WORDMATE 2.1C (Overload Patch Tape Rev A)	4.3	1.1,2.3	1.3,5.3	6.0
TRANSACTION LOGGING 1.2	N/A	2.3	5.3	6.0



- Janet Altman

Accessing Documents in REALITY Integrated Office

Several calls have been placed inquiring about gaining access to word processing documents that have been created under a different User ID.

Suppose, for example, USER1 had created word processing documents through REALITY Integrated Office. However, since then, USER1 has left the company and now USER2 needs to have access to USER1's documents. USER2 attempts to edit one of USER1's documents, either through "1. Create or Modify Document Header" or "2. Edit Document," and receives the error message "This document cannot be accessed by you--try again!" How can USER2 access USER1's documents?

Very easily. The REALITY Integrated Office System Adminstrator can change a user's access rights to documents by exiting to TCL in the OA account and then entering the following:

```
:ED LETTER-FILE document-name <cr>
.TOP .G14  <cr>
014 USER1
.R/USER1/USER2 <cr>
014 USER2
.FI
'document-name' filed in file
  'LETTER-FILE'.
```

Please note that attribute 14 in the LETTER-FILE may only contain one User ID.

- Janet Altman

Slaving Print Jobs With REALLINK

REALLINK is a terminal emulator which emulates PRISM 4 terminals (and later PRISM models through PRISM 9) in addition to a number of other terminals. One of the capabilities of the PRISM terminals is that a printer can be connected to the auxiliary port. This printer can then be used as a slave printer allowing print jobs to be printed right at that particular terminal. In being a PRISM emulator, REALLINK also has this capability if there is a printer connected to the parallel port on the PC. Here is one way this can be accomplished.

First, a form-queue must be created for the port that the PC is connected to. This can be accomplished by selecting CREATE-QUEUE (Option 1) of the SP-STATUS screen. You will then be prompted for a form-queue name, a device type, and a device number. The form-queue name can be whatever you want to name your form-queue as long as a form queue does not already exist with that name. The device type will be "PORT" and the device number will be the port the PC is connected to.

After the form-queue has been created, the following DATA/BASIC program (we call it "SLAVE.PTR") needs to be entered into a file:

```
SLAVE.PTR
001 PRINT @(-17)
002 PERFORM "PORT-DESPOOL (X)"
003 PRINT @(-18)
004 END
```

Line 001 of this program tells the terminal that all output to the screen will now be redirected to the slave printer. Line 002 executes the PORT-DESPOOL verb which despools print jobs to the port. The (X) option on the PORT DESPOOL verb is used to bring the port back to TCL when all the print jobs for that port have been despooled. Line 003 of the program tells the terminal that all output will now go back to the screen. When the program has been entered, compile and catalog the program.

Next, assign the port you are on (the PC port) to the newly created form queue by using the SP-ASSIGN verb. Now you are ready to run your reports or whatever creates the print jobs that you want to print on the slave printer. When your reports are completed, run the DATA/BASIC program. Your reports should now print out on the slave printer and, when they are finished printing, you should be brought back to TCL with all output going back to the screen.

• Bryan Glassick

Series 7000

Quickstarts on Series 7000

Many M7000 users are already taking advantage of the ease of quickstarts. For those of you who aren't, let's discuss what they are and what they can do for you.

Quickstarts can eliminate repeating the same instructions to the system over and over on tasks that you perform daily, such as batch transfer or communications. A quickstart is a file containing the system commands in the order that you would do them by hand, with pauses inserted for keying the commands that change. A quickstart also can reduce the errors caused by the wrong commands being given to the system.

But quickstarts are not a panacea to be used instead of teaching operators how to perform system functions the long way. All supervisors or operators who do batch transfers etc., should first learn to do these jobs by manually keying in the commands. Then if for some reason you can't use the quickstart, you can still do your work manually.

If you need to brush up on the quickstart commands please check the front of your ECL manual in section two. There is a quickstart class being presented in March 1990 that would be helpful. If you have any questions about quickstarts, please contact the Series 7000 Group through our toll-free Central Dispatch number and ask for Helen James or myself.

• Ann Connelly

COMMS

Current COMMS Releases

The Product/Release matrix shown below details the current release of software for the various communications products.

Any software fixes which may be required on McDonnell Douglas hardware will only be produced for the most current release.

If you plan to upgrade your McDonnell Douglas Computer system to the next hardware system or operating system release, contact your ISO or analyst to make sure you have the required communications software prior to the upgrade. If in doubt, have your ISO or analyst contact the McDonnell Douglas Field Service Data Communications Group at (800) 678-3399.

