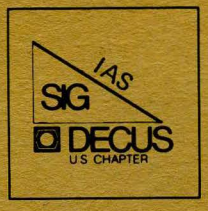
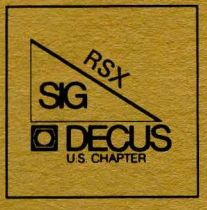
DECUS

The DeVIAS Letter



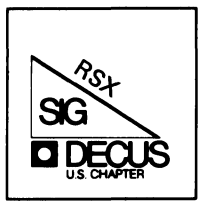


The RSX Multi-Tasker

February 1984

Printed in the U.S.A.

Section One



The RSX Multi-Tasker

"Table of Contents"

"From the Editors"

Most of the material this issue contains is either written by the editors, received last month just after deadline, or copied from Symposia Session notes or scribe material. About 45 pages of the issue were written by the editors or Ralph Stamerjohn, the SIG Chairman.

We have not received a single contribution, letter, question, -- not nothin -- in the last month. I refer to the period between December 9, 1983 and January 9, 1984. That is a bad time for extracurricular activity, I realize, but even so I expected something.

Reader! We desperately need your help. Please send in at least your questions or complaints; while they may not seem constructive, they will stimulate other readers to respond with answers or comments, and soon the Multitasker will be bulging its bindings again.

Update B for RSX11M V4.1 has been out for a while as I write this, and it is supposed to be released this month (January) for M-PLUS V2.1. What has been your experience in installing it? What did you gain? What did you lose? Were there any problems? Does it appear to be a reliable release? Let other users know. It is rumored (by Carl Friedberg of the VAX SIG) that these updates incorporate vectored resident libraries for FCS (at least). Is it true? Do they work, and do they help? How? How do you like EDT V3.0?

We want to make the Multitasker a valuable resource once again, but we are entirely dependent on your contributions. Please send machine-readable material if your contribution exceeds one page.

Submitting Multi-Tasker Articles

If you share our belief in RSX, then show it by sending articles, letters, questions, and discoveries to the Multitasker. Let's make it a rewarding experience to read every issue.

ACCEPTABLE MEDIA FORMAT

We are quite flexible in the formats in which articles can be submitted. All articles should be sent in some machine readable format, accompanied by hard copy if possible. The following media are acceptable:

Magnetic tape:
800, 1600 or 6250 BPI
FLX, BRU, PIP, or VMS BACKUP

Floppy Disk:
RX01 or RX02
ODS-1 or ODS-2

Any media sent to us will be returned to you. It would help if you include a return address label.

In addition, for last minute submissions arrangements can be made to dial in to a special account on Allen Watson's VAX. Please phone him directly for details.

FORMAT OF ARTICLES

The contents of the media should be in RUNOFF format, preferably. If you don't know RUNOFF then send straight unformatted text. The RUNOFF we use is VAX Digital Standard Runoff; John Clement's version on the RSX tapes has done a good job of maintaining compatibility with DSR. We use the following formats for most articles:

```
.PAGE SIZE 58,72
.layout 3,2 (page numbers at bottom)
.RIGHT MARGIN 72
.LEFT MARGIN 5
.SET PARAGRAPH 5,1,2
.AUTOPARAGRAPH
.TITLE RSX MULTITASKER
.NO SUBTITLE
```

This allows you 67 characters per line. Many program examples and command files will have to be edited if they exceed 67 characters per line; it would be very helpful if you tested things through RUNOFF first using these parameters and edited any long lines into what you would prefer they be. Otherwise we will have to do it, and you will have to take our decision on where to break lines. To make our editing easy, put all the above commands right at the top of your file while you are testing, and then remove them when you are done, replacing them with this:

```
.REQUIRE "[MULTITSK]ARTICLE.HED"
```

DEADLINES

Deadlines for the Multitasker are as follows, and will be strictly observed:

Submissions:	Second Friday of preceding month
Mail to DECUS:	Third Friday of preceding month

This means, for example, that the deadlines for the March issue are, for articles to reach us, February 10; for our camera ready copy to be mailed to DECUS, February 17. Sometimes (but not usually) material received between the second and third Fridays will be able to be included in the issue. Most often we will put the issue together the weekend after the second Friday and mail it to DECUS the following Monday or Tuesday.

Please notice that for material to reach us (the editors) by the second Friday of a month, you will have to mail it to DECUS about two weeks earlier. They will then re-mail the material to us. Although this seems inconvenient it keeps a single mailing address for Multitasker material even if editorship changes.

If you have some time-dependent material and are afraid it will not reach us in time through normal channels, phone one of us directly and we will help you get it to us.

All submissions should be sent to:

Multitasker Editor
c/o DECUS
One Iron Way
MR2-3/E55
Marlboro, MA 10752

Allen A. Watson
(201) 646-4111

Dominic DiNollo
(914) 968-2500 Ext. 2207
Multitasker Editors

"Hints 'N Things"

"Hints 'n Things" is a month potpourri of helpful tidbits and rumors. Readers are encouraged to submit items to this column. Any input about a way to make life easier on RSX is needed. Please beware that tiems in this column have not been checked for accuracy. Send any contributions to Multi-Tasker - "Hints 'N Things", c/o DECUS, One Iron Way, MR2-3/E55, Marlboro, MA 01752.

RSX-11M PLUS V2.1 Problems

Carl Friedberg
Seaport Systems, New York

I recently completed two separate M Plus V2.1 Sysgens. Some problems which have occurred are:

- Although non-privileged tasks built without any external library support should be transportable from earlier versions of M Plus or M, TKBBIG (from + V1.0 or M V3.2) dies during initialization with a trap 4 (probably execution of a halt [0]). (We are running on a 768. KW 11/70).

  A quick look through the post mortem dump suggests that much of low core has been scribbled on. TSC has so far been unable to duplicate the problem. Their suggested workaround was to use TKB from V2.0 (which we never sysgen'ed). Aside from a message that module OVDAT multiply defined $$RTS, it does seem to work. This strange set of circumstances comes about if you have decided not to upgrade to BASIC PLUS-2 V2xx; then you have to use version 1.0 SYSLIB, BASRMS, and RMS OLB's, as well as using the earlier task builder. The current task builder on V 2.1 has been completely rewritten to support I/D overlays, version 2 RMS, etc, and was not usable for this environment.

- Standalone BRU ([6,54]BRUSYS) seems to have some problems, particularly when creating or restoring IMAGE (multiple volume floppy disk) save sets; this is documented in the Dispatch. However, the previous workaround of installing BRU non-checkpointable and then fixing it in memory, along with MOU and DMO, no longer works, as BRU is now OVERLAID (from disk, natch) and can't be fixed in memory.

There are two major overlay legs, and it looks like a difficult chore to create memory resident overlays. DEVELOPERS: How about documenting this kind of change in the release notes; and can we build BRU with user I/D mapping to get rid of the overlays. Also, how about shipping updates to [6,54]BRUSYS with autopatch, whenever [3,54]BRU is upgraded.

- To work around the previous item, we built a bootable MPlus system on a single RX-50 diskette (800 blocks). Article coming next issue on that subject.

FLX Record Lenght Problem

From North Texas LUG Newsletter

In using FLX to copy RT-11 files to RSX V4.0, I discovered a gremlin in FLX. FLX will not accept a record length longer than 512 bytes. if it comes to the end of 512 characters without a <CR>, it declares that the record is not formatted ASCII (a bit frightening the first time that occurs) and throws away the record. The file is then closed at the end of the previous record. Also, if your last record does not end with a <CR>, it will be lost even though it is less than 512 characters.

To overcome this, one needs to write a program to go into RT-11 file and count the characters since the last <CR>. If the count gets around 500, you will want to put an unusual sequence (such as <ESC><CR>) at that point in the line. This will create records less than 512 bytes. After you transfer the file to RSX, you then use a program to search for every occurrence of the <ESC>CR> and remove them. Your file will then look like it did before you transferred it.

RSX SIG - 1983 Activity Report

Ralph Stamerjohn
RSX SIG Chairman

This report covers RSX SIG activities for the calendar year 1983. This period saw the SIG in transistion, a process which is still underway. Key areas of transition include a split into two SIGs, a new operating procedure, an almost total change in leadership, and a changing focus in Digital's RSX operating systems.

At the start of 1983, the SIG included the IAS operating system. A new IAS SIG has been formed through the efforts of Bob Curley and other DevIAS members. This properly positioned IAS within the DECUS structure. The two SIG's will work closely on common interest and share the Multi-Tasker for at least the next six months.

The RSX SIG revised it operating procedures in May, 1983, clarifying the election procedures and definition of the Executive and Steering Committees. These changes were adopted by the membership in late summer.

Under these new procedures, an election of five officers to the Executive Committee was scheduled for the Fall. However, after four of the nine nominees had to withdraw for various reasons, the five remaining people were declared elected by the DECUS Chapter Administrator. Following procedure, the five elected a new chairman (Ralph Stamerjohn) and he appointed the remaining three seats. The current Executive Committee (and Steering Committee) are shown in another article in this issue of the Multitasker.

The SIG has experienced almost 100% turnover at the Executive Committee level in the last two years. The current Executive Committee has not yet had an opportunity to meet and work together.

The RSX SIG was active in 1983 in the major areas of DECUS: publications, symposia, and library. It also celebrated its tenth anniversary with a formal banquet at the Las Vegas Symposium.

The Multi-Tasker did suffer difficulties, failing to publish on a monthly schedule for the first time in eight years. A new editor, Charles Goodpasture, experienced personal problems and the changing SIG leadership caused a delay in getting a new editor. Four issues were published in 1983 and new co-editors, Allen Watson and Dominic DiNollo, have been found.

RSX symposia activities continued quite well in 1983 under the direction of our Symposia Coordinator, Jim Hopp. The SIG no longer draws hundreds of people to its meetings, but a core group of around 300 to 400 attended the main RSX sessions.

The SIG continued its successful library efforts with two substantial SIG tapes and some renewed submissions to the DECUS library, most notable a portable spreadsheet from Glenn Everhart.

RSX SIG Steering Committee Members

Ralph Stamerjohn
RSX SIG Chairman

FROM THE EDITOR

This is an update to the list of steering committee members that appeared in the December, 1983 issue. It contains several additional addresses and corrections to previously published addresses. If a name appears more than once in the list, the address is given only the first time.

The following is the current composition of the RSX SIG Executive and Steering Committees as of this date. Positions in the left-most column are executive committee members.

