VIM

INC.

NOS/BE TO NOS MIGRATION A WHITE PAPER

JANUARY, 1980

VIM

INC.

AR/Z

VIM INC.

PRESIDENT

Mr. James F. Presti (BCS)
Boeing Computer Services
Mail Stop 36-04
P.O. Box 24346
Seattle, Washington 98124
206-433-1432

February 4, 1980

To: VIM Membership

Subject: NOS/BE to NOS Migration White Paper

At VIM 30, in May of 1979, the VIM Board of Directors established a task force to address issues related to NOS/BE to NOS migration. These issues had been the subject of innumerable discussions in VIM as well as at many of your sites. It was felt that appropriate attention in the form of a task force might generate information sufficient to allow factual comparisons and escape the paralyzing effects of extensive rhetoric.

With the establishment of the task force, Abe Levine of Rockwell Intl., was asked to serve as chairman. The enclosed White Paper is evidence of the fine work done by Abe and the task force. Abe has presented the findings of the task force to the Operating Systems Committee and will present these findings in a VIM 32 session in San Francisco in late April. At that conference, it will be determined if any further activities are in order for the task force.

On behalf of the VIM Board of Directors and the VIM membership, I would like to thank Abe, the entire task force, Rockwell Intl., and the support from the CDC Los Angeles office.

J. F. Presti
VIM President

JFP/bjs

Enclosure

cc: T. N. Burt for VNL
A. S. Levine

FORWARD

As all of you are quite aware by now, Control Data has decided to standardize it's software product offerings around the NOS operating system. This has created quite a bit of anxiety among the installations still using the NOS/BE operating system. Therefore, on May 1979 at VIM 30 at Phoenix, the VIM board of directors created the NOS/BE to NOS migration task force. Its purpose is to bring into focus many of the issues effecting a site's decision whether or not to convert (or migrate) to NOS. As part of its activities the task force has created this white paper in order to aid installations make the correct decision for their site.

Though all of the chapters form one complete document, they can also stand alone as separate entities. Thus, the different chapters can be distributed separately to different members of the organization. The NOS/BE-NOS differences and the MACRO comparison document were specifically designed to be used as separate migration aids.

No conclusions are presented in this white paper. Each individual installation will have to make its own decision to migrate for its own unique reasons. It is hoped, however, that the material contained herein will aid the site make an intelligent and informed migration decision.

I would like to take this opportunity to thank the many people who devoted the time and effort to make this paper possible. Thanks to Lorraine Minor (NRDC) for preparing, distributing and analyzing the questionnaire as well as helping in the differences document. Thanks to Tom McCune (Rockwell) for his help in developing the MACRO differences document and for general support in other chapters as well. Thanks to Ron Dawson, Don Rowland, Len Sweet and Jerry Sherman (all from Control Data, Los Angeles) and to Don Larson, Joe Loy and Frank Minor (all from Rockwell International) for their help in designing, executing and analyzing the benchmarks.

Abraham S. Levine
Chairman, NOS/BE to NOS Migration Task Force

INTRODUCTION

So you want to convert NOS? Maybe you have decided to tough it out with NOS/BE. Before you make up your mind, you should consider the issues. Of primary concern is what the computing requirements are at your site.

Do you support many interactive terminals on your machine?

Is batch turnaround of primary importance?

Do you have to interface to foreign mainframes?

Do you have a requirement for systems like MSS or LCN?

Most probably, "all of the above" would be your answer to those questions. In that case, a closer analysis of your job mix is in order. Are your jobs CPU bound? Do they use large amounts of CM/LCM/ECS? Maybe they use large amounts of disk space. Your answer to the above is probably "yes" in all cases. In that case, a detailed analysis of each system is in order. Does each system satisfy all your present and future computing requirements? If so, a conversion is not for you. The answer is probably somewhere in between. Then you must decide which features are most important to you. Performance is also a consideration. CEMS/PSR support is also important. NOS/BE is obviously not going to receive its fair share of Control Data's attention.

After having performed the same analysis, you now decided that NOS is for your site. Have you considered what a conversion is going to cost? One site said that it cost them over a quarter of a million to convert. How long will it take to convert? By the time you do, there could be a new mainframe announcement. Have you considered how many people in your staff you could devote to a conversion? Maybe you will have to hire new people. Your systems programmers will have to learn NOS fairly quickly in order to expedite your conversion. Your user coordinators and applications programmers will have to get started learning and rewriting many programs, especially the accounting programs.

After reading this section, you are probably very tired thinking about your conversion. Just think about all the lost sleep, bad dreams, late (or all) night computer time, arguments with the spouse and kids, or mother-in-law. If you are still reading this, then NOS is for you.

Seriously, though, an NOS migration is a complicated decision to make. Large amounts of money will be spent in the migration itself. Be sure that the benefits justify the cost. Included in this White Paper are some handy documents that should aid in making an informed, intelligent and correct decision regarding a migration. Happy Reading!

7766B
AL

ANALYSIS

Many factors must be weighed in deciding whether or not to convert (or migrate) to NOS. These factors must be analyzed in terms of the benefit to the corporation in making an operating system switch. Presumably an operating system change will only be undertaken if the benefits outweigh the costs incurred in the conversion process. The benefits to the corporation can be broken down into two parts: The benefit to the computer installation and the benefit to the user. Though an argument can be made that benefits to the installation accrue as well to the user community, that is not always the case. The impact of a conversion may be such that the productivity and ability of the user to do his job in an acceptable time frame may be impaired. This could cost a corporation a small fortune. Thus, user benefits must be analyzed on their own merit. It must be shown that a conversion will, in the long run, increase productivity and therefore save the divisional entity and the corporation money.

The benefit to the installation:

It is in this area that a conversion to NOS is most attractive. Most of the new NOS features are in the areas that aid in the management and expansion of the computer and its resources. These are: job administration, security, communications software, additional hardware support and operating system internal design changes. Some of these new features will indirectly benefit the ultimate user as well. The current as well as future security features will enhance the user's ability to protect his data files and programs. While NOS/BE also provides file and program security, its philosophy is different. NOS/BE requires explicit password protection while NOS uses implicit protections. This seems to be the current trend in security software. New communications features will allow more users to access the central host computer through networking techniques. Support of more terminals will also benefit the user.

Control Data has stated that NOS/BE will support new hardware (such as disks) as long as it does not become a major project to do so. MSS is a case in point. Double density 819 disks are another. In both cases it is a major undertaking to provide the necessary software support. However, 885 support is present in both systems. Since the useful life of these devices will be at least a few years, a purchase/lease of these disks could forstall an expensive NOS conversion. SCOPE2 sites with 176 mainframes and double density 819s could never convert to NOS/BE if they so desired. A NOS conversion is feasible if not actually practical for other reasons. For those 176 NOS/BE sites with field upgradeable 819s NOS becomes attractive. The cost of upgrading to double density is quite low. Considering the enormous transfer rate of these disks and the storage capacity (800 million characters) many 844s could be returned and a power and space savings realized.

MSS should be considered on its own merits before a conversion to NOS is investigated. It has been shown, for example, that MSS does not substantially reduce a site's tape requirements and mounting. It does, however, reduce the number of private packs and their associated pack mounting. MSS would provide the installation with a nice archiving medium. Anyway, once the MSS analysis is complete NOS should then be considered for all other reasons as well.

The area of communications is extremely important. Vast sums of money can be lost if careful analysis is not made. A site must plan its requirements for some years ahead and decide in this context if a NOS conversion is desirable. More protocols, remote front-ends, and foreign hosts will be supported under NOS. However, it is possible that such capabilities are not needed by the site. Then a conversion will not be necessary based on communications needs alone. For sites planning networks from scratch, a make or buy decision is in order. What would it cost to design and implement a custom network? What are the benefits? What is the cost of converting to NOS and using CCP and/or RHF? Since communications networking software is long in coming from Control Data, many sites have already implemented their own communications system. What would it cost to convert these systems to NOS? Maybe such systems are old enough and cost enough to maintain that they should be discarded. Most probably, however, sites with their own front-ends and possibly networks, will need to find other justifications in order to migrate to NOS.

More and more installations are finding it necessary to communicate with foreign host mainframes. In many cases the CYBER is not the main computer at the installation (it is brand I). For them, networking usually means hooking into SNA. Loosely coupled networks (LCN) could be the answer. It will allow communication between foreign mainframes and peripherals. LCN software will only be supported under NOS and its Remote Host Facility. However, LCN will not communicate with SNA. It is designed to communicate with JES only. Considering the potential of the product VIM should take an active interest in LCN and guide Control Data in the design in order to best satisfy member installations' requirements.

As computer mainframes and peripheral equipment become faster and sites increase their workload, stress on the operating system internal design increases as well. Many NOS/BE sites are currently experiencing service degradations due to limitations in critical operating system resources. They are: FNTs, RBT chains, terminal IDs, number of permanent files and number of control points. The current NOS design solves some of the problems. NOS supports 131071 unique username/password combinations. NOS/BE supports only 1296. Control Data, however, is investigating the terminal ID problem and may implement a solution that expands the limit to 4095 IDs. This may be enough to allow sites to remain on NOS/BE and still expand their remote user base. To those sites who already have some 16000 unique permanent files on their default device set, NOS may be a godsend. There is no such limit on the number of permanent files. NOS also supports more control points, but not that much more. After using control points for BATCHIDO, WHAM, RBF, IAF, MAGNET, TAPF, etc., not much extra remains. Currently NOS has the same FNT problems as does NOS/BE. However this is due to be solved in NOS R6.

NOS is not without its design limitations as well. EST size, MST and TRT locations can be a major problem for sites with large disk configurations. Disk overflow, permanent file name sizes and cycles are others, for example. A site will have to weigh the limitations of both systems and the future enhancements of each before making its decision. It's sometimes like choosing the lesser of two evils.

General level of support and responsiveness by Control Data toward either system must be analyzed. Many feel that NOS/BE support from Control Data is minimal. Many feel that NOS/BE sites are subsidizing NOS with their monthly CEMS charges. Operating system stability (both present and future) is of prime importance to an installation. Is NOS a more stable system? Are PSRs answered in a more timely fashion? These questions must be answered before a conversion is decided upon.

What about performance? This affects both the user and the installation. While those are important to the installation as well, the number of jobs and sessions per day are also important. The system that allows the most efficient use of the hardware is the one that will keep down the cost of CYBER computing.

The benefit to the user:

In this area the benefits are less profound. One of the reasons is that both systems share basically the same product set. Thus an FTN compilation is the same on both systems. A site must approach an analysis of user benefits from the following points: feature differences benefitting or hindering the user and performance differences (batch turnaround and interactive response time).

NOS seems to have more "goodies" control cards than does NOS/BE (e.g., COPYEI, TCOPY). NOS certainly has a different permanent file philosophy than NOS/BE. NOS/BE supports longer permanent file names and PF cycles. NOS supports both MODIFY and UPDATE. NOS/BE supports just UPDATE. There are Control Data products that will only be supported under NOS. ASCII character processing is different under NOS. NOS has more editors than NOS/BE (some feel that NOS/BE has no editor at all - pardon the editorial opinion). Both systems support the common decks and their macros. NOS interactive processing and commands are much different from INTERCOM. One could almost migrate from a GE440 system to NOS and not notice much of a difference (well, the hardware is faster). User libraries are much more convenient under NOS/BE. In any event, a site will have to examine the reference manuals and the feature difference document contained in this white paper to extract further details.

Performance may be what sells or kills the conversion in the eyes of the user. If he feels that he can get faster response and turnaround from NOS then he will accept it. Benchmarks will need to be run to prove the case one way or the other. Sites with high interactive usage will probably have an easier time selling NOS because of its normally faster response time. However, the effort involved in converting batch users is much less than for interactive. After all, the interactive user needs to learn a new language. For many batch users a new set of accounting cards may be the extent of the conversion.

Stability will be a major concern to users. A fast system that hangs alot is a highly negative influence (as it should be). Perhaps a series of "stress" benchmarks could weed out operating system bottlenecks and weak points. Recoverability will also be important to users. How well does NOS recover from hangs? How much is saved? The last thing a user wants to do is lose his edit session when the system goes down.

How friendly is NOS versus NOS/BE? What happens if a user hangs up his telephone? How are the error messages? Are they cryptic? Are they understandable? These are also valid and important user concerns?

Finally, how much will it cost the user to convert to NOS? How much free computer time will he receive? How much of his time will be spent converting rather than producing? What kind of service impact will the conversion have? Considering general user skepticism toward operating system change benefits these concerns may override all others.

Summary:

It is hoped that this section will aid sites in clearly analyzing whether or not to convert. It is neither an easy nor inexpensive decision. The key point to remember is which system satisfies the corporation's current, as well as future, computing requirements the best.

7769B/AL

QUESTIONAIRE ANALYSIS

A questionnaire was distributed during the summer to many VIM sites. The purpose of the questionnaire was to see first how many NOS/BE sites were planning to convert to NOS. NOS/BE sites were also asked to list the features that must be present in NOS before thay would even consider converting and whether these features must be supplied in standard software. Finally, NOS/BE sites were asked to list the local modes currently in their system. This was to see how many of the enhancements are present in NOS and how many could never be installed due to its architecture. NOS installations were asked to comment on NOS/BE features they did not want to see in NOS. This question was very significant in that many NOS sites did not want features deemed essential to NOS/BE installations.

Judging by the answers to the first question it would appear that the overwhelming majority of NOS/BE sites have no intention of converting to NOS. Of the 30 respondents, 17 indicated that they are not going to convert. Four were not sure and nine intended to convert. Of those planning a conversion, two were converting immediately, three are planning a 1980 conversion, two are waiting for NOS R6, one for mid-1981 and one site is patiently waiting until 1984 to convert. A few sites have deferred their migration plans pending a new mainframe announcement by Control Data. They felt that if this new system is going to require a major conversion from NOS they would rather convert one time only from NOS/BE to the new system. One site felt that the thrust of a migration analysis should be in this area.

The responses to the second question dealing with essential features were quite varied (some 75 different essential features were listed). This was to be expected with an open-ended question like this. No list of features was presented to the respondents to choose from. Some NOS deficiences have been so publicized as to make some answers almost automatic to many. In addition, a significant percentage of the NOS/BE respondents are not that well-versed in NOS. This could lead to the exclusion of important but not well-known NOS deficiencies. The features receiving the most attention (double-digit responses) were:

- Device overflow as a means to prevent a job from aborting if the device is full (14 respondents).
- Longer permanent file names, not necessarily 40 characters (14 respondents)
- User-initiated permanent file archive (12 respondents)
- Job-card priority (10 respondents)
- Permanent file cycles (10 respondents)

As for who should be responsible for implementing these features it was generally felt that Control Data should implement these in the standard system. In fact, for the top five features listed here only one site was willing to receive code other than standard. This was for job card priority support. Some sites were willing to develop their own code if it were not available from other sources.

The next question dealt with the local enhancements sites have installed in their operating system. The intent of the question was to see how many of the NOS/BE local mods are either presently in NOS or easily integrated by the site. The NOS installations were asked the same question to determine if the mods were features present in NOS/BE or local. By and large it seems that many NOS local modifications 13 are current NOS/BE features. NOS/BE modifications, on the other hand, seem designed to enhance features already present in the operating system. Of the 18 most popular local mods only 4 are present in NOS. Ten of the 18 modifications will soon be present in NOS. Of the 10 easy-to-install modifications only one (performance measurement code) is also on the NOS local mod list and it will soon be standard software. Thus, converting sites will have some coding to do in order to install their local enhancements in NOS. Additionally, seven of the 10 NOS-type local mods found in NOS/BE are high on the list of features essential to a migration effort.

The last question was asked of the NOS sites only. They were asked to list NOS/BE features they did not want added to their NOS system. Though some wrote that they did not know enough about NOS/BE to answer the question many sites presented detailed lists with accompanying explanations. The overwhelming negative vote was cast against the NOS/BE permanent file structure and many of its features (long permanent file names, cycles). It was generally felt that the NOS permanent file structure is much more efficient and faster. Some typical comments were: "The NOS method for economizing directory searches is much better than allowing 'random' ID's with the resulting expense" and "I consider the solution to have private permanent files with the permissions to individual users better than the password-controlled solution of NOS/BE." Interestingly, the longer permanent file names feature was tied for first in rating by the NOS/BE sites. This conflict will create some obvious problems for Control Data in choosing which features to add to NOS to encourage a migration.

As a side note it should be observed that two sites (both universities) stated that the loss of 40-character permanent file names in converting to NOS was hardly felt. At worst it was a mild inconvenience. No industrial installations have yet commented on this, however.

It is known, though, that a large service bureau is running KRONOS/NPS with well over 100 permanent file disks. The absence of long permanent file names doesn't seem to be a major problem to them.

There were three other significant negative votes cast against NOS/BE features. All received the same number of votes (3). They are: Device overflow, a stack processor and "dirty coding". Many respondents to the questionnaire were afraid that code would be installed in NOS that came directly from NOS/BE. Such code would not adhere to coding standards and would cause instabilty in NOS. In general, concern was voiced that any new feature not impact the reliability of the system. Many other NOS/BE features were mentioned as undesirable by NOS sites but they were not listed as essential NOS features by NOS/BE sites.

What can be concluded from this questionnaire? It would be very important to know why most of the responding NOS/BE sites are not planning to convert. Is it because of NOS feature deficiencies or a "Hell no I won't go!" attitude? If the questionnaire were to be sent again, say in six months, how many of the sites would have changed their mind as more information about the NOS system and future features became available? I suspect that the more NOS is talked about and analyzed the less fearsome it becomes.

NOS/BE sites will have to install a fair amount of code to install their local enhancements in NOS. Some of these can be obtained from other sites as well as QSS code from Control Data. Though almost every site wrote that all such mods which are essential standard NOS features must be installed and supported by Control Data, many of the small ones will probably be locally coded and installed if necessary. Perhaps the greatest amount of effort will be spent on an item barely listed in the NOS/BE local enhancement section of the questionnaire. That is local accounting, both off-line and charge algorithm. A significantly higher percentage of the NOS sites listed local accounting as one of their local enhancements. The greater capabilities of NOS is that area probably requires a greater sophistication of accounting programs. It is partly because of these programs that some NOS sites do not want any tampering with the permanent file names or other features. Some accounting programs will have to be rewritten. To summarize, no matter which position you take on the conversion and the necessary features, many sites will have to expend quite a bit of man-hours installing local code to smooth-out a migration or upgrade their NOS system to accommodate new features.

Finally, as expected, device overflow and long permanent file names were deemed the most essential features to be added to NOS before a conversion would be attempted. The addition of these features, however, is being vehemently opposed by NOS sites as being unecessary, unstable and requiring conversion on their part as well. The opposition to device overflow is not as strong as the NOS/BE vote is to incorporate it in NOS. The opposition to longer permanent file names is much stronger on a percentage basis. While NOS sites are against any name greater than seven characters, many NOS/BE installations are willing to compromise and accept some smaller number, around twenty. This was also established at a closed VIM meeting in July, 1979.

Control Data, it seems, is caught in a classic dilemma. Does it add features in order to encourage a migration and alienate its established NOS base or refrain from adding certain features and have the migration fall flat? Compromise will certainly be the order of the day.

7757B/AL

NOS/BE Migration Project Office

The NOS/BE Migration Program Office was established to provide a focal point within the corporation for the accumulation and dissemination of information pertaining to customer conversions from NOS/BE to NOS. More specific responsibilities include:

- development of training classes and literature required to meet the combined needs of internal Control Data and customer personnel.
- assisting the development of specific site migration strategies.
- technical presentations/discussions related to software differences and work-arounds.
- sponsoring the development and distribution of conversion aid utilities.

The conversion aids currently offered are designed to be: stand-alone utilities; to be used only during actual conversion; will generally be discarded when the conversion is complete.

Plans for the conversion aids include:

- standardization by the first quarter of 1980
- packaging as a single product with multiple aids
- requirement of a NOS license and right to use with PSR support at no charge
- additional aids to be added when available

The currently available conversion aids are:

- BELOAD - multiple NOS/BE permanent file load onto a NOS system
- LOADBE - single permanent file load
- HELPBE - on-line user assistance

In addition to conversion aids, the Migration Project Office will sponsor training classes in the following areas:

- NOS Application and Software Education (a current standard CDC offering)
- usage difference class
    - how to do the same (NOS/BE) job on NOS
    - control statements
    - interactive
- SCOPE 2 - NOS file conversion guide (available soon)
- magnetic tape difference pamphlet

7678B/AL

NOS/BE-NOS MIGRATION: SOFTWARE QSS (s)

QSS NO.	ITEM
20864	Job Priority (NOS R4)
21692	Job Class Priority (NOSR4/485)
21700	Common I/O Queues (NOSR4/485)

SPL CAT. NO.	ITEM
Q142-02	PF Archive (NOS R4/485)
Q153-02	Tape Reservation System (NOS R4)
Q169-01	TEXTJAB 2.0
Q321-01	Job Dependency (NOS L498)
Q327-03	7000 NOS Station (NOS R4/485, SCOPE 2.1.5)
Q356-01	Remote Operator Station

NOS - NOS/BE Comparison Document

A. Introduction

The purpose of this document is to provide an unbiased tool to be used in the decision whether or not to convert to NOS. Contained in this document are basic differences and comparisons between NOS and NOS/BE. Many of the differences and comparisions may not apply to all sites since many installations have added local modifications and enhancements to their system.

As this is not a purely original document, many people must be thanked for their prior work in preparing comparison documents for their own sites. Additional people must be thanked as well for their help in preparing this document. Many thanks to:

Steve Hallstrom	-	University of Washington
Helga Koste	-	Industrieanlangen - Betriebsges
Gregg Townsend	-	University of Arizona
John Stephens	-	McMaster University
Lorraine Minor	-	NSRDC
Tom McCune	-	Rockwell International

B. JOB DECK STRUCTURE

1. General Job Deck Structure

The basic job deck structures for NOS/BE and NOS are the same. Both use the logical record concept, where the job deck may be broken into logical records, and the first logical record contains the control statements.

NOS further allows a job deck to contain files, where each file may contain one or more logical records. NOS/BE does not permit this feature.

1.1 Delimiter Cards

Both NOS and NOS/BE use the 7-8-9 and 6-7-8-9 cards for end-of-record and end-of-job terminators respectively. NOS terms the 6-7-8-9 card as the EOI or end-of-information card while NOS/BE calls it the EOF or end-of-file card.

NOS provides an additional delimiter card, the EOF (6-7-9) card, allowing another level of delimiting.

NOS/BE provides record level numbers which in a sense provide 16 more levels of delimiting, although not all utilities differentiate between level numbers. These level numbers are entered on the 7-8-9 (EOR) cards.

1.2 Logical Records

The structure of logical records within the job deck is again the same for NOS and NOS/BE with the exceptions noted under delimiter cards.

The first logical record contains the control statements, and subsequent logical records contain miscellaneous types of data (program source, raw data, etc.) for both NOS and NOS/BE. Both systems permit similar manipulation of the input file, including copying, rewinding, skipping records, etc.

1.3 Card Formats

The delimiter card differences have already been discussed.

The 026/029 mode options for interpreting Hollerith cards are present on both NOS/BE and NOS, but NOS offers further differentiating possibilities: mode can be indicated on the job card, and/or on EOF card, and/or on a special "change conversion" card (5-7-9).

The rules for reading binary cards (checksumming, word count, sequence number convention) are the same.

The convention for handling "literal input" or "free form binary" (accepting cards in full 80 column binary format whether or not column 1 has 7-9 punch) is available on both NOS/BE and NOS, but the delimiter cards are different.

2. Job Identification, Accounting and Resource Control Cards

In both systems the control card record contains the job's control cards which direct the processing that is to occur. Under NOS/BE the Job Card and Account card are the first cards in a deck and are always required. Under NOS the Job and User Cards are required to be the first two cards in a deck. Depending on other factors two more cards (Charge and Resource Cards) may be required.

```
NOS/BE -   jobname, Tt, IOs, CMfl, Pp, Dym, MTn, NTn,
           Stxxx.  Accounting Card (Installation-Defined)

NOS    -   jobname, Tt, CMfl, Pp.
           USER, usernum, passwrd, familyname.
           CHARGE, chargenum, projectnum.
           RESOURC, rt1=u1,.....,rtn-un.
```

NOS/BE's Job Card includes fields for specifying time limits, central memory requirements, job priority, and tape drives required.
NOS Job Card contains equivalent for central processor time limit and central memory requirements. The Job Card differences are:

1. Priority - NOS has defined a priority field but it is disregarded by the system. NOS/BE uses the priority as a factor of choosing which job to initiate next and which output file to print/punch next.

2. Time limit on NOS defaults by 100_8 seconds. Under NOS/BE, default is installation parameter. Time limit on NOS is for each job step; for NOS/BE it is for the entire job.

3. Default field length under both systems is either 50000_8 words, or installation defined. But, as under NOS/BE, this field length is changed for different job steps by the required field length in a loader table (54-table).

4. NOS has no equivalent field for NOS/BEs:

Dym — Job dependency (Job dependency is handled differently in NOS. CDC has mentioned possible support of Job Dependency as either a special product or standard software by late 1979.)

CPp — Selects a specific central processor on a dual processor system. Under NOS, the USECPU control statement provides an equivalent function with the additional flexibility that the user can change the central processor specification at any point in the job.

STxxx — Provides a destination id for multi-mainframe configurations.

IO+ — Defines the I/O time limit, based on dynamic accumulation of I/O channel time.

5. The NOS equivalents for NOS/BE's tape scheduling parameters MTk and NTk are specified on the RESOURC card.

The tape and disk resource scheduling for NOS/BE and NOS have several basic differences:

a. NOS/BE requires that the maximum number of devices in use at any one time always be pre-scheduled on the Job Card, else the job is aborted. NOS permits a user to use a single device (maximum of one device in use at any one time) without pre-scheduling it.

b. The Resource Card need not be at the beginning of the job deck in NOS. It can occur anywhere. If no device is assigned to the job at the point where a Resource Card requests devices, the devices are reserved for the job if available or the job is rolled out awaiting availability. If the user is not validated for the requested devices, the job is aborted.

c. NOS/BE does not initiate a job until the scheduled devices are available. NOS will initiate a job regardless of whether or not the devices are available, as the system does not interpret the Resource Card until the job has been initiated. If a device is already assigned and the newly requested devices are not available, the job may be aborted or rolled out.