COMMS Product Release	Series 4700	Series 6000	Series 9000	Series 14	Series 18
MCC 3.1 (Rev 4)	N/A	1.1, 2.2	N/A	N/A	N/A
MCC (2.3) 3.1 (Rev 4)	N/A	2.3	N/A	N/A	N/A
MCC (2.3) 3.1 (Rev 5)	N/A	2.3	N/A	N/A	N/A
MCC (6.0) 3.1 (Rev 3)	N/A	N/A	N/A	N/A	6.0
FTU 1.2 (Rev I)	4.3	1.1, 2.3	1.2, 5.1, 1.3, 5.3	2.3	6.0
M3800 (2780) 2.1 (YA)	4.2	N/A	1.2, 5.1	N/A	N/A
M3800 (2780) 1.3 (A)	4.3	N/A	1.3, 5.3	N/A	N/A
M3800 (SNA) 2.1 (YA)	4.2	N/A	1.2, 5.1	N/A	N/A
M3800 (SNA) 5.3 (A)	4.3	2.3	1.3, 5.3	N/A	N/A
5750 COMMS	4.2	N/A	1.2, 5.1	N/A	N/A
TCL (5750) COMMS (Rev 2)	N/A	N/A	1.3, 5.3	N/A	N/A
2602 BISYNC	4.3	N/A	N/A	N/A	N/A

Notes:

5750 Communications Software resides on the Sysgen tapes. If the software is not on the tape, it may be retrieved from a previous Sysgen tape.

TCL-COMMS software is run on the 5750 Communications Terminal, in lieu of the 5750 software described above.

2602 Bisync runs only on Series 4700 systems. The software is included on the Sysgen tape.

• Richard Yeh

McDonnell Douglas Field Service Company (MDFSCO) has provided data communication services for years. This section introduces some of the new offerings in wide area networks and IBM connectivity.

Network Services

The term "Network Services" is used in the industry to describe a wide range of design, implementation and support services. Within the telecommunication industry, design and implementation ser-

vices are typically known as Systems Engineering. The **Data Communications Group,** a part of the Product Support group of MDFSCO, provides these services. Following is a brief explanation of these services.

1. **Analysis of Customer's Needs**

   Designing a network involves determining the customer's needs. This includes listing every piece of hardware, software, cable and miscellaneous parts. The network requirements are established from the customer's business demands. The following factors are evaluated:

   A. For individual terminal and computer ports, the speed, flow control, data format, interface type, priority, service type, privilege etc.

   B. Logical partition of a network, "user groups" and their respective privileges.

   C. For each trunk, the speed, protocol type, modem and Telco interface cables, etc.

   D. Multi-vendor equipment interface, if any, and mutual compatibility

2. **Data Traffic and Analysis**

   MDFSCO will perform the network traffic analysis from the information provided by the customer. In the event the customer is unable to estimate the terminal usage, one of the widely applied industry models is used to establish the trunk bandwidth and related parameters. This service is crucial to ensure satisfactory network performance. The individual port and trunk speeds are established with close customer consultation.

3. **Network Map and Physical Layout**

   MDFSCO will prepare a detailed network map depicting the geographical locations, all major network components and equipment (Telco, switch, multiplexor, modem, terminal etc.) and all pertinent information to give the viewer an overview of the network complexity.

4. **Configuration Design**

   This service involves creating the "Application software" which provides the custom-built intelligence for the Network Processor. The data gathered in service items (1) and (2) above are used here to create the network configuration. The quality of this design is directly dependent upon the accuracy of the input data.

   This service includes entering the customer's specific parameters for each multiplexor, switch, modem, computer and terminal port. Product Support personnel prepare the software module in the Network Support Center and ship the module to the site or dial directly into the customer's network processor to enter the parameters.

   MDFSCO works very closely with the network users to ensure the network meets the customer's requirements.

5. **Network Implementation Plan and Schedule**

   This service details the implementation plan and coordination among MDFSCO engineers, the customer and the multi-vendor personnel. It provides the schedule of various phases of network installation and testing.

6. **Network Acceptance Criteria**

   MDFSCO will establish the acceptance criteria for the network performance from the data gathered during the design phase and the customer-approved implementation plan.

   It is common to experience constant changes in a large network environment. This, however, makes the task of establishing the acceptance criteria like shooting at a constantly moving target. Thus, a date is agreed upon with the customer to "freeze" the configuration requirements. The implementation of all the subsequent changes are deferred as a separate service until after the customer accepts the network as "Installed and Implemented."

- Niki Jhaveri

We will detail in the next issue our IBM Network Environment and related Network Services, including the Monthly Maintenance Support Program and the Installation and Network Implementation guidelines.

SNA: System Network Architecture (Part 3)

In the last issue we discussed the IBM environment prior to IBM's introduction of SNA in 1974. IBM users demanded some control of the chaos created by the lack of data communication standards.

In this issue we will look at the fundamental building blocks upon which the architecture is designed. At first, these concepts may seem very abstract and even irrelevant to most users. However, to understand the operating procedures in a SNA network, a basic understanding of the architecture is necessary.