CHAIRMAN:
Ralph Stamerjohn
412 Falaise
St. Louis, MO 63141

LOBBYIST:
Kerry Wyckoff
LDS Church
125 North State St.
Salt Lake City, UT 84103

MENU COORDINATOR:
Allen Bennett
Clark Equipment Company
Automated Systems Division
525 North 24th St.
Battle Creek, MI 49016

PLANNING COORDINATOR:
Terry Medlin
GEJAC Inc.
P.O. Box 188
Riverdale, MD 20737

BUDGET:
(open)

LONG-RANGE PLANNING:
(open)

RSX <-> SIG REPRESENTATIVES:

RSX <-> IAS:
Ray French
Boeing Commercial Airplane Co.
P.O. Box 3707, MS 6F-21
Seattle, WA 98124

RSX <-> VAX:
Joe Sventek

RSX <-> PC:
(open)

PUBLICATIONS COORDINATOR:
Allen Watson
The Record
150 River Street
Hackensack, NJ 07602

MULTI-TASKER CO-EDITORS:

Allen Watson

Dominic DiNollo
Loral Electronics
Engineering Computer Center
Ridge Hill, Yonkers, NY 10710

SYMPOSIA SESSION NOTES:
S. Reid Madsen
Weidner Communications
1673 West 820 North
Provo, UT 84601

SCRIBE SERVICE:
(open)

DECUS STORE/BOOKS:
(open)

SOFTWARE COORDINATOR:
Glenn Everhart
RCA GSD
Engineering 206-1
Route 38
Cherry Hill, NJ 08358

TAPE COPY COORDINATOR:
Glenn Everhart

TAPE COPY DISTRIBUTION:
(open)

TAPE COPY INDEXING:
(open)

DECUS LIBRARY:
(open)

SPECIAL PROJECTS:
Liz Bailey
Tennessee Valley Authority
222 CEB
Muscle Shoals, AL 35660

AUDIO/VISUAL:
(open)

VOLUNTEER RECRUITMENT DATA BASE:
Liz Bailey

SYMPOSIA COORDINATOR:
Jim Hopp
Swift and Company
1919 Swift Drive
Oak Brook, IL 60521

SYMPOSIA SCHEDULING:
Jim Hopp

PRE-SYMPOSIA SEMINARS:
Steve Mylroie
PRLS/Signetics, MS:0265
P.O. Box 3409
Sunnyvale, CA 94086

CALLS FOR PARTICIPATION:
(open)

SOFTWARE CLINIC:
Edward Cetron
The University of Utah
Center for Biomedical Design
3168 Merrill Engineering
Salt Lake City, UT 84112

FACILITY/MANPOWER MANAGER:
Gary Maxwell
Scientific Research Management Corp.
848-3 E. Gish Road
San Jose, CA 95112

POSTER PAPERS:
(open)

VOLUNTEER COORDINATOR:
Nancy Pallett
Cameron Shaw Associates
6735 Telegraph Rd. S 15
Birmingham, MI 48010

VOLUNTEER NETWORK:
Nancy Pallett

JOB DESCRIPTIONS:
(open)

WORKING GROUP COORDINATOR:
Jeff Hamilton
E-Systems Greenville Division
P.O. Box 1056, CBN27
Greenville, TX 75401

Data Acquisition,
Simulation and Process Control
Allen Bennett

DECUS Library
Glenn Everhart

RSX-11M Unsupported Versions
Bill Burton
Texas Research Institute
1300 Moursand
Houston, TX 77030

Runoff
Chuck Spalding
Adept Technology, Inc.
1202 Charleston Rd.
Mountain View, CA 94043

SIG Tape Collection
Jim Neeland
Hughes Research Lab
3011 Malibu Canyon Rd.
Malibu, CA 90265

SRD
Bob Turkelson
NASA/Goddard Space Flight Center
Mail Code 935
Greenbelt, MD 20771

System Performance and Accounting
Paul Sorenson
American Electric Power
Interactive Graphics Section
1 Riverside Plaza
Columbus, OH 43215

DIGITAL REPRESENTATIVE: (Spring)
Not known

DIGITAL REPRESENTATIVE: (Fall)
Laine Heiser
Digital Equipment Corporation
110 Spit Brook Rd. ZK01-3/J35
Nashua, NH 03062

DECUS REPRESENTATIVE:
Deborah Kleiner
DECUS/U. S.
One Iron Way, MR2-3/E55
Marlboro, MA 01752

FROM THE EDITOR

How about some volunteers to fill those twelve open positions? If you are interested you are qualified; contact Ralph Stamerjohn.

"Working Group News"

Jeff Hamilton
Working Group Coordinator
(214)454-4175

Date of this report: 06JAN84

The working group chairmen are as follows:

RSX-11M Unsupported Versions:
Bill Burton
Texas Research Institute
1300 Moursand
Houston, Texas 77030

System Performance and Accounting
Paul Sorenson
AEP, Interactive Graphics
1 Riverside Plaza
Columbus, Ohio 43215

DECUS Library
Bruce Zielinski
RCA
Marne Highway M/S 138-2
Moorestown, N. J. 08057

SIG Tape Collection
Glen Everhart
RCA Government Systems Division
Route 38
Cherry Hill, New Jersey 08358

SRD
Bob Turkelson
NASA/Goddard Space Flight Center
Mail Code 935
Greenbelt, Maryland 20771

Data Acquisition, Simulation and Process Control (DASPC)
Allen J. Bennett
Clark Equipment Co.
P.O.Box 3000
Battle Creek, Mich. 49016

Runoff
Chuck Spaluing
Adept Technology Inc.
1202 Charleston Rd.
Mountain View, Calif. 94043

The Unsupported Version working group is currently planning sessions for the Spring Symposium for past unsupported versions of 11M/11M+. An article has been submitted to the Multitasker to show how to modify the magtape driver to provide better execution of BRU3.2.

The System Performance and Accounting working group is continuing to prepare an index of the past RSXSIG tapes as it applies to System performance and accounting features that can be provided. There is a session being planned for the Spring Symposium for the System Performance and Accounting working group to discuss further work to be done. A method will be supplied to the Spring 84 tape of a method of providing pass offspring capability in CCL to 3.1 and 3.2. A method is also shown of passing more than 80 bytes on a command line. This is not on the Fall 83 tape, but those desperate enough can contact Paul Sorenson.

The DECUS Library working group have continued efforts to construct a tape to provide to the DECUS library of the best software off of the past RSXSIG tapes. The tape should go out to the working group for evaluation in the second week of January.

The SIG Tape Collection working group is currently in the process of developing the summary of the Fall 83 tape. The starting packets have been sent to the tree structure. (Only half of the LUGS responded and those not responding should be dropped but will not be for now, due to the problems of mail, etc.) An article has been submitted to the multitasker with information about the Fall 83 tape. A correction to the SRD submitted to the Fall83 tape just made it in under the wire. It appears Swedish Pascal will be included due to the fact of it incorrectly building under 4.0 and 4.1. The preliminary Fall83 tape has been shipped to the DECUS library and will be updated when the tree delivered version goes out.

There are many site problems with the LUGS which order tapes. A site with 1 tape drive and 20 Mbytes of disk storage will have trouble copying a RSXSIG tape. This also slows the tree down.

The SRD working group has supplied a NEW and improved SRD to the Fall 83 RSXSIG tape. A Multitasker article has been submitted describing it and all the neat things that can be done with it. This version is V6.4.

The DASPC working group has continued its efforts to lobby DEC for more real time development into RSX systems. This will continue into the future. Representatives from the LABS SIG, VAX Realtime Working Group and DASPC continued in their efforts to form a new SIG, but the status is unknown as of this writing. A seperate session is also planned for this working group.

The Runoff group has continued its effort to consolidate desirable features of several versions of Runoff into an "official" version. The latest Runoff working group version has been submitted to the DECUS Library. Telephone discussions have been held to review the charter that is held for the Runoff working group. Discussions were also held on the desirability of making Runoff more closely aligned with DSR. No sessions other than the general session is planned for the Spring Symposium.

If you are interested in providing information to a special working group concerning problems or ideas in that area, please get in touch with the working group chairman of that group.

A Primer for RUNOFF

Allen A. Watson
Multitasker Editor

NOTE

At the Las Vegas DECUS Symposium in October, 1983, I gave a session entitled "The Hows and Whys of RUNOFF". Afterward, one gentleman came up to me and asked if I could either provide him with a copy of my overhead slides or publish them in the Multitasker. The talk I gave was based upon the following primer which I wrote for use in my own company, and the slides were hand-written; they would reproduce poorly. Rather than publish the slides, I felt publishing the entire primer might serve the same purpose, and do it better.

1.0 AN OVERVIEW OF RUNOFF

RUNOFF (RNO) is a program to make the preparation of memos, letters, and documents easier and quicker. RNO reads an input file containing the text of the document and, interspersed in the text, RUNOFF commands that tell RNO how to format the text. You input the text pretty much in free-form, and RNO takes care of making it look pretty.

RUNOFF is available from several sources. The latest RSX SIG tape usually has at least one version. The DECUS library has a version dated January, 1982. And for RSX users who may sometimes use a VAX, RUNOFF is present in the VMS system where it is known as DSR (Digital Standard Runoff).

RUNOFF is supported by the RSX SIG RUNOFF Working Group. There is a brief report on the activity of the working groups in this issue.

1.1 HOW RUNOFF IS USED

The basic usage of RNO is presented in Figure 1.

FROM IDEA TO DOCUMENT

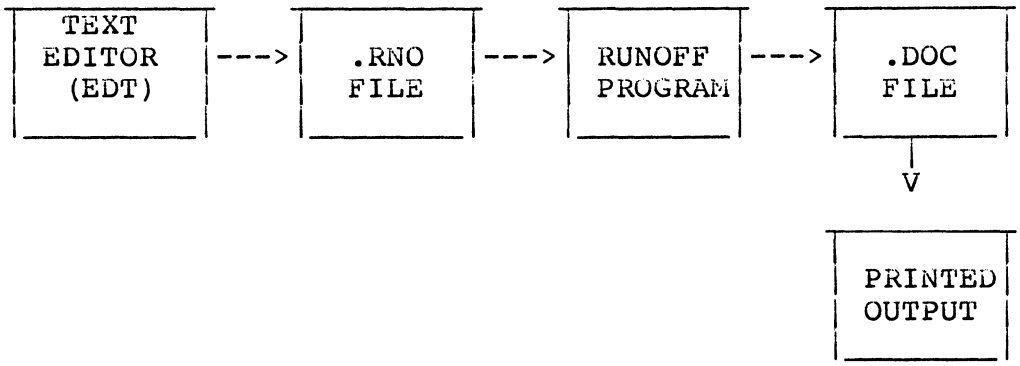


FIGURE 1

Using a text editor program (EDT or TECO) you create a Runoff source file with a file extension of ".RNO". You include text and formatting commands. Next, you run the RNO program, for example "RNO MEMO1=MEMO1". RNO creates an output file called, in this case, "MEMO1.DOC". (Some versions, such as DSR, default to a file type of ".MEM".)

Now you review the output file, either at your terminal or by printing it. If changes are necessary, you go back to the text editor and make the changes in the .RNO file, and run RNO again. When you are satisfied with the appearance of the output file, you

can print it for final use.

It's important to remember that all changes must be made to the Runoff source file with the extension of ".RNO", and not to the .DOC file that RNO produces.

1.2 THE ADVANTAGES OF RUNOFF

RUNOFF allows you to easily correct and revise documents. When you insert a sentence, a few words, or several paragraphs, RNO will automatically reformat everything else. RNO paginates the output: that is, if you say there are 56 lines to a page, RNO will put 56 lines on a page. If you add text to a page, RNO will move an equal amount of text onto the next page. Pages are numbered automatically, as are sections, etc. RNO can indent text easily; it can do numbered lists; it can produce an index and a Table of Contents. RNO can underline text like this with just a simple command. It can center lines of text within the margins, like this:

this text is centered
and so is this
and this

It can even produce footnotes at the bottom of a page, automatically.*

In general, RNO can produce documents with a much neater appearance than most of us can produce at a typewriter. You can use a video terminal to create and edit the text, or to revise it, which is much easier than revising a hard copy document.

2.0 FORMAT OF THE RUNOFF INPUT FILE

All text in the file is taken as source text except those lines which have a period in column one. Lines beginning with a period are understood to be RUNOFF commands.

For example, here is a short section of an input file with a command calling for two spaces (lines) between two paragraphs:

```
This is an example of some text within a source file.
This is the first short
paragraph. It has several lines.
.skip 2
This is a second paragraph of text within the source
```

* This is an example of a footnote.

file.
As you can see, the previous line was a command
to RNO to output two spaces.