The User Card in NOS specifies the user's user number and password. If user account validation is active, the card must follow the job card. The user number is used in system bookkeeping and defines the user's permanent file catalog area. The password may be up to seven characters and its value is whatever the user selected, when his user number was originally entered into the system (placed on VALIDUS file). The user may be allowed to change his password via a PASSWOR command, depending on his validation privileges.

Other privileges or attributes are kept in the VALIDUS file associated with each user, such as:

- Equipment usage - Maximum number of magnetic tapes and removable disk packs allowed.

- File usage - Maximum number of permanent files allowed. User allowed to create DA or IA permanent files. Maximum size of IA permanent files. Access to system files.

- Machine usage - Maximum central processor time. Maximum central memory space.

- Terminal usage - Which terminals are available for individual users.
  Terminal type.
  Transmission mode.

- Priorities and the number of deferred batch jobs, etc.

The Charge Card permits the entry of a charge number and project number. User numbers may be validated to either require the Charge Card following a User Card or to make it optional.

The information on the Charge Card is checked against the PROFILB file. This is a system file containing:

o list of valid charge numbers

o list of valid project numbers within each charge number

o set of constraints for using each project number, including:

  - which users are allowed to use it
  - hours of the day during which use is permitted
  - limit of amount of processing time

o master user number for each charge number who is allowed to query and change project numbers, etc., for that charge number.

The RESOURC Card is used to schedule magnetic tapes and removable packs for the job. The card may occur more than once in the job deck to increase or decrease scheduled devices. The Resource Card is not required if only one device is to be used at any one time during the job.

3. Other Control Cards

3.1 Control Card Format

The general format for NOS/BE and NOS control cards is very similar. Both use an initial keyword of 1 - 7 characters followed by an optional list of parameters, terminated by a period or closed parenthesis. Some differences exist in the rules about separator characters and regarding the control language for NOS.

a. Under NOS, a $ preceding the initial keyword forces the system to not search the job's local file names for a match with the keyword, but rather search the system library immediately. Therefore, even if a local file of the same name is present, it will not be executed. Without the $, local files are searched first.

b. Parameter field separators in NOS/BE may be:

> any character with a display code value greater than 44 except *) $. and blank. The , and (are the preferred separators.

In NOS the valid separator characters are:

> \+ - = / = , (

and any character with a display code value greater than 44 except *) $. and blank.

c. NOS ignores any imbedded spaces except in literals. Under NOS/BE, a blank may serve as the separator after the initial keyword and is significant in literals, but is otherwise ignored. Blanks are preserved in comment fields which fall after the terminator character.

3.2 Control Card Flow

The control card flow for NOS and NOS/BE is similar in that the control card record is read sequentially by the system and each statement directs the next activity.

Both systems support CCL (Cyber Control Language).

Both NOS/BE and NOS have provisions for dropping into a separate stream of control cards within the control card record in the event of an abort by using the EXIT card. NOS/BE allows for options of the EXIT card - EXIT, EXIT(S), EXIT (C) and EXIT (U).

NOS provides a NOEXIT card which suppresses the transfer of control to the statement following the next EXIT card if an error occurs. The error processing can be resumed with the ONEXIT card.

Both systems allow comments on control cards, either on normal control cards following the terminator character or on a special COMMENT. card. NOS treats a card containing an * in column 1 as a comment card. In both systems, these comments are merely copied onto the job's dayfile and serve as information only.

C. CHARACTER SETS AND CONVERSION

1. Cards Read

The EOR card is the same in both systems. The NOS/BE EOF card corresponds to the NOS EOI card. They are used for the same purpose (deck terminators). NOS has an EOF card which is used as a file terminator as opposed to a deck terminator.

1.1 Coded

Under NOS and NOS/BE the conversion mode is either 026 or 029, depending upon an installation parameter. The user can select an alternate conversion mode in both cases.

1.2 Binary

Standard binary decks are the same. The NOS absolute binary corresponds to the NOS/BE free-form binary except in the way they are flagged in the deck. NOS identifies this type of binary with a card with 5,7,9 in column 1 and 4,5,6,7,8,9 in column 2. The binary deck is terminated with a similar card. NOS/BE identifies the deck with a card with all rows punched in column 1 and all rows punched in any other column with the remaining columns blank.

NOS/BE produces a message indicating a conversion mode change, but NOS doesn't.

2. Cards Punched

2.1 Coded

The keypunch mode under NOS/BE is either 026 or 029 depending on an installation parameter. Under NOS the keypunch mode depends on the job origin type. If the job is of local batch origin, decks are punched in the initial keypunch mode (e.g., mode specified on the job card or set by system default). For all other job origin types, decks are punched in the system default keypunch mode. However, the DISPOSE request allows the user to specify that decks be punched in either 026 or 029 mode, regardless of the job's keypunch mode.

2.2 Binary

No significant differences. Both standard binary and absolute/free-form are punched.

3. Printed Data

The following carriage control characters are available under NOS/BE, but not under NOS.

A - Page eject after printing.
B - Skip to last line after printing.
I-L - Skip to channels 7,8,9,10 respectively, after printing.
X-Z - Skip to channels 8,9,10 respectively, before printing.
9 - Skip to channel 7 before printing.
PM - Stop printing and display remainder of line on B display. To continue, the operator must type GO.

4. Character Sets Used

Under NOS/BE and NOS, the 026 and 029 character set can be set up for either 63 or 64 characters and they are identical.

D. CONTROL CARDS

This section covers control cards that are not covered elsewhere. Only NOS/BE control cards that do not exist under NOS or exist in a different format are discussed. The following compares standard NOS/BE and NOS control cards.

	NOS/BE	NOS
1.	EXIT.	
	EXIT(S)	EXIT.
	NOT AVAILABLE	NOEXIT.
	NOT AVAILABLE	ONEXIT.
	(C) (U)	NOT AVAILABLE
2.	LIMIT(N)	Not Available Limits are set through entries in the validation file.
3.	SWITCH(S)	SWITCH (S1, S2, ..., Sn) ONSW (S1, S2, ..., Sn) identical OFFSW (S1, S2, ..., Sn)
4.	COMBINE (file1, file2, n)	PACK (file1, file2, x)
5.	COMPARE (file1, file2, n, lev, e, r)	VERIFY (file1, file2, P_1, P_2, ...p_n) VFYLIB (file1, file2, NR)
6.	COPY (file1, file2)	COPY (file1, file2, x, c)
7.	COPYBCD (file1, file2, n)	This can be accomplished with COPYBF or COPYCF declaring the tape as NOS-coded and setting the record size to 148 characters.
8.	COPYBF (file1, file2, n) COPYBR (file1, file2, n) COPYCF (file2, file2, n) COPYCR (file1, file2, n)	COPYBF (file1, file2, n, c) COPYBR (file1, file2, n, c) COPYCF (file1, file2, n, fch, lch) COPYCR (file1, file2, n, fch, lch)
9.	COPYN (f, file, file1, ... file10) with directives: REWIND (file) SKIPF (file, n) SKIPR (file, n) WEOF (file)	LIBEDIT provides this feature under NOS

10.	COPYSBF (file1, file2)	COPYSBF (file1, file2, n)
11.	COPYXS (file1, file2, n)	NOS supports x format tapes
12.	DISPOSE (file x) DISPOSE (file, x=ky) DISPOSE (file, x=ky)	DISPOSE (file1=q_1, file2=q_2, ... file$_n$=q_n / ot-usernumber) OUT.
13.	SKIPF (file, n, lev, m) SKIPB (file, n, lev, m) BKSP (file, n)	SKIPF (file, x, m) SKIPFB (file, x, m) SKIPR (file, n, lev, m) BSKP (file, n, m)
14.	REQUEST, file, dt. REQUEST, FILE, APF, SN-setname, VSN=vsn.	REQUEST (file, CK) CB
15.	MODE(m)	MODE(m,n) CYBER 170 only
16.	SUMMARY.	SUMMARY. identical SUMMARY (OP=$p_1p_2...P_n$, JN=jobname,FN=file1,O=file2) SUMMARY ($p_1p_2...p_n$) same as ENQUIRE(...)

E. JOB SCHEDULING

NOS/BE and NOS define classes of jobs for scheduling purposes. NOS defines job classes or the basis of the following job origins: local unit record equipment, remote batch terminals, interactive terminals, or operator console. NOS/BE defines job classes on the basis of the following types of activity and speed of response required: express, batch, batch with tapes, interactive, graphics, multi-user jobs, and ECS jobs.

NOS/BE and NOS associate scheduling parameters with each class which can be modified from the console during normal system execution. This feature allows the system to be tuned in reaction to externally applied requirements. NOS defines three queues within each class: these queues are governed by the scheduling parameters and include the input queue, the rollout queue, and the output queue. NOS/BE does not distinguish between job classes except in considering jobs that use central memory. Its scheduling parameters define a set of priorities to control the activity of jobs in each class.

Once jobs have been scheduled for execution, both systems provide a minimum control point time (quantum for NOS/BE and CM and CPU time slices for NOS). The systems differ inasmuch as NOS/BE prevents swapout of jobs with non-mass storage equipment assigned. These jobs are only rolled out, freeing memory, but tieing up a control point and FNTs. NOS does not impose this restriction. As a result, it allows all jobs to be completely swapped out.

A few other important differences are mentioned below:

1. Input Queue Scheduling

In NOS, input queue scheduling is a part of the integrated scheduler. All jobs (with the exception of TRANEX "transaction" jobs) initially reach a control point by being scheduled from the input queue. In NOS/BE input queue scheduling is not done by the integrated scheduler and only batch jobs initially reach a control point by being scheduled from the input queue.

2. Tape Job Scheduling

NOS - No previewing of a job's tape requirements while it resides in the input queue. Operator does not control entry of tape jobs into the input queue.

NOS/BE - The operator may preview tape job requirements and control the entry of tape jobs into the input queue.

3. Swapping

NOS - Provision is made for a job being swapped out until an event occurs or a time period elapses. This provision may be used either by the system or the user job. Also jobs waiting for access to permanent files and unavailable tapes are swapped. In addition, TTY I/O causes swap out of jobs waiting for terminal I/O and does not put them in the swap queue until this I/O is completed.

NOS/BE - Jobs are either swapped from a control point or rolled out. A rolled out job remains at the control point but gives up all but 100B words of its field length. A job will be swapped unless operator action is required and/or non-allocatable devices are assigned to the job. In these cases, a rollout is performed.

A job is swapped or rolled out for the following:

1. Waiting for permanent file access
2. Waiting for a tape
3. Waiting for interactive terminal I/O
4. Waiting for operator action

4. Dependent Job Processing

NOS - Dependent job features are:

1. SUBMIT control card
2. Procedure files
3. Timed event rollout

NOS/BE - The job dependency feature allows the user to structure job dependent trees which are controlled either through control cards or macro calls.

F. SYSTEM INTERACTION

1. System Requests

RA+1 calls used for system requests are the same; however, the routines used for making the calls differ. NOS/BE routine CPC is called to make RA+1 calls. In the case of file action requests, all FET processing is done by CPC including handling of abnormal conditions and entry of an OWNCODE routine. If the FET is busy, the program is put into recall status. NOS routine SYS= is called to make RA+1 calls. It does not do any

FET processing. Special routines (CIO=, OPE=, LFM=, etc.) exist for this purpose. They check for busy and will put the program into recall status, but they do not handle abnormal conditions. This is up to the user. The routines call SYS= to make the RA+1 call.

2. User/System Communications

In both cases, RA through RA+101$_8$ is used for user/system communication. The differences are as follows:

- When parameters are stored, the delimiter is stored in the lower 6 bits of the word. NOS stores display code equivalent while NOS/BE stores code identifiers. On common products the NOS/BE convention is followed.

FILE PROCESSING

1. File Structure

Generally, file structures are the same under both systems:

BOI...EOR...EOF...EOI

Differences exist in the format of the latter three delimiters:

NOS/BE:

	EOR	EOF	EOI
Cards	7,8,9 Punch	6,7,8,9, Punch	6,7,8,9 Punch
Mass Storage	Short PRU with Lvl No. 0-16	Zero Length RPU with Lvl No. 17	End of Data
Tape	Short PRU with Lvl No. 0-16	Zero Length PRU with Lvl No. 17	Double EOF

NOS:

	EOR	EOF	EOI
Cards	7,8,9 Punch	6,7,9 Punch	6,7,8,9 Punch
Mass Storage	Short PRU linked to next PRU	Zero Length RPU w/link to next PRU	Zero length PRU with no link
Tape	Short PRU with Lvl No. 0 or Lvl No. 0-16 if format is I or SI	Zero Length PRU with Lvl No. 17	Double EOF

NOS supports level numbers on tape in that, when read, they are detected and removed.

2. File Definition and Usage

2.1 FET

The File Environment Table is a communications medium between a CP program and the system processors.

NOS/BE uses the FET for passing tape labeling parameters and for all I/O requests. All other file action requests use parameter lists, i.e., permanent file requests use the FDB.

NOS uses the FET for all file requests -- tape labeling, I/O, permanent file, etc.

2.2 CIO

Under both systems, CIO performs I/O requests. In general, they are the same, including function codes.

3. Sequential/Random Processing

NOS/BE provides the user with macros (WRITOUT, WRITIN, READIN) for maintenance of random file directories. They set up the directory, insert the random address in the FET and make the I/O request. NOS users must perform these functions. Similar macros do not exist. NOS/BE converts random files to sequential files when copying to tape. NOS maintains the random file structure even on tape.

4. Disk Pack Files

Under NOS/BE there exist public devices, which can be accessed by all jobs and can hold permanent files, and private device sets, which consist of a group of RMS devices and can hold permanent files. One of the devices in a set is designated as master and it holds all the tables related to that set.

Under NOS every permanent file a user creates resides either in his family of permanent file devices, which consists of 1 to 63 RMS devices, or on an auxiliary device. An auxiliary device can, but must not necessarily, be a disk pack that can be physically removed. An auxiliary device is a self-contained permanent file device and may be defined as public or private. Anyone permitted to use auxiliary devices who supplies the appropriate pack name can create, replace and access files on a public device. Only one user, the owner, can create and replace files on a private auxiliary device, but others may access those files as permitted by the owner.

MAGNETIC TAPE PROCESSING

1. Scheduling of Tapes

NOS/BE - The number of 7-track and 9-track tapes (and their varying densities) to be used concurrently is specified on the job card.

NOS - The number of 7-track and 9-track tapes to be used concurrently is specified on the RESOURC card, which may appear anywhere in the job stream.

2. Tape Labels

Both systems support labeled and unlabeled tapes. The ANSI X3.27-1969 tape label standard applies to both NOS and NOS/BE.

2.1 Trailer Labels

Trailer labels are always present on NOS/BE standard tapes and NOS I and SI tapes. NOS/BE S,L tapes and NOS F,S,L tapes also have compatible trailer label sequences.

2.2 Header Labels

NOS/BE and NOS header labels are the same except for the following fields:

VOL1 - Field 4 - Accessibility
NOS/BE does not use Field 4. NOS allows the user to enter this field during BLANK labeling. Field 4 controls future labeling of the tape.

VOL1 - Field 7 - Owner ID
User can supply up to 14 characters. If the user omits an entry in this field, NOS/BE inserts blanks while NOS inserts a family name and user number.

HDR1 - Field 7 - Generation Number
This field is not used by NOS/BE (blank). NOS allows the users to insert four characters: default is 0001.

HDR1 - Field 13 - System Code
Under NOS/BE, the default for this field is blank. Under NOS, the default is yyy-nn, where yyy is the version of the O/S and where nn is the EST ordinal of the unit on which the file was written. The EST ordinal is reported in read error recovery operations on NCS to aid in tracking inoperative tape drives.

2.3 Extended ANSI Label Processing

Extended ANSI label processing is the same at the user level under both systems.

2.4 3000 Series Labels

While 3000 series labels are supported by NOS/BE, they are not supported by NOS. Under NOS, 3000 series labels can be treated as non-standard labels and will be skipped.

2.5 End-of-Reel Processing (Write)

NOS provides a default end of reel option which is compatible with end-of-reel processing on NOS/BE. It also provides two other selectable end-of-reel processing options for compatibility with other systems. These additional options are not available on NOS/BE.

2.6 Termination Processing (Write)

Termination processing for write operations is the same under both systems.

3. Record Levels

The binary record terminator is different:

FIELD	NOS/BE	NOS
0 - 5	Level Number	Level Number
6 - 11	Zero	Zero
12 - 35	35522754B	PRU Number
36 - 47	5523B	PRU Byte Count

The coded record terminator is the same under both systems.

NOS/BE supports level numbers for standard binary and coded tape formats. On READ/WRITE operations the level number in the FET is used. NOS supports this same procedure for I and SI tape formats. For all other formats, level numbers do not exist or are written as zero and ignored when read.

NOS/BE uses level number 17 as an EOF. NOS also does, but only for I and SI tape formats.

4. Tape Formats

NOS/BE	NOS
Standard Binary	SI Binary (NOS/BE compatible)
Standard Coded	SI Coded (NOS/BE compatible-no longer supported)
S, L	S, L (NOS/BE compatible)
	I Internal Binary
	X External Binary (KRONOS 2.0 compatible)
	E Line Image Mode-Coded
	B Blocked Mode-Coded
	F Foreign - Mixed Mode

5. Parameter Definitions

Differences exist in tape format selection. Under NOS/BE, S,L formats are selected. If not given, standard format is assumed. Mode is always controlled by the mode flag in FET. Under NOS, one of eight formats (I,X,E,B,SI,S,L,F) is selected. The format dictates the mode except for SI,S,L in which case mode is controlled by the mode flag in the FET.

6. System Communication

Both systems use the FET to hold label information (NOS/BE REQUEST macro uses parameter list). The format is notably different.

7. Magnetic Tape Control Cards

7.1 Assignment of Magnetic Tapes

NOS/BE	NOS	(Parameters differ)
VSN	VSN	
LABEL	LABEL	
REQUEST	REQUEST	
	ASSIGN	

7.2 Multi-File Processing

NOS/BE	NOS	(Parameters differ)
LISTMF	LISTLB	
POSMF	POSHF	

8. Automatic Assignment of Tapes

Under NOS/BE and NOS, automatic assignment of tapes is requested by the VSN control statement and the VSN field in the file header information.

The NOS/BE operator can preview incoming jobs to determine whether tapes will be necessary. The NOS operator can view VSN information for the current requests after jobs have been event-rolled because the requested tape is not mounted. This prevents unnecessary job queuing and tape mounts for jobs which abort before tape usage or conditionally request tapes. There is no operational penalty for NOS for rolling out a job awaiting tapes since the control point is released for reassignment. Tape units are managed by MAGNET's control point which functions like a stack processor for tapes.

1. The following kinds of file access are supported by NOS and NOS/BE.

System	Supported File Access
NOS/BE	Direct access
NOS	Direct access and indirect access

Indirect files are customarily limited to small files (typically 64 PRUs) by user validation. When a user GETs an indirect file, NOS makes a local file copy. A REPLACE replaces the permanent file with the local copy. Although it is not apparent to the user, each user's indirect access files are maintained in what is physically one file; the user's indirect file can be allocated in units as small as a single PRU.

2. Permanent File Directories

NOS has no directories or catalogs. The system equates the user number to a user index which in turn points to the catalog in which the user's permanent files are defined. The user index is also used to subdivide catalogs and minimize search time. When a user requests access to a specific permanent file, the system searches permanent mass storage by looking for a specific permanent file name and user index.

NOS/BE maintains a permanent file directory or catalog for each device set. When a user requests access to a specific permanent file, the system searches for a match on permanent file name. The directory is divided into sub-directories and search time is minimized because the user supplies an owner identifier (ID). The ID is hashed and the resulting value identifies the associated sub-directory.

3. Permanent File Devices

NOS organizes users and their permanent files into mutually exclusive families. Each PF device must be a member of a family or a removable private, shared, or public pack. Although users from different families may share access to a removable pack, they cannot access each other's permanent files. NOS maintains up to eight separate PF directories or catalogs on each family according to a bit mask (device mask) set when the device is initialized. NOS equates each username to a user index which maps directly to a specific device and catalog track where the directory of the user's files is located. When a user requests access to a file, NOS goes directly to the proper device and catalog track to search. Since a user's catalog entries must all reside in a single catalog track chain, only a single track chain need be searched for the file name and user index match, thus providing quick access to the file. This multiple catalog technique is flexible enough to allow isolation and grouping of users within a family to manage permanent file space and enhance performance and security.

Permanent file devices are defined at deadstart time. The definitions are maintained in CMR (NOS/BE) or CMRDECK (NOS), or they can be entered from the central-site operator console.

Under NOS, definition of the EST entry specifies only whether the device is available for system files. Additional data either recovered from the physical device or entered from the console qualifies the device as a permanent file device and identifies residency of the catalogs. This data also defines residency of indirect and direct access files and whether or not the device is removable.

Under NOS/BE, all permanent file devices belong either to a public or private device set. The members of a public set can individually have system, permanent file and queue attributes, which are set during initial deadstarts. The public sets remain mounted at all times. Private sets are removable and can consist of a master device and optional member devices. When working with a private set, the master device must always be mounted; the member devices can be mounted and dismounted as needed.

3.1 Interchangeable Permanent File Devices

The availability of interchangeable permanent file devices allows all permanent file devices to be switched to an alternate system without interrupting system operation. This capability requires that the catalog/directory defining permanent files be available under an alternate system.

Permanent file ownership under NOS as defined by family and user name allow a site significant reconfiguration options. An entire permanent file family may be brought on or taken off line without a deadstart. One of the prime uses for this capability is in machine backup. If a machine goes down, its permanent file families may be brought on line on other mainframes with no conflict/overlap problems in user names, permanent file names, validation, etc. Some sites with multiple mainframes configure their workloads into at least two families on each mainframe so that it is possible to off-load each family to a different machine or to the same machine during different time periods based on available resources.

Under NOS/BE, interchangeable permanent file devices are available in a limited sense by using the device set concept. The default permanent file set cannot be easily interchanged, and there is not a correspondence between groups of users and particular device sets as there is in the NOS 'family' concept.

3.2 Removable Permanent File Devices

The availability of removable permanent file devices allows users to retain permanent files on devices other than those within the public permanent file system.

NOS

Under NOS, EST entries are made for removable devices (auxiliary devices) at deadstart. The entries can be logically turned off. Permanent file definitions such as packname, public, private, etc. are obtained from the device when it is mounted. A user name/password combination is associated with private packs; however, only the packname is assigned to public packs.

Users access removable packs via the packname which must appear in all permanent file requests or the PACKNAM control statement. If the user omits the packname, the system uses default family permanent files. Removable packs cannot be used for local (non-permanent) files. Removable packs can be scheduled by the user.

NOS also schedules removable packs in conjunction with tapes to prevent a deadlock condition. Device type parameters are provided on the RESOURC statement for this purpose. The handling of this feature is similar to tape drives in that the operator tape preview display informs the central site operator of requests.

NOS/BE

NOS/BE provides removable permanent file devices through the set concept. Under this concept, the system groups all rotating mass storage into one or more device sets (either public or private). Public sets can be defined at deadstart time. All public set member devices must be mounted at all times. The public sets are the default sets used for all types of mass storage requests. For example, the default PF set is accessed when a user accesses a permanent file but omits a setname. Public set attributes are system, default permanent file, queue and scratch.

A user can create his own private set and add or delete members. To access a set, the user must request that the master device (containing the directory) be mounted. This logical mount allows several jobs to access the same set simultaneously. The system controls access through permanent file passwords only. Users refer to the set by supplying the setname. The REQUEST control statement or REQUEST macro directs local files to the set. All permanent file commands can be used to manipulate private and public sets. If the user requests a set but omits a setname, the system makes the default set available. The user can request that members of a private set be mounted and dismounted. Automatic mount requests for set members are generated by the system when the user attempts to access a file residing on a device not mounted.

NOS/BE does not provide scheduling of private device sets.

3.3 Device Overflow

Under NOS, a file must reside entirely on a single logical device which may consist of up to eight physical drives. Although NOS does not provide for device overflow as NOS/BE does, the validation restrictions applied to users' mass storage utilization plus the automatic assignment of new files to the least full device may minimize occurence of device full conditions. Additionally, those jobs requesting a track on a full device do not abort but merely await an available track. Hopefully, some job will soon release a scratch file or the operator can dump some permanent files or queue files to allow the hung job to obtain tracks and complete processing. Failing these measures, the operator must abort the job.

4. Permanent File Identification

4.1 Permanent File Name

NOS/BE permanent file names are 1-40 alphanumeric characters and must be unique within all permanent files cataloged by the user (owner ID).

NOS names are 1-7 alphanumeric characters and must be unique within all permanent files created by the user (user number). The request is aborted if the name is not unique.

4.2 Logical File Name

Logical file names identify a file while it is assigned to a control point. NOS/BE names are 1-7 alphanumeric characters beginning with an alpha. NOS names are 1-7 alphanumeric characters with no restriction on the first letter.

4.3 Sub-Dividing Files

NOS/BE allows for cataloging up to 5 files (cycles) under a single permanent file name. Each cycle is identified by a unique number in the range of 1-999. NOS does not have this capability.

5. Permanent File Security

5.1 Owner Identification

The owner of a NOS/BE file is established by the owner ID which is entered by the user when the file is cataloged, attached or purged.

The user/charge number identifies the owner of a NOS file. It is entered at log-on time (terminal) or on the USER card (batch). An alternate user number may be specified when the file is accessed.

5.2 Passwords

NOS/BE provides for the definition of four passwords, each associated with a mode of access (READ, MODIFY, EXTEND, CONTROL). Access to the file is granted only in the mode associated with the passwords given. For example, a user can READ and EXTEND a file only if he gives the READ and EXTEND passwords when the file is attached. A fifth password can also be defined for additional security (TURNKEY). Although this password is not associated with a specific mode, the file cannot be accessed in any mode unless it is given. The owner of the file is not distinguished from other users and therefore must specify the desired passwords when accessing the file.