Why an Architecture?

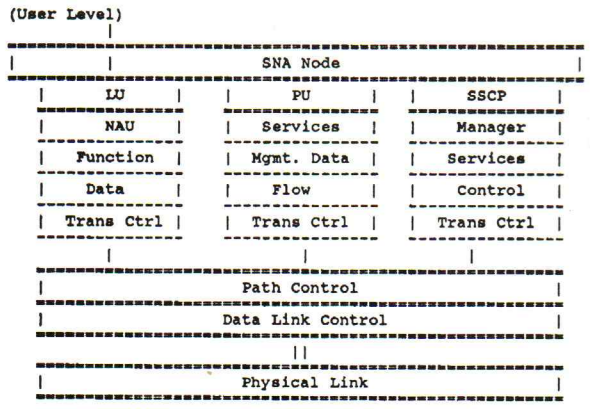
IBM needed a network architecture to specify the control and the flow of data between an application subsystem (e.g., CICS, TSO, IMS/DC) and an end user at the other end. Defining such an architecture is a very complex and substantial undertaking. IBM proposed an architecture that defined protocols, standards and message formats to which different machines and software packages must conform. When new products conformed to the architecture, they would then be compatible, and could be interlinked to share data, programs and resources of the network. The architecture, however, needed to be designed to evolve with the fast-changing technological environment.

The requirements of an architecture to be stable and yet evolving are conflicting. One technique to minimize this conflict is to compartmentalize the architecture into component parts. Each component is then treated as an independent entity to resolve its specific problem. This permits an enhancement of one component without affecting any other component. It also minimizes the impact on the architecture as a whole to maintain its stability. IBM developed SNA based upon such independent components called "layers."

SNA Layers

SNA offers six layers that specify the architecture. The figure below shows the SNA layered structure. Each of the layers and its functions are discussed briefly.

SNA layers and network organization.



The **PHYSICAL CONTROL** layer specifies the physical and the electrical characteristics of the interface to the network hardware (i.e. modems, terminals, computers). The most common interface is RS-232C for data transmission and RS-366 for automatic calling. These standards specify how a computer or terminal (i.e., DTE) interacts with a modem or an auto call unit (i.e., DCE).

The **DATA LINK CONTROL** layer is responsible for the management of the physical link including error recovery and data flow control. This layer could be the Synchronous Data Link Control (SDLC) or IEEE 802.3 in a local area network environment.

The **PATH CONTROL** layer is responsible for the management of the routes which interconnect the network components.

The **TRANSMISSION CONTROL** layer keeps track of status of sessions that are in progress. It controls the pacing of data flow within a session, and sees that the units of data that make up a message are sent and received in the proper sequence.

The **DATA FLOW CONTROL** layer maintains the overall data integrity of the flow of data during a session between two network addressable units. It involves determining the mode of sending and receiving and managing groups of related messages. This layer maintains control of the end-to-end connection between an application (e.g., CICS, TSO) and the end-user (i.e., 3270 or 3777 terminal).

The **FUNCTIONAL MANAGEMENT** layer provides high level SNA services such as coordinating and controlling the activities of the network as a whole. It controls all the sessions that are active. It also coordinates the interface between the network user and the network, as well as the presentation of information to the user.

The **NAU SERVICES MANAGER** layer provides services to the functional management layer and the data flow control and the transmission control layers.

SNA Fundamental Concepts & Components

This section introduces some concepts fundamental to SNA. It will help the users develop a vocabulary and recognize the messages on the terminal screen while establishing a session with an IBM mainframe.

IBM defines three "entities" known as **Network Addressable Units (NAUs)** to represent participating devices and programs in a SNA network. Each NAU uniquely defines and represents a device or a program and is identified by a unique address.

The three NAUs are: Logical Unit, Physical Unit and System Services Control Point.

- **Logical Unit (LU)**: An important function in a SNA network is to implement a virtual or logical path between users so they can communicate with one another. To establish such a path, each user must gain access to the SNA network. SNA

defines LUs as a point of access for end-users to interact with the SNA network.

- **Physical Unit (PU)**: An SNA network consists of computers, controllers, and terminal devices. They, however, are not part of architectural definition, but are the devices used to implement the network. Instead, SNA uses PUs to represent actual devices to the SNA network. A PU provides the services needed to manage and use a particular type of device to handle physical resources.
- System Services Control Point (SSCP): A SSCP is an entity to provide the services needed to manage a SNA network. It establishes and controls the interconnections among network users. This unit is considered the "brain" that controls the network.

Cont'd on last page

CUSTOMER ED.

Expanded Customer Education Schedule

Central Support is pleased to present the following Customer Education schedule. Course offerings have been expanded, both in terms of subject matter and location. In most cases, on-site training is also available. If your Independent Sales Organization (ISO) does not offer a class that you are interested in, contact us at (714) 566-4803 for course information. If you would like to see classes that are not listed, please let us know what they are.