RUNOFF has certain characters that have special meaning to the program. They cannot be used as normal text characters to appear in the output, unless you precede them with a special "quote" character (known as an Accept Flag). The meaning RNO gives to these characters will be discussed later, but for now you should be aware that the following characters get special treatment:

_	Underbar	(quote character)
#	Number sign	(explicit space)
&	Ampersand	(underscoring)
^	Circumflex	(mode lock)
\	Backslash	(mode unlock)
=	Equal sign	(hyphenation disable)
<	Less than	(capitalize next word)
>	Greater than	(subindex)

If you want to use any of these characters in the text of your file they must be preceded by the "quote character", which is the underbar. Thus, to have a number sign in the text like this -- #33 -- it must be typed in looking like this -- _# -- with a preceding underbar.*

3.0 BASIC RUNOFF COMMANDS

This section presents a selected subset of commands that can be used to prepare simple documents. RNO has over 85 commands, but you can do most simple documents using just the commands in the subset presented here.

As a matter of fact, all of the basic commands have default settings. This means that you can actually process a text file through RNO without imbedding any commands at all. The default settings are listed in the RUNOFF manual for the version you are using.

To get a quick idea of what RNO does, enter a short text file of two or three paragraphs, separating the paragraphs by blank lines, and then use RNO to process it, specifying your terminal as the output file, like this:

```
RNO TI:=filespc
```

You will see that RNO reformats all of the text. Any extra

* The underbar key is the shift of the hyphen.

spaces or lines are stripped out so that the output will appear to be all one paragraph, with even left and right margins. The left margin is at column 5, and the right margin is at column 65. Note that the blank lines have disappeared, and RNO has inserted spaces between words to force the left and right margins to be even. In some cases, RNO has hyphenated words if that makes a better fit.

RNO's hyphenation algorithm is very simplistic so it may make some bad hyphenation choices. If a high degree of literacy is important to your document you may need to override RNO's hyphenation choices; commands exist to do this.

3.1 AUTOMATIC PARAGRAPHING

Now edit the same test file and insert this line at the very beginning:

```
.AUTOPARAGRAPH
```

Try RNO TI:=filespec again. As you will see, RNO now recognizes your paragraph breaks (indicated by two consecutive RETURN's) and automatically indents the first line of each paragraph five spaces. You may notice that the first paragraph is not indented; if this occurs, it is because that paragraph was not preceded by two RETURN's, and therefore RNO did not recognize it as the start of a paragraph. If you wish, you can re-edit the source file to include a blank line after the ".AUTOPARAGRAPH" command; now RNO will indent the first paragraph as well.

When AUTOPARAGRAPH is in effect, RNO will recognize any blank line or formfeed followed by text, or a line starting with a space or tab as the start of a new paragraph. If the blank line or formfeed is followed by a command line (starting with a period), no paragraph is started.

This one command allows normally typed text to be justified without any further special commands.

3.2 PAGE LAYOUT

Page layout refers to the way in which margins are set for the top and bottom of a page, as well as right and left margins. It also involves the number of lines used by RNO for the "page header". RNO automatically outputs page numbers at the top of each page. The page number is put on the first line, and then (in the default setting), four blank lines are output.

3.2.1 The PAGE SIZE Command -

The Page Size command tells RNO how large the paper is that you are using. It defines how many lines to print per page, and how many characters to print on each line. It looks like this:

```
.page size 58,70
```

The example informs RNO to use 58 lines per page and 70 characters per line. The default page size setting varies from version to version; in many versions it is 58 lines per page and 60 characters per line. Some of the versions on the RSX SIG tapes allow you to specify what the defaults will be when you Task Build RNO.

3.2.2 The MARGIN Commands -

Once the intial page layout is set, you can alter the margin settings on the fly as you go. For example, to produce a paragraph indented on both sides you could set the left margin five higher, and the right margin five less. If you know the margin settings this can be done with absolute values. For example, if the left margin is at 0 and the right at 65, you could issue these two commands:

```
.LEFT MARGIN 5
.RIGHT MARGIN 60
```

> This is the result of executing the two margin commands. As you can see, the text has been indented on both sides. This serves to set it off nicely from the surrounding text.

If you do not know the current margin settings you can use relative arguments, like this:

```
.LEFT MARGIN +5.RIGHT MARGIN -5
```

The above example also illustrates the fact that multiple commands may appear on a single line.

To reset the margins to their original, default setting just issue the margin command with no argument:

```
.RIGHT MARGIN
.LEFT MARGIN
```

The margin commands (like most RNO commands) can be abbreviated. For RIGHT MARGIN you may use RM, and for LEFT MARGIN, LM. It is not required to put spaces between the command and the numeric argument, so that the following are legal RNO commands:

```
.LM+5
.RM-5
```

Command abbreviations are shown in the manual for each command.

3.2.3 The SPACING Command -

Another frequent need is for double- or triple-spaced documents. This is accomplished with the .SPACING command. Normally RNO will single-space your document. If you want double-spacing, include the command .SPACING 2 or .SP 2 before any text in your file. The number can range from 1 to 5. You can change back and forth in the course of your document.

3.3 TEXT FORMATTING COMMANDS

Text formatting commands are used to alter the format of text within a page.

3.3.1 The INDENT Command -

Sometimes you just want to indent one line, as for example in the illustration of the command that follows immediately below. The indentation of the command line is accomplished by preceding the text with an ".INDENT 5" command. The numeric argument is followed by a semi-colon to separate it from the text that follows:

```
.INDENT 5;_.INDENT 5
```

In this particular case the "quote character" or underbar is required to force RNO to accept the period as text because on a command line, all periods are assumed to start commands.

When the left margin has been set at other than zero, negative indents are possible. This produces what typesetters call a "hanging indent", where the first line sticks out to the left and everything else is indented beneath it. Here is an example of a hanging indent, followed by an approximation of what the text looks like in the source file, including the command.

```
RUNOFF HAS the capability of producing hanging indents such as is
          shown in this paragraph. The left margin is set at 10,
          and then a negative indent (.i-10) is done just before
          the first line of the paragraph to force it back to
          column 0. The rest of the text is justified against
          column 10.
```

Now, here is the way it looked in the source file.

```
.lm +10 .s
.i-10,RUNOFF HAS the capability of producing hanging
indents such as is shown in this paragraph. The left
margin is set at 10, and then a negative indent
(.i-10) is done just before the first line of the
paragraph to force it back to column 0. The rest of
the text is justified against column 10.
.lm-10
```

This technique can be used along with "explicit spaces" (number sign characters) to force a series of words to line up at a left margin while the text describing them is justified to a different left margin. Here is a short example:

```
LM      LM is an abbreviation for the
        LEFT MARGIN command. It changes
        the setting of the left margin.

ELI     ELI is an abbreviation for the
        END LITERAL command. It ends
        the literal mode started by a
        preceding LITERAL command.
```

3.3.2 The BREAK Command -

When you want to force a line to end without being justified, use the .BREAK command. This is useful in a short list of items, like this:

```
This is line one.
This is line two, and you can see it's not justified.
This is line three
```

There were two RNO command lines added to produce that: ".BR" between lines one and two, and between lines two and three. Without the Break commands, RNO would have ignored the line endings and justified all three lines on one or two lines.

3.3.3 The SKIP And BLANK Commands -

SKIP (abbreviation .S) causes a "break" just as .BREAK does, but in addition outputs at least one blank line. If you say ".SKIP 3", three blank lines are output. (If you are double-spacing or more, SKIP will output the appropriate number of double or triple spaces. E.g. in double-space mode, .SKIP 2 outputs four blank lines.)

BLANK does the same thing as SKIP except that the number is not multiplied by the spacing; .BLANK 2 puts out two blank lines regardless of whether you are double-spacing or not.

When you are writing a paragraph and want to follow it with an indented example, as we have been doing throughout this document, the easiest way is to enter a line containing ".S", and then a line with ".I5;text". This would produce something looking like this:

text

3.3.4 The CENTER Command -

The .CENTER command (.C) causes a BREAK and centers the following text. The text is centered within the current margins, over column "right margin plus left margin, total divided by two". There are variants on the CENTER command not discussed here; see the manual for more information. The text must be separated from the command by a semicolon, like this:

.c;THIS IS CENTERED TEXT

which produces the output:

THIS IS CENTERED TEXT

This makes a nice way to do headings for tables or title pages of a document.

3.3.5 The PARAGRAPH Command -

This command is used to establish the format for output paragraphs, not to start a new paragraph. It determines how a paragraph looks when AUTOPARAGRAPH is in effect. It takes three main arguments:

.P [<ind>],[<skip>],[<start>]

"ind" sets the number of spaces the first line of a paragraph is to be indented. By default this is 5; in this document I have set it to zero.

"skip" is the number of blank lines between paragraphs; the default is 1.

"start" is the minimum number of lines at the start of a paragraph which must appear on a page at the bottom; the default is 2. That means that when a paragraph starts, if there are not at least two lines remaining on the page, RNO will start a new page

before beginning the paragraph.

The setting of the PARAGRAPH command I inserted at the start of this document was:

```
.P 0,1,2
```

3.3.6 The LITERAL Command -

The .LITERAL (.LIT) command disengages filling and justifying, and outputs the input text exactly as entered in the input file, with the exception of RNO's special characters, which still must be preceded by a "quote character". Any extra spaces included in the input will also be included in the output, and no commands will be recognized except .END LITERAL or .el, which is the abbreviation for .END LITERAL.

Don't forget to use the ".END LITERAL" command after the part you want output literally, or all the rest of your file will be output exactly as you input it, including the RNO commands!

The LITERAL command is useful for outputting tables and columnar material. Tabs can also be used.

3.3.7 Underscoring Text -

To cause RNO to underline text in the output file, for emphasis, you must imbed special characters in the text. These special characters are:

```
^&   -- to start underlining
\&   -- to stop underlining
```

For example, if this paragraph ^&includes underscoring characters like this, the output would be underlined\&. Now I will repeat the paragraph with the characters actually there to do the underlining.

For example, if this paragraph includes underscoring characters like this, the output would be underlined. Now I will repeat the paragraph with the characters actually there to do the underlining.

Notice that the spaces between words are not underlined.

3.4 SECTIONS, NOTES, AND LISTS

As you have been reading this document you have doubtless noticed that it is segmented into sections with numbered headings, such as the one above this paragraph. This has been done (almost) automatically by RNO; all of the numbering, spacing and so on was done by RNO as the result of just one command, called a "HEADER LEVEL" command. RNO can also do other types of automatic formatting for "notes" and "lists", which we will now discuss.

3.4.1 The HEADER LEVEL Command -

To understand header levels, you have to think of a document as having "levels" of sections.

When you start a document, you are at "level 1" for headings. Level 1 headings are numbered "1.0", "2.0", "3.0" and so on. When you want a sub-heading under one of these, you call for a "level 2" heading. These would be numbered "2.1", "2.2", "2.3" and so on. If you want a sub-sub-heading under section 2.3, you would call for a "level 3" heading, which would produce numbers "2.3.1", "2.3.2", "2.3.3" and so on.

The "HEADER LEVEL" command is used to do this (abbreviation ".HL"). It takes the form:

```
.HEADER LEVEL n text
```

where "n" is a number from 1 to 5 (five is the highest level of heading allowed), and "text" is the text for the section heading.

The HEADER LEVEL command line is usually followed by a blank line, or an indented one, to signify to RNO that the next text starts a new paragraph.