NOS provides for the definition of one password. It is not associated with a mode of access but all users, except the owner, must specify it when accessing the file.

5.3 Access Controls and Level of Access

To access a file under NOS/BE, the file name, owner ID, passwords and cycle (optional) are specified. As explained above, the passwords control the mode of access. The file owner must also specify these attributes.

Under NOS, the file category, as specified when the file is created, defines the information required to access the file:

PRIVATE — File name, user number, password. Explicit permission and mode of access must be given via the PERMIT command.

SEMI-PRIVATE — File name, user number, password. Explicit permission is not required, but a general mode of access may be defined. The system records in the owner's catalog the user number of each user who accessed the file, the number of accesses and the date and time of the last access.

PUBLIC — File name, user number, password. No explicit permission. The system records the number of times the file was accessed but does not record user numbers or the last access date and time.

The mode of access is defined when the file is created or when explicit permission is given. A mode may be specified when the file is accessed. It must be the same or a subset of the mode for which permission has been granted.

The owner of a file is not restricted by the above controls. Only the file name is required to access the file; however, he may specify a mode of access to control file activity while attacheo.

The file or user permission modes are:

WRITE — Allows the user to write, read, append, execute, modify and/or purge the file (direct and indirect access files).

MODIFY — Allows the user to modify, append, read and/or execute a direct access file.

APPEND — Allows the user to append information to the end of the file (direct and indirect access file).

READ — Allows the user to read and/or execute the file (direct and indirect access file).

READMD — Allows the user to reao and/or execute the file while another user may currently be accessing the same file in MODIFY mode (direct access file only).

READAP — Allows the user to read and/or execute the file while another user may currently be accessing the same file in APPEND mode (direct access file only).

EXECUTE	Allows the user to execute the file. The file must be in absolute format (direct and indirect access or relocatable file).
NULL	Removes permission previously granted (direct and indirect access file).

5.4 Multi-User Access

NOS/BE provides the following types of multi-access:

Multi-Read

Multi-Modify (installation option).

Multi-Read with Single Extend or Modify.

Under NOS, indirect access files allow more than one user to access a file concurrently. This capability is facilitated by the fact that each user accesses a working copy of the file. However, the system does not provide interlocks for replacement and append on indirect files.

NOS direct access files can be accessed currently by multiple users quite similarly to NOS/BE's permanent files, but there are the following differences:

1) Under NOS, "read" and "read and allow modify" are separate modes. Thus, the user accessing the file only for reading can control whether concurrent modification access is allowed. The NOS/BE user accessing a file only for reading cannot prevent concurrent modification without preventing all modes of concurrent access.

2) An installation option under NOS/BE will allow concurrent (multiple) modifiers. NOS does not support a similar capability.

6. Parameter Definitions

The following parameters are only available under NOS/BE:

RP = Retention period
AC = Account Number (not used at UW)
FO = Validity check for Direct Access-Index Sequential Files
CY = Cycle number
LC = Lowest cycle
SN = Setname

7. Permanent File Control CARDS

7.1 Create

NOS/BE: CATALOG
NOS: DEFINE, SAVE

7.2 Access

NOS/BE: ATTACH
NOS: ATTACH, GET, OLD

7.3 Modify

NOS/BE: ALTER, RENAME, EXTEND
NOS: APPEND, CHANGE, PERMIT, REPLACE

7.4 Purge

NOS/BE: PURGE
NOS: PURGE, PURGALL

7.5 Statistics

NOS/BE: AUDIT
NOS: CATLIST

8. Permanent File Macros

All NOS/BE control cards are available as macros. In addition, the PERM macro enables a user to determine what permissions have been granted to a file and whether or not the file is permanent. All NOS control cards except PURGALL are available as macros. In addition, ALTER and PERM are available as functions (RA+1 calls coded by user - no macros): PFC and PRM.

9. Permanent File Utilities

NOS/BE:	DUMPF	NOS:	PFDUMP
	LOADPF		PFLOAD
	TRANSPF		PFCAT
	AUDIT		PFATC
			PFCOPY

DUMPF and TRANSPF are saved as permanent files and are available to users that know the appropriate passwords. LOADPF and AUDIT are retained on the system library and are available for all users. AUDIT is restricted to just the user's files.

NOS utilities are available only to SYSTEM origin jobs. This means that they can be called from the console or by users that have system origin privileges (entry in validation file).

9.1 PFDUMP and PFLOAD Parameters

Pi		Description	
FM	=	Family name	
PN	=	Pack name	
DN	=	Device number	
T	=	Archive file name	
LO	=	T	Files processed
		C	Catalog files
		E	Errors
L	=	Output file name	
OP	=	C	Creation
		A	Last access
		M	Last modification
		I	Indirect access files
		D	Direct access files
		B	Before time and date
		P	Purge after dump
NT	=	Nine track	
DE	=	Density	
NR	=	No rewind	
NU	=	No unload	
SF	=	Number of files to skip	
DT	=	Date	
TM	=	Time	
UI	=	User index	
PF	=	Permanent file name	
VF	=	Verify file name	
V	=	Verify	
UN	=	User number	

9.2 DUMPF Parameters

MO=n Dump mode:

n	Mode
1	Back-up mode. Permanent file tables and all associated mass storage space are intact. RD password required. Default.
2	Archive dump. Mass storage space is released, but permanent file tables remain with the files marked as being on an archive tape. MD password required.
3	Destructive dump. All permanent file tables and mass storage spaces are released as the files are dumped. CN password required. The central site operator receives notification when a mode 3 dump is attempted and must authorize continuance of the dump.

I=lfn_1 Logical file name of directive file for MO=1 dump; 1-7 letters or digits beginning with a letter. All other parameters except MO, SN, CL, and PW are ignored. If lfn_1 is not specified, directives for MO=1 are on INPUT.

LF=lfn_2 Output listing file. Default is OUTPUT.

CL Complete list option selected. All files in the permanent file directory are listed. If CL is omitted, information is listed only for files which are dumped.

DP=a Dump type:

a	Type
A	All files meeting criteria of other parameters. Default.
X	All files meetings criteria parameters only if their expiration dates are equal or less than current date.
C	All files meeting criteria of other parameters only if they have been modified, renamed, created, or extended since the last DP=C dump.

ID=name Dump files with this owner.

PF=pfn Dump files with this permanent file name. ID should be specified also; if it is not specified, ID=PUBLIC is assumed.

CY=cy Dump cycle cy of file identified by PF and ID. CY is ignored and the dump continues if this cycle is not found of if PF and ID have not also been specified.

SN=setname Dump files from device set with this name; 1-7 letters or digits beginning with a letter.

VSN=vsn Dump files from this device of device set specified by SN; 1-6 letters or digits with leading zeros assumed. VSN is ignored if SN is omitted.

IN=ddd Dump files inactive this number of days; 1-3 digits. Can be qualified by a TI parameter.

JN=yyddd — Dump files inactive since this ordinal date; 5-digit ordinal date format. Can be qualified by TI parameter.

LA=minddyy — Dump files not attached on or after this date; 6-digit month-day-year format. Can be qualified by TI parameter.

DA=yyddd — Dump files created, modified, renamed, or extended after this date; 5-digit year-and-day-of-year format. Can be qualified by TI parameter.

CD=mmddyy — Dump files created, modified, renamed, or extended after this date; 6-digit month-day-year format. Can be qualified by TI parameter.

TI=hhmm — Time qualifier for date parameters; 4-digit 24-hour clock format. If date parameters are not specified, TI is ignored.

9.3 LOADPF Parameters

LP=x — Files to be loaded:

x — Significance

A — Load all files. Existing files are not replaced unless the file is incomplete or not disk resident. Default.

R — Replace existing files. Both X and R can be specified in the form LP=X,R.

P — Load archived files (files with entries in permanent file tables but file residence on tape).

X — Do not load expired files.

O — Permanent file dump tape is in SCOPE 3.2 or 3.3 format. If LP=O is not specified, the tape is assumed to be a SCOPE 3.4 permanent file dump tape. The O option can be used with other LP parameters in the form LP=R,O,X.

LF=lfn$_1$ Name of the file on which listing is to appear, 1-7 letters or digits beginning with a letter. Default is OUTPUT.

CL Complete list option selected. All files on the dump tape are listed. If CL is omitted, only loaded files are listed.

SN=setname Name of device set to which files are loaded, 1-7 letters or digits beginning with a letter. Master device of this set must be previously mounted.

VSN=vsn Volume serial number of the device onto which permanent files are loaded, 1-6 letters or digits with leading zeros assumed. Parameter SN must also be included, and the master device of the set must be previously mounted.

ID=name Load files with this owner.

PF=pfn Load files with this permanent file name. ID=owner is also required.

CY=cy Load cycle cy of file specified by PF and ID. CY is ignored and the load continued if this cycle is not found, or if PF and ID are not specified.

I=lfn$_2$ Logical file name of directive file, 1-7 letters or digits beginning with a letter. If I is specified but not equivalenced, file INPUT is used.

PW=pw EX password for DUM.

IN=ddd Load files inactive this number of days; 1-3 digits. Can be qualified by a TI parameter.

JN=yyddd Load files inactive this number of days; 1-3 digits. Can be qualified by a TI parameter.

LA=mmddyy Load files inactive since this ordinal date; 5-digit ordinal date format. Can be qualified by TI parameter.

LA=mmddyy Load files not attached on or after this date; 6-digit month-day-year format. Can be qualified by TI parameter.

DA=yyddd Load files created, modified, renamed, or extended after this date; 5-digit year-and-day-of-year format. Can be qualified by TI parameter.

CD=mmddyy	Load files created, modified, renamed, or extended after this date; 6-digit month-day-year format. Can be qualified by TI parameter.

TI=hhmm	Time qualifier for date parameters; 4-digit 24 hour clock format. If date parameters are not specified, TI is ignored.

LOADER

1. General Description

NOS/BE and NOS both use the CYBER Loader. Therefore relocatable loading, overlay generation and segmentation are the same.

If the program does not use specific system features (macros, RA+1 calls etc.), it should be possible to load the relocatable binaries under either system. For safety reasons, however, it is recommended to re-assemble or compile the program.

2. Libraries

Under NOS/BE, user libraries can be generated and maintained by the EDITLIB program. The libraries can be used for satisfying external during relocatable loading and as a source of relocatable and absolute programs for selective loading.

Under NOS, user libraries can be generated by LIBGEN and maintained by LIBEDIT. They are used for satisfying externals during relocatable loading and as a source of relocatable programs for selective loading.

LIBEDIT can be used to create and maintain a random file of relocatable and absolute programs; however, there is no provision for selective loading of absolute programs. The user would have to do a GTR (GET RECORD) to get the specific program and then do a load.

The directives for EDITLIB and LIBEDIT are totally different.

LIBEDIT files are just random files containing records of any type, indexed by name and type. This type of library is also used by MODIFY and recognized by GTR.

LIBGEN files can meaningfully contain only relocatable subroutine and capsules. LIBGEN files have both an entry point index at the front and a LIBEDIT-type index at the end.

To put things in perspective, the NOS deadstart tape is just one big LIBEDIT library copied to tape. It contains several LIBGEN libraries (FORTRAN, SYSIO, etc.) as components.

LIBEDIT

The LIBEDIT program, in most basic form, functions like COPYL (with maintenance of the random index added); but LIBEDIT accepts directives for insertion, deletion, reordering, etc. and is MUCH more powerful. LIBEDIT maintains the deadstart tape.

LIBGEN

LIBGEN maintains the only type of libraries recognized by the CYBER Loader. Unlike NOS/BE EDITLIB, LIBGEN can only build a library from scratch; it cannot modify one. Modifying a library is generally done by using LIBEDIT to build a new file with programs replaced or added, then calling LIBGEN to build yet another file in library form.

SYSEDIT

The NOS SYSEDIT program modifies the running system. Individual PP and absolute CP programs can be replaced, but individual routines in relocatable libraries such as SYSIO cannot. The entire library must be replaced. SYSEDIT has no facility for generating a deadstart tape from the running system.

Global Library Sets

The LIBRARY control card can be used to establish a global library set under either NOS or NOS/BE; but the global library set is not searched under NOS when processing control cards. Global library sets under NOS are good only for supplying subroutines or capsules.

Relocatable Loading

Functioning of the CYBER Loader is essentially similar on both systems. Library searching is significantly faster under NOS because of the difference in library structure.

Absolute Program Loading

CYBER Loader under NOS loads absolute programs from files just fine. It can't load them from libraries because LIBGEN doesn't put their entry points in the index.

Overlay Loading

Overlay generation functions similarly under both systems. There are differences in the LDV PP call to load overlays. The new "Fast Overlay Loading" system functions identically on both systems.

Segments and Capsules

Segment and Capsule loading were first introduced to NOS by the CYBER Common Loader and are therefore compatible.

3. Control Cards

As the loader is the same, the same control cards are used for loader processing.

4. Debugging Aids

The DMP control card/macro under NOS/BE dumps control point area, exchange package, CM (relative/absolute).

The NOS DMP control card/macro dumps relative CM (in octal or octal/display code equivalences) or exchange package.

Under NOS/BE and NOS TRAP (TRACK/FRAME) is provided for snapshot dumps (CYBER Loader feature).

CHECKPOINT/RESTART

1. Description

Under both systems the checkpoint/restart feature allows a job to be restarted at some intermediate point of execution.

2. Checkpoint Media

NOS/BE uses magnetic tape, with dumps written immediately after the previous dump (multiple dumps). A specific dump can be selected for restart.

NOS uses either tape or mass storage. Dumps can be written at beginning of information, end of information (multiple dumps) or alternate between two mass storage files. Specific dumps can be referenced by number during restart.

3. Checkpoint Control Card

NOS/BE checkpoints all local files.

NOS will checkpoint all local files or only those named in the parameter list.

4. Checkpoint Operator Command

Same. All local files are included in the checkpoint.

5. Checkpoint Macro

The parameter list structure is the same. The files given are copied to the checkpoint dump in the manner specified.

NOS/BE copies only mass storage files. Tapes are repositioned at restart time. The disposition of up to 42 mass storage files can be specified.

NOS will copy both mass storage and tape files to the checkpoint dump. The disposition of up to 77 files can be specified. A fifth disposition code is available to indicate that only positioning information for the file is to be retained. It is not copied.

6. RESTART Control Card

The restart procedure is the same but the options available are different:

NOS/BE:	NOS:
- Checkpoint file name.	- Checkpoint file name.

NOS/BE:	NOS:
- Checkpoint number.	- Checkpoint number.
	- Use INPUT file from checkpoint or current INPUT file. - Do not abort if all files needed are not available. - If file exists, the copy in the dump is not used.

TERMINAL SUPPORT

The communications subsystems available under NOS/BE and NOS are:

NOS/BE

INTERCOM - All standard NOS/BE communications facilities are considered part of INTERCOM, as all communications drivers operate under the single INTERCOM subsystem. Currently supported multiplexors include:

2550 Communication Processor (TTY, UT200, HASP, BYSYNCH) - INTERCOM 5

NOS

Two subsystems handle communications. IAF processes TTY or interactive communications. RBF supports Remote Job Entry.

a. IAF - 2550 Communication Processor

b. RBF - 2550 Communication Processor (UT200, HASP)

1. Low Speed Interactive Terminals

NOS/BE supports the low speed terminals along with the voice grade batch terminals under a common sub-system, while NOS supports these two classes of terminals under separate sub-systems.

1.1 Terminal Types

Both support ASCII terminals (including block mode) with standard and APL printing. NOS additionally supports correspondence code terminals.

1.2 Login Procedures

On NOS/BE, terminals specify a "username" and "password" at login. NOS terminals specify a "user number" and "password" at login.

For the NOS user facilities are available to enter a charge number via the CHARGE command.

Both NOS and NOS/BE compare the user number and password against the system file for validity. Also, the matching entry contains certain information about this user's attributes and privileges.

1.3 Terminal Control

1.3.1 Terminal Oriented Controls

Both have equivalent functions for line delete, character delete, interrupting output, aborting program, paper tape input and end of input line.

NOS additionally has provisions for text mode, transparent input, binary input, ASCII mode (128 character set), echoplexing, output character parity control and dynamic rubout count control (pad characters between lines of output).

1.3.2 Commands

Although there exists some similarity between the NOS/BE INTERCOM commands and NOS time sharing commands, there is frequently not a direct relationship. NOS commands are more "typically" time sharing oriented.

Both systems provide access to nearly all of the batch control statements.

1.4 Character Sets

NOS/BE provides several character sets:

- 64 character set where ASCII control codes are stripped out or interpreted and all lower-case ASCII characters (140_8 and above) are folded into corresponding upper case.
- ASCII-128 character set. Some ASCII control codes are stripped out or interpreted. All other ASCII characters are delivered to the program in a 7-bits-in-12 format (all characters take 12 bits in memory). No NOS equivalent.
- ASCII-256 character set. All ASCII characters are delivered to the program with all 8 bits intact. No interpretation is done. NOS "binary" is similar but not identical.

NOS provides several modes in which a user may operate:

- Normal code - 61 character set on input, full 128 on output.
- ASCII - Nearly full 128 character set on input (except for certain control characters like LF, CR, etc.), full 128 on output.
- Transparent - Full 128 character ASCII on input and output.
- Binary - Full 8 bit binary transparent on input and output.

The ASCII mode is an escape code convention where some characters are represented by a single six-bit byte and others are represented by two six-bit bytes. Product set support of this format is limited. NOS has conversion tables, in addition to ASCII, for correspondence code, correspondence APL code and Memorex APL code.

1.5 Submitting Batch Jobs

Both NOS/BE and NOS allow local files to be routed to the I/O queues. NOS/BE allows users to retrieve the files associated with their terminal ID from the output queue. NOS has no such facility. NOS has a submit command which simplifies creating a job from a terminal.

1.6 Miscellaneous

o NOS/BE provides a form of carriage control for output. NOS has no equivalent.

o NOS provides means for monitoring one terminal with another terminal. NOS/BE has no equivalent.

o Central site operator may send messages to terminals under both systems, either individually or to all logged on terminals (broadcast). NOS additionally has welcome message at log-on that can be changed; NOS/BE can call a system bulletin.

o NOS has a disable terminal control feature to limit the interrupt control user has while interacting with a program. NOS/BE has no equivalent.

2. Voice Grade Batch Terminals

The single most obvious difference between NOS/BE and NOS in this area is that NOS/BE provides both batch and interactive access from Mode 4 (User 200) terminals while NOS provides only batch.

2.1 Terminal Types

NOS/BE supports both the BCD and ASCII UT200 while NOS supports only the BCD version. Both also support HASP multileaving protocol terminals.

2.2 Terminal Control

Equivalent, except that NOS/BE provides "output paging" screen control, as data may be displayed on the screen.

2.3 Batch Processing Commands

Roughly equivalent. NOS/BE allows card deck to be read into a local file rather than just being submitted as a batch job. NOS/BE also allows output files to be evicted.

NOS RBF supports plotters as well as readers, printers, and card punches.

2.4 Status Display

NOS/BE H display allows user to select queue type for display. NOS has B display showing system activity, which is not available under NOS/BE.

2.5 Interactive Processing

NOS/BE users must connect files to read and write at the terminal. In NOS the files INPUT and OUTPUT always do I/O to the terminal.

NOS/BE programs must flush output buffers by calling CIO before reading from the terminal. NOS checks the output buffer and empties it if an input request is made.

NOS/BE sends all Dayfile messages to the terminal and they may be interspersed with program output. NOS sends only the last and it is sent after all other output. NOS has a facility to list the Dayfile of the terminal session. NOS/BE has no such feature.

NOS has a primary file. Lines entered which begin with a numeric are written to this file. Lines beginning with alpha characters are treated as commands. The primary file is sorted and duplicate line numbers eliminated allowing fast editing. NOS/BE has no such feature.

3. Network Products

Network Products provide support for a large variety of terminals, with terminal type being largely transparent to the application program.

Network Products include facilities for generating Network Application programs, certain Network Applications are provided and these implement basic interfaces for the terminal users. The three major Network Application programs are:

3.1 IAF - Interactive Facility

Provides a versatile command structure, with the implementation of several subsystems to provide different timesharing environments. The subsystems provide command formats varying from simple to sophisticated, e.g. the FTNTS System. The timesharing FORTRAN subsystem requires the following commands to create and run a program:

NEW - to initialize the program file

AUTO - to place information in the file

RUN - to compile and execute the program

SAVE - to save a copy of the program

Other subsystems available are BASIC (Basic Language Programs), BATCH (for entering control cards as under Intercom), NULL (for performing simple file and editing operations), EXECUTE (for executing previously compiled programs).

3.2 RBF - Remote Batch Facility

RBF provides remote batch capabilities for the Network Access Method (NAM). This capability includes the transfer of batch files from terminal to host and back again, and also provides the user with a batch terminal command language.

The command language allows the user to control and status the flow of his batch files, and control the execution of his batch jobs. The user can control items such as file disposition to other users, repeat counts, job priority, etc.

3.3 TAF - The Transaction Facility

TAF provides a generalized transaction processing capability with the following features:

Multithread

High Volume Processing

TAF Data Manager

Total Extended

Terminal Independence

3.4 NAM - Network Access Method

NAM provides security at the network level before a user may access a host computer through the network. A user must be validated to use one of the Network Application Programs, eg TAF, IAF, RBF, etc.

NAM also provides a feature called the Interactive Virtual Terminal (IVT) and the Batch Virtual Terminal (BVT). This feature allows for more flexible terminal definition and increases the scope of terminals that can be supported by NAM.

3.5 NOS Multi-Mode System

3.5.1 NOS supports the following modes either concurrently or as necessary:

Local Batch

Remote Batch

Deferred Batch

Timesharing

Transaction

Network Processing

3.5.2 Timesharing provides support of the following facilities:

3.5.2.1 Support of up to 512 on-line terminals.

3.5.2.2 Support of interactive FORTRAN, COBOL, BASIC, APL, two text editors and one line editor.

3.5.2.3 Access to all batch capabilities via deferred batch mode.

3.5.2.4 Support of communication lines at standard speeds up to 9,600 BPS and inter-node links up to 19.2 KBPS.

3.5.2.5 Support for both synchronous and asynchronous terminals.

3.5.2.6 Support of APL terminals.

3.5.2.7 Extensive permanent file capabilities.

3.5.2.8 "Conversational" FORTRAN Extended and COBOL.

3.5.3 Transaction Processing provides support of the following facilities:

3.5.3.1 512 terminals through NAM.

3.5.3.2 Two (2) data management products are presently available TAF Data Manager and CINCOM's Total.

3.5.3.3 TAF runs without the need for a dedicated system.

3.5.4 A common system interface exists between all modes of the NOS operating system. No special conversion is required between tape files and disk files, and vice versa.

Timesharing
Remote Batch
Deferred Batch
Local Batch

Products
Files
System Commands
Data
Control Language
Magnetic Tapes

3.5.5 Primary/Secondary terminal files

A timesharing user can declare a file he wishes to work on, to be a Primary file. This allows the user to issue timesharing commands that will default to the file that is declared as the Primary terminal file.

All other files attached to a user's terminal are considered as Secondary terminal files and must referenced explicitly by the user when issuing commands.

3.5.6 Additional Sundry Features - Text Editing

3.5.6.1 Upper lower case text processing.

3.5.6.2 Text editing through batch.

3.5.6.3 Ellipsis in search strings.

3.5.6.4 String movement.

3.5.6.5 Merge secondary file into text at specified point.

3.5.6.6 Count occurences of string.

3.5.6.7 Format text for printing.

4. NOS vs NOS/BE Interactive FTN Programs

4.1 CONNEC/DISCON on NOS/BE has a mode option for ASCII which is not available with NOS. One must use COMPASS and include control bytes with every line to talk ASCII on NOS. If one sets the terminal with a COMPASS macro or before execution into the NOS "ASCII" character set mode the peculiar 6/12 'half-ASCII' character set results.

4.2 Since NOS/BE protects the interactive program from sending control information except when transparent mode has been invoked, the NOS interactive program by contrast must be careful not to inadvertently send control-bytes.

eg

```
          I=4
          print 101,I
      101 format (R2)
```

will log the terminal off.

The difficulty becomes exacerbated trying to control any intelligent (e.g. graphics) device.

4.3 Interrupt processing accepts "STOP" upon waiting for INPUT on NOS as a command to terminate the program. On INTERCOM the characters "STOP" were quite valid data.

4.4 Carriage-control was expected by INTERCOM and is not handled by NOS.

4.5 Type-ahead is not provided on NOS. Aside from the human convenience, this is a real problem for intelligent devices trying to transmit information before the system is ready to accept it.

4.6 An interactive program on NOS/BE could be sure of seeing an error message to the line of the dayfile. On NOS one sees only the last half dayfile message for some products this contains no information.

The order of appearance of information from multiple-connected files is random on NOS whereas on INTERCOM it was done in 'real-time' not as the program is rolled-out.

4.7 The INTERCOM write to a connected file was performed by CIO just as for any other device. On NOS, CIO calls IRO and output is part of the rollout process and copied from the rollout-file to the terminal until complete. Thus a different program design may be necessary with NOS relative to NOS/BE not to 'flog' the system to death rolling out for every write.

5. Editors

In NOS R4, two feature editors are supported: EDIT and XEDIT. In addition to these editors, the NOS Interactive Facility (IAF) supports certain in-line editing capabilities. These in-line editing capabilities will be extended in NOS R6.

NOS/BE supports one editor: EDITOR.

The following subsections describe the major capabilities available with these editors. Specific command capabilities are not discussed.

3.5.1 NOS EDIT and XEDIT

EDIT and XEDIT characteristics compare as follows:

1) Both editors are usable in batch and interactive mode.

2) Files can be either display code or 6/12 ASCII.

3) Both editors are context oriented, although a few line number command capabilities are available.

4) Both editors must be explicitly called and terminated. (However, XEDIT allows limited in-line editing as the user can enter multiple editing commands on the same line as the editor call itself).