MCDONNELL DOUGLAS
CUSTOMER EDUCATION SCHEDULE

	DEC				JAN					FEB				MARCH				APRIL				
COURSES OFFERED	4	11	18	25	1	8	15	22	29	5	12	19	26	5	12	19	26	2	9	16	23	30
INTRO TO REALITY O/S 4 Days $800/Person	NJ SL	SA				FL SA	SL	NJ		SA	AT NJ		SL	NJ	SA	SL			SL	DC SA	NJ	
ADVANCED REALITY O/S 4 Days $800/Person	SA	SL						SL			SA				NJ		DC				SA	
ACCELERATED REALITY O/S 4 Days $800/Person													DC									
INTRO TO DATA/BASIC 4 Days $800/Person		DC					SA			NJ				SL						SL		
ADVANCED DATA/BASIC 4 Days $800/Person			DC						SL			NJ				SA			NJ			
ACCELERATED DATA/BASIC 5 Days $900/Person	DC																					
PROC PROGRAMMING 2 Days $450/Person																	DC					
SYSTEM INTERNALS & MAINT. 4 Days $900/Person																					AT	
WORDMATE WORD PROCESSING 2 Days $400/Person								SA														
REALISM SHELL 4 Days $800/Person						SA																
REALISM DEVELOPER 4 Days $800/Person										SA												
INTRO TO DOS FOR REALLINK* 1 Day $200/Person														SA								
REALLINK* 1 Day $200/Person														SA								
BASIC SOVEREIGN OPERATION 4.5 DAYS $1000/PERSON								SA														
SOV. FORMATTING/KEY BASIC 5 Days $1000/Person												SA										
SOV. REFORMAT/QUICKSTARTS 4 Days $1000/Person																	SA					
SOV. BASIC PROGRAMMING 4.5 Days $1000/Person																					SA	

LOCATION CODES: AT = ATLANTA, GA; DC = WASHINGTON, DC; FL = TAMPA, FL; NJ = SECAUCUS, NJ; SA = SANTA ANA, CA; SL = ST LOUIS, MO

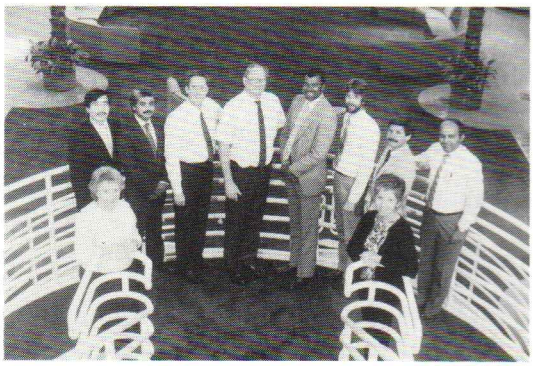
*** Please note: You can register for each or both REALLINK classes, depending upon your interest.**

Central Support:

(Left to right, front row) Grace Varela, Susan Ewing; (Back row) Cherylnn Pope, Gary Moote, Teri Garrett, Greg Mount, Jim Smith, Bob Aull, Brad DeWitt, Susie Seipel, Linda Denney, Jan Heppert. (Not pictured: Mike Bingman, Sam Craighead, Chuck Dorich, Will Edwards, and Van Harris.)

Applications Support:

(Left to right) L.W. Abel, Terry Smithton, Steve Gill, Janet Altman, Bryan Glassick.



COMMS/Sovereign Support:

(Left to right) Ann Connelly, Richard Yeh, Mauro Mendoza, Steve Champeau, Steve Moore, Randy Jordan, Brent Barrett, Craig Steigler, Niki Jhaveri, Helen James.

Sustaining Hardware Support:

(Left to right) Ray Van Sluis, Mark Onoda, Ralph Montalvo, Steve Gill, Keith Peterson, Robert Slagle, Jim Kirk, Brian McKinstry. (Not pictured: Gary Monroe.)

Cont'd from page 14.

SNA End Users

In SNA, the term "end user" refers to either a person or an application program residing in mainframes, controllers or personal computers. End users reside outside the architecture and access the network through a LU.

In the next issue, we will conclude the description of the SNA architecture. We will also put the building blocks together to form a network depicting the specific IBM and MDCSC products.

- Niki Jhaveri

GooFiEs

Central Support
ON-LINE

Published for System Software Users

L.W. Abel, Editor
Linda Denney, Asst. Editor
Will Edwards, Technical Advisor
Pat Dwight, Production Consultant
Cheryl L. Bouwens, Production
Bryan Glassick, Cartoonist

Please write to the Editor, ON-LINE, Mailstop FS200, 1801 East St. Andrews Place, Santa Ana, Ca. 92705.