For example, suppose RNO encountered the following HEADER LEVEL commands as it precedes through an input file:

```
.HL 1 MAIN SECTION
.HL 2 First sub-section
.HL 2 Second sub-section
.HL 3 First sub-sub section
.HL 3 Second sub-sub section
.HL 1 SECOND SECTION
.HL 2 Another sub-section
.HL 2 Last sub-section
```

This would produce the following output at the start of each section:

1.0 MAIN SECTION
1.1 First sub-section
1.2 Second sub-section
1.2.1 - First sub-sub section
1.2.2 - Second sub-sub section
2.0 SECOND SECTION
2.1 Another sub-section
2.2 Last sub-section

The HEADER LEVEL command also automatically outputs three blank lines before the heading line, tests to be sure at least 5 lines remain on the page, and then outputs the heading line. You can see the effect of this command throughout this document. The test for 5 lines insures that a heading will not appear on a page without at least two lines of text on the same page.*

When you use HEADER LEVEL commands, most RSX versions of RNO will automatically produce a neatly formatted Table of Contents at the end of your document, using the text from all the HEADER LEVEL commands and recording the page number on which each section starts.**

3.4.2 The LIST Command -

The .LIST command starts an indented list with one blank line between each element of the list. The elements of the list will be numbered sequentially. Each element in the list must be preceded with a ".LIST ELEMENT" command (.LE) and the list is terminated with an ".END LIST" (.ELS) command.

For example, here is a short list:

1. This is the first element of a list.

2. This is another element.

3. This is the next-to-last.

4. This is the last element.

---

* There is a way this could still happen: if you input two HEADER LEVEL commands in a row, for example a ".HL 1" line followed by a ".HL 2", with no text for level 1. For this reason it is a good practice to include at least one line of text for each header level.

---

* This can be disabled using the .DISABLE CONTENTS command. The VAX version of RUNOFF requires that a separate program, TOC, be run to produce a Table of Contents.

This is the way the list appeared in the input file:

```
.list
.le;This is the first element of a list.
.le;This is another element.
.le;This is the next-to-last.
.le;This is the last element.
.end list
```

The nice part of using the .LIST command is that you don't have to worry about the numbers; RNO does it for you. If you add a new element in the middle of the list, everything else is automatically re-numbered when you RUNOFF the file again. And all of the indenting and spacing is done automatically, too.

A variant of .LIST produces an unnumbered list, with a "mark character" instead of a number, like this:

o This is the first element of a list.

o This is another element.

o This is the next-to-last.

o This is the last element.

The only difference in the input file between this and the previous example is that I said ".LIST 'o'" instead of just ".LIST"; when RNO sees a character after the .LIST command it uses that as a mark character instead of numbers.*

3.4.3 The NOTE Command -

Frequently you want to call some short paragraph to the reader's attention by indenting it left and right and heading it with a special heading. RNO has a simple command to do all this for you: the .NOTE command.

HOW THE NOTE COMMAND LOOKS

> The .NOTE command produces output that looks like this! This is a note. The text is indented on both sides so it stands out to you as you are reading.

---

* See the manual for restrictions on characters that can be used as mark characters. One RSX version supports this feature with slightly different syntax.

The input file for the above note looked like this:

```
.NOTE HOW THE NOTE COMMAND LOOKS
The .NOTE command produces output that looks like this! This is a
note. The text is indented on both sides so it stands out to you
as you are reading.
.end note
```

If there is no text following the .NOTE command on the same line, RNO will center the word "NOTE" over the text of the note.

3.5 PAGE HEADING COMMANDS

Unless you suppress it, RNO automatically outputs page numbers at the top of each page. You can also have RNO output a document TITLE on each page, and in addition, SUBTITLES. You can see what that looks like by looking at the top of this page.

3.5.1 The TITLE Command -

If you want each page to have a title, include the ".TITLE" command near the start of your document, preferably before any text. If you have any text before the TITLE command, the title will not appear on the first page. This command takes any text on the same line and outputs it at the top of each page along with the page number. Example:

```
.TITLE RUNOFF PRIMER
```

3.5.2 The SUBTITLE Command -

Subtitle works like title: the text is taken as a subtitle to appear beneath the title on the next page.

3.5.3 The AUTOSUBTITLE Command -

AUTOSUBTITLE takes all the work out of it for you and makes the SUBTITLE command unnecessary. Just include this at the start of your input file, and then any time you use a "HEADER LEVEL" command, the text from that section heading will also become the subtitle on the next page!

3.6 PAGINATION COMMANDS

Sometimes you will want to override the normal flow of RNO's output to force the start of a new page. The following commands allow you to do that.

3.6.1 The PAGE Command -

The .PAGE command causes a break and an advance to a new page. You would use this when you want to start a table that you know will take most of a page.

3.6.2 The TEST PAGE Command -

The .TEST PAGE command (.TP) allows you to force the start of a new page if there are not enough lines left on the current page to output what you want. For example, if you are going to output a "literal" section that is 12 lines long and you want to be sure all twelve lines are on the same page, you would precede the literal section with a ".TP 12" command. If twelve lines remain on the current page, nothing happens; but if not, RNO will start a new page.

TEST PAGE will interpret the number you give according to the spacing; for example, if you are in ".SPACING 2" mode, RNO will test for 12 double lines, or 24 lines. If you are not single spacing but want to test for a specific number of lines, use the ".TEST LINES" command (.TL).

4.0 PRODUCING AN INDEX

When you are writing a fairly long document such as this one, it is useful to have an index to it. RNO can do that for you, but not without a little work.

If you include an index entry like this: .INDEX term, RNO will save each entry as it finds it, remembering what page it was on, and at the end will produce an index for you. All of the sorting and formatting of the index is entirely automatic.

You can index terms in the middle of a paragraph by including the .INDEX command in the text, as long as the "dot" of the command occurs in column one.

In producing the index RNO will distinguish between upper and lower case, so be sure you type the index entry the same each time.

You can also have sub-entries in the index. If you look at the "RUNOFF" entry in the index you will see an example of sub-entries. Sub-entries are produced by entering index commands like this:

```
.x RUNOFF>Usage
```

You can create a "main" entry with no page number associated by using the ".ENTRY" command (.Y), for example:

```
.Y MAIN ENTRY
```

which can then be referenced in sub-entries such as:

```
.X MAIN ENTRY>Subtopic
```

The easiest way to get used to indexing a document is to do it, use RNO outputting to your terminal, and see the result.

5.0 OTHER COMMANDS

In this primer I have discussed only the commands I use most frequently in producing documents. There are scores of others, and in addition, there are execution time switches that can be used to alter the parameters under which RNO executes.

Just to whet your appetite some more, let me end with a list of some of the commands we have not covered.

RNO COMMANDS NOT COVERED IN THIS PRIMER

- ALTERNATING TITLE -- flip-flops title and page number on alternate pages, as in this document, so page numbers are always at outside edge of a double-sided document. (This command has different names in different versions of RUNOFF; the VMS version uses .LAYOUT.)
- SUBPAGE -- special page numbering for revisions to documents.
- CHAPTER -- starts a new chapter for chapter-oriented documents.
- APPENDIX -- Starts a new appendix, titling with "APPENDIX A", then "B", etc.
- EVEN and ODD -- forces to next even or odd page. (Not in all versions.)
- RIGHT JUSTIFY -- forces text against right margin.

- o FOOTNOTE -- used to produce footnotes, as seen in this document.

- o BLOCK -- used to force output of a block of text as a single unit on a page: the current page if there is room, or if not, defer output until the next page. (Not in all versions.)

- o FIGURE -- used to reserve space on a page for something to be drawn or pasted in later.

- o OPTION -- one of a set of commands used to include conditional input in an input file, so that a single source file can be used to produce several documents with varying content. Some versions have a full set of IF and ELSE commands.

APPENDIX A
EXERCISES FOR RUNOFF

The exercises in this Appendix are optional, and are given to help you use some of the RNO commands discussed. References are given to the section of the primer that discusses the commands in each exercise.

6.0 NO COMMANDS (SECTION 3.0)

Using EDT or TECO, enter a file containing two or three paragraphs, separated by blank lines, with no RUNOFF commands. Output it to your terminal with RNO by typing "RNO TI:=your-file-name". Notice how RNO has justified the text into even margins, and has removed any extra spaces and all blank lines.

7.0 AUTOMATIC PARAGRAPHING (SECTION 3.1)

Add the .AUTOPARAGRAPH command at the beginning of your file, followed by a space, and output it with RNO to your terminal again. Notice that now RNO recognizes the blank lines as paragraph breaks. The first line of each paragraph is indented five spaces.

8.0 CHANGING PARAGRAPHING (SECTION 3.3.5)

Now add a .PARAGRAPH command at the start of your file. Set the paragraph indent to 0 and the "skip" parameter to 2, like this: ".P 0,2,2". Output the file again. This time there is no indent for the first line, and there are two blank lines between paragraphs instead of one.

9.0 CHANGING PAGE LAYOUT (SECTION 3.2.1)

At the top of your file, add a line like this: ".PAGE SIZE 20,50". This tells RNO to output only 20 lines per page, and to justify in fifty columns. Again, output to your terminal to see the results.

10.0 CHANGING MARGINS IN THE FILE (SECTION 3.2.2)

Remove the ".PAGE SIZE" command used in the previous exercise. Now, before the second paragraph in your file, enter two commands:

.LM +5 .RM-5

Be sure to leave a blank line between the commands and the start of the paragraph so RNO will know it is a paragraph start.

At the end of the second paragraph, enter the commands ".LM .RM" to reset the margins to their original values.

Output the file to your terminal to see the result.

11.0 SPACING (SECTION 3.2.3)

At the top of your file insert the line ".SPACING 2". Output the file to see the result.

12.0 USING INDENT (SECTION 3.3.1)

Start a new file. This is to be a list of items, indented 5 spaces for each new column. The first column has no indent, the second is at 5; the third at 10, and the fourth at 15. Using only the indent command, try to reproduce the output you see below:

TABLE OF INDENTED ITEMS
    First item
    Second item

```
          Part 2A
          Part 2B
     Third item
          Part 3A
               3A sub one
               3A sub two
          Part 3B
     Fourth item
```

Each line except the first must begin with an indent command, for example, ".I10;Part 2A". Keep outputting the results to your terminal until you get the table to look right.

13.0 NEGATIVE INDENTING (SECTION 3.3.1)

Start a new file. This file should begin with the command ".LEFT MARGIN 13". Use the ".PARAGRAPH" command to set the indent for new paragraphs to "-13", like this: ".P -13".

Now enter several short paragraphs describing some RUNOFF commands like LAYOUT, SPACING, and AUTOSUBTITLE (use the Table of Contents to get other names). Start each paragraph with the command name followed by explicit spaces to make the command name twelve characters long, like this:

LAYOUT###### - the LAYOUT command is used to set the top, bottom left and right margins on a page.

AUTOSUBTITLE - this command causes the text of each Header Level to be used as the subtitle when a new page is started.

SKIP######## - the SKIP command causes a break and skips the indicated number of lines.

When you output this to your terminal, because of the left margin being set at 13 and the negative indent for each paragraph, the result should look like this:

LAYOUT - the LAYOUT command is used to set the top, bottom left and right margins on a page.

AUTOSUBTITLE - this command causes the text of each Header Level to be used as the subtitle when a new page is started.

SKIP - the SKIP command causes a break and skips the indicated number of lines.

14.0 CENTER AND SKIP (SECTIONS 3.3.3 AND 3.3.4)

Edit the file from the previous exercise and add a document title, centered, as the first line, like this:

```
.C;Description of RUNOFF commands
```

Cause four blank lines to follow the title by adding the command ".SKIP 4".

15.0 UNDERLINING (SECTION 3.3.7)

Change the title line in the previous example like this:

```
.C;^&Description of RUNOFF commands\&
```

Output it again. You will not see any underlining at your terminal! Now output it to a file by typing "RNO filename=filename", and then print the file on a lowercase printer. Remember to print filename.DOC, not filename.RNO. Look at the printed output and you will see the title is now underlined.