5) Neither editor can intermix NOS commands with editing commands. (Certain permanent file commands can be entered under XEDIT.)

6) XEDIT allows multiple editing commands per line.

7) XEDIT users can edit multi-file or multi-record files. EDIT operates on a single record file only.

8) Text file line lengths are limited to 150 characters for EDIT and 160 characters for XEDIT. Lines greater than these limits are truncated.

9) XEDIT user can perform many sophisticated tasks with a minimum of keystrokes. EDIT commands on the average are longer than XEDIT commands.

10) XEDIT allows optional verification by the editor when the editing command actually changes a file. EDIT does not support such verification.

5.2 NOS In-Line Editing (Including NOS R6)

The following capabilities characterize the NOS in-line editor.

1. The in-line editor is a high performance editor intended for a casual user.

2) Only interactive mode editing can be performed.

3) Files can be both display code and 6/12 ASCII (8/12 in R5).

4) The in-line editor is oriented to line numbered files (file <u>must</u> be line numbered).

5) The editor operates in-line. As a result, a user does not need to differentiate editing and NOS commands.

6) Users can enter only a single editing command per line.

7) The editor operates on the primary file only (a single record file).

8) Text file line lengths are limited to 150 characters.

5.3 NOS/BE Editor

The following features are available under the NOS/BE EDITOR.

1) EDITOR is usable in interactive mode only.

2) It operates on display code files only.

3) This editor is oriented to line numbered files.

4) Users must explicitly call and terminate EDITOR.

5) Users can intermix NOS/BE and editing commands. (EDITOR passes unrecognizable (non EDITOR) commands to NOS/BE.)

6) Users can issue only a single command per line.

7) EDITOR operates or multi-record or multi-file files.

8) Text file lines can contain up to 510 characters (controllable by FORMAT specification).

COMPILERS AND OTHER PRODUCTS

1. Languages

1.1 FORTRAN

NOS/BE and NOS support the same FTN (V4 and V5) and MNF compilers. (MNF was developed at the University of Minnesota which currently runs NOS.) NOS has Time-sharing FTN (FTNTS) which is supported by Telex to provide fast easy program development.

The old RUN compiler exists at both NOS/BE and NOS sites which have no choice but to support it.

1.2 PASCAL

Both systems support the same PASCAL compiler.

1.3 COBOL

Both systems support the same COBOL compiler.

1.4 BASIC

Both systems support the same BASIC compiler. On NOS, BASIC runs as a subsystem of the interactive system, and programs can be entered and run without using an editor. In addition NOS supports permanent file commands from within a BASIC program.

1.5 ALGOL

Both systems support the same ALGOL compiler.

1.6 APL

NOS supports APL for both APL and ASCII terminals. NOS/BE support for the same APL is available by special arrangement.

1.7 SIMULA

SIMULA is supported under NOS by special arrangement.

1.8 PL/I

Both NOS and NOS/BE support a subset of PL/I.

1.9 LISP

LISP is available from the University of Texas for both systems.

2. Record Manager

Record Manager, and its replacements, Basic Access Method and Advanced Access Method, are supported by both systems.

3. UPDATE and MODIFY

UPDATE is supported on both NOS/BE and NOS. In addition, NOS has MODIFY which provides similar capabilities but has more features including program library utilities for merging or extracting decks.

4. Library Utilities

NOS/BE supports libraries through EDITLIB. NOS has several library maintenance programs:

LIBEDIT	- Edit a library - add/replace decks
LIBGEN	- Generate a library
GTR	- Extract records from a library

5. Applications Packages

Below is a list of products either available on both NOS/BE and NOS or just on NOS. Bear in mind that even though a product may be supported on either system, it may not cost the same to use. Control Data introduced usage pricing for certain applications at level 501. This could either raise or lower the price depending on the monthly usage.

Products available on both NOS/BE and NOS:

UPDATE
COMMON UTILITIES
LOADER
COMPASS
CRM
FORM
CCL
CID
SORT/MERGE
8-BIT SUBROUTINES
FTN 4
COBOL 4
COBOL 5
COBOL 4/5 CA
FTN 5
F45
BASIC
CRM

FACTORY FORMAT UTILITY
SYMPL
ALGOL-60
CYBER CROSS

CDCS
DDL
FDBF
QU3
DATA CATALOGUE 2
PL/I
IMSL
TOTAL UNIVERSAL 1
GPSS-V
APEX-III
APT IV 2
TOTAL/ATHENA 1
PERT/TIME 2
TOTAL UNIVERSAL 2
TIGS
UNIPLOT

PRODUCTS ONLY AVAILABLE IN NOS:

MODIFY
NAM
IAF
NETWORK UTILITIES
EXEDIT 3
RBF
TOTAL-EXTENDED 1
TOTAL-EXTENDED 2
PDS/MAGEN 1
SIMSCRIPT II.5
AD-2000
GTICES/STRUDL
GTTABLE
PROPLAN

SAMPLE JOBS
(from University of Washington)

1. FORTRAN batch compilation and execution, using IMSL library

```
         NOS/BE                                 NOS
-------------------------------    ----------------------------------

jobname.                           jobname.
ACCOUNT(acctnum,password)          USER(usernum,passwrd)
                                   CHARGE(chargnum,projnum)
FTN.                               FTN.
PUBLIC(INSLPTN)                    PUBLIC(IMSLFTN)
LDSET(LIB=IMSLFTN)                 LDSET(LIB=IMSLFTN)
LGO.                               LGO.
7/8/9                              7/8/9
  FORTRAN program                    Fortran program
7/8/9                              7/8/9
  data                               data
6/7/8/9                            6/7/8/9
```

2. FORTRAN batch compilation and execution; data from permanent file

```
         NOS/BE                                 NOS
-------------------------------    ----------------------------------

jobname.                           jobname.
ACCOUNT(acctnum,password)          USER(usernum,passwrd)
                                   CHARGE(chargnum,projnum)
ATTACH(CLASS,ID=EE374)             ATTACH(CLASS/UN=EE374) or
                                   GET(CLASS/UN=EE374)
MNF.                               MNF.
LGO(CLASS)                         LGO(CLASS)
7/8/9                              7/8/9
  Fortran program                    Fortran program
6/7/8/9                            6/7/8/9
```

3. SPSS batch execution; data from 7-track labeled tape

```
         NOS/BE                                 NOS
-------------------------------    ----------------------------------

jobname.                           jobname.
ACCOUNT(acctnum,password)          USER(usernum,passwrd)
                                   CHARGE(chargnum,projnum)
LABEL(TAPE,VSN=xxxx,R)             LABEL(TAPE,VSN=xxxx,PO=R)
COPYBF(TAPE,MYDATA)                COPYBF(TAPE,MYDATA)
RETURN(TAPE)                       RETURN(TAPE)
REWIND(MYDATA)                     REWIND(MYDATA)
SPSS(D=MYDATA)                     SPSS(D=MYDATA)
7/8/9                              7/8/9
  SPSS control statements            SPSS control statements
6/7/8/9                            6/7/8/9
```

4. Create a permanent file from cards and attach it in a later job.

```
         NOS/BE                               NOS

jobname.                          jobname.
ACCOUNT(acctnum,password)         USER(usernum,passwrd)
                                  CHARGE(chargnum,projnum)
REQUEST(MYFILE,*PF)
COPYBF(INPUT,MYFILE)              DEFINE(MYFILE)    COPYBF(INPUT,MYFILE)
                                                 or
CATALOG(MYFILE)                   COPYBF(INPUT,MYFILE)  SAVE(MYFILE)
7/8/9                             7/8/9
  data file to be                   data file to be
  cataloged                         cataloged
6/7/8/9                           6/7/8/9

jobname.                          jobname.
ACCOUNT(acctnum,password)         USER(usernum,passwrd)
                                  CHARGE(chargnum,projnum)
ATTACH(MYFILE)                    ATTACH(MYFILE) or GET(MYFILE)
  .                                 .
  .                                 .
  .                                 .
```

5. Request a 7-track unlabeled Stranger tape for input and a 7-track labeled tape for output

```
         NOS/BE                               NOS

jobname,MT2.                      ¨obname.
ACCOUNT(acctnum,password)         USER(usernum,passwrd)
                                  CHARGE(chargnum,projnum)
                                  RESOURC(MT=2)
REQUEST(OLD,VSN=xxxx,S)           LABEL(OLD,VSN=xxxx,F=S,LB=KU)
LABEL(NEW,VSN=yyyy,               LABEL(NEW,VSN=yyyy,W,PO=W)
W,RING,PW=zzz)
  .                                 .
  .                                 .
  .                                 .

                                  NOS has no equivalent for UW
                                  tape password
```

6. Interactive program modification and compilation

```
         NOS/BE                               NOS

cr                                cr
LOGIN,acctnum,passwrd             family,usernum,passwrd
                                  BAT
```

```
UEDIT,CREATE                   NEW,MOD
ADD                            TEXT
%*IDENT MOD1                   *IDENT MOD1
*/THIS IS AN UPDATE MODE       */THIS IS AN UPDATE MOD TO
  TO DECKA                       DECKA
*D DECKA.24                    *D DECKA.24
*I DECKA.63                    *I DECKA.63
   TW2=0.                        TW2=0.
*C DECKA%                      *C DECKA
                                ETX (use ETX key or
                                control-C to terminate TEXT)
SV,MOD,P                       PACK
END                            SAVE,MOD
ATTACH,OLDPL,MYPL              GET,OLDPL=MYPL
REQUEST,NEWPL,*PF
CONNECT,OUTPUT
UPDATE,Q,I=MOD,L=4             UPDATE,Q,I=MOD,L=4
CATALOG,NEWPL,MYNEWPL          SAVE,NEWPL=MYNEWPL
FTN,I,L=LIST                   FTN,I,L=LIST
BATCH,LIST,PRINT,MYNAME        DISPOSE,LIST=PR
LOGOUT,Q                       BYE

(output will be identified     (output will be identified by
as MYNAMxx)                    system-generated jobname
                               based on user number)
```

7. Creation and execution of small interactive FORTRAN program

```
            NOS/BE                                   NOS
___________________________           ______________________________

PLEASE ENTER LOGIN                    USER NUMBER: usernum,passwrd
login,acctno,passwrd                  TERMINAL:  NN, TTY
10/26/78 LOGGED IN AT                 RECOVER/USER ID: id
   10.11.31 with user-id              N ftnts,new,npwr2
   G4 equip/port 13/005
COMMAND-connect input
COMMAND-connect output
COMMAND - uedit create
BEGIN TEXT EDITING                    T  auto
?a                                    00100 program npwr2 (input,
                                        output)
ENTER TEXT                            00110 5 print *, "enter number please:"
/;program npwr2 (input,output)        00120 read *,r
5;print *," enter number please:"     00130 rsw = r*r
;read *,r                             00140 print *,r," squared ",rsq
;rsw = r*r                            00150 go to 5
;print *,r," squared ",rsq            00160 end
;go to 5                              00170 (escape) *DEL*
;end/                                 (return)
READY
?rs /rsw/,/rsq/                       T 00130 rsq = r*r
?1*                                   lnh
```

```
    PROGRAM NPWR2 (INPUT,OUTPUT)
5   PRINT *," ENTER NUMBER PLEASE:"
    READ *,R
    RSQ = R*R
    PRINT *," SQUARED ",RSQ
    60 to 5
    END
-END OF INFORMATION-
?sy npwr2,o,p
FILE SAVED
FILE CATALOGED
CYCLE NO. 1
?end
END TEXT EDITING
COMMAND - rewind lgo
COMMAND - mnf,i-npwr2,k
    TIME USED = 0.055 CPU SECONDS
    CORE USED = 044000B OCTAL WORDS
COMMAND - lgo
  ENTER NUMBER PLEASE: -1.01
  -1.010000000  SQUARED  1.02010000000
  ENTER NUMBER PLEASE: %@
USER ABORT
COMMAND - logout,q
```

```
00100 PROGRAM NPWR2 (INPUT,OUTPUT)
00110 5 PRINT *," ENTER NUMBER PLEASF·"
00120 READ *,R
00130 RSQ = R*R
00140 PRINT *,R," SQUARED ",RSQ
00150 GO to 5
00160 END

T rnh
  ENTER NUMBER PLEASE:
I  -1.01
  -1.01 SQUARED 1.0201
  ENTER NUMBER PLEASE:
I stop
  *TERMINATED*
* save
T save
T bye
```

Control Data Publication	Publication No.
Network Products Transaction Facility Version 1 CYBER Record Manager Data Manager Reference Manual	60456710
Network Products Network Access Method Version 1 Reference Manual	60499500
Network Products Network Access Method Version 1 Internal Maintenance Specification	60490110
Network Products Remote Batch Facility Version 1 Reference Manual	60499600
NOS Version 1 Installation Handbook	60435700
NOS Version 1 Operator's Guide	60435600
NOS Version 1 Reference Manual Volume 1	60435400
NUS Version 1 Reference Manual Volume 2	60445300
NOS Version 1 System Maintenance Reference Manual	60455380

NOS Version 1 System Programmer's Instant	60449200
NOS Version 1 Time-Sharing User's Reference Manual	60435500
NOS Version 1 Export/Import Reference Manual	60436200
TAF/TS Version 1 Reference Manual	60453000
TAF/TS Version 1 User's Guide	60436500
TAF/TS Version 1 Data Manager Reference Manual	60453100
TAF/TS Version 1 CYBER Record Manager Data Manager Reference Manual	60456700
6400/6500/6600 Computer System Reference Manual	60100000

P. RELATED PUBLICATIONS

For further information concerning CYBER 170, CYBER 70,and 6000 Series Computer Systems, the NOS time-sharing systems, and the user interface for NOS, consult the following manuals.

Control Data Publication	Publication No.
CYBER 170 Computer Systems Reference Manual	60420000
CYBER 170 Computer Systems Models 720, 730, 750, and 760 Model 176 (Level B)	60456100
CYBER 70/Model 71 Computer System Reference Manual	60453300
CYBER 70/Model 72 Computer System Reference Manual	60347000
CYBER 70/Model 73 Computer System Reference Manual	60347200
CYBER 70/Model 74 Computer System Reference Manual	60347400
Modify Reference Manual	60450100
Network Products Interactive Facility Version 1 Reference Manual	60455250
Network Products Transaction Facility Version 1 Reference Manual	60455340

Network Products Transaction Facility Version 1 User's Guide	60455360
Network Products Transaction Facility Version 1 Data Manager Reference Manual	60455350

6532B
AL

Document Title: Permanent File Commands
Comparison Table

Source of Material: Technical Information Bulletin (TIB-85)

Supplies and Author: User Services
Academic Computing Center
University of Virginia

Date of Publication: 5/11/79

Editing: Document reformatted; local references removed; page numbers checked against current NOS manual.

Comments: This table is highly dependent on the revision level of the users manual to be accurate.

PERMANENT FILE COMMANDS

COMPARISON TABLE

NOS/BE COMMAND	NOS EQUIVALENT: DIRECT ACCESS	NOS EQUIVALENT: INDIRECT ACCESS	NOS MANUAL PAGE REFERENCES	ACTION
ATTACH	ATTACH	GET	1-8-07 1-8-14	Access contents of a permanent file
CATALOG	DEFINE	SAVE	1-8-12 1-8-18	Create a permanent file
PURGE	PURGE, PURGALL	PURGE, PURGALL	1-8-17 1-8-16	Remove a permanent file from the system
RENAME	CHANGE	CHANGE	1-8-11	Alter parameters associated with a permanent file
EXTEND	Position and write at EOI	APPEND	1-8-7	Add information to the end of a permanent file
AUDIT	CATLIST	CATLIST	1-8-9	Generate list of all permanent files in a user's library
ITEMIZE	ITEMIZE, CATALOG	ITEMIZE, CATALOG	1-14-3	List information about the structure of a file
COMBINE	PACK	PACK	1-7-27	Remove logical record boundaries from a file
Purge file and re-catalog	Position and write at BOI	REPLACE	1-8-18	Replace the contents of a permanent file

Document Title: Control Card Comparison Table

Source of Material: Technical Information Bulletin (TIB-86)

Supplies and Author: User Services
Academic Computing Center
University of Virginia

Date of Publication: 5/16/79

Editing: Document reformatted; local references removed; page numbers checked against current NOS manual.

Comments: The accuracy of the page references is dependent on the revision level of the users manual. The user can however update the table with his/her own manual.

CONTROL CARD COMPARISON TABLE

NOS/BE command	NOS equivalent(s)	References, NOS Reference Manual V.1	Action
ALTER	No equivalent under NOS	No equivalent under NOS	Change contents of a permanent file
ATTACH	ATTACH, GET	1-8-7 1-8-14	Access a permanent file
AUDIT	CATLIST	1-8-9	List information about a permanent file
BEGIN	BEGIN	1-4-28	Call and execute CCL procedure file
BKSP	BKSP	1-7-3	Bypass a number of logical records
CATALOG	DEFINE,SAVE	1-8-12 1-8-18	Retain a file as a permanent file
CKP	CKP	1-11-1	Take checkpoint dump
COMBINE	PACK	1-7-27	Combine system-logical-records into a single record
COMMENT	COMMENT	1-6-3	Place comments in dayfile
COMPARE	VERIFY	1-7-47	Compare compiler and of binary files
Compiler Calls	Compiler Calls	See Reference Manual for language	Access compiler and compile source
COPY	COPYEI	1-7-15	Copy to end-of-information
COPYBF	COPYBF	1-7-11	Copy binary files
COPYBR	COPYBR	1-7-12	Copy binary records
COPYCF	COPYCF	1-7-12	Copy coded files
COPYCR	COPYCR	1-7-14	Copy coded redord
COPYSBF	COPYSBF	1-7-16	Copy shifted binary file
.DATA	.DATA	1-4-26	Write CCL data statements to file
DISPLAY	DISPLAY	1-4-17	Place evaluated CCL expression in dayfile
DISPOSE	DISPOSE	1-7-18	Release files to output queues
DMP	DMP,DMD	1-9-1	Dump central memory
EDITLIB	LIBGEN,LIBEDIT	Appendix C	Create and maintain program library file
ELSE	ELSE	1-4-9	Conditional CCL skip
ENDIF	ENDIF	1-4-8	End conditional CCL branch
ENDW	ENDW	1-4-11	End CCL loop
EXECUTE	EXECUTE	See LOADER Reference Man.	Execute contents of a file

EXTEND	No equivalent under NOS	No equivalent under NOS	Add data to end of a permanent file
EXIT	EXIT,NOEXIT, ONEXIT	1-6-8	Error condition branch
FILE	FILE	See BASIC ACCESS METHODS (BAM) Reference Manual	Describe file characteristics
IFE	IFE	1-4-8	Conditional CCL branch
NOS/BE jobcard	NOS jobcard, USER, CHARGE, RESOURC	1-3-1 ff	Jobname, resource, and accounting information
LABEL	REQUEST, LABEL, BLANK	1-10-18 1-10-14 1-10-13	Request labeled magnetic tapes.
LDSET	LDSET	See LOADER Reference Man.	Set loader options
LGO	LGO	See LOADER Reference Man.	Load and execute
LIBLOAD	LIBLOAD	See LOADER Reference Man.	Load library programs
LIBRARY	LIBRARY	See LOADER Reference Man.	Define global library
LISTMF	LISTLB	1-10-15	List contents of label on labeled tape
LOAD	LOAD	See LOADER Reference Man.	Load contents of file
MAP	MAP	See LOADER Reference Man.	Generate Loader Reference Map
MODE	MODE	1-6-13	Define exit error conditions
NOGO	NOGO	See LOADER Reference Man.	Complete load without execution
OVERLAY	OVERLAY	See LOADER Reference Man.	Generate overlays
PAUSE	PAUSE	N/A	Send message to central site operator
.PROC	.PROC	1-4-22	Define beginning of CCL procedure file
PURGE	PURGE,PURGALL	1-8-17 1-8-16	Remove permanent files from system
REDUCE	REDUCE	See LOADER Reference Man.	Set dynamic job field length management
RENAME	CHANGE	1-8-11	Alter permanent file parameters
REQUEST	REQUEST,LABEL, BLANK	1-10-16 1-10-9 1-10-7	Request labeled or unlabeled magnetic tapes
RESTART	RESTART	1-11-2	Resume execution of checkpointed job
RETURN	RETURN	1-7-32	Release lfn from job: decrement resource count
REVERT	REVERT	1-4-33	Exit CCL procedure
REWIND	REWIND	1-7-33	Rewind lfn to B.O.I.
RFL	RFL	1-6-23	Set field length for job

SKIP	SKIP	1-4-7	Skip CCL statements
ROUTE	ROUTE	1-7-33	Release lfn to designated input/output queue
SEGLOAD	SEGLOAD	See LOADER Reference Man.	Generate segmented load
SET	SET	1-4-12	Assign value to CCL register
SKIPB	SKIPR	1-7-8 1-7-3	Skip logical records backward
SKIPF	SKIPR, SKIPEI SKIPR	1-7-38	Skip logical records backward
SLOAD	SLOAD	See LOADER Reference Man.	Load selected programs
SUMMARY	SUMMARY, ENQUIRE	1-6-32 1-6-4	Return system information
SWITCH	SWITCH,ONSW	1-6-33	Set sense switches
SYSBULL	No equivalent under NOS	No equivalent under NOS	Access system bulletin files
UNLOAD	UNLOAD	1-7-45	Release lfn from job; do not decrement resource count
VSN	VSN	1-10-18	Assign tape VSN to local file
WHILE	WHILE	1-4-11	Begin CCL loop

Document Title: Control Statement Comparison

Source of Material: NOS/BE-NOS

Supplier and Author: Martin Marietta, Data Systems

Date of Publication: N/A

Editing: Addition of other common products

Comments: None

NOS/BE-NOS

CONTROL STATEMENT COMPARISON

The following write-up will indicate where to look in NOS for a capability equivalent to a NOS/BE control statement. Where "same" is indicated the same control statement is used under NOS but the parameter syntax may differ.

This write-up may be used as a guide to NOS commands. A NOS reference manual should be consulted for more detail.

NOS/BE	NOS
ACCOUNT	NOS USER and CHARGE card
ATTACH	Same, but can be used for direct access files only. See NOS reference manual for parameter differences from the NOS/BE ATTACH statement. For indirect access files, see the OLD and GET NOS commands.
AUDIT	Use CATLIST
BEGIN	Same
BKSP	Same
CATALOG	For direct access files, use DEFINE. For indirect access files, use a SAVE (new file only) or REPLACE.
CKP	Same
COMBINE	PACK
COMMENT	Same
COMPARE	VERIFY
COPY	Same
COPYBCD	COPY
COPYBF	Same
COPYBR	Same

COPYCF	Same
COPYCR	Same
COPYN	No Direct Equivalent
COPYSBF	Same
COPYXS	COPY
COPYL	Same
COPYLM	Same
DISPLAY	Same
DISPOSE	Same
DMP	Same
EDITLIB	LIBGEN, LIBEDIT
ELSE	Same
ENDIF	Same
ENDW	Same
EXECUTE	Same
EXIT	Same
EXTEND	APPEND
GETPF	Use ATTACH
IFE	Same
ITEMIZE	Same
LABEL	Some major differences
LIMIT	LIMITS
LISTMF	LISTLB
LOAD	Same
MAP	Same

MODE	Same
MOUNT	PACKNAM
PAUSE	No NOS Equivalent
PURGE	Same, file must have write permission
REDUCE	Same
RENAME	CHANGE
REQUEST	REQUEST, ASSIGN
RESTART	Same
RETURN	Same
REVERT	Same
REWIND	Same
RFL	Same
ROUTE	Same
SET	Same
SETNAME	PACKNAM
SKIP	Same
SKIPB	BKSP
SKIPF	SKIPR
SUMMARY	Same
SWITCH	Same
SYSBULL	No NOS Equivalent
TRANSF	Equivalent capability available under NOS as a special product from CDC.
UNLOAD	Same
UPDATE	Same
VSN	Same
WHILE	Same

The following standard products are common to both NOS and NOS/BE and differs only in minor areas dictated by differences in the Operating System.

FTN
COBOL
SORTMRG
CYBER RECORD MANAGER
CYBER LOADER
CYBER CONTROL LANGUAGE (CCL)
ALGOL 68
BASIC
APL
COMPASS
UPDATE
COBOL5
FTN5
GPSS

NOS FEATURES

The following command summary shows features that are available on NOS but not on NOS/BE.

NOS COMMAND	DESCRIPTION
APPEND	Appends local file(s) to an existing indirect access permanent file.
ASSIGN	Assigns a file to the specified device.
BLANK	Gives a tape a blank label.
CALL	Allows the insertion of a procedure file at the specified position in the control statement stream.
CATALOG	LISTS INFORMATION ABOUT EACH RECORD IN A SPECIFIED FILE.
CLEAR	Releases all the user's current working files.
COMMON	Used to create or access a library type file.
CONVERT	Converts records from one character set to another.
COPYEI	Copies one file to another.

COPYX Enables user to specify certain conditions when copying logical records.

CTIME Issues the accumulated CPU time for a job to the user's dayfile.

DAYFILE Writes the user's dayfile to a specified file.

DMD Produces dump similar to DMP but with the display code equivalences to the right of the octal representations.

DOCMENT Extracts either the external or internal documentation from a file containing COMPASS source code.

DMDECS ECS memory dump, four words per line with display code equivalences to the right of octal presentations.

ENQUIRE Gives information about the system to the user.

ENTER Allows a series of control statements on one line.

EVICT Release file space for a specified file but does not release file attachment to the job.

FCOPY Converts a file from one code set to another.

FILE Determines the attributes of a file.

GTR Appends records selected from one file to the end of another using flexible selection directives.

GOTO Transfers control within the control statement file.

IF Processes a dependent statement if the conditions given in an expression are true.

KRONREF Generates a cross reference listing of system symbols used by decks on a MODIFY OPL.

LBC Loads binary data of an unknown format.

LDI Copies file to mass storage and submits the job(s) in the file.

LENGTH Gives the user the current status of one of his local files.

LIST 80 Compresses COMPASS assembler output to 80 columns for listing on 8-1/2 by 11 inch printer paper.

LOC Reads octal line images from file INPUT into user specified CM locations.