Try underlining some of the other text in your file.

16.0 USING THE LITERAL COMMAND (SECTION 3.3.6)

You may add this to any of your files, or create a new one.

Insert a ".LITERAL" command. On the next line, start inputting some tabular text, such as that which follows:

DRIVE NUMBER	USE	ACCESS FOR CPUA	CPUB	CPUC
0	CSI MASTER	R/W	--	--
1	CSI PDUAL	R/W	--	--
2	SHADOW	--	--	R/W
3	BUSINESS	--	RO	R/W
4	DATA PACK	--	RO	R/W
5	PROGRAM DEV	--	R/W	RO

You can use tabs to input it.

End the table with an ".END LITERAL" command and input a few lines of normal text so you can see that normal justification has resumed. Output the file to your terminal. (Be sure your terminal tabs are set at the default positions, every eight columns.) If the results do not look right, replace the tabs with spaces and try RNO

again.

17.0 LIST AND NOTE (SECTIONS 3.4.2 AND 3.4.3)

Using the same file, add a NOTE at the top of the file. Put ".NOTE" on the first line, type a short paragraph, and then add a line saying ".END NOTE". Output it to see the result.

Now at the end of the file, add a ".LIST" command, and then create a list of several elements, each one starting with ".LE;" or ".LIST ELEMENT;". The element can be one word, one line, or several paragraphs long. End the file with the command ".END LIST" or ".ELS". Output the results.

Finally, change the ".LIST" command to use a hyphen as a mark character instead of numbering.

NOTE

The remaining commands, for the most part, require a multi-page document for efficient testing. If you have done all of the above exercises, you are ready to start using RNO to do your own documents. That will give you plenty of practice with HEADER LEVEL, INDEX,TITLE, and PAGE.

Concerning Versions of RUNOFF

Allen A. Watson
Multitasker Editor

Runoff is an old program. When I first started programming over fifteen years ago my first project was converting a version of Runoff written at MIT in PL/1 that ran in batch mode on an IBM 360 to a time-shared version of PL/1. It had to run in a 16K user partition.

Seven or eight years ago I ran into Runoff again, this time on a DEC 11/70 running Unix. Runoff is still in use on Unix systems, frequently known as NROFF or ROFF. The LBL Software Tools package by Joe Sventek (found on the RSX and VMS SIG tapes) contains another version, called FORMAT.

On VAX systems there is Digital Standard Runoff Version 2.0, a supported product.

The RSX world is dependent on volunteer labor to maintain Runoff. Early RSX systems were distributed with Runoff included among the unsupported utilities; this distributed version was the de facto standard version. Once users get source code, though, changes start happening, and this is what happened to RSX Runoff. Versions began to proliferate: a version to support Diablo printers, a version to support another printer, a version that did Tables of Contents, one that did indices, another that did multiple indices, and so on.

A Runoff Working Group was formed with the purpose of maintaining and enhancing Runoff, and -- it is hoped -- consolidating the best features of the many versions. Charles Spalding is the current WG Coordinator (see the Working Group Report elsewhere in this issue). As yet, however, there is not a "standard" working group version of Runoff.

For the last two years every RSX SIG tape has had releases of two different verions of Runoff, both excellent. Chuck Spalding submits one of them in UIC [307,50]. John Clement of Rice University submits the other version in UIC [332,12]. I have used both versions and found them almost equally useful; as a consequence I use both versions regularly depending on which features I need for the document I am working on. Those of us who met with the Runoff Working Group in Las Vegas expressed a strong hope that Chuck will be equal to the task of integrating these two versions.

If you want the most recent Runoff, just get the most recent RSX SIG tape and look in those two UICs. Both versions come with installation command files, source code, and full, well-written documentation (produced with Runoff, of course). The version recently submitted to the DECUS library was Charles Spalding's version, the same revision level as the one that will be on the Fall 1983 RSX SIG tape, so if you can't get a SIG tape you can order Runoff by itself from the library.

Advantages of the two versions

Both Spalding and Clement keep on making improvements to their versions. New features seem to appear on every tape (every six months) and it is hard to keep up with them. I once began a comparison chart for Spalding, Clement and DSR, but unfortunately I have lost it. The versions that I am now using and that I will discuss here all date from early 1983.

Spalding's Runoff

1. The Runoff Manual seems more complete and better organized.

2. Supports sub-indexing.

3. Supports alternating titles (flush left on even pages, flush right on odd pages).

4. Supports automatic subtitles based on Header Level commands.

5. Implements a base left margin. That is, you can set base left margin to column 5, for example. Subsequent margin setting commands will all start at column 5.

6. Extensions to TEST PAGE and PARAGRAPH for cleaner handling of end of page conditions.

7. EVEN and ODD commands to force output onto an even or odd page.

8. Allows redefinition of the Quote Character so that the underscore may be used freely in text.

9. DISABLE OUTPUT and ENABLE OUTPUT to turn output off and on; useful to output only selected portions of a file.

10. Extensions to LIST and NOTE commands for better control of line spacing. The LIST command supports the form ".LIST ,x" where "x" is a character to be output instead of consecutive numbers.

11. A FOOTNOTE LINE command to automatically output a separator between text and footnotes.

12. The BLOCK command, which denotes a block of text. If space exists on the current page it is output immediately. Otherwise it is deferred to the next page on which it will fit.

13. The FIGURE command allows inclusion of a figure caption.

14. Multiple Tables of Contents for separate referencing of text, tables, and figures.

15. A CONTENTS command to include a reference in the TOC even though not in a Header Level command.

16. More user control over the format of Tables of Contents; ability to force output of a TOC at any time.

17. Supports optional input in the sense of "if option 2, include this text". Multiple versions of a document can be produced from the same source file.

18. Allows processing of multiple input files into a single output file.

19. Allows appending output to an existing file.

Clement's Runoff

1. Command line control to output only certain chapters or appendices.

2. Command line control to right shift output n characters.

3. Has overstrike flag character to allow overstriking.

4. Supports user definition and use of string substitution, e.g.:

```
.DEFINE SUBSTITUTE /SIG$/Special Interest Group
```

   Each time SIG occurs in the text the string "Special Interest Group" would be substituted for it.

5. Break flag character for better control of word breaks.

6. Allows centering of multiple lines of text with single command.

7. Automatically computes the number of lines needed for footnotes.

8. Supports several DISPLAY commands -- NUMBER, CHAPTER, APPENDIX, SUBPAGE, LEVELS, ELEMENTS -- that allow user control of formats for these items. Formats supported are Decimal Numbers, Uppercase Letters, Lowercase Letters, Mixed Letters, Uppercase and Lowercase Roman Numerals, and Roman mixed.

9. Allows more user control of format of header levels.

10. Supports seven different page layouts for titles and page numbering.

11. The LOCK command locks in basic page formatting parameters to aid in merging several files (created with different parameters) into one document.

12. Supports change bars.

13. Allows redefinition of all flag characters.

14. Implements user definition of escape sequences. Using this mechanism you can define non-printing escape sequences to control a variety of printers such as the Diablo or NEC Spinwriter.

15. Supports the REQUIRE command for inclusion of input text from another file. Five levels of nesting allowed.

16. PAGE SIZE command expanded to allow setting defaults for top and left margins.

17. SET PARAGRAPH command to set parameters for paragraphing without actually causing a break or starting a new paragraph.

18. More consistently compatible with VAX Digital Standard Runoff (a matter of opinion).

As you can see both versions have much to recommend them. Note that I have listed only things that appear in one version and not in the other. Both versions have many more useful features in common.

RSX-11M V4.1 SYSGEN on a VAX

Frank J. Nagy, Laura Vanags, Lin Winterowd
Fermi National Accelerator Laboratory
P. O. Box 500 Mail Stop 306
Batavia, IL 60510

We have successfully completed a SYSGEN for RSX-11M V4.1 using the MCR CLI under VAX/VMS. For the most part, the instructions in the RSX SYSGEN Manual appropriate to online SYSGEN's are applicable to VAX/VMS. However, some preparatory work must be undertaken to successfully perform the SYSGEN. These include changes to be made in SYSGEN2.CMD and SYSGEN3.CMD dealing with assignment of the logical names TKL and VMR. In addition, some of the privileged .BLD files must be changed to avoid bugs in VMS MCR (VMS V3.4 was used).

In the past we have performed RSX V3.2 and V4.0 SYSGEN's under VMS V2.5 and V3.0. DEC seems to have come full circle on us as the V3.2 SYSGEN required modifications to the command procedures to perform correctly on the VAX. In contrast, the V4.0 SYSGEN required minor changes to some of the .BLD files to bypass a bug in MCR in VMS V2.5; with VMS V3.0 this bug was fixed and RSX V4.0 could be SYSGEN'ed on the VAX without modifying the command procedures. With RSX V4.1 we seem to be back to having to avoid a substantial MCR bug and having to fix problems in the SYSGEN procedures seemingly unrelated to possible MCR problems. We

certainly wish DEC would get its act together on this issue and make a commitment to support RSX SYSGEN's under the new VAX-11 RSX layer product to be introduced with VMS V4.0.

Our RSX distribution is the RL01/RL02 distribution kit on 9-track magnetic tape. The VAX is equipped with RL02 disk drives which were used for the SYSGEN itself. The following DCL command sequence was used to copy the baseline distribution tape onto disk:

```
$ MOUNT/NOWRITE/FOREIGN MT0:
$ MOUNT/FOREIGN DL1:
$ MCR BRU
RSX> /BACK:<label>/REW/DENSITY:1600/VER
From: MT0:
To:   DL1:
```

This sequence was repeated to copy the RSX V4.1 distribution kit onto six RL02 disks.

Before attempting a SYSGEN, several RSX V4.1 utility tasks were copied from the baseline system disk onto the VAX system disk (i.e., [1,54]MAC.TSK was copied to MAC.TSK in SYS$RSXROOT:[V41TSKS]). A series of ASN statements (under the MCR CLI) were then placed in the pre-SYSGEN command procedure:

```
$ ASN SYS$RSXROOT:[V41TSKS]MAC.TSK=MAC
```

where SYS$RSXROOT points to __DRA0:[RSX.]. These assignment statements cause the overlays for the tasks to be loaded much faster, as the task images on the VAX RM80 disks are used rather than those on the RL02 baseline disk. A listing of the pre-SYSGEN command procedure (RSXSYSGEN.CMD) is included with this note.

Modifications were made to SYSGEN2.CMD and SYSGEN3.CMD allowing the TKL/VMR assignments to be made either externally or within the SYSGEN command procedures. In SYSGEN2.CMD, replace both occurances of

```
.IFF $MLH .IF <SYSTEM> EQ 5 ASN SY:[1,54]TKB.TSK;1=TKL
```

with

```
.IFF $MLH .IF <SYSTEM> NE 5 .GOTO <label>
.SETS FJNSYM "'F$LOGICAL(""TKL"")'"
.IFF $MLH .IF FJNSYM  = "" ASN SY:[1,54]TKB.TSK;1=TKL
<label>
.IFF $MLH <SYSTEM> NE 5 .IFF TKLX INS SY:[1,54]TKB.TSK;1/TASK=...TKL
```

In addition, near the end of SYSGEN2.CMD, replace

```
.IFF $MLH .IF <SYSTEM> EQ 5 ASN SY:[1,54]VMR.TSK;1=VMR
```

with

```
.IFF $MLH .IF <SYSTEM> NE 5 .GOTO FJNVV3
.SETS FJNSYM "'F$LOGICAL(""VMR"")'"
.IFF $MLH .IF FJNSYM  = "" ASN SY:[1,54]VMR.TSK;1=VMR
.FJNVV3: .IFF $MLH .IFF $ONL .IFINS VMR ASN SY:=LB:
```

and replace

```
.IFF $MLH .IF <SYSTEM> EQ 5 ASN =TKL
.IFF $MLH .IF <SYSTEM> EQ 5 ASN =VMR
```

with

```
.IFF $MLH .IF <SYSTEM> NE 5 .GOTO FJNVV4
.SETS FJNSYM "'F$LOGICAL(""TKL"")'"
.IFF $MLH .IF FJNSYM <> "" ASN =TKL
.SETS FJNSYM "'F$LOGICAL(""VMR"")'"
.IFF $MLH .IF FJNSYM <> "" ASN =VMR
.FJNVV4:
```

In SYSGEN3.CMD, replace

```
.IF <SYSTEM> EQ 5 .IFF $MLH ASN SY:[1,54]TKB.TSK'TKBVER'=TKL
```

with

```
.IF <SYSTEM> NE 5 .IFF $MLH .GOTO FJNVV1
.SETS FJNSYM "'F$LOGICAL(""TKL"")'"
.IFF $MLH .IF FJNSYM  = "" ASN SY:[1,54]TKB.TSK'TKBVER'=TKL
.FJNVV1: .IF <SYSTEM> EQ 5 .GOTO 660
```

and replace

```
.IF <SYSTEM> EQ 5 .IFF $MLH ASN =TKL
```

with

```
.IF <SYSTEM> NE 5 .IFF $MLH .GOTO FJNVV2
.SETS FJNSYM "'F$LOGICAL(""TKL"")'"
.IFF $MLH .IF FJNSYM <> "" ASN =TKL
.FJNVV2: .IF <SYSTEM> EQ 5 .GOTO NOREM
```

Our initial attempt at performing the RSX V4.1 SYSGEN failed. The problem was due to the VAX/VMS intercept of the character '@' and the resulting attempt at invoking an associated indirect command procedure. Both ICPBLD.BLD and ICQBLD.BLD contain the parameter line

```
.DATA @'$OD'ICPCOMBLD.ODL
```

Upon encountering the '@' character, VMS immediately attempted to execute a command procedure ('$OD'ICPCOMBLD.ODL). This, of course, failed and as a result the .ODL file was not created. During the build of privileged tasks, ICP.OBJ was not found and a fatal TKB error resulted. The following modification was made to both ICPBLD.BLD and ICQBLD.BLD:

```
.IF <SYSTEM> NE 5  .DATA @'$OD'ICPCOMBLD.ODL
.IF <SYSTEM> EQ 5  .DATA "@"'$OD'ICPCOMBLD.ODL
```

The quotation marks around the '@' symbol (now interpreted as a literal character) inhibited an attempt at immediate execution of the command procedure. This then necessitated answering YES to the "pause after building the .ODL and .CMD files" request to allow the pair of quotation marks to be editted out of the ICPBLD.ODL file before proceeding with the privileged task builds.

To actually do the SYSGEN, login again using the "/CLI=MCR" qualifier with the user name. This will establish MCR as the default command line interpreter (CLI) allowing the processing of RSX indirect command files. The RSX V4.1 baseline system is mounted, the UIC is set to [200,200] and the first SYSGEN command file is executed:

```
> MOU/SHARE DL1:RSXM35
> SET /UIC=[200,200]
> @SYSGEN
```

All phases (I, II, and III) of SYSGEN are then performed as if online to an RSX system. The RSX Autoconfigure program will not be run, as is to be expected. The final step of booting and saving the newly SYSGEN'ed system must be performed on the target PDP-11 system, as must the test run of UETP used to checkout the newly created system.

One note of caution, since our PDP-11's do not include any of the K-series peripherals, we have never done an RSX SYSGEN with those features. We therefore do not know if those portions of the RSX SYSGEN will be performed correctly under VAX/VMS.

Listing of file RSXSYSGEN.CMD

```
.;
.; RSXSYSGEN.CMD
.;
.;  Used to make the logical name assignments to do an RSX SYSGEN
.;  on the VAX under VMS.  From the VAX-11/RSX-11M USER''S GUIDE
.;
.; Modification History:
.;
.;  01-Jul-81       FJN     Make sure MAC used from SYS$SYSTEM:
.;  03-Jul-81       FJN     Straighten out assignments and
.;                  make sure utilities used from SYS$SYSTEM
.;  05-Jul-81       FJN     Use BIGTKB.TSK on SYS$SYSTEM:
.;  25-Jan-82       ADT/KJC Modify message concerning spooled maps
.;                  and the MOU/SHARE command for clarity.
.;  08-Jul-82       FJN     Ask question re RSX V4.0 SYSGEN to use xxx4.TSK
.;                  utilities on VAX system disk instead of V3.2 forms.
.;  23-Jul-83       FJN     Moved from SYS$MGR_UTIL: to RSXMANAGER account
.;                  and edited to move distribution utilities and to
.;                  replace V3.2 stuff by V4.1 stuff.
.;  07-Sep-83       FJN     Fix text on .SETS commands, define TKL for V4.1
.;
.ENABLE SUBSTITUTION
.;
.; Get RSX target device if not entered as a parameter
.;
.IF P1 EQ "" .ASKS P1 Target device name
.;
.; Make sure the device name is ended by a colon
.;
.SETS TEMP P1[<STRLEN>:<STRLEN>]
.IF TEMP NE ":" .SETS P1 P1+":"
SHOW SYMBOL P1
.;
.; Set default to target device
.;
SET DEFAULT 'P1'
.;
.; Assign SY, SY0, LB, and LB0 to target disk
.;
ASN 'P1'=SY:
ASN SY:=SY0:
ASN 'P1'=LB:
ASN LB:=LB0:
.;
.; Equates MP0 and MP, also TK0 and TK
.;
ASN MP:=MP0:
ASN TK:=TK0:
.;
.; If doing RSX V4.0 SYSGEN, use different assignments for utilities.
.;
```

```
.ASK V4 Doing an RSX-11M V4.0 SYSGEN
.IFT V4 .GOTO RSX4
.;
.; What follows is specific for V4.1 gens
.;
.SETS where "SYS$RSXROOT:[V41TSKS]"
.GOTO COMMON
.RSX4:
.;
.; What follows is specific for V4.0 gens
.;
.SETS where "SYS$RSXROOT:[V40TSKS]"
.COMMON:
.;
.; Assigns equivalence names for utilities kept on VAX system disk(to allow
.; speedier overlays).  Note that these were copied/built from RSX baseline
.; system or RSX system libraries and cannot reference VAX/VMS device and
.; directory names.
.;
ASN 'where'TKB.TSK=TKB
ASN 'where'TKB.TSK=TKL
ASN 'where'VMR.TSK=VMR
ASN 'where'MAC.TSK=MAC
ASN 'where'LBR.TSK=LBR
ASN 'where'PIP.TSK=PIP
.;
.; Use EDT from VAX/VMS system disk (native mode) as itself and also
.; as the EDI editor.
.;
ASN SYS$SYSTEM:EDT=EDT
ASN SYS$SYSTEM:EDT=EDI
.;
.; Set process privileges needed to do a SYSGEN
.;
SET PROCESS/PRIVILEGES=(SYSPRV,LOG_IO,CMKRNL,EXQUOTA)
;
; Note that the baseline system MAC and TKB are used.   The listings
; and maps cannot be sent to a non-RSX device or to magtape.  The maps
; may be put on the target disk ('P1').  The assembly listings may be
; placed on a scratch disk (RL01/RL02).  Each disk must be mounted
; SHARE'd to allow the printouts to be spooled to the printer:
;
;    > MOU/SHARE DL2:LISTINGS
;    > MOU/SHARE 'P1'RSXM35
;    > SET /UIC=[200,200]
;    > @SYSGEN
;
EXIT
```

AST's for Beginners

Dominic DiNollo
Loral Electronic Systems
Engineering Computer Center
Ridge Hill
Yonkers, New York 10710

An AST is an Asynchronous System Trap. An AST is a software interrupt mechanism which allows a program to recognize an external event, service the event, and return to the section of the code which was interrupted. The occurrence of the event is random in nature, that is it happens at no particular time and during the execution of any instruction in the program. Several examples of events which can be recognized and serviced by AST's are I/O completion, power failures, timer expiration, and memory parity errors. Many other directives allow the programmer to specify the address of an AST service routine.

The AST mechanism allows a task to behave like a simple operating system. A program can be executing a "background" function and be interrupted to work on a more important function just as an operating system would stop running a user's program to handle a request from a device driver. This interrupt mechanism increases the through-put of a program just as it does for an operating system. For example, a simple system is needed to collect and analyze data. One method could be a two program approach: one to collect the data and one program to analyze it. In many cases this could be accomplished by one program with increased through-put over the two program approach. This task could perform the data analysis as a "background" function and be interrupted to gather more data from a device when it is ready. Very little time is wasted; the time between data collections is put to good use.

When experimenting with AST's several things must be kept in mind:

- The general purpose registers are not saved by the Executive when an AST is recognized. If you need to use them then you must save and restore them.

- AST execution is a distinguishable task state. The Executive knows the task is servicing an AST.

- An AST can not be interrupted by another AST. The additional

AST's are queued to the task by the executive. They are released to the task in FIFO order. The de-queueing of the AST's will resume when the current AST service routine is exited if AST recognition was not disabled.

- Recognition of AST's can be disabled. The programmer may not want certain sections of the code to be interrupted such as when updating a common region shared by several tasks. RSX provides this.

- Recognition of AST's can be enabled by Executive directive.

- A task is returned to the state it was in prior to AST recognition if not altered by the service routine. The service routine may alter the task's state if necessary.

- The stack must be cleaned before the service routine is exited. The data pushed on to the stack is dependent on the event which triggered the AST. All data up to but not including the directive status word (DSW) must be popped from the stack.

- Exit from an AST service routine to via Executive directive.

More detailed information on specific AST's and finer details are available in the Executive Reference Manual. Asynchronous System Traps are very useful. I encourage all to use this powerful feature of RSX when appropriate to your application.