LOCK Enables the user to prevent writing on a file.

LO72 Allows reformatting of files.

MFL Resets the maximum field length for a job step.

MODIFY Edits a Modify-formatted program library.

NEW Creates a primary file.

NOEXIT Suppresses the transfer of control to the statement following the next EXIT statement if an error occurs.

NORERUN Signals job may not be rerun.

NOTE Creates a file contining lines of data. The data lines are entered as character strings on the same line as the NOTE statement.

NUM Determines if a string is numeric or not.

OFFSW Clears the pseudo-sense switches for reference by the user's program.

OPLEDIT Removes modification decks and identifies from a Modify-formatted program library file.

OUT Releases output files from the control point to the output queue.

PACK	Packs a specified file and copies it to another.
PACKNAM	Directs subsequent perm file requests to the specified auxiliary device.
PASSWOR	Changes the user's password.
PBC	Writes one record from the specified area of CM to file PUNCHB.
PERMIT	Allows a user to explicitly permit another user to access a private or semi-private file in his permanent file catalog.
PRIMARY	Makes a local file the primary file, or it creates an empty primary file.
PROTECT	Used to activate or deactivate preservation of a user's ECS field length between job steps.
PURGALL	Purges all permanent files in a user's catalog that satisfies the criteria specified by the parameters.
RBR	Loads one binary record from a specified file.
RERUN	Job may be rerun.
RESEQ	Resequences source files which have leading sequence numbers or adds sequence numbers to an unsequenced file.
RESOURC	Necessary in any job that uses more that one tape or pack concurrently.
ROLLOUT	Places job in rollout queue.
RTIME	Gives accumulated time since the last system dead-start.
SETASL	Specifies new account block SRU limit.
SETCORE	Presets each word within the field length.
SETID	Assigns a new identification code for the specified file.

SETJSL	Specifies the job step SRU limit for each remaining step of the job.
SETPR	Specifies a new CPU priority for a job.
SETTL	Specifies a new CPU time limit for subsequent job steps.
SKIPEI	Positions specified file at the EOI.
SKIPFB	Skips backwards a given no. of files.
SKIPR	Skips forward a given no. of records.
SORT	Sorts a file of line images or statements in numerical order based on the leading line numbers.
SS	Sets current subsystem being used by a job.
STIME	Issues the accumulated SRU value for the job to the user's dayfile.
SUBMIT	Submits a file to the system as a batch job.
TCOPY	Copies tapes to mass storage and vice versa.
TDUMP	Lists a file in octal and/or alphanumeric form.
UNLOCK	Rescinds the LOCK command. Clears the write interlock bit for the specified file.
UPMOD	Converts an Update-formatted source library file to a modify formatted file.
USECPU	Specifies which central processor to use.
VFYLIB	Performs a binary comparison of two files.
WBR	Writes a binary record from CM to a file at its current position.
WRITEF	Writes a specified no. of file marks on the named file.
WRITER	Writes a specified no. of empty records on the named file.

Document Title: NOS User Concerns

Source of Material: N/A

Supplier: Martin Marietta, Data Systems

Date of Publication: 4/06/79

Editing: None

Comments: This document references concerns at a local installation so it is important to discern which questions may be pertinent to your site and which refer to local software or modifications and equipment.

NOS USER CONCERNS

CRITICAL CONCERNS

1. Will the RUN24 compiler be available under NOS? There is a need to compile and execute using RUN24 and its libraries (MARLIB and RUNLIB).

   We recommend that RUN24, RUNLIB and MARLIB be installed on NOS as direct access permanent files under user number 'LIBRARY'. They will not be eligible for archiving and their use will be restricted to users on the AEROSPACE family.

2. There is a need for stacking of commands in the interactive mode under NOS.

   This capability does not exist under NOS and CDC has no plans in this area. We see no way of providing it short of major operating system revision. We would like to point out that this is not a critical concern of current NOS users. We recommend no change in this area.

3. The changing of account numbers (F207, F210, etc.) without changing user number.

   This capability exists under NOS. We recommend strongly that each user be limited to one and only one user number. This user may use any one of several account numbers for which he is validated. This account number will be the charge number specified on the Charge Card. Interactive users may switch charge numbers without logging out by issuing a new charge command.

4. There is a need for the DD280 plot package under NOS. We recommend the installation of DD280 under NOS.

5. There is a need to access and track jobs other than by the last alphanumerics in the job name that is automatically assigned by the system. The user needs to be able to specify 3 characters of the job name.

   We recognize that it is difficult for a user to tell one job from another when he has several in the system at once. We recommend an active search at other NOS sites for local code that addresses this problem.

6. There is a need to execute runs in the interactive mode. This is a feature of NOS. We strongly recommend that during prime shift interactive user be limited to a max

core size of 100 - 130K octal. This is a global parameter set by operations to prevent degradation of response time during heavy use periods.

SIGNIFICANT CONCERNS

7. Please submit all default parameters to SCUT for approval. We recommend all parameters be presented to SCUT but point out that final decision on their values must lie within Data Systems. We recommend reasonable large limits rather than no limit on the parameters. In the area of perm file limits we recommend limits that will allow the permanent files to be kept under control both in number and size. These parameters should be based on user need and also on capacity and overhead consideration. We recommend a procedure that allows users with excess requirements to be accommodated but only with Data Systems management approval.

8. Job name of a session must not change upon system recovery. It is unclear how this is accomplished under NOS.

   Job name will not change upon System Recovery. The users position in an internal system terminal table might change but his job name won't.

9. There is a need for cycles (5 cycles under NOS/BE) of permanent files under NOS. Please address proposed methods of insuring integrity of both direct and indirect permanent files.

   There is no provision for cycles under NOS. CDC points out that file integrity and system stability are both much higher under NOS than NOS/BE. The dispersal of catalogs (directory) under NOS provide faster perm file functions and thereby a smaller window of time during which a user is vulnerable to file loss in case of a system hang. Users concerned about this might wish to:

   (a) Catalog file under name B.

   (b) Purge old version file A.

   (c) Rename B to A.

10. There is a need for 40 character permanent file names. We see no way to provide this under NOS. Examination of internal tables shows no area where such an entity might be kept. We recommend no changes in this area of the system.

11. There is a need to be able to convert NOS/BE permanent files to NOS using an automated procedure.

CDC is providing a utility for loading NOS/BE perm files onto NOS. We recommend that Data Systems provide client assistance in this area. This person or persons will develop automated procedures around the above utilities and would assist users on an individual basis in file conversion.

We foresee perhaps one procedure under NOS/BE to consolidate user files onto one dump tape and one procedure under NOS to load the users files. In addition users could independently bring over files one at a time using the above utilities.

12. Will a RUN24 program execute interactively under NOS?

    RUN24 programs will execute interactively under NOS in the same manner they do under NOS/BE. However, file interaction presents the same problems it does under NOS/BE.

13. What is the NOS replacement for TRANSF? Will it be available on NOS?

    We recommend the installation of job dependency software under NOS available from CDC.

14. There is a need for an alphabetized CATLIST.

    We recommend that this feature be developed and installed by Data Systems.

15. There is a need for additional parameters to identify and document files.

    We see no way to provide this feature under NOS. No internal table space exists to accommodate anything like this. Users might wish to develop their own "directory" file containing useful information about their other files.

16. There is a need for the EXIT (U) control card.

    We recommend that the system be modified to provide this feature. Many NOS sites have done so. We see a small change to 1AJ as the only needed modification.

IMPORTANT CONCERNS

17. Senator is currently our best editor. Will it be available under NOS?

    We recommend the installation of SENATOR under NOS. We could convert ours or get a version already running at a NOS site. The latter would probably be preferable.

18. Bye is used for exiting from Senator. Will this conflict with bye as a substitute for logout?

There is no conflict here. "BYE" in Senator will get you out of SENATOR.

19. What are the program mods required to run NOS/BE FORTRAN IV (FTN and RUN24) programs under NOS? Please address source, relocatable and absolute forms of programs.

a. RUN24 Source - No mods required unless program calls 'XCOF' which will not be implemented.

b. RUN24 Relocatable - Should work - tests in progress.

c. RUN24 Absolutes - Should work unless they call routine such as XCOF, INFO, CPTIM, JOBNO, etc. that access the control point area. For these types relinking is required.

d. FTN Source - Some mods required to convert to FTNV. (ANSI77STD). A conversion utility will be available.

e. FTN relocatables - Recompilation required.

f. FTN absolutes - Recompilation required.

20. Will MACE programs have to be recompiled?

See Item 19 (b), (c) above.

21. NOS system procedures for accessing other permanent files is cumbersome.

Other users files are easily accessable if they are 'PUBLIC'. System design provides a large degree of inherent security. We recommend no changes in this area.

22. The control cards under NOS have a syntax that is different than NOS/BE. NOS does not treat the first blank as a separator. This is imbedded in the NOS design and the large number of products and programs that do their own control card cracking lead us to recommend no changes in this area.

23. There is a need for the NOS/BE job dependency feature under NOS.

See Item 13.

24. The charging algorithm needs to be maintained. Will the same cost parameters appear on the dayfile?

    The same cost parameters will appear on the dayfile. The message syntax is different. A NOS 'System Second' is called an 'SRU'. More detail is available under NOS. For example, Disk and Tape I/O are reported separately.

25. What is the equivalent of PAGE?

    We recommend installation of a program called 'SCAN' from the University of Virginia which is an improved 'PAGE' program.

26. There is a need to edit multirecord and multifile permanent files under NOS.

    This can be done with XEDIT and SENATOR. See page 1-1 of XEDIT Ref. Man.

27. There is a need to access a permanent file with one command regardless of file type.

    For those users requiring this feature we recommend a procedure call.

28. There is a need to be able to retrieve TRS files.

    TRS files may be retrieved in same manner as other NOS/BE files. Once on NOS, no new files will be put on TRS.

29. There is a need to use the FILES command.

    The ENQUIRE,F. command replaces the FILES command.

30. The IF, END CASE, IFC, ENDIFC, ELSEC, and CASE CCL commands are needed.

    We recommend the installation of these programs under NOS to the extent possible, i.e., the IF(PF) card can only refer to direct access files.

31. There is a need for MFL to be greater than MAXFL.

    You may use core up to what is on your job card or what you are validated for. We recommend each user be validated for 377700B unless he wants less.

32. What is the time limit with EXIT (P) if a job hits a time limit?

The user is given 10 octal seconds after a time limit error. We recommend this be changed to 24 octal as it is on NOS/BE.

33. There is a need for the NOS/BE Q, J, and MYQ commands under NOS.

    We recommend that such a program be written and installed. CDC might wish to restrict it to certain classes of users, i.e., aerospace but not commercial.

34. There is a need for setting permanent file attribute defaults to desirable values (i.e., PUBLIC category type and READ usage mode).

    We believe the standard defaults (private, write) are appropriate. We do not recommend change here. Many people will want their files to be private rather than public and since they are private and available only to this user, a default of write is appropriate.

35. What is the substitute command for the assets command?

    The ENQUIRE command replaces the assets command.

36. Search option in the page mode.

    See Item 25.

37. NOS uses different product sets (i.e., MODIFY and LIBEDIT).

    Both Update and Modify are available under NOS. LIBEDIT functionally replaces EDITLIB.

38. Is PAUSE available under NOS? If not, can it be implemented under NOS?

    "PAUSE" is not available under NOS. It is not as easily implemented as under NOS/BE which is probably why it isn't there. We are against such a function in a multiprogramming environment and do not recommend its installation.

39. There is a need to define a usernum on DEFINE, SAVE and REPLACE commands under NOS.

    We see no way to provide this under NOS. The built in security dictates that each user create files only under his own user number. We recommend no change in this area.

40. There is a need for forms control.

Forms control will be provided via local mods to the "ROUTE" control card. The NOS 'DISPOSE' command will be de-installed.

41. There is a need to LOAD both types of files.

This is a standard feature of NOS.

42. There is a need for the priority request on the job card.

This does not currently exist under NOS. CDC will implement it around mid 1980. We are searching other NOS sites for local code to provide this capability.

44. How can a user input a job so that the output is directed to a specified output device?

This may be done with the ROUTE control card.

45. Will MACE tapes be readable under NOS?

The COPYM utility will be available under NOS for copying MACE tapes.

46. The default procedure should be PROFIL.

We recommend that this be done. It is a CCL installation parameter.

TOLERABLE CONCERNS

47. What is the substitute command for this situate command?

The "USER" command may be used to see if another user is logged in. The "DIAL" command may be used to send messages to other terminals.

48. Page forward or backward in the page mode.

See Item 25.

49. Will there be a minimum size allowed for direct access files? Suggested limit could be 3000 words.

There will be no minimum. However, users should be aware that DA files are allocated in multiples of 640 PRUs. Users requiring small DA files will not be inhibited but should be aware of the potential waste.

50. There is a need for level numbers on end of records/end of files.

We see no way to provide this under NOS. Its extremely limited use under NOS/BE precludes it from further consideration.

51. Can the NOS job structure be changed to the MCC NOS/BE structure?

We see no way to provide this capability. We believe it is extremely important that NOS remain as standard as possible in order to take advantage of future upgrades and new features.

52. There is a need for a retention period on permanent files.

Infinite retention is available as under NOS/BE. It is up to the user to purge his files when no longer needed.

53. Will SCOPE carryovers associated with CCL be available under NOS CCL?

See Item 31.

54. Do we need new software for the DATA100 or Harris terminal?

200UT Protocol will still be available. Also HASP Protocol and probably 3780/2780.

55. Does the user have access to the timed event file?

Users may enter the timed event queue with the ROLLOUT macro.

56. Will block mode be available for editing files?

Block mode will be available as this is strictly a terminal hardware feature. This means user may transmit any line on his screen but only one line at a time. Page transmit will not be available.

57. There is a need for a well organized audit. CATLIST is poorly organized.

See Item 14.

58. Could BATCH be implemented on NOS?

We recommend that all routing of files be done with the ROUTE control card. DA files will have to be copied prior to routing. We do not recommend implementing 'BATCH'.

59. The users need to be warned about the changes in COPYBR.

    We plan to issue an impact statement covering all non-obvious impacts.

60. How does a SYSTEM count ATTACH of PUBLIC files toward ARCHIVAL?

    PUBLIC files, i.e., files under user number "LIBRARY", will be made ineligible for archiving.

61. How can one CATLIST cross user lines?

    You may CATLIST the files of other users that they have allowed you to access. "CATLIST,LO=F,UN=USERABC."

Document Title: Basic Features of NOS/BE Available in NOS

Source of Material: Feasibility Study of Conversion to NOS Attachment B/University of Washington Correspondence

Supplier and Author: Steve Hallstrom
Academic Computer Center
University of Washington

Date of Publication: 10/17/78

Editing: Local references removed; format remains primarily the same.

Comments: In cases where local code was necessary to provide feature, a no was indicated for feature available.

Basic Features in NOS/BE Available in NOS

CYBER Hardware Support

	NOS	NOS/BE
Unit Record Equipment		
405 Card Reader	Yes	Yes
415 Card Punch	Yes	Yes
580 Printers	Yes	Yes
Magnetic Tape Equipment		
607 Tape Drive	No	Yes
65X Tape Drive	Yes	Yes
66X Tape Drive 1600 BPI	Yes	Yes
67X Tape Drive 6250 BPI	Yes	Yes
Rotating Mass Storage		
841	Yes	Yes
844-21 Single Density	Yes	Yes
844-41 Double Density	Yes	Yes
819 on 176 only		
Single Density	Yes	Yes
Double Density	Yes (R6)	
Multiplexors		
6676 Teletype	Yes	Yes
6671 Teletype	Yes	Yes
UT200	Yes	Yes
Mixed	No	Yes
6673/6674 Wide Band	No	Yes
7077/791 LCC	No	Yes
Communication Processors		
2550-100 6676 Emulation	Yes	Yes
2550-2, 2552-1 Teletype	Yes	Yes
UT200	Yes	Yes
Mixed		
Teletype Speeds		
110-300 Autobaud	Yes	Yes
Correspondence Terminals	Yes	No
600 Baud	Yes	No
1200 Baud	Yes	Yes
2400 Baud	No	Yes
4800 Baud	No	Yes
9600 Baud	No	Yes

General Job Control

	NOS	NOS/BE
Deck Structure		
Logical Records	Yes	Yes
Record Levels	0,17(2)	0-17(16)
Multi-Files	Yes	No
Mode Select (029, 026)	Yes	Yes
CYBER Control Language	Yes	Yes
System Bulletin	No	Yes
Job Card Priority	No (SPL)	Yes
Job Dependency	No (SPL)	Yes
User Validation		
User Name, Account Number	Yes	No
Password	Yes	No
User Profile	Yes	No
Charge Numbers	Yes	No
Project Number	Yes	No
Family, Device Set	Yes	Yes
Limit Total Resource Usage	Yes	No
Limit Access by Time of Day	Yes	No

Accounting

	NOS	NOS/BE
Application Control	Yes	No
CPU/CM/ECS	Yes	Yes
Tape/Disk I/O	Yes	No
Channel Time	No	Yes
Mass Storage Used	Yes	Yes
System Seconds	Yes	Yes
Cards in/Lines out	Yes	Yes
Terminal Character I/O	Yes	No
Tape/Pack Mounts	Yes	No

Tape Processing

	NOS	NOS/BE
Multi-File Tapes	Yes	Yes
Recognition by VSN/LABEL	Yes	Yes
Tape Scheduling Overcommitment	Yes	Yes
Preview	Yes	Yes
Labels		
ANSI	Yes (Latest Standard)	Yes
Non-Standard	Yes	Yes
3000	No	Yes
Z-Labels	No	Yes
File Access Control		
Owner Field	Yes	Yes
Expiration	Yes	Yes
Ring/NoRing	Yes	Yes
Extended Label Processing	Yes	Yes
Formats		
I (NOS Internal, BLK-ID on Tape)	Yes	No
SI (NOS/BE Internal)	Yes	Yes
S (Stranger)	Yes	Yes
L (Long Stranger)	Yes	Yes
F (Foreign)	Yes	No
Binary/EBCDIC	Yes	Yes

Permanent Files

	NOS	NOS/BE
NAME (length)	Yes(7)	Yes(40)
User Number, Account Number	Yes	No
ID	No	Yes
Cycles	No	Yes
Access Controls		
Read	Yes	Yes
Append only	Yes	No

Modify	Yes	Yes
Write	Yes	Yes
Execute	Yes	No
Read (allow modify)	Yes	Yes
Read (allow append)	Yes	No
Explicit Permission to Other Users	Yes	No
Record of Usage by Others	Yes	No
Dump/Load Utilities		
Selective	Yes	Yes
Incremental	Yes	Yes
Available to Users	No (R6)	Yes
TRANSPF	No (R6)	Yes
Audit, Catalog	Yes	Yes

ECS Support

System Support	PF Residence	I/O Buffering
Swap Files	Yes	Yes
User Access	Yes	Yes
System Recovery	Yes	No

Scheduler Features

	NOS	NOS/BE
Classes		
Batch		
Local	Yes	Yes
Dependent (Uses Tapes)	No	Yes
Remote	Yes	No
Express (Operator Initiated)	No	Yes
Interactive	Yes	Yes
Multi-User	Yes	Yes
Graphics	No	Yes
System	Yes	No
Network	Yes	No

Swapping Control		
Jobs with Tapes	Swap	Roll
Event Related (User Controlled)	Yes	No
Wait Permanent File	Yes	Yes
Wait Tape	Yes	Yes
Wait Interactive I/O	yes	Yes
Wait Private Pack	Yes	Yes
Wait Multi-Mainframe	No(SPL)	Yes
Wait Staged File (7600)	No	Yes
Schedule By		
CPU Priority	Yes	Yes
Quantum Priority	Yes	Yes
CM Priority	Yes	Yes
Priority Aging	Yes	Yes
Job Card Priority	No(SPL)	Yes

Multi-Mainframe Support

Configurations Supported	4	2
Link Media		
ECS	Yes	Yes
6683 Satellite Coupler	Yes	Yes
RMS	No(SPL, R6)	No
Tape Staging (7600)	No(SPL)	Yes
Shared RMS		
Permanent Files	Yes	Yes
I/O Queues	No(SPL, R6)	Yes
Removable Devices	Yes	Yes
Load Leveling	No(SPL, R6)	Yes
Permanent File Staging	No	Yes
I/O File Routine	No (R6)	Yes
Symmetric Link	Yes	Yes

R6 - Scheduled to be included in Release 6 of NOS.

SPL - Available in a Special Product Library. Cost is negotiable.

Document Title: NOS-NOS/BE Feature Comparison

Source of Material: N/A

Supplier and Author: N/A

Date of Publication: N/A

Editing: Extracted from more comprehensive document; change in format and spelling corrections.

Comment: Supplier asked not to be referenced.

NOS - NOS/BE FEATURE COMPARISON

NOS	NOS/BE
Large variety terminal support.	Similar.
User or default definition of terminal characteristics.	Not currently available.
Continued operation of 2550 if host goes down.	Not currently available.
Availability of on-line 2550 diagnostics.	Except at initialization. Not currently available.
Availability of accounting data for individual batch terminals.	Not standard.
Provision of network validation before granting access to host operating system.	Login validation.
The facility is available to allow generation of specialized network application programs.	Similar facility provided via multi-User Job/System Control Point features.
Provided network applications include:	Intercom.
IAF - Interactive Facility	
RBF - Remote Batch Facility	
TAF - Transaction Facility	TAF function not standard.
Host operating system provides three levels of accounting:	Interfaces provided for user written accounting routines.
Charge Number - allocated by a site to a department.	
Project Number - allocated by a department to project controllers and controlled by same.	
User Name - each individual user has a unique user name, which is charged for machine usage under a Charge and Project number.	

NOS	NOS/BE
Tape management is centralized in one subsystem to provide tape user security.	Tape security on an individual basis is not provided.
Magnet provides tape usage, accounting and overcommitment algorithm.	Facilities provided for user written accounting routines. Wider range of tape scheduling options.
Magnet allows tape user to have greater control over tape processing options.	Not to the same extent.
Removable pack usage does not require specific use of MOUNT control card.	MOUNT card required, VSN and SETNAME must also be known.
If removable pack is requested, and is not mounted the job is aborted unless the user requests not to be aborted.	User is swapped out unconditionally until pack is mounted.
When the last direct access file on a removable pack is RETURN'ed the job's association with the pack is terminated.	Removable packs must be specifically DSMOUNT'ed by the job, or End of Job Processing.
A removable pack may be locked for new user access, so that a pack may be dismounted after all current users have completed operations on the pack.	Not available.
Permanent file usage allows for both Direct Access and Indirect Access permanent files.	Direct Access style files available.
Indirect access permanent files provide for allocation of files in 64 word blocks.	Indirect access to files available on Multi-Mainframe configurations only.
File structures between tape and disk, disk and disk, and disk and tape are preserved.	System/Scope random file structures destroyed by copying. Affects utilities such as UPDATE and EDITLIB.

Large number of utilities available. Very few control card differences compared to NOS/BE.	Not as many utilities available.
Extensive resource control and management provided at the user level, and entry points to the network and host operating system.	Not available.
Controls are provided on file sizes both local and permanent network application programs, central memory, CP time, tape, and removable pack usage.	Not available as standard.
Permanent file security: Users may only access their own files, unless specifically authorized to access another users files.	Requires more conscious control on part of user. Default is no security.
Permanent files may be created with the following modes being specified: Write, Read, Modify, Append, Execute.	Similar facilities available via use of passwords for: Read, Control, Extend, Turnkey.
Permanent files may be given individual user permissions or blanket user permissions of Private, Semiprivate, or Public, plus the facility of nominating a password.	Blanket permissions available through passwords.
The accounting facilities maintain a separate system account file.	Accounting information maintained in system day file.
IAF provides several subsystems for timesharing users. Batch, Execute, FTNTS, Basic, Null, APL. Each of these subsystems has a set of simple commands for performing simple program development.	Subsystems not available.
The Batch subsystem provides access to all of NOS control cards.	Intercom provides effectively what the Batch subsystem provides.
Three text editors are available, line editor, and two powerful text editors which may be used interactively or from a batch job.	Editor is available as a Multi-User Job.

A processor to reformat a text file and submit to the central site for processing as a batch job.	Batch command provides similar but has no reformatting capability.
The facility for declaring a terminal to be used for a specific application, and also to autologin the terminal.	Application dependence not available. Autologin possible as nonstandard.
The timesharing user can declare a file to be a Primary terminal file. All commands by default will reference this file unless specifically directed to another work file.	Not available.
A subsystem may be associated with an Indirect Access file, when that file is retrieved as a Primary terminal file the subsystem will be selected automatically.	Not available.
Not directly available, can be implemented in a batch job stream.	Job card dependency identifier.
All jobs executing are treated as queue files.	Limited on number of currently executing tape jobs requesting tape assignment.

6542B
TM

NOS/BE - NOS MACRO EQUIVALENCES

SECTION THREE (NOS REFERENCE MANUAL)
INPUT OUTPUT

	NOS RM	NOS/BE RM
CIO OPEN AND CLOSE FUNCTIONS		
OPEN	2-3-6	7-47
CREATES A FILE OR DETERMINES INFORMATION ABOUT A FILE		
CLOSE	2-3-12	7-49
TERMINATES OPERATIONS ON A FILE		
CLOSER	2-3-14	7-51
CLOSES A MAGNETIC TAPE FILE		
CIO READ FUNCTION		
RPHR	3-3-15	7-57
CAUSES ONE PRU TO BE TRANSFERRED TO THE CIRCULAR BUFFER		
READ	2-3-16	7-54
READS INFORMATION INTO THE CIRCULAR BUFFER		
READSKP	2-3-17	7-56
READS INTO CIRCULAR BUFFER UNTIL AN EOR OR EOF IS FOUND		
READCW (READC ON NOS/BE)	2-3-17	5-22 (NOS/BE SPRM)
PERFORMS A NON-STOP READ OF PRUS BOUNDED BY CONTROL WORD		
READLS	2-3-19	5-23 (NOS/BE SPRM)
READS THE GROUP OF RMS LOGICAL RECORDS SPECIFIED BY A LIST		
RPHRLS (ALTERNATE FORMAT OF READLS IN NOS/BE)	2-3-20	5-23 (NOS/BE SPRM)
READS THE GROUP OF PRUS AS SPECIFIED BY A LIST		
READNS	2-3-21	7-55
READS A FILE FROM CURRENT POSITION TO EOF		
READN	2-3-21	7-57
READS DATA FROM S/L TAPE INTO CIRCULAR BUFFER		
READEI	2-3-22	
READS DATA INTO CIRCULAR BUFFER UNTIL EOI OR FULL BUFFER		

SECTION THREE
INPUT OUTPUT (cont.)