The following program demonstrates the random occurrence of AST's during program execution. Run this program several times and you will see the address of the next instruction to be executed when the program returns from the service routine.

```
        .TITLE  AST
;
;       Program to demonstrate the random interruption of a program
;       by an Asynchronous System Trap
;
;       Extract necessary Macros
;
        .MCALL  DIR$,MRKT$,QIOW$,ASTX$S,EXIT$S
;
;       Set up Directive Parameter Blocks
;
MARK:   MRKT$   ,1,2,MRKAST
OUTTTY: QIOW$   IO.WLB,5,5,,,,<OUTMSG,OUTLEN>
;
;       Set up local storage
```

```
;
OUTMSG: .ASCII  <12><12><15>/The program was interrupted at /
        .ASCII  /location /
OUTPC:  .BLKB   6.
OUTLEN = .-OUTMSG
;
        .EVEN                   ; Code must start on word boundary
;
;       Start of Code
;
START:
        MOV     #1,R5           ; Set looping flag
        DIR$    #MARK           ; Issue Mark Time
;
;       LOOP UNTIL INTERRUPTED
;
LOOP:
        .REPT   1000.           ; Repeat instruction 1000 times
;
        NOP
;
        .ENDR                   ; End Repeat Block
;
        TST     R5              ; Are we finished
        BEQ     EXIT            ; YES - Exit
        JMP     LOOP            ; Loop Again
EXIT:
        EXIT$S                  ; Exit Program
;
MRKAST:                         ; Start of Mark Time AST Service Routine
;
;       Convert Return PC value to Octal ASCII and print
;
;       Task Stack is as follows
;
;       SP+10   Event Flag Mask Word
;       SP+6    PS of Task prior to AST
;       SP+4    PC of Task prior to AST
;       SP+2    Directive Status Word
;       SP+0    Event Flag Number
;
;

        MOV     #OUTPC,R0       ; Address to put converted number
        MOV     4(SP),R1        ; Get return PC from Stack
        MOV     #1,R2           ; R2 <> 0 Leave leading zeroes
;
        CALL    $CBOMG          ; Convert binary to Octal ASCII
;
        DIR$    #OUTTTY         ; Output to terminal
;
        CLR     R5              ; Clear looping flag
;
```

```
        TST     (SP)+           ; POP Stack down to DSW
;
;       Task Stack as follows:
;
;       SP+6    Event Flag Mask Word
;       SP+4    PS of Task prior to AST
;       SP+2    PC of Task prior to AST
;       SP+0    Directive Status Word
;
;
        ASTX$S                  ; Exit AST Service Routine
;
        .END    START           ; End of Program
```

Cheap (Free) RSX Networks

David R. Birkenmeyer
Clark Equipment Company
Battle Creek, MI

R. J. Hopp, Session Chairperson
Swift & Company
Oak Brook, IL

Reported by Joy Weese, DECUS Scribe Service

The "networks" mentioned are actually software mechanisms for point-to-point transfers of files. First, though, the processors must be connected with a wire. This can be done by purchasing a null modem cable and swapping the transmit and receive lines of the two processors or by linking the processors with modems.

There is a problem, known as the MCR (or "huh, what?") loop, with having two processors hooked together when each one thinks the other is a terminal. If one processor sends a syntax error to the other processor (which it thinks is a terminal), the second processor will receive that error message and think it is a syntax error and will send a message to the other processor (its terminal). They will rapidly exchange messages until the system pool is drained, which takes a very short time. To solve this problem one of the processors must be made a "slave" when making the physical connection.

The first step in writing the software for file transfer is to create a program which the speaker calls VTT. This program turns the terminal of one processor into a virtual terminal for the other processor. This program can be used to test for a valid connection between processors.

There are two kinds of interprocessor communications. The first is the "One Sided" Link which has software on only one of the processors. This kind of link can operate only in ASCII mode. The "VTT" mode is inherent in the program and it is bootstrappable but there is no error checking. The second kind is the "Two Sided" Link with cooperating software on each end. ASCII and binary mode transfers are possible and error checking can be implemented, but it is not bootstrappable.

There are several "One Sided" Link packages available from the DECUS tapes. "TALK" (DECUS--Spring '82 RSX SIG Tape [352,2], submitted by: Bob Turkelson, NASA/Goddard Space Flight Center) is a MACRO package which features bi-directional data transfers and is configurable for PDP/RSX-11M, VAX/VMS and Sigma 9. However, "TALK" requires a dedicated DL-11 (not known to RSX) and assembly time parameter customization. "XMITR" (DECUS--Spring '81 (updated Fall '81) RSX SIG Tape [312,315], submitted by: Glen Evernart) is written in MACRO and FORTRAN. It claims the ability to do bi-directional data transfers simultaneously. "XMITR" is IAS oriented and may need RSX customization.

"SNDRCV" (DECUS--Fall '80 RSX SIG Tape [301,44], submitted by: unknown) is a "Two Sided" link package written in MACRO. Multiple files are queued. Error checking and recovery are featured. "SNDRCV" is not bi-directional and requires an operator on each end. The speaker's "Two Sided" link package is called "XFR" and is not yet available on DECUS tapes. This MACRO package transfers and preserves attributes on all file types, but it has no RMS file transfers (at present). "VAXNET" (DECUS--Fall '81 RSX SIG Tape [343,51], submitted by: Project Software and Development, Inc.) is a FORTRAN and MACRO menu-driven package which does communication between the VAX/VMS and the PDP/RSX. It will not do RSX to RSX transfers.

RSX-11M Taskbuilder Tutorial

Brian McCarthy
Digital Equipment Corporation
Nashua, NH

Jonn Vilandre, Session Chairperson
University of Minnesota
Minneapolis, MN

Reported by Marc Caffee, DECUS Scribe Service

This session dealt with the use of the TKB utility. In particular techniques involved in using overlays and libraries were discussed.

It's not unusual for a program to start out rather small and as time goes on the program mushrooms to the point where it uses a substantial amount of memory which in turn causes an overall slowdown in system performance. The solution to such a problem in RSX involves using overlays. The four steps in developing an overlay structure are to develop the overlay structure, sketch in the total path, isolate transitions, and finally generate the ODL tree.

In developing the overlay structure the first thing to remember is that you can't move sideways in the tree. The next thing to remember is not to go overboard on writing separate in your programs. If you do this the processor spends more time loading and unloading the separate branches than it should with the result being hampered performance.

There are four types of transitions to be isolated: the root segment, moving down a level, moving sideways across a boundary, and moving up a level. To move down you use"-*(", to move sideways use a ",", and moving up requires a ")". For example you might have a structure that looks like A-*(B-*(C,D),E). Here A is the root segment, B and E are separate branches, and C and D are down from B. C and D lie beside each other. In an ODL file the formatting is similar to that of MACRO-11. You always need at least a .ROOT and a .END command. TKB support two types of loading mechanisms: autoload and manual load. The * signifies the use of auto load. If you use an ODL file you use the /MP switch on the input file when you taskbuild (A=A/MP). This switch TKB to look for an ODL file named "A" and name the output file "A". When you use this switch you do not use the "" switch to get options; you will automatically get options when you specify a ODL file.

You can imagine that in a long program the ODL file could be quite complicated and cumbersome. To alleviate this problem you use the .FCTR command. We might say for example:

```
        .ROOT    A-*(BFCTR,E)
BFCTR   .FCTR    B-*(C,D)
        .END
```

This does precisely the same thing as the previous example except we have used multiple lines to define the ODL file.

Another feature of overlaying is path loading. By this we mean that all the modules in a particular segment are autoloaded when one of the modules is called. However, this may not really be necessary. By removing the "*" you eliminate the autoload. For example:

```
.ROOT A-(B-*(C,D),*E)
.END
```

In this example module B is not autoloaded with the rest of the modules. It is also possible to load data segments in an ODL file. Another feature is the ability to create two independent overlay trees, which are called co-trees. An example of a co-tree ODL is:

```
.ROOT A-*(B-*(C,D),E), FCS-*(OPEN,CLOSE)
.END
```

Here we have used the FCS library as a co-tree.

It is also possible to use TKB to access commons. A common allows you to share data or code between different tasks. Switches which you use are -HD, PI, -PI, LI, CO. Also the LI and CO switches specify whether you want to access a common or a library. The first step in accessing a common is to built the common. In MACRO-11 the .PSECT directive is used and in Fortran the BLOCK DATA statement is used. You then compile the common and then you build it with TKB. Since it is not a task you must specify the -HD switch. You should also specify Stack=0 in the options. For further details look at the TKB manual. To access the common from a Fortran program you sue a COMMON statement. When you taskbuild in the options sections specify RESCOM=[common name]. You then install the common and then run the task. In RSX-11M the common must be in it's own partition. This is not the case for M+.

There is also the capability for device commons within TKB. These are built essentially the same as a library and they allow Fortran access to a device. These may also be used with the CINT$. directive.

TKB also allows access to object libraries. The rationale for doing this is that TKB is more efficient in fetching object files out of a library. The procedure is to compile the files and then create a library and install the modules into the library. In TKB you specify the name of the library followed by the LB switch.

Other points of interest, especially in Fortran, are the following options which can enhance TKB performance:

ACTFIL = n	number of files simultaneously open
EXTSCT = n	section name
FMTBUF = n	size of FMT buffer
STACK = n	stack size
UNITS = n	number of logical units
MAXBUF = n	number of files in largest record which you can read or write to

There are several features which M+ has. In particular the /MU switch separates read only and read-write sections and the /FD switch separates I and D space tasks. M+ also supports libraries which run in supervisor mode.

Factors which will affect TKB performance are:

1. The table size estimator

2. Placement of the work file

3. The memory size TKB can work in

4. Use of libraries

There are other defaults which can change TKB performance however you should proceed with caution.

New Queue Manager Features and How to Use Them in RSX-11M/M-PLUS

Cathy Ziegelmiller
Digital Equipment Corporation
Nashua, NH

Reported by Marc Caffee, DECUS Scribe Service

During this session several of the features which appeared in RSX-11M V4.0 and RSX-11M-PLUS are presented. Before we look at these features it might be helpful to explain what the queue manager (QMG) does and what is its relationship the despooler (LPP). As an illustrative example consider an office manager and the employees. Essentially, the manager finds out what needs to be done and then gets one of the office workers to perform the task. In general the queue manager is not unlike the office manager and LPP can be likened to the employees. In order to better understand just how the queue manager goes about its work we will look at some of the available switches and options and some pitfalls in using them.

One of the premier areas of confusion is whether to use a jobswitch or fileswitch. It should first be mentioned that a job is merely a group of files. As an example suppose you want two copies of a file printed. If you use the CO:2 (MCR) jobswitch and request the file to be deleted you will only get one copy of the file. However, if you use this switch as a file switch the file is submitted twice to the printer and subsequently deleted.

Features of the queue manager which will be covered include the ability to establish an input spooler, shared spooling, submission notification, naming options, and task build options.

To set up an input spooler it is first necessary to install $CKP. The next step is to create an input queue and finally you initialize the processor. One of the available switches will allow you to specify another terminal for console output.

By using the /SP/SHR (MCR) switches you can have shared spooling. There are, however several rules to remember:

1. The terminal cannot be attached (QMG will attach the terminal when a job is coming).

2. The terminal cannot be logged in.

3. There must be an existing queue.

4. Spool a console terminal in startup command, prior to changing the UIC.

Once the terminal is spooled as shared, you can log on, and other tasks can attach to it.

Using the Print or Submit commands will cause a submission notification on your terminal. To suppress this message use the /NM switch.

Job numbers are now 3 digits long and range from 1-999. Also processor names now match their queue names.

There are now task build options which allow you to change QMGBLD.BLD and LPPBLD.BLD. By changing QMGBLD.BLD the location and name of the queue file can be changed. Also the job attributes can be changed. Finally there are LA50, LA100, and LN01 options in RSX-11M V4.1 and RSX-11M-PLUS V2.1. By changing LPPBLD.BLD you can define delete checking, change the automatic deletion of certain files, and vary form width settings. Also in V4.1 and V2.1 the LA50, LA100, and LN01 options are available in addition to the ability to do form feeds at the end of a job.

It is also possible to write your own despooler. The executive directives which will help you do this are SRDA$, ASTX$S, DSAR$, RCVD$, STOP$, RCST$, WTSE$, WTLO$, SDAT$, and USTP$. QMG manager modules to be used include QMGSYN and FPRIV. The first step in writing a despooler is to initialize the despooler. To do this you must have a corresponding queue, an installed despooler task, and an associated device. When the despooler is initialized the QMG will send a startup to the despooler, activate the despooler task, enter the despooler into the symbiotic control block, and attach the device (except where shared despooling is specified).

Some details which you should remember are always slave the despooler task and build the despooler with QMGSYM. Also, you may allow the despooler to be nonprivileged. It is also recommended that you include a message handler and that you include lots of comments.

KWIC Index of Multi-Tasker Articles

Dave Birkenmeyer
Clark Automated Systems
P.O. Box 3000
Battle Creek, MI 49016

The following pages contain an up to date KWIC cross reference of all the Multi-Tasker articles since August 1980. The May and June issues of 1980 contained a similar index for the preceding issues.