	NOS RM	NOS/BE RM
CIO WRITE FUNCTIONS		
WPHR WRITES ONE PHYSICAL RECORD FROM CIRCULAR BUFFER	2-3-23	7-63
WRITE WRITES THE CONTENTS OF THE CIRCULAR BUFFER	2-3-23	7-61
WRITER EMPTIES CIRCULAR BUFFER AND WRITES AN EOR	2-3-24	7-62
WRITEF EMPTIES CIRCULAR BUFFER AND WRITES AN EOF	2-3-24	7-63
WRITECW (WRITEC ON NOS/BE) PERFORMS A NON-STOP WRITE OF PRUS BOUNDED BY CONTROL WORDS	2-3-25	5-25 (NOS/BE SPRM)
REWRITE PERFORMS A REWRITE OF RECORDS ON AN RMS FILE	2-3-26	7-67
REWRITER PERFORMS A REWRITE OF RECORDS ON AN RMS FILE INCLUDING AN EOR	2-3-27	7-67
REWRITEF PERFORMS A REWRITE OF RECORDS ON AN RMS FILE INCLUDING AN EOF	2-3-28	7-67
WRITEN WRITES NON-STOP ON S/L TAPE FILES	2-3-28	7-64
FILE POSITIONING MACROS		
BKSP BACKSPACES A FILE ONE LOGICAL RECORD	2-3-29	7-72
BKSPRU BACKSPACES A FILE ONE PHYSICAL RECORD	2-3-30	7-73
REWIND REWINDS THE FILE BACK TO BOI	2-3-30	7-73
UNLOAD UNLOADS THE FILE AND RELEASES THE RESOURCE	2-3-31	7-74
RETURN (CLOSE RETURN ON NOS/BE) UNLOADS THE FILE AND RELEASES THE RESOURCE	2-3-33	7-50
POSMF OPENS AND/OR POSITIONS STANDARD ANSI-LABELED MULTIFILE MAGNETIC TAPE SETS TO A MEMBER OF THE SET	2-3-34	7-48

	NOS RM	NOS/BE RM
FILE POSITIONING MACROS (cont.)		
EVICT SIMILAR TO UNLOAD EXCEPT THAT THE FNT AND LFN STILL EXIST	2-3-39	7-74
SKIPF SKIPS FORWARD LOGICAL RECORDS	2-3-39	7-71
SKIPFF (EQUIVALENT TO SKIPF LEVEL 17 ON NOS/BE) SKIPS FORWARD FILES	2-3-39	7-71
SKIPEI (EQUIVALENT TO SKIPF OF 777777B ON NOS/BE) SKIPS TO EOI	2-3-41	7-71
SKIPB SKIPS BACKWARD LOGICAL RECORDS	2-3-41	7-71
SKIPFB (EQUIVALENT TO SKIPB FILES ON NOS/BE) SKIPS BACKWARD FILES	2-2-42	7-71
DATA TRANSFER MACROS		
READC (NO NOS/BE EQUIVALENT IN FUNCTION) READS ONE CODED LINE FROM BUFFER TO WSA WITHOUT BLANK FILL	2-3-46	*
WRITEC (NO NOS/BE EQUIVALENT IN FUNCTION) WRITES ONE CODED LINE FROM WSA TO BUFFER WITHOUT BLANK FILL	2-3-46	*
READH (READIN ON NOS/BE) TRANSFERS DATA FROM BUFFER TO WSA WITH BLANK FILL	2-3-47	7-58*
WRITEH (WRITOUT ON NOS/BE) TRANSFERS DATA FROM WSA TO BUFFER WITH BLANK FILL	2-3-47	7-65*
READO (NO NOS/BE EQUIVALENT) READS ONE WORD FROM BUFFER INTO X6 REGISTER	2-3-47	*
WRITEO (NO NOS/BE EQUIVALENT) WRITES ONE WORD FROM X6 REGISTER INTO THE BUFFER	2-3-48	*
READS (NO NOS/BE EQUIVALENT) READS A LINE INTO THE WSA AT ONE CHARACTER PER WORD	2-3-48	*
WRITES (NO NOS/BE EQUIVALENT) WRITES A LINE FROM THE WSA TO THE BUFFER AT ONE CHARACTER PER WORD	2-3-48	*
READW (READIN ON NOS/BE IS CLOSE) READS INTO WSA FROM BUFFER AND DOES READ-AHEAD	2-3-49	7-58*
WRITEW (WRITOUT ON NOS/BE IS CLOSE) WRITES FROM WSA INTO BUFFER AND WRITES AHEAD	2-3-49	7-65*

* SEE COMPASS MANUAL FOR DETAILS ON COMMON MACROS.

SECTION FOUR

	NOS RM	NOS/BE RM
LOCAL FILE MANAGER		
RENAME (NO NOS/BE EQUIVALENT)	2-4-2	
CHANGES THE NAME OF THE LOCAL FILE NAME WITHOUT CHANGING THE PFN THAT IS ATTACHED		
RELEASE (DISPOSE ON NOS/BE)	2-4-5	7-75
RELEASE FILES TO ANY OF THE OUTPUT QUEUES		
LOCK (NO NOS/BE EQUIVALENT)	2-4-6	
PREVENTS WRITING ON A FILE		
COMMON (NO NOS/BE EQUIVALENT)	2-4-4	
CHANGES FILE TYPE TO LIBRARY		
UNLOCK (NO NOS/BE EQUIVALENT)	2-4-7	
CLEARS WRITE LOCK-OUT BIT ON A FILE		
STATUS	2-4-7	7-21
RETURNS CURRENT POSITION AND STATUS OF A LOCAL FILE		
REQUEST	2-4-10	7-42
REQUEST EQUIPMENT FOR A JOB		
SETID (ROUTE IS CLOSE ON NOS/BE)	2-4-12	
SETS THE DEFERRED DISPOSITION CODE FOR CERTAIN FILES		
ASSIGN (SPECIAL LDL CALL ON NOS/BE)	2-4-12	
ALLOWS AC		
CESS TO USER OR SYSTEM LIBRARIES		
ENCSF	2-4-13	7-15 (NOS/BE SPRM)
REPLACES THE CONTROL STATEMENT FILE		
PSCSF (CONTROLC ON NOS/BE)	2-4-14	7-29
ALLOWS CONTROL OF EXECUTION OF JOB STATEMENTS		
LABEL	2-4-15	6-24
USED FOR TAPE LABELING		
GETFNT (STATUS IS CLOSE ON NOS/BE)	2-4-21	7-23
ACCESS FNT ENTRIES FOR LOCAL FILES		
PRIMARY (NO NOS/BE EQUIVALENT)	2-4-23	
ALLOWS THE USER TO CREATE OR CHANGE A PRIMARY FILE		
FILINFO	2-4-24	7-24
RETURNS INFORMATION ABOUT A FILE		

SECTION FIVE

	NOS RM	NOS/BE RM
PERMANENT FILE MANAGER		
SAVE (NO NOS/BE EQUIVALENT) ENABLES THE USER TO RETAIN A COPY OF A LFN IN THE PF SYSTEM AS AN INDIRECT FILE	2-5-9	
GET (NO NOS/BE EQUIVALENT) GENERATES A WORKING COPY OF AN INDIRECT ACCESS FILE	2-5-10	
PURGE PURGES ANY TYPE OF PERMANENT FILE	2-5-11	7-81
CATLIST (AUDIT (CONTROL CARD) ON NOS/BE) QUERIES THE CATALOGS THE USER IS ALLOWED TO ACCESS	2-5-12	
PERMIT (NO NOS/BE EQUIVALENT) ENABLES A USER TO PERMIT ANOTHER USER TO ACCESS A PF IN HIS CATALOG	2-5-17	
REPLACE (NO NOS/BE EQUIVALENT) ENABLES THE USER TO PLACE A COPY OF HIS LFN AS AN INDIRECT FILE	2-5-19	
APPEND (NO NOS/BE EQUIVALENT) ALLOWS THE USER TO APPEND DATA TO THE END OF HIS INDIRECT FILE	2-5-20	
DEFINE (REQUEST *PF IS CLOSE ON NOS/BE) ENABLES THE USER TO SPECIFY A FILE AS A DIRECT ACCESS PF	2-5-21	
ATTACH ATTACHES A DIRECT ACCESS FILE	2-5-25	7-81
CHANGE (RENAME ON NOS/BE) CHANGES SEVERAL PF PARAMETERS IN THE USER'S CATALOG	2-5-26	7-81

SECTION SIX

CONTROL POINT MANAGER

THERE ARE NOT NOS/BE EQUIVALENT MACROS TO THE ENTIRE SECTION EXCEPT FOR GETJCI AND SETJCI

SECTION SEVEN	NOS RM	NOS/BE RM
QUEUE FILE MANAGER		
RERUN (NO NOS/BE EQUIVALENT)	2-7-2	
ALLOWS JOB TO BE RERUN IN EVENT OF A SYSTEM FAILURE		
NORERUN (NO NOS/BE EQUIVALENT)	2-7-3	
RESETS THE NO-RERUN BIT PREVIOUSLY SET BY A RERUN MACRO		
SUBMIT (ROUTE ON NOS/BE)	2-7-3	7-76
ENABLES THE USER TO SUBMIT A LOCAL FILE TO THE INPUT QUEUE		
SPECIAL SYSTEM CALL TO QFM (REQUEST ON NOS/BE)	2-7-5	7-42
ALLOWS ASSIGNMENT OF A FILE TO A QUEUE DEVICE		
SECTION EIGHT		
FILE ROUTING		
ROUTE	2-8-4	7-76
PROVIDES MANY DIFFERENT FORMS OF FILE DISPOSITION. MOST PARAMETERS ARE THE SAME FOR BOTH SYSTEMS. HOWEVER, SOME NOS/BE PARAMETERS SUCH AS TID ARE IGNORED ON NOS. SEE THE MANUALS FOR MORE DETAILS.		
SECTION NINE		
SYSTEM FILE MANAGER		
DAYFILE (NO NOS/BE EQUIVALENT)	2-9-3	
ENABLES THE USER TO ACCESS SYSTEM DAYFILES		
ESYF (NO NOS/BE EQUIVALENT)	2-9-4	
ENABLES THE USER TO SET A LOCAL FILE AS A SYSTEM FILE		
RDVT (FILINFO ON NOS/BE)	2-9-4	7-24
SECTION TEN		
JOB CONTROL		
CONTROL (CONTRLC ON NOS/BE)	2-10-2	7-29
ALLOWS THE USER TO READ THE NEXT CONTROL STATEMENT		
EXCST (NO DIRECT NOS/BE EQUIVALENT)	2-10-3	
ENABLES THE USER TO EXECUTE A CONTROL STATEMENT SUPPLIED BY THE USER'S PROGRAM		

SECTION TEN (cont.)

JOB CONTROL (cont.)

	NOS RM	NOS/BE RM
CHECKPT INITIATES A SYSTEM CHECKPOINT OF THE USER'S FIELD LENGTH AND FILES	2-10-3	7-40
REPRIEVE GENERATES AN RA+1 CALL TO RPV	2-10-9	7-39
RECOVR ADDED FOR NOS/BE COMPATIBILITY ONLY. MAY NOT BE SUPPORTED UNDER FUTURE NOS RELEASES. USE REPRIEVE INSTEAD.	2-10-10	7-30

SECTION ELEVEN

SYSTEM/LOADER REQUESTS

	NOS RM	NOS/BE RM
ABORT USED TO ABORT THE PROGRAM AND TRANSFER TO AN EXIT CARD IF ONE EXITS	2-11-1	7-14
CLOCK RETURNS THE CURRENT TIME OF DAY IN DISPLAY CODE	2-11-2	7-19
DATE RETURNS THE CURRENT DATE IN DISPLAY CODE	2-11-2	7-19
EDATE (NO NOS/BE EQUIVALENT) TAKES THE PACKED DATE AND CONVERTS IT TO DISPLAY CODE	2-11-2	
ENDRUN TERMINATES THE PROGRAM	2-11-3	7-15
ETIME (NO NOS/BE EQUIVALENT) TAKES THE PACKED TIME AND CONVERTS IT TO DISPLAY CODE	2-11-4	
GETMC ALLOWS THE USER TO OBTAIN INFORMATION ABOUT MACHINE CHARACTERISTICS	2-11-4	7-16
JDATE RETURNS THE CURRENT JULIAN DATE	2-11-5	7-19
MEMORY ALLOWS THE USER TO DETERMINE OR CHANGE THE AMOUNT OF CM OR ECS (LCM) ASSOCIATED WITH THE JOB	2-11-6	7-17
MESSAGE ALLOWS THE USER TO DISPLAY A MESSAGE ON THE DAYFILE	2-11-8	7-18

SYSTEM/LOADER REQUESTS (cont.)	NOS RM	NOS/BE RM
MOVE (NO NOS/BE EQUIVALENT) MOVES A BLOCK OF DATA FROM ONE CM LOCATION TO ANOTHER	2-11-10	
PDATE (NO NOS/BE EQUIVALENT) RETURNS THE CURRENT DATE AND TIME IN BINARY PACKED FORMAT	2-11-10	
RECALL ALLOWS THE USER TO RELINQUISH THE CPU	2-11-11	7-19
RTIME RETURNS THE REAL-TIME CLOCK READING	2-11-11	7-19
STIME (NO NOS/BE EQUIVALENT) ALLOWS THE USER TO DETERM?INE HIS ACCUMULATED SRU'S	2-11-12	
SUBR (NO NOS/BE EQUIVALENT) ENABLES THE USER TO DISTINGUISH BETWEEN ENTERING A SUBROUTINE AND EXITING FROM A SUBROUTINE EVEN THOUGH CONTROL IS TRANSFERRED TO THE SAME ADDRESS	2-11-13	
SYSTEM ALLOWS THE USER TO CONVENIENTLY CALL PP PROGRAMS	2-11-13	7-12
TIME RETURNS THE ACCUMULATED CPU TIME USED BY THE JOB	2-11-15	7-19
OVERLAY (NO NOS/BE EQUIVALENT) GENERATES AN USER REQUEST TO LOAD THE SPECIFIED OVERLAY. LOADREQ IN THE LOADER MANUAL IS CLOSE BUT THE USER MUST CREATE THE PARAMETER TABLE. OVERLAY GENERATES ALL THE BIT SETTINGS FOR THE USER.	2-11-17	
LOADD (NO NOS/BE EQUIVALENT) ALLOWS THE USER TO LOCATE FDL CAPSULES OR CCL PROCS	2-11-19	
LOADQ(NO NOS/BE EQUIVALENT) LOADS FDL CAPSULES OR OVERLAYS FROM SPECIFIED FILES	2-11-21	

NOS/BE

SECTION SEVEN (NOS/BE REFERENCE MANUAL)
NOS/BE MACROS

	NOS RM	NOS/BE RM
SYSTEM COMMUNICATION MACROS		
SYSCOM DEFINES STANDARD SYMBOLS AND MACROS	7-11	2-1-8
SYSTEM ALLOWS THE USER TO CONVENIENTLY CALL PP PROGRAMS	7-12	2-11-13
SYSTEM ACTION MACROS		
ABORT ABORTS THE USER PROGRAM	7-14	2-11-1
ENDRUN TERMINATES THE USER PROGRAM WITHOUT ABORT	7-15	2-11-4
GETMC ALLOWS THE USER TO OBTAIN INFORMATION ABOUT MACHINE CHARACTERISTICS	7-16	2-11-4
MEMORY ALLOWS THE USER TO DETERMINE OR CHANGE THE AMOUNT OF CM OR ECS (LCM) ASSOCIATED WITH THE JOB	7-17	2-11-8
MESSAGE ALLOWS THE USER TO DISPLAY A MESSAGE ON THE DAYFILE	7-10	2-11-8
RECALL ALLOWS THE USER TO RELINQUISH THE CPU	7-19	2-11-11
CLOCK RETURNS THE CURRENT TIME OF DAY IN DISPLAY CODE	7-19	2-11-2
DATE RETURNS THE CURRENT DATE IN DISPLAY CODE	7-19	2-11-2
JDATE RETURNS THE CURRENT JULIAN DATE	7-19	2-11-5
RTIME RETURNS THE REAL TIME CLOCK READING	7-19	2-11-11

	NOS RM	NOS/BE RM
SYSTEM ACTION MACROS (cont.)		
TIME RETURNS THE ACCUMULATED CPU TIME USED BY THE JOB	7-19	2-11-15
IOTIME (NO NOS EQUIVALENT) RETURNS ACCUMULATED IO TIME USED BY THE JOB	7-19	
STATUS PROVIDES A USER PROGRAM INFORMATION ABOUT SYSTEM RESOURCES (FILESTAT, GETACT ARE FORMS OF THE STATUS MACRO)	7-21	2-4-7
FILINFO RETURNS INFORMATION ABOUT A FILE	7-24	2-4-24
GETJCI TRANSFERS CCL JOB CONTROL INFORMATION TO THE USER'S JOB	7-26	2-6-30
SETJCI TRANSFERS CCL JOB CONTROL INFORMATION TO THE SYSTEM FROM THE USER'S JOB	7-27	2-6-31
TRANSF (NO NOS EQUIVALENT) INITIATES DEPENDENT JOBS	7-29	
CONTROLC (CONTROL ON NOS) ALLOWS THE USER TO MANIPULATE HIS CONTROL STATEMENTS TO SOME DEGREE	7-29	2-10-2
RECOVR ALLOWS THE USER TO REGAIN CONTROL OF HIS PROGRAM UNDER THE SPECIFIED TERMINATION CONDITIONS. SUPPORTED ON NOS ONLY FOR NOS/BE COMPATABILITY. USE REPRIEVE INSTEAD.	7-30	2-10-10
REPRIEVE GENERATES AN RA+1 TO CALL TO RPV	7-39	2-10-9
CHECKPT INITIATES A SYSTEM CHECKPOINT OF THE USER'S FIELD LENGTH AND FILES	7-40	2-10-3
REQUEST REQUEST EQUIPMENT FOR A JOB	7-42	2-4-10

SECTION SEVEN (cont.)
NOS/BE MACROS

	NOS RM	NOS/BE RM
CIO OPEN AND CLOSE FUNCTIONS		
OPEN	7-47	2-3-6
CREATES A FILE OR DETERMINES INFORMATION ABOUT A FILE		
POSMF	7-48	2-3-34
OPENS AND/OR POSITIONS STANDARD ANSI-LABELED MULTIFILE MAGNETIC TAPE SETS TO A MEMBER OF THE SET		
CLOSE	7-49	2-3-12
TERMINATES OPERATIONS ON A FILE		
CLOSER	7-51	2-3-15
CLOSES A MAGNETIC TAPE FILE		
CIC READ FUNCTIONS		
READ	7-54	2-3-15
READS INFORMATION INTO THE CIRCULAR BUFFER		
READNS	7-55	2-3-21
READS A FILE FROM CURRENT POSITION TO EOF		
READSKP	7-56	2-3-17
READS INTO CIRCULAR BUFFER UNTIL AN EOR OR EOF IS FOUND		
RPHR	7-57	2-3-15
CAUSES ONE PRU TO BE TRANSFERRED TO THE CIRCULAR BUFFER		
READN	7-57	2-3-21
READS DATA FROM S/L TAPE INTO CIRCULAR BUFFER		
READIN (READH ON NOS)	7-58	2-3-47
TRANSFER DATA FROM BUFFER TO WSA WITH BLANK FILL		
CIO WRITE FUNCTIONS		
WRITE	7-61	2-3-23
WRITES THE CONTENTS OF THE CIRCULAR BUFFER		
WRITER	7-62	2-3-24
EMPTIES CIRCULAR BUFFER AND WRITES AN EOR		
WRITEF	7-63	2-3-24
EMPTIES CIRCULAR BUFFER AND WRITES AN EOF		
WPHR	7-63	2-3-23
WRITES ONE PHYSICAL RECORD FROM CIRCULAR BUFFER		
WRITTEN	7-64	2-3-28
WRITES NON-STOP ON S/L TAPE FILES		

SECTION SEVEN (cont.)
NOS/BE MACROS

	NOS RM	NOS/BE RM
CIO WRITE FUNCTIONS		
WRITOUT (WRITEH ON NOS) TRANSFERS DATA FROM WSA TO BUFFER WITH BLANK FILL		
REWRITE PERFORMS A REWRITE OF RECORDS ON AN RMS FILE	7-67	2-3-26
REWRITER PERFORMS A REWRITE OF RECORDS ON AN RMS FILE INCLUDING AN EOR	7-67	2-3-27
REWRITEF PERFORMS A REWRITE OF RECORDS ON AN RMS FILE INCLUDING AN EOF	7-67	2-3-28
WRITIN (NO NOS EQUIVALENT) SIMILAR TO WRITOUT WITHOUT SOME OF THE RESPONSIBILITIES	7-69	
FILE POSITIONING MACROS		
SKIPF SKIPS FORWARD LOGICAL RECORDS	7-71	2-3-39
SKIPB SKIPS BACKWARD LOGICAL RECORDS	7-71	2-3-41
BKSP BACKSPACES A FILE ONE LOGICAL RECORD	7-72	2-3-29
BKSPRU BACKSPACES A FILE ONE PHYSICAL RECORD	7-73	2-3-30
REWIND REWINDS THE FILE BACK TO BOI	7-73	2-3-30
UNLOAD UNLOADS THE FILE AND RELEASES THE RESOURCE	7-74	203031
EVICT SIMILAR TO UNLOAD EXCEPT THAT THE FNT AND LFN STILL EXIST	7-74	2-3-39
DISPOSE (RELEASE ON NOS) RELEASE FILES TO ANY OF THE OUTPUT QUEUES	7-75	20405
ROUTE PROVIDES MANY DIFFERENT FORMS OF FILE DISPOSITION. MOST PARAMETERS ARE THE SAME FOR BOTH SYSTEMS. HOWEVER, SOME NOS/BE PARAMETERS SUCH AS TID ARE IGNORED ON NOS. SEE THE MANUALS FOR MORE DETAILS.	7-76	2-8-4

SECTION SEVEN (cont.)
NOS/BE MACROS

	NOS RM	NOS/BE RM
PERMANENT FILE MACROS		
FDB (NO NOS EQUIVALENT)	7-81	
SETS UP THE FILE DEFINITIN BLOCK		
PERM (NO NOS EQUIVALENT)	7-85	
DETERMINES WHAT PERMISSIONS HAVE BEEN GRANTED TO A FILE AND WHETHER THE FILE IS PERMANENT		
ALTER (NO NOS EQUIVALENT)	7-81	
CHANGES THE EOI FOR A PERMANENT FILE		
ATTACH	7-81	2-5-26
ATTACHES A PERMANENT FILE TO A JOB		
CATALOG (DEFINE ON NOS IS THE CLOSEST THING)	7-81	2-5-21
DIRECTS THE SYSTEM TO SAVE A FILE AS A PERMANENT FILE		
EXTEND (NO NOS EQUIVALENT FOR DIRECT FILES)	7-81	
ADDS OR CHANGES DATA ON AN EXISTING PERMANENT FILE		
GETPF (NO NOS EQUIVALENT)	7-81	
RETRIEVES A PERMANENT FILE RESIDING ON ANOTHER MAINFRAME		
PURGE	7-81	2-5-11
PURGES A PERMANENT FILE		
RENAME (CHANGE ON NOS)	7-81	2-5-26
CHANGES SEVERAL PF PARAMETERS IN THE USER'S CATALOG		
SAVEPF (NO NOS EQUIVALENT)	7-81	
SAVES A FILE AS A PERMANENT FILE ON ANOTHER MAINFRAME		

Converting Applications Program to NOS

Below is a partial list of applications converted to NOS by the University of Virginia. Use this as a guide to conversion difficulties at your site.

PROGRAM NAME	COMMENTS	TIME ESTIMATE
MATINV	Simple recompilation	
UNENAN	Simple recompilation	
SEQ	Minor mods to COMPASS routine	1 Hour
ECTA	Simple recompilation	
IDA	Some difficulty, changes to OVERLAY calls, file manipulation, information satements	10 Hours
TSP	Considerable difficulty; eventually obtain mod from another site	15 Hours
ESP	Simple recompilation	
MPOS	Conversion abandoned; purchased new version from supplier	20 Hours
RNF	Conversion abandoned; obtained new version from another site	50 Hours
COPYSQ	Modified extensively, removed SCOPE request card processor; new version available	40 Hours
PLOTFTN	Minor modifications to FTN ENCODE statements, otherwise trivial	1 Hour
PLOT10	Major project to make use of transparent I/O and List of Files, buffer flush	80 Hours
COMP10	Local UVA version to PLOTFTN no problems	
LISP	Did not attempt to convert, got NOS version, installed successfully	5 Hours
FORTPFM	Didn't attempt to convert, modified an old KRONOS program	10 Hours
SPSS	Version 7.0, had to insert a couple of entry points, and increase the field length in several procedures; had a lot of trouble when trying out various mods after ATTACHing a previously created SPSS with M=W; until we finally figured out that the SPSS Preload utility was adding on to the end of the old file;	30 Hours

PROGRAM NAME	COMMENTS	TIME ESTIMATE
SPSS (cont'd)	thereafter always used a PURGE, SPSS/NA, DEFINE, SPSS sequence; also had trouble with LIBEDIT replacing the (0,0) Main Overlay, therefore, we routinely use GTR to extract all REL records before using LIBEDIT	
PASCALR	Didn't try to convert; got a new NOS version from another site	3 Hours
DAREP	Had to make several modifications to the COMPASS subroutines which do error processing; had to rewrite the procedure file	6 Hours
MANOVA	Recompilation	
ORGCHT	Recompilation	
TIDY2	Recompilation	
JHVIEW	Recompilation	
STATMTR	Recompilation	
NLWOOD	Recompilation	
MLTQUAL	Have still not been able to convert this	12 Hours
ICPSR tapes	We had created about 8 of them as SI-coded tapes, therefore had to use TCOPY to create I-format tapes from the original ones	
LBMAINT	Needed a complete rewrite	10 Hours
SYMAP	No programmer comments but 5 to 8 hours time	

ORGCHT	ORGANIZATION CHART PLOTTING PROGRAM
MATINV	MATRIX INVERSE
PASCALR	PASCAL REVISED
RNF	INTERACTIVE TEXT FORMATTER
UNENAM	HARMONIC MEAN ANALYSIS OF VARIANCE
SEQ	SEQUENCE ANY 80-CHARACTER CARD IMAGE FILE
ECTA	EVERYMAN's CONTINGENCY TABLE ANALYSIS
IDA	INTERACTIVE DATA ANALYSIS
TSP	TIME SERIES PROCESSOR
ESP	ECONOMETRIC SOFTWARE PACKAGE
MPOS	MULTI-PURPOSE OPTIMIZATION SYSTEM
COPYSQ	COPY SEQUENTIAL FILES
PLOTFTN	SUBROUTINE FOR COMPLOT DP-7 DIGITAL PLOTTER
PLOT10	TEKTRONIX GRAPHICS PACKAGE
COMP10	EMULATE PLOTFTN ON GRAPHICS TERMINALS
LISP	LIST PROCESSING LANGUAGE
FORTPFM	FORTRAN PERMANENT FILE MANIPULATIONS
SPSS	STATISTICAL PROGRAMMING PACKAGE FOR THE SOCIAL SCIENCE
DAREP	DIFFERENTIAL ANALYZER REPLACEMENT, PORTABLE
TIDY2	CLEAN UP FORTRAN SOURCE PROGRAMS
JHVIEW	3-D PLOTTING PACKAGE
STATMTR	MATRIX ALGEBRA SUBROUTINE PACKAGE
ICPSR	INTER UNIVERSITY CONSORTIUM DATA RESOURCES
SYMAP	SYNAGRAPHIC COMPUTER GRAPHING

BENCHMARK

INTRODUCTION

No analysis would be complete without a performance comparison. The object of the benchmark was to compare the speed of each operating system in completing certain groups of jobs. Thus, the system that finished the group of jobs first was the winner. Additionally, analysis was performed on the manner in which the jobs were processed. This would give an indication of the characteristics of the system schedulers. These characteristics would affect the turnaround of large and small jobs. Sites with an overabundance of either type could draw the appropriate conclusion from our benchmarks.

DESCRIPTION OF THE BENCHMARK

The benchmark is divided into six categories (groups) of measurement:

1. FORTRAN compilations

2. Relocatable loading, generation of overlays and segments, overlay and fast overlay execution

3. Execution of scientific applications and other high-CPU programs

4. Execution of programs almost totally I/O bound

5. Execution of jobs that create, attach and purge 250 permanent files

6. All of the above in simultaneous execution

Except for the last group of jobs each group was tested by itself. Thus operating system strengths and weaknesses could be determined. It was also felt that this kind of testing would allow the reader to identify with the job mix most closely related to his site. An obvious group missing from the benchmark was the COBOL jobs. There is no COBOL compiler available at Rockwell and licensing agreements prevented us from simply copying one from another site. Communications was also not tested. This is next to impossible to do without a simulator and another mainframe. Table 8 has a complete list of each benchmark group, the jobnames of the jobs that comprise them and a description of each job.

Many of the jobs in each group were submitted more than once during the benchmark of the groups. For example, jobs G2REL1, G2REL2, G2REL3 and G2REL4 were submitted 50 times each. This was done in order to generate a substantial load on the system and provide a lengthy benchmark. Only the group 5 jobs were executed only once. The jobs were submitted via a feeder job and routed them into the input queue. A parameter was given to the procedure of fact, the "FEEDER" job executed almost through the life of each benchmark. Jobs were not held in the input queue until the feeder job completed and then released for execution. However, the feeder job was not locked in either. It was swapped on occasion. It was necessary to run the benchmark this way due to FNT problems. On some benchmarks there was no way to stack all the jobs without running out of FNTs. There will be more details on the feeder procedure later in this document.

The group 1 benchmark was designed to test the ability of each system to process large amounts of jobs which did nothing more than compile programs. In one sense, the FTN 4 compiler was tested as well. All the programs used the FTN compiler. The program source came from either purchased applications (IMSL, SAP, NASTRAN) or local subroutines developed over many years at Rockwell. Counting the multiple times the group 1 jobs were submitted a total of 4,244 routines were compiled during the group 1 benchmark.

Group 2 jobs tested simple relocatable loading and overlay and segment generation. Execution of regular and fast overlayed jobs was also tested. The simple relocatable load jobs (G2REL4) concentrated on satisfying externals located on different system libraries. They consisted of about 100 or so call statements. Each job had the call statements in different sequence in order to randomize the library search pattern. The segment generation test consisted of generating the 44 segments comprising the SAP application from the relocatable binaries. One of the overlay generation tests involved the generation of the 126 overlays needed to form the ASKA application.

An extra dimension was added to the overlay test. That was the comparison of overlay generation and execution of identical programs. The difference in the programs was that one job used the fast overlay loader and the other did not. The total number of overlays (MAIN, PRIMARY and SECONDARY) was 25. The overlays were loaded in random order. It should be noted here that although 844-21 disks were configured to contain execution files on NOS, the absolutes generated were copied to other files in order not to confine the programs to one disk. The same was done on NOS/BE.

The scientific applications benchmark was the longest single test (except for the composite, of course). It consisted of NASTRAN (NASA Structural Analysis Program) and SAP (Structural Analysis Program) executions. In addition, a noise program was added to the mix. This noise program used quite a bit of CPU time and a fair amount of I/O time as well. The noise job itself was quite small (22.4K). These jobs ensured almost complete CPU and disk channel usage through the life of the test. They also forced NASTRAN and SAP to compete more heavily for these resources, as one would expect in a heavily scientific environment. NASTRAN used 250K of CM while SAP used 355K. Direct access LCM (ECS) was not used. The NASTRAN version used was obtained from NASA. It is version 17.5. The NASTRAN binaries came straight from the released NASA tapes. The SAP program was obtained from the University of Southern California. No mods were made to either application to accommodate the different systems.

Operating system ability to process large numbers of jobs with tremendous input/output requirements was the object of the group 4 test. Job G4SORT sorted 100,000 150-character records using a 19-character key. Incidentally, PSR ST40390 was present in both systems. Job G4MIP worked with a MIP-AK-based file with 20 alternate keys. The file contained 400 records. The G4TECS job executed a program that wrote 20000 words, skipped back, read the record back and compared the two buffers. This was done 100 times.

The last group tested the quickness of each system in handling permanent files. Four jobs were submitted, each using the same procedure with different IDs. The control cards used were slightly different due to system JCL differences. Otherwise they were exactly the same. The procedure cataloged (defined on NOS) 250 permanent files, re-attached them and then purged the files. Thus 1000 permanent files were manipulated.

Finally, group 6 was run. Group 6 consisted of all-of-the-above. This obviously test the abilities of each system to handle a huge workload. In fact, this test processed 323 jobs. Needless to say, having a CYBER 176 around allowed us to process extreme workloads in a reasonable amount of time. This last group would probably take 15-20 hours on a CYBER 173.

A note about the operating systems themselves: the NOS version is 1.4 PSR level 501/498. No mods were made to any of the programs other than some constant changes outlined in the installation handbook (such as default permissions, magnet recall time, and the like). The product sets were at level 498 with identical local mods in both systems (a mod to FCL to generate a LDSET macro to search an extra library to satisfy externals). The NOS/BE system is Rockwell's level 499 system with a different CMR (smaller) to reflect the configuration in effect. IP.C176 was set to 1 and IP.819 to zero for this test. No speed-up code was installed in the NOS/BE system. PP and CP program residencies for each system were made as similar as possible (see the LIBDECK and MOVE DECK listings). Rockwell's system was used instead of the default system in order to minimize system build time. There wasn't enough time to test out a newly built NOS/BE system. Since hardly any local mods are in the system in the first place, it was felt that this system would be representative enough.

HARDWARE CONFIGURATION

Computer	CYBER 176
Memory	262,144 words
LCM	524,288 words
PPS	20
FLPPS	6 (not used--they drive 819s which are not supported on NOS)

Peripheral equipment used in this test:

Channel	Equipment
06,07	4-679 (1600,6205)--used for PF dump/load, not benchmark
11,13	7-669,1-667--used for PF dump/load not benchmark
20,21	3-844-41 (full track)
22,23	2-844-41 (full track
32	3-844-21 (half track)
33	4-844-21 (half track)

```
*         INSERT IPARAMS MODIFICATIONS HERE.............................. IPARAMS    15
 IP.ACNT  EQU    2             MUST HAVE AN ACCT CARD FOLLOWING JOB CARD  ==IPARAMS   2
 IP.CP    EQU    9                                                        ==IPARAMS   3
 IP.CPLM  EQU    10B                                                      ==IPARAMS   4
 IP.CR    EQU    96D                                                      ==IPARAMS   5
 IP.C64.2 EQU    1                                                        ==IPARAMS   6
 IP.CSET  EQU    IP.C64.2                                                 ==IPARAMS   7
 IP.ECSB  EQU    1                                                        ==IPARAMS   8
 IP.IQD   EQU    3                                                        ==IPARAMS   9
 IP.IUSID EQU    2RAC          LOWEST NON-HARDWIRED INTERCOM USER ID      ==IPARAMS  10
 IP.LVF   EQU    6                                                        ==IPARAMS  11
 IP.MECS  EQU    1000B         MAX ECS FOR JOB IS 1000000B.               ==IPARAMS  12
 IP.MFL   EQU    377700B       MAXIMUM AMOUNT OF FL ALLOWED A USER        ==IPARAMS  13
 IP.MMS   EQU    0                                                        ==IPARAMS  14
 IP.MPPU  EQU    20D                                                      ==IPARAMS  15
 IP.MPR   EQU    5                                                        ==IPARAMS  16
 IP.MSLM  EQU    0                                                        ==IPARAMS  17
IP.NDFS   EQU    1                                                        ==IPARAMS  18
 IP.NJFL  EQU    30B                                                      ==IPARAMS  19
 IP.OPRI  EQU    1                                                        ==IPARAMS  20
 IP.PFRP  EQU    5             DEFAULT RETENTION PERIOD                   ==IPARAMS  21
 IP.POSFL EQU    10B                                                      ==IPARAMS  22
IP.SCHDE  EQU    1                                                        ==IPARAMS  23
 IP.SECS  EQU    1000B         DEFAULT MAX LCM IS ONE MILLION OCTAL       ==IPARAMS  24
 IP.SFL   EQU    377700B       DEFAULT FL IF NONE SPECIFIED ON JOB CARD   ==IPARAMS  25
 IP.SPR   EQU    4                                                        ==IPARAMS  26
 IP.XJ    EQU    2                                                        ==IPARAMS  27
 IP.C176  EQU    1                                                        ==NOSBE    37
 IP.819   EQU    0                TURN OFF THE 819                        ==NOSBE    38
 MODEL    CMICRO 3,(176)                                                  ==IPARAMS  29
 PR.IDLE  EQU    0             IDLE MODE CPU PRIORITY                     ==IPARAMS  30
 PR.BATCH EQU    PR.IDLE+1     NORMAL BATCH JOB CPU PRIORITY              ==IPARAMS  31
 PR.INT   EQU    PR.BATCH      CPU PRIORITY FOR INTERCOM = BATCH.         ==IPARAMS  32
```

```
*          INSERT MODIFICATIONS HERE .................................... CMRIP       1
 L.EST     EQU     200B                                                   ==CFIG      2
 N.2550    EQU     2              TWO NPU'S CONFIGURED                    ==CFIG      3
 N.SYNCA   EQU     10             NUMBER OF SYNCRONOUS PORTS ON 2550A     ==CFIG      4
 N.1200A   EQU     10             NUMBER OF 1200 BAUD PORTS ON 2550A      ==CFIG      5
 N.300A    EQU     12             NUMBER OF 300 BAUD PORTS ON 2550A       ==CFIG      6
 N.AUTO3A  EQU     23             NUMBER OF MODE 3 AUTO-BAUD PORTS ON 2550A ==CFIG    7
 L.2550A   EQU     N.SYNCA+N.1200A+N.300A+N.AUTO3A+3                      ==CFIG      8
 N.SYNCB   EQU     10D            NUMBER OF SYNCHRONOUS PORTS ON 2550B    ==CFIG      9
 N.2400B   EQU     1              NUMBER OF 2400 BAUD ASYNC ON 2550B      ==CFIG     10
 N.1200B   EQU     10             NUMBER OF 1200 BPS PORTS ON 2550B       ==CFIG     11
 N.300B    EQU     16             NUMBER OF 300 BPS PORTS ON 2550B        ==CFIG     12
 L.2550B   EQU     N.SYNCB+N.2400B+N.1200B+N.300B+1+1                     ==CFIG     13
 L.ITABL   EQU     4                  2 NPUS + 2 EMPTY                    ==NOSBE     5
 L.FNT     EQU     1510B                                                  ==NOSBE     6
 N.DEVICE  EQU     12B                                                    ==CFIG     16
 N.RBR     EQU     60D                                                    ==CFIG     17
 N.SPRPP   EQU     4                                                      ==CFIG     18
IP.ECSTP   EQU     1                                                      ==CFIG     19
 L.ECSSWP  EQU     10B                SWQP ANY BATCH JOB TO ECS           ==NOSBE     7
IP.SYSL1   CMICRO  20,(ROCKWELL NOS/BE 1.3)                               ==SYSLAB    2
IP.VER     CMICRO  10,(LEVEL 488E)                                        ==SYSLAB    3
IP.SYSE    CMICRO  10,(  10/29/79)                                        ==SYSLAB    4
IP.CMPID   CMICRO  10,(176)                                               ==CFIG     22
 L.APF     EQU     250D                                                   ==NOSBE     8
 N.SETS    EQU     2                                                      ==NOSBE     9
 N.VDDT    EQU     16D                                                    ==CFIG     25
L.SCHJDT   EQU     600B                                                   ==CFIG     26
 N.ESD     EQU     256D                                                   ==CFIG     27
 N.SD      EQU     63D                                                    ==CFIG     28
 N.RBTC    EQU     512D                                                   ==CFIG     29
 N.RQS     EQU     100B                                                   ==NOSBE    10
 N.CP      EQU     17B                                                    ==CFIG     31




*                                                                         CMR      1526
*          SCHEDULER PARAMETER SETTINGS                                   CMR      1527
 MAXNBA    EQU     50B                                                    ==NOSBE    11
*                                                                         CMR      1528
```

```
**                                                                        CMR       2319
*                                                                         CMR       2320
*        INSERT EST ENTRIES HERE ........................................ EST          1
CC       EST    CH=02,EQP=0,UNIT=0                                        ==EST        2
DS       EST    CH=10,EQP=7                                               ==EST        3
 FE      EST    CH=03,EQP=7,ESTO=3,MUX=MUX1-T.ITABL                       ==EST        4
FE       EST    CH=04,EQP=7,ESTO=4,MUX=MUX2-T.ITABL                       ==EST        5
CR       EST    CH=12,EQP=4                                               ==EST        6
LR       EST    CH=12,EQP=6,EC=A6                                         ==EST        7
NT       EST    CH=(6,7),EQP=0,UNIT=0,UNITS=2,MOD=(ATS,GCR),ESTO=20       ==EST        8
NT       EST    CH=(7,6),EQP=0,UNIT=2,UNITS=2,MOD=(ATS,GCR)               ==EST        9
NT       EST    CH=(11,13),EQP=0,UNIT=0,UNITS=4,MOD=(MTS,BID),ESTO=30     ==EST       10
NT       EST    CH=(13,11),EQP=0,UNIT=4,UNITS=3,MOD=(MTS,BID)             ==EST       11
MT       EST    CH=(13,11),EQP=0,UNIT=7,UNITS=1,MOD=(MTS,BID),ESTO=37     ==EST       12
AZ       EST    NAME=844DA,CH=(20,21),EQP=0,UNIT=1,SN=NOSBE,VSN=NBE101,   ==NOSBE     12
,TYPE=F,MOD=(PF,QUE,SYS,SCR),ESTO=101                                     ==NOSBE     13
AZ       EST    NAME=844DB,CH=(20,21),EQP=0,UNIT=2,SN=NOSBE,VSN=NBE102,   ==NOSBE     14
,TYPE=F,MOD=(PF,QUE,SYS,SCR),ESTO=102                                     ==NOSBE     15
AZ       EST    NAME=844DC,CH=(21,20),EQP=0,UNIT=4,SN=NOSBE,VSN=NBE104,   ==NOSBE     16
,TYPE=F,MOD=(PF,QUE,SYS,SCR),ESTO=104                                     ==NOSBE     17
AZ       EST    NAME=844DD,CH=(22,23),EQP=0,UNIT=5,SN=NOSBE,VSN=NBE115,   ==NOSBE     18
,TYPE=F,MOD=(PF,QUE,SCR),ESTO=115                                         ==NOSBE     19
AZ       EST    NAME=844DE,CH=(23,22),EQP=0,UNIT=7,SN=NOSBE,VSN=NBE117,   ==NOSBE     20
,TYPE=F,MASTER=(PF,QUE,SCR,SYS),NF=9080D,NM=50D,MOD=(PF,QUE,SCR),ESTO=1   ==NOSBE     21
,17                                                                       ==NOSBE     22
AY       EST    NAME=844A,CH=32,EQP=0,UNIT=0,SN=NOSBE,VSN=NBE160,MOD=(,   ==NOSBE     23
,PF,QUE,SCR),ESTO=160                                                     ==NOSBE     24
AY       EST    NAME=844B,CH=32,EQP=0,UNIT=1,SN=NOSBE,VSN=NBE161,MOD=(,   ==NOSBE     25
,PF,QUE,SCR),ESTO=161                                                     ==NOSBE     26
AY       EST    NAME=844C,CH=32,EQP=0,UNIT=2,SN=NOSBE,VSN=NBE162,MOD=(,   ==NOSBE     27
,PF,QUE,SCR),ESTO=162                                                     ==NOSBE     28
AY       EST    NAME=844G,CH=33,EQP=0,UNIT=0,SN=NOSBE,VSN=NBE170,MOD=(,   ==NOSBE     29
,PF,QUE,SCR),ESTO=170                                                     ==NOSBE     30
AY       EST    NAME=844H,CH=33,EQP=0,UNIT=1,SN=NOSBE,VSN=NBE171,MOD=(,   ==NOSBE     31
,PF,QUE,SCR),ESTO=171                                                     ==NOSBE     32
AY       EST    NAME=844I,CH=33,EQP=0,UNIT=2,SN=NOSBE,VSN=NBE172,MOD=(,   ==NOSBE     33
,PF,QUE,SCR),ESTO=172                                                     ==NOSBE     34
AY       EST    NAME=844J,CH=33,EQP=0,UNIT=3,SN=NOSBE,VSN=NBE173,MOD=(,   ==NOSBE     35
,PF,QUE,SCR),ESTO=173                                                     ==NOSBE     36
         SPACE     4                                                      CMR       2322
         ENDTABLE  EST                                                    CMR       2323
```

```
**                                                                      CMR       2446
*                                                                       CMR       2447
*         INSERT INTERCOM MULTIPLEXOR TABLE DEFINITION HERE ............ MUX          1
 MUX1     MUX2550 1                                                     ==NOSBE      1
          EMPTY                                                         ==NOSBE      2
 MUX2     MUX2550 1                                                     ==NOSBE      3
          EMPTY                                                         ==NOSBE      4
                                                                        CMR       2449
                                                                        CMR       2450
          ENDTABLE   ITABL                                              CMR       2451
```

CMRDECK

```
NAME=80/01/01 ROCKWELL NOS 1.4
VERSION=LEVEL 501A
FNT=1000.
NCP=27.
EQ02=DS,ON,7,0,10.
EQ03=NP,ON,7,1,03,1.
EQ04=NP,ON,7,2,04,2.
EQ05=CR,ON,4,,12.
EQ06=LR-5,ON,6,,12.
EQ10=DL-N3,ON,0,01,02,04,20,21.
EQ11=DL-N2,ON,0,05,07,22,23.
EQ12=DI-3,ON,0,0,32.
EQ13=DI-4,ON,0,0,33.
EQ14=DE,ON,LE,2000.
EQ20=NT-4,ON,0,0,06,07,,,11.
EQ30=NT-4,ON,0,0,13,11,,,20.
EQ34=NT-3
EQ37=MT-1,ON,0,7,11,13,,,20.
MSAL,B=12,13.
MSAL,L=10,11,12,13.
MSAL,P=12,13.
MSAL,O=12,13.
MSAL,I=12,13.
MSAL,T=10,11,12,13,14.
MSAL,R=14,11,10,12,13.
PF=10,F,252,377,NOS,50,2.
PF=11,F,125,377,NOS,51,2.
PF=12,F,0,177,NOS,52,0.
PF=13,F,0,177,NOS,53,0.
PF=14,F,0,0,NOS,54,4.
SYSTEM=10.
ASR=11,14.
FAMILY=10.
LBC,FT,20,21,22,23.
LBC,HT,32,33.
UEC=1000
TEMP=14,10,11,12,13.
```

The NOS/BE system used for the benchmark executed with the following PP program in CM residence:

1AJ 1DL 1RN 1SP 1SX 1S5 3DØ

3SY 4DØ 7ID 8DA 8DM 8DN 8XA

8X8 A

The following programs had ECS residency:

ACT APR CEM CIØ CØN DSM DSP

EPF FAD FIN FNT IAP IUP JAC

LDD LDL LDQ LDV LDW LPF MAC

MEM MSG MUJ PAK PFA PFC PFE

PFP QAC QAF REQ TBL 1BR 1CT

1EV 1FC 1IB 1IM 1IQ 1IR 1IT

1MH 1NP 1PK 1QP 1SC 1SI 1SØ

2CC 2CS 2FC 2IA 2IB 2IR 2IW

2ND 2NP 2PK 2RN 3II 3IM 3IR

3IW 3MN 3ND 3NP 3RQ 3TT 3T1

3T2 4ND 4NP 5ND 5NP 6MN 6ND

6NP 6RD 6PM 7AJ 7EC 7ND 7RQ

8D8 8DC 8DF 8DH 8DL 8DP 8DR

8DV 8DX 8ND 8XB 8XC 8XD 8XE

8XG 8XI 8XL 8XM 8XN 8XØ 8XP

8XQ 8XR 8XS 8XT 8XU 8XW PFCCP

LØADER

The remainder of the operating system was resident on a full-track 844-41 dual access disk. The standard residency was used for CMR segments and library name tables.

LIBDECK

```
*CM     PP/9AA,9AB,9AM
*CM     PP/1DC
*CM     PP/CIO,1AJ,ODF
*CM     PP/9A1,9A5,9A6,9A7 (DSD RELATED)
*CM     PP/6DP,7DP,7MP,7RP,7SP,6DE,7DE,BCS,BCF
*CM     PP/3MB, (TAPE ERROR RECOVERY)
*AD     14,PP/1DS,9AC,9AD,9AE,9AF,9AG,9AH,9AI,9AJ
*AD     14,PP/9AK,9AL,9,AR,9AS,9AT,9A8,9A9,9BA,9BE,9BF
*AD     14,PP/9BI,9BJ,9BK,9BL,9BN,9BO,9BP,9BQ,9BU
*AD     14,PP/1DL
*AD     14,PP/0RF,0FA,0RP,0BF,0AV,0AU
*AD     14,PP/QAC,3QS
*AD     14,PP/1CJ,1CK,TCS,LDR,1MA,2MA,3AA,3AB,3AD,3AE,3AF,3AG
*AD     14,PP/LDD,LDQ
*AD     14,PP/9BV,CMS,3PF,3PO,PFU,3FA,3SZ
*AD     14,PP/DSP,1CD,1IO,6DI,7DI,7WI,OTI,OPI,OCI,7EP
*AD     14,PP/2CA,2CB,2CC,2CD,2CE,2CF,2CG,2CH,2CI
*AD     14,PP/1RI,3RG,3RH,3RI,1RO,3RP,3RQ
*AD     14,PP/LFM,3LB,3LF,3LG
*AD     14,PP/PFM,3PA,3PB,3PD,3PE,3PG,3PH,3PI,3PK
*AD     14,PP/RPV,CPM,3CA,3CB,3CC
*AD     14,PP/1TA,3TC,3TD,3TE,3TK,3TF,3TJ,TLX,1TO,2TO
*AD     14,PP/1MT,3MG,3MH,3MH,3MI,3ML,3MN,3MT
*AD     14,PP/1IO,1SJ,1SP
*AD     14,PP/1LS
*AD     14,ABS/FILES,PFILES,CATLIST,CTL2,CTL3,EDIT,RESEX
*AD     14,ABS/ACCFAM,MODIFY,LOADER,CHARGE,COPYB,RWF
*AD     14,OVL/LDC
*AD     14,ABS/COMPASS
*AD     14,OVL/COMP3$,COMP3$A,MSORT
```

IPRDECK

```
TDEN=PE.
TDTR=NT.
KEYPM=29.
CSM=64.
LOCK.
EI200.
VALID.
QUEUE,SY,IN,OP7757,LP700,UP3000.
QUEUE,SY,RO,OP6000,LP100,UP1000.
SERVICE,SY,PR70,CP400,CM200,FL3777,AM3777,EC7777.
SERVICE,SY,EM1000,FC0,CS1,FS1,DS0.
QUEUE,BC,IN,OP2400,LP2000,UP4010.
QUEUE,BC,RO,OP2400,LP1010,UP4004.
QUEUE,BC,OT,OP200,LP100,UP7000.
SERVICE,BC,PR30,CP400,CM200,FL3777,AM3777,EC7777.
SERVICE,BC,EM10000,FC0,CS1,FS1.
QUEUE,EI,IN,OP3400,LP2400,UP4010.
QUEUE,EI,RO,OP3400,LP1400,UP4006.
QUEUE,EI,OT,OP200,LP100,UP7600.
SERVICE,EI,PR30,CP400,CM200.
QUEUE,TX,IN,OP4000,LP3770,UP7006.
QUEUE,TX,RO,OP4004,LP3740,UP7000.
QUEUE,TX,OT,OP200,LP100,UP7000.
SERVICE,TX,PR30,CP40,CM10.
QUEUE,MT,IN,OP6774,LP6700,UP7400.
QUEUE,MT,RO,OP6774,LP4000,UP7400.
QUEUE,MT,OT,OP6000,LP100,UP7700.
SERVICE,MT,PR31,CP400,CM60.
QUEUE,NS,IN,OP7374,LP7360,UP7500.
QUEUE,NS,RO,OP7374,LP7350,UP7500.
QUEUE,NS,OT,OP500,LP100,UP7700.
SERVICE,NS,PR73,CP400,CM200.
DELAY,JS1,CS10,AR1000.
QUEUE,SY,OT,OP400,LP100,UP7700.
```

```
DSD,0,MAI%X.QREC(NK)
DSD,3,AUTO.
SCP.
MS VALIDATION.
PF VALIDATION.
SRST=20.
NAM.
IAF.
RBF.
```

DESCRIPTION OF THE SUBMITTAL PROCEDURE

The jobs were submitted by means of a job which was running during each of the benchmark runs. The job first attached a file for each unique job to be submitted for the run. These files were the complete job stream for the respective jobs. Then, by means of the CYBER Control Language's control registers, a control loop was executed which submitted the desired number of jobs of each type of the input queue. The same job deck was used on the NOS/BE test as on the NOS test with the exception that the ACCOUNT card used on the NOS/BE test was exchanged for a USER control card used on the NOS test. The proper NOS/BE or NOS control statements were executed in the job by determining which system the job was running on from the CCL 'SYS' function.

Two instream procedures were used for the processing of the job. One was 'AT' which attached the job streams to be submitted to the input queue. The other was 'SUB' which was used to route the job to the input queue, create a hard copy of the input job, and to return the file when it was not needed any more.

A large loop was executed several times in the job. Within this loop were 5 smaller groups representing the groups which were tested in the benchmark. For groups 1 through 5 only the jobs in one of the smaller groups were submitted within the large loop. However, for the composite run, a few jobs from each group were submitted each time through the large loop. Thus the system was not flooded with all the jobs of one type giving a more realistic input job mix. For many of the groups, more than one copy of a specific job was submitted for each time through the large loop. In particular, for group two which consisted of small loader jobs, up to ten copies of some of the jobs were submitted in each large loop with fifty copies for the benchmark.

```
GROUPX,T7777.JOB TO SUBMIT BENCHMARK JOBS. SET,R1=GROUP.
USER(ADON,ELLA)
COMMENT.
COMMENT. SET R1 EQUAL TO THE BENCHMARK NUMBER.
SET,R1=6.
REWIND,OUTPUT.
COPYBR,INPUT,SUB.
COPYBR,INPUT,AT.
REWIND,AT,SUB.
IFE,R1.EQ.1.OR.R1.EQ.6,A1.
AT,G1SCC,GROUP1.
AT,G1NAST,GROUP1.
AT,G1SAP,GROUP1.
AT,G1IMSL,GROUP1.
AT,G1NFM,GROUP1.
ENDIF,A1.
IFE,R1.EQ.2.OR.R1.EQ.6,A2.
AT,G2SAP,GROUP2.
AT,G2REL1,GROUP2.
AT,G2REL2,GROUP2.
AT,G2REL3,GROUP2.
AT,G2REL4,GROUP2.
AT,G2ASKA,GROUP2.
AT,G2OVL,GROUP2.
AT,G2FOL,GROUP2.
ENDIF,A2.
IFE,R1.EQ.3.OR.R1.EQ.6,A3.
AT,G3NAST,GROUP3.
AT,G3SAP,GROUP3.
AT,G3NOISE,GROUP3.
ENDIF,A3.
IFE,R1.EQ.4.OR.R1.EQ.6,A4.
AT,G4SORT,GROUP4.
AT,G4MIP,GROUP4.
AT,G4TECS,GROUP4.
ENDIF,A4.
IFE,R1.EQ.5.OR.R1.EQ.6,A5.
AT,G5PF1,GROUP5.
AT,G5PF2,GROUP5.
AT,G5PF3,GROUP5.
AT,G5PF4,GROUP5.
ENDIF,A5.
SET,R1G=1.
WHILE,R1G.LE.6,ENDSUB.
COMMENT.+++++++++++++++++++++++++++++++++
COMMENT. BEGINNING OF LOOP AGAIN
DISPLAY,R1G.
COMMENT.+++++++++++++++++++++++++++++++++
IFE,R1.EQ.3.OR.R1.EQ.6,GP3.
SUB,G3SAP,5.
```

```
SUB,G3NOISE,6.
SUB,G3SAP,4.
SUB,G3NOISE,5.
SUB,G3NAST,2.
ENDIF,GP3.
IFE,R1.EQ.1.OR.R1.EQ.6,GP1.
SUB,G1SCC,5.
SUB,G1SCC,5.
SUB,G1SCC,5.
SUB,G1SCC,4.
SUB,G1NAST,5.
SUB,G1NAST,4.
SUB,G1SAP,4.
SUB,G1IMSL,4.
SUB,G1NFM,5.
SUB,G1NFM,5.
SUB,G1NFM,4.
ENDIF,GP1.
IFE,R1.EQ.4.OR.R1.EQ.6,GP4.
SUB,G4SORT,4.
SUB,G4TECS,7.
SUB,G4SORT,4.
SUB,G4TECS,7.
SUB,G4SORT,3.
SUB,G4TECS,6.
SUB,G4MIP,3.
ENDIF,GP4.
IFE,R1.EQ.2.OR.R1.EQ.6,GP2.
SUB,G2ASKA,6.
SUB,G2SAP,6.
SUB,G2REL1,6.
SUB,G2REL1,6.
SUB,G2REL1,6.
SUB,G2REL1,6.
SUB,G2REL1,6.
SUB,G2REL1,6.
SUB,G2REL1,6.
SUB,G2REL1,6.
SUB,G2REL1,6.
SUB,G2REL2,6.
SUB,G2REL2,6.
SUB,G2REL2,6.
SUB,G2REL2,6.
SUB,G2REL2,6.
SUB,G2REL2,6.
SUB,G2REL2,6.
SUB,G2REL2,6.
SUB,G2REL2,6.
SUB,G2REL3,6.
SUB,G2REL3,6.
```

```
SUB,G2REL3,6.
SUB,G2REL3,6.
SUB,G2REL3,6.
SUB,G2REL3,6.
SUB,G2REL3,6.
SUB,G2REL3,6.
SUB,G2REL3,6.
SUB,G2REL4,6.
SUB,G2REL4,6.
SUB,G2REL4,6.
SUB,G2REL4,6.
SUB,G2REL4,6.
SUB,G2REL4,6.
SUB,G2REL4,6.
SUB,G2REL4,6.
SUB,G2REL4,6.
SUB,G2REL1,5.
SUB,G2REL2,5.
SUB,G2REL3,5.
SUB,G2REL4,5.
SUB,G2SAP,5.
SUB,G2ASKA,5.
SUB,G2OVL,2.
SUB,G2FOL,2.
ENDIF,GP2.
IFE,R1G.LE.1,GT1G5.
IFE,R1.EQ.5.OR.R1.EQ.6,GP5.
SUB,G5PF1,1.
SUB,G5PF2,1.
SUB,G5PF3,1.
SUB,G5PF4,1.
ENDIF,GP5.
ENDIF,GT1G5.
SET,R1G=R1G+1.
ENDW,ENDSUB.
IFE,SYS=NOSB,REW.
COMMENT. ******************************
COMMENT. END OF JOB GENERATION
ELSE,REW.
NOTE(OUTPUT,NR)/1 END OF JOB GENERATION.
NOTE./1  END OF JOB GENERATION.
ENDIF,REW.
```

```
.PROC,SUB,XX,N=1.
IFE,R1G.LE.N,DONE.
IFE,FILE(XX,AS),S.
COMMENT. -------------------------
COMMENT. ROUTE N COPIES OF XX TO INPUT
COMMENT. -------------------------
IFE,R1G.EQ.1,COPY.
REWIND,XX.
COPYSBF,XX.
ENDIF,COPY.
REWIND,XX.
COPYBF,XX,JOB.
ROUTE,JOB,DC=IN.
IFE,R1G.EQ.N,NRET.
RETURN(XX)
ENDIF,NRET.
ELSE,S.
COMMENT. *************************
COMMENT. XX NOT SUBMITTED
COMMENT. XX NOT ATTACHED TO THIS JOB
COMMENT. *************************
ENDIF,S.
ENDIF,DONE.
REVERT.
```

```
.PROC,AT,FILE,U.
COMMENT.  ATTACH FOR FILE GROUP U
IFE,SYS=NOSB,ENOSBE.
ATTACH,FILE,ID=ADON.
ELSE,ENOSBE.
ATTACH(FILE/UN=ADON,NA)
ENDIF,ENOSBE.
REVERT.
EXIT.
REVERT.
```

Benchmark Results

The following Tables 1-6 show the comparison, by group, between NOS/BE and NOS with respect to real time in seconds, CPU time in seconds, CPU utilization computed by CPU/R.T., and performance ratios relative to the NOS/BE data.

For the 23 unique jobs (each job was run multiple times for a total of 323 jobs) run in the composite group 6, the average throughput (real time) is shown for each job in Table 7. Included is the average real time for NOS and NOS/BE along with the performance ratio relative to NOS/BE.

TABLE 1. Group 1 (45 jobs)

OP SYS	REAL TIME	CPU TIME	% CPU UTILIZATION
NOS	1968	871.3	44.2
NOS/BE	3480	907.4	26.1
RATIO	0.57	0.96	

TABLE 2. Group 2 (225 jobs)

OP SYS	REAL TIME	CPU TIME	% CPU UTILIZATION
NOS	1146	281.5	24.5
NOS/BE	1794	362.6	20.2
RATIO	0.64	0.78	

TABLE 3. Group 3 (21 jobs)

OP SYS	REAL TIME	CPU TIME	% CPU UTILIZATION
NOS	3593	2294.2	63.8
NOS/BE	3657	2287.5	62.6
RATIO	0.98	1.003	

TABLE 4. Group 4 (31 jobs)

OP SYS	REAL TIME	CPU TIME	% CPU UTILIZATION
NOS	2285	518.6	22.6
NOS/BE	2499	607.0	24.3
RATIO	0.91	0.85	

TABLE 5. Group 5 (5 jobs)

OP SYS	REAL TIME	CPU TIME	% CPU UTILIZATION
NOS	584	2.1	0.3
NOS/BE	587	17.3	2.9
RATIO	0.99	0.12	

TABLE 6. Group 6 (323 jobs)

OP SYS	REAL TIME	CPU TIME	% CPU UTILIZATION
NOS	8702	4016.3	46.1
NOS/BE	9413	4166.4	44.3
RATIO	0.92	0.96	

JOB NAME	JOB COUNT	NOS REAL TIME	NOS/BE REAL TIME	RATIO
G1SCC	16	1231	1224	1.01
G1NAST	8	2529	2942	0.86
G1SAP	4	3798	3118	1.22
G1IMSL	4	5102	3865	1.32
G1NFM	12	956	1331	0.72
G2ASKA	10	1214	1388	0.88
G2SAP	10	811	1374	0.59
G2REL1	50	243	1199	0.20
G2REL2	50	131	1199	0.11
G2REL3	50	218	1168	0.19
G2REL4	50	72	1173	0.06
G2OVL	2	1911	1920	0.99
G2FOL	2	1760	3457	0.51
G3SAP	8	5150	4939	1.04
G3NOISE	10	823	1482	0.56
G3NAST	2	8148	9018	0.90
G4SORT	9	3068	1918	1.60
G4TECS	18	1192	1187	1.00
G4MIP	3	6395	5336	1.20
G5PF1	1	1286	3130	0.41
G5PF2	1	1199	3316	0.36
G5PF3	1	746	3340	0.22
G5PF4	1	1175	3300	0.36
(AVERAGE)	323	906	1560	0.58

TABLE 8. BENCHMARK JOB DESCRIPTIONS

Group 1 FORTRAN Compilations

G1SCC	Compile 121 FORTRAN routines from local PL
G1NAST	Compile 186 FORTRAN routines from NASTRAN PL
G1SAP	Compile 352 SAP programs from PL
G1IMSL	Compile 488 IMSL routines from PL
G1NFM	Compile large program from PL

Group 2 Execute jobs that exercise the loader

G2ASKA	Load ASKA system (generate 126 overlays)
G2SAP	Load SAP system (generate 44 segments)
G2REL1	Load program which has many calls to routines from multiple libraries
G2REL2	Variation of the calling sequence on G2REL1
G2REL3	Variation of the calling sequence on G2REL1
G2REL4	Variation of the calling sequence on G2REL1
G2OVL	Load and execute large program with 25 overlays. Overlays called in random order.
G2FOL	Same as G2OVL except that the fast overlay loader was used.

Group 3 Execute scientific job mix

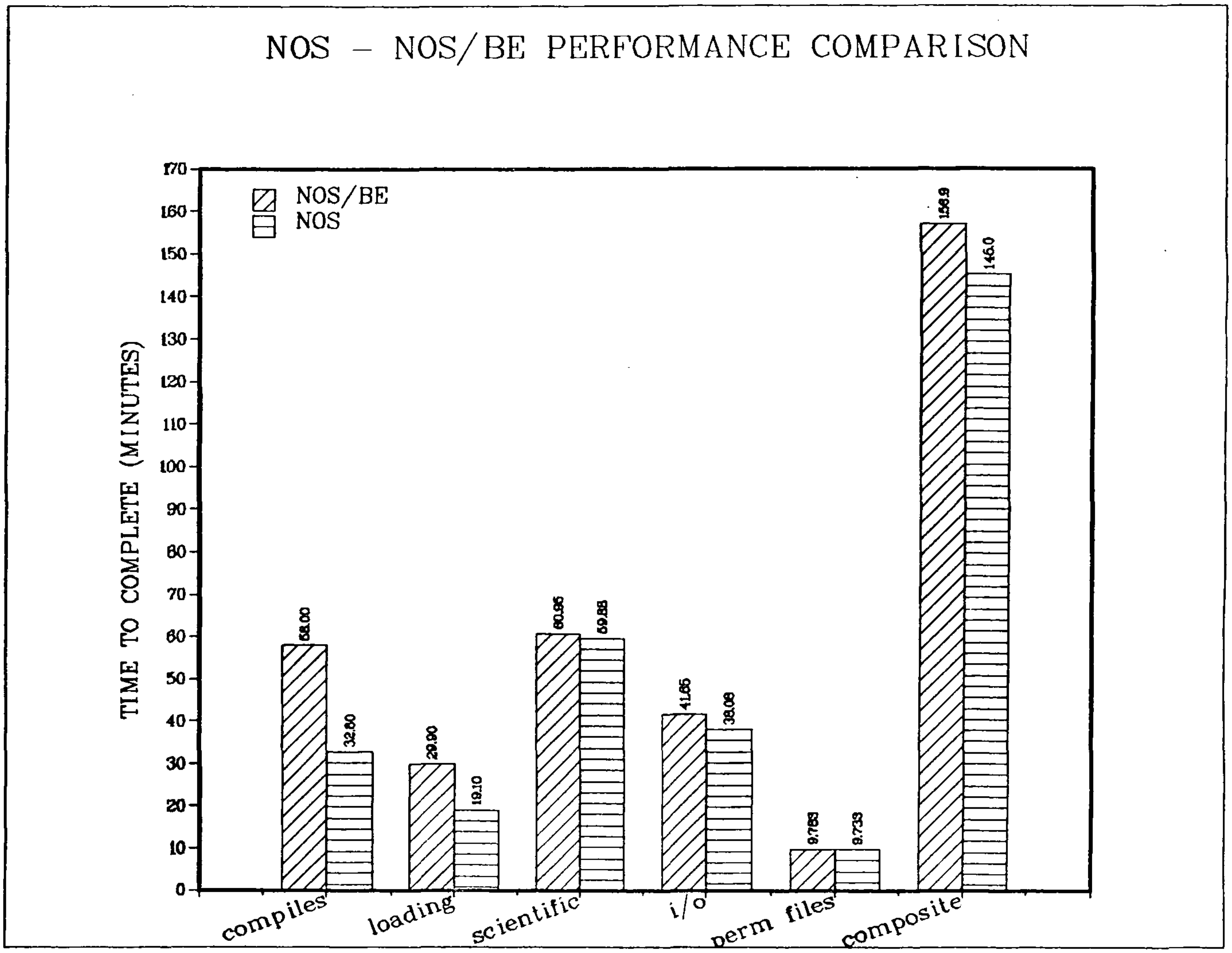
G3SAP	Execute SAP program (355K)
G3NAST	Execute NASTRAN program (250K)
G3NOISE	Execute high CP, medium IO program (22.4K)

GROUP 4 Execute jobs with high I/O requirements

G4SORT	SORT 100000 150-character records with 18-character key
G4TECS	WRITE 20000 words, read back, compare--100 records
G4MIP	Generate MIP/AK-based file with 20 alternate keys--400 records

TABLE 8. BENCHMARK JOB DESCRIPTIONS (con't)

Group 5		Execute jobs that manipulate permanent files
	G5PF1	Catalog (define on NOS) 250 permanent files, attach them, purge them
	G5PF2	Same as G5PF1 with different ID/USERNUMBER
	G5PF3	Same as G5PF1 with different ID/USERNUMBER
	G5PF4	Same as G5PF1 with different ID/USERNUMBER
Group 6		All of the above


NOS – NOS/BE PERFORMANCE COMPARISON
▨ NOS/BE
☰ NOS
TIME TO COMPLETE (MINUTES)
0
10
20
30
40
50
60
70
80
90
100
110
120
130
140
150
160
170
58.00
32.80
29.90
19.10
60.95
59.88
41.65
38.08
9.763
9.733
156.9
146.0
compiles
loading
scientific
i/o
perm files
composite

COMMENTS ON OPERATING A NOS SYSTEM

During the NOS benchmark many operational differences were found. These differences are the type that difference documents usually do not elaborate on. The reader can judge for himself how severe or critical these differences are:

- No backspace of a print file is available.

- The size of the file is not displayed on the H,O display.

- The DISPLAY,FNT ordinal is real nice and handy.

- PFDUMP does not use reprieve on tape errors. Thus the dump must be started from scratch instead of EOT simulated as on NOS/BE.

- No way to know which tape a permanent file was dumped to. The VSN does not appear in the report. Restoration from back-up tape is difficult.

- In dumping permanent files to more than reel of tape, ensure that the VSNs are specified in advance. Otherwise all tape loading must be to the same unit. Thus you must wait for the current reel to unload before starting on the next one.

- PURGALL is a very handy command.

- No FNT threshold is present. A count like in the NOS/BE S display is also nonexistent. We ran out of FNTs during one of the benchmarks.

- Cannot lockin time-critical jobs.

- No page eject on start of the banner page. Thus part of the dayfile from the preceding job appears on the top of the banner page of the next one.

- No control card such as PAUSE on NOS/BE to inform operations.

- BLANK only blanks one tape. Must reinitialize it for every tape.

- Must blank label all scratah tapes because the system will not overwrite an unexpired tape label.

- Love the K display.

- Can't track job by jobname.
- When operations ends the printer, no message appears on the output file.
- Can't have more than one banner page and dayfile.
- NOS provides the capability to assign specific file types to designated mass storage (and ECS/LCM) devices. Such files include Rollout, LGO, Output, Primary, etc.

BENCHMARK OBSERVATIONS

This section presents an attempt to explain some of the results of the benchmark. Though careful analysis was performed on the results, other explanations than those presented here may be valid as well.

It appears that the main reason NOS won all the benchmark runs was that its system was distributed upon multiple 844 spindles. Logical device I/O was composed of three 844-41 drives. When NOS writes on a logical device it writes a logical track (343B PRUs) on the first physical device, then the second, etc. Thus the system was evenly spread across the three packs. Additionally, programs such as the FTN compiler were also spread among the three devices. Contention, therefore, is substantially reduced. NOS/BE, on the other hand, writes as much of the system as possible to one device before overflowing to another. Thus, all the contention is directed toward one unit. It is clear that NOS would win any benchmark which involved high system disk activity. This fact was borne out by the results of the group 1 test in particular, which involved substantial FTN compiler overlay loading off the system device. Group 2 required heavy searching of system libraries to resolve external references for relocatable loads. NOS also won this test by a wide margin for much the same reason.

There may be one drawback in the NOS method of distributing file data across many spindles. This disadvantage may appear in jobs which read/write blocks of data which exceed the logical track size of either 14528 (844-41) or 7168 (844-21) words or in swapping. Many extra disk seeks may be necessary to accommodate the large I/O request. NOS/BE, on the other hand, would be able to satisfy the I/O request on the same unit. To prove this theory, however, it will be necessary to run group 4 again with strict control over file residency and job mix.

Speaking of I/O, it will probably be a shock to every NOS/BE analyst to read thaat NOS won the group 4 benchmark. It was (and still is, I think) widely believed that the NOS/BE stack processor concept would out-perform the "every PP for itself" method of NOS I/O. Well, NOS won. The reason for this surprising win is not in the stack processor theory itself. In an ECS (LCM)-based system, RMS I/O is first processed by the CMR segment CPCI/O. The I/O scheme starts with the user program issuing an XJ with the CI/O call in RA+1. Then CPCI/O is loaded from ECS. It in turn loads CP4ES to issue the stack request which is processed by the CM-resident CPSPM. CPSPM, after choosing the best request to work on from the multitude received, passes the request to the PP program 1SP to actually perform the I/O. Since all the system CPU programs execute in monitor mode, only one program can execute at a time. In an environment loaded with I/O requests, many of the exchange jumps from the user programs will wait for the other user's I/O request to be passed to the request stack or 1SP. Hence, you revert to a serial processing technique for RMS requests. NOS, on the other hand, uses PPCI/O for its I/O processing. It loads the appropriate overlays to perform the actual input/output. As long as there are sufficient numbers of PPS the benefits of parallel processing are present.

To successfully run this test on NOS, however, it is mandatory to have a computer with 20 PPS. There were almost no idle PPS for the duration of the group 4 benchmark, whereas on NOS/BE the PPS were barely used. This will represent a major headache to sites wishing to convert to NOS based on mainframes with less than 20 peripheral processors and a heavy I/O workload.

As expected, the elapsed time to run the group 6 (composite) benchmark was less than the sum of the individual benchmarks (groups 1-5). What was most remarkable was the performance of NOS/BE in handling the total job mix. NOS executed group 6 in 9.1 percent less time than the sum of groups 1-5. NOS/BE, on the other hand, saved 21.6 percent. Two possible explanations present themselves. The first is that with this job mix the load on the system disk is less pronounced. The group 1 jobs were not running alone and just loading overlays. They had to share resources with all the other groups which did not necessarily require system disk access. The other possible explanation is the performance of the NOS/BE scheduler. It may be that the NOS/BE scheduler is more adept at processing large numbers of different kinds of jobs. One must bear in mind, however, that the default scheduling parameter were used in both systems. With proper tuning the scheduler performances could be dramatically changed one way or the other.

With all this discussion, it must be remembered that in the end, NOS still won group 6. The margin of victory, however, was extremely slight. As pointed out in Table 7 the individual job turnaround on NOS/BE is higher than NOS. A possible reason lies in the concept of limits on the concurrent execution of jobs, e.g. the MAXN on NOS/BE. NOS treats each input queue job as a job simply residing in one of the scheduler's queues awaiting execution with its own scheduling queue priorities. NOS has no limits on the number allowed to execute. The MAXN on our NOS/BE system was set to 50B.

The following are miscellaneous observations:

- Though the group 5 (permanent file test) elapsed time was nearly the same for both systems, NOS/BE used much more CP time in its permanent manipulations.

- NOS provided more usable field length (742500 vs. 727300). Though every attempt to made to equalize the field lengths (including dropping Janus), we were unsuccessful. The RBT chain usage exacerbated the problem.

- The smaller number of control points (17B vs. 27B) on NOS/BE did not seem to adversely affect the benchmark results. On both systems, though, there were occasions when all control points were busy. If you plan to run an interactive workload in addition to a heavy batch load, the extra control points on NOS would probably be a benefit.

- Ran into an FNT critical situation five times on NOS/BE during the group 6 benchmark. This was without 819s thus proving that the FNT problem is not only related to CYBER 176 installations as some would have us think. NOS ran out of FNTs also once, thus forcing us to re-run the group 6 benchmark using PURGALL to evict the output files. APR,11 was used on NOS/BE to prevent a queue buildup. This function wwould be nice to have on NOS as well. A display of the number of FNTs remaining and possibly a FNT threshold would also be nice features to have on NOS.

CONCLUSION

It would be very nice to tell you that you should or should not convert to NOS. That, however, is neither prudent nor fair to you or your site. Each site will have to make its own decisions for its own reasons. It is hoped that this white paper clearly presented many of the important facts and differences needed to know in order to intelligently decide whether or not to convert. The benchmark presented in this paper is just one of many possible types of benchmarks needed to conclusively rate the systems. It is suspected, however, that the results obtained in this benchmark will be present in other benchmarks as well.

Well over 1500 man-hours were spent preparing this white paper (including the benchmark). It is not known how many man-hours were spent preparing the individual different documents that were used to prepare this one. The obvious conclusion is that the research and analysis that each site will have to invest before a conversion is even attempted is enormous. If this white paper has accomplished nothing else, let it be a reminder to all that a migration to NOS or even a decision to consider a conversion should be thought through very carefully. It is perhaps the biggest decision that you or your management will make this year.

ye8973A-9