>>> SOFTWARE	PERFORMANCE REPORTS	AUG-80
-OR RSX-11M >>> F11ACP	PERFORMANCE MEASUREMENTS F-	OCT-81
>>> SYSTEM TUNING,	PERFORMANCE OPTIMIZATION	JAN-81
>>> SOFTWARE	PERFORMANCE REPORTS	SEP-80
-SGENS UNDER VAX/VMS >>>	PERFORMING RSX-11M V3.2 SY-	JUL-82
>>> SHORTCUTTING SYSGEN	PHASE 2 AND 3	AUG-82
>>> REESE'S	PIECES	JAN-82
-ING I/O >>> IAS	PIP EXISTING WITH OUTSTAND-	JUL-81
>>> BASIC	PLUS 2 V1.6 PATCHES	AUG-80
>>> RSX-11M	PLUS CTS-11 DRIVER	APR-81
>>> RSX-11M	PLUS ICS/ICR SUPPORT	APR-81
-ESSION >>> RSX-11M/M	PLUS QUESTION AND ANSWER S-	JUN-81
-RROR >>> FORTRAN IV	PLUS V3.0 RUNTIME SYSTEM E-	SEP-80
>>> RSX-11M	PLUS VERSATEC DRIVER	FEB-81
>>> A	PO.E.M.	MAY-82
--11M >>> FLOATING	POINT EMULATOR CRASHES RSX-	SEP-80
>>> RSX-11M	POOL FRAGMENTATION	FEB-82
>>>	POOL MONITOR PROGRAM	SEP-80
>> MAKE USE OF SECONDARY	POOL UNDER RSX-11M-PLUS	APR-82
>>>	POWER FAILURE RESTART	AUG-82
>>> DISK DRIVERS AND	POWERFAIL RECOVERY	MAR-81
>>> PATCH TO INITL TO	PRE-ALLOCATE I/O PACKETS	AUG-80
>>> ANAHEIM	PRE-SYMBOSIUM SEMINARS	AUG-82
>>> SOFTWARE	PREFORMANCE REPORTS	AUG-82
TAPE ABSTRACTS >>>	PRELIMINARY SPRING '82 SIG	JUL-82
-OR TECHNICAL MANUSCRIPT	PREPARATION >>> A SYSTEM F-	OCT-81
>>>	PRESERVE, DSC, BRU	OCT-81
>>> FALL 1982 SYMPOSIUM	PREVIEW	SEP-82
>>> IAS	PRINT QUEUE PROBLEM	MAR-81
>> FORTRAN ACCESS TO THE	PRINT SPOOLER	DEC-80
-METIC MOD TO THE SIMPLE	PRINT SPOOLER ...COS-	SEP-80
-ND >>>	PRM PATCH TO INSTALL COMMA-	AUG-81
>>> IAS PRINT QUEUE	PROBLEM	MAR-81
RSX011M-PLUS V2.0 BOOT	PROBLEM >>>	SEP-82
>>> TASK BUILDER	PROBLEM	SEP-81
>>> SRD COMMAND FILE	PROBLEM	SEP-81
>>> SRD	PROBLEM	AUG-81
>> RSX-11M ERROR LOGGING	PROBLEMS	JUL-82
>>> AUTOPATCH E	PROBLEMS	FEB-82
>>> WONDERFUL HARDWARE	PROBLEMS	JUN-81
>>> DDT,FPEM	PROBLEMS	MAR-82
>>> RX02/RX03	PROBLEMS IN RSX-11M V4.0	SEP-82
-1M V4.0 ON-LINE >>>	PROBLEMS IN BRINGING RSX-1-	SEP-82
>>>	PROCEDURAL TEST LANGUAGE	MAR-81
-R >>>	PROCESS CONTROL QUESTIONAI-	JUL-82
>>> RSX	PRODUCT PANEL	JAN-82
>>>	PROGRAM	OCT-83
-G THE SIZE OF A FORTRAN	PROGRAM >>> REDUCIN-	FEB-82
-LETED FILE RECLAIMATION	PROGRAM >>> A DE-	DEC-80
>>> POOL MONITOR	PROGRAM	SEP-80
>>> USER	PROGRAM CORRECTIONS	MAY-82
>>> TALK	PROGRAM CORRECTIONS	APR-82
>>> UND - A	PROGRAM TO UNDELETE FILES	JUL-81
>> USING ODT IN OVERLAID	PROGRAMS	AUG-80
-REQUENTLY ORDERED DECUS	PROGRAMS >>> MOST F-	MAY-82

```
           >>> 1980 MENU  RESULTS                          FEB-81
-NTEREST GROUP 1979 MENU  RESULTS> RSX/IAS SPECIAL I-     AUG-80
                     >>>  RETIRED VERSIONS OF RSX-11      FEB-82
-M                   >>>  RETIRED VERSIONS OF RSX-11-     MAR-82
  RSX-11M V4.0 SYSGEN ON  RK05                    >>>     SEP-82
             >>> RSX-11M  RK07 SYSGEN                      APR-81
        >>> USING UNIBUS  RM02 EMULATOR ON PDP 11/70      DEC-80
-JUSTING MEMORY SIZE FOR  RMDEMO                 >>> AD-  SEP-82
-R RSX-11M V4.0      >>>  RMDEMO I/O DISPLAY PAGE FO-     SEP-82
                     >>>  RMDEMO PATCH                     FEB-81
          >>> USING V3.0  RMDEMO WITH RSX-11M V4.0        AUG-82
-SER WRITTEN DRIVERS AND  RMS BLOCK LOCKING      >>> U-   SEP-82
                     >>>  RMS-11K WORKAROUNDS              JAN-82
                     >>>  ROADMAP OF RSXMC.MAC            OCT-80
-ES WITHOUT LIGHTS   >>>  ROTATING LIGHTS FOR MACHIN-     MAY-82
>>> FORTRAN CALLABLE FNB  ROUTINES                        FEB-81
                     >>>  RSX 10TH ANNIVERSARY           OCT-83
                     >>>  RSX PRODUCT PANEL              JAN-82
                >>> DECUS  RSX SIG LIBRARY NEWS          AUG-82
                 >>> 1981  RSX SIG MENU RESULTS          MAY-82
               >>> COMMON  RSX SPELLING ERRORS           MAY-82
    >>> THE JOURNEY FROM  RSX TO VMS                     MAR-82
    >>> THE JOURNEY FROM  RSX TO VMS -- PART II          MAY-82
 >>> RETIRED VERSIONS OF  RSX-11                         FEB-82
-4           >>> RUNNING  RSX-11D V6.2 ON A PDP 11/4-   JUN-81
-NSTALLING MOUSETRAPS IN  RSX-11M                >>> I-  SEP-82
  >>> BOOTING RT-11 FROM  RSX-11M                       APR-81
 >>> RETIRED VERSIONS OF  RSX-11M                       MAR-82
   >>> UNIBUS MAPPING IN  RSX-11M                       SEP-82
    >>> THE MELLOWING OF  RSX-11M                       JUL-82
-RMANCE MEASUREMENTS FOR  RSX-11M   >>> F11ACP PERFO-   OCT-81
  POINT EMULATOR CRASHES  RSX-11M       >>> FLOATING    SEP-80
-ON-PROTECTED MULTI-USER  RSX-11M     >>> BUILDING N-   OCT-81
                     >>>  RSX-11M ANNOUNCEMENT          OCT-81
-S           >>> READING  RSX-11M BRU TAPES UNDER IA-   FEB-82
-NCES       >>> RT-11 TO  RSX-11M CONVERSION EXPERIE-   JUL-82
                 >>> NEW  RSX-11M DIRECTIVES            FEB-82
-LEMS                >>>  RSX-11M ERROR LOGGING PROB-   JUL-82
                 >>> 24K  RSX-11M EXECUTIVE             JUN-81
-CTED INTERRUPTS     >>>  RSX-11M HANDLING OF UNEXPE-   SEP-80
                     >>>  RSX-11M MAGIC SESSION         JUN-81
                     >>>  RSX-11M MULTIUSER TASKS       FEB-81
             >>> FORCING  RSX-11M OR IAS CRASH DUMPS     AUG-81
                     >>>  RSX-11M PLUS CTS-11 DRIVER     APR-81
-ER                  >>>  RSX-11M PLUS VERSATEC DRIV-   FEB-81
-RT                  >>>  RSX-11M PLUS ICS/ICR SUPPO-   APR-81
                     >>>  RSX-11M POOL FRAGMENTATION     FEB-82
-ANSCRIPT OF SPRING 1980  RSX-11M Q&A            >>> TR-  NOV-80
-ACE                 >>>  RSX-11M QMG/SPOOLER INTERF-   MAR-81
-R SECTION           >>>  RSX-11M QUESTION AND ANSWE-   FEB-82
-R UPDATE    >>> CHICAGO  RSX-11M QUESTION AND ANSWE-   JAN-81
                     >>>  RSX-11M RK07 SYSGEN           APR-81
             >>> HOW THE  RSX-11M SHUFFLER WORKS         APR-82
                     >>>  RSX-11M SPEAKS DCL            AUG-80
          >>> STANDALONE  RSX-11M SYSTEMS               JUN-81
```

```
       >>> CORE TUNING A  RSX-11M SYSTEM                  MAY-81
-SHOP                >>>  RSX-11M SYSTEM TUNING WORK-     JUN-81
-LER  >>> A USER WRITTEN  RSX-11M TIMESHARING SCHEDU-     FEB-82
     >>> CONVERSION FROM  RSX-11M TO VAX/VMS             AUG-82
                 >>> TRUE  RSX-11M TYPEAHEAD             MAR-81
 HIDDEN COTCHAS      >>>  RSX-11M V3.2 QUICKIES AND      NOV-80
  VAX/VMS >>> PERFORMING  RSX-11M V3.2 SYSGENS UNDER     JUL-82
-D UNDOCUMENTED...   >>>  RSX-11M V3.2 DOCUMENTED AN-    OCT-80
       >>> FIELD TESTING  RSX-11M V4.0                   MAR-82
-AX                  >>>  RSX-11M V4.0 SYSGEN ON A V-    SEP-82
>> RX02/RX03 PROBLEMS IN  RSX-11M V4.0                   SEP-82
-PORT                >>>  RSX-11M V4.0 FIELD TEST RE-    FEB-82
-BLE MANUALS         >>>  RSX-11M V4.0 MACHINE READA-    AUG-82
-MO I/O DISPLAY PAGE FOR  RSX-11M V4.0      >>> RMDE-    SEP-82
           >>> USING THE  RSX-11M V4.0 .BLD FILES        SEP-82
                     >>>  RSX-11M V4.0 RELEASE NOTES     MAY-82
  FIXES              >>>  RSX-11M V4.0 SOFTWARE TOOL     SEP-82
>>> PROBLEMS IN BRINGING  RSX-11M V4.0 ON-LINE           SEP-82
  USING V3.0 RMDEMO WITH  RSX-11M V4.0           >>>     AUG-82
  (CONT)             >>>  RSX-11M V4.0 RELEASE NOTES     AUG-82
-EMENTS              >>>  RSX-11M V4.0 KMSKIT ENHANC-    JUL-82
-5                   >>>  RSX-11M V4.0 SYSGEN ON RK0-    SEP-82
-ENCES       >>> ATLANTA  RSX-11M V4.0 SYSGEN EXPERI-    SEP-82
                     >>>  RSX-11M VFY                    APR-81
 OF SECONDARY POOL UNDER  RSX-11M-PLUS >>> MAKE USE      APR-82
 AN EXPOSITION ON SAV IN  RSX-11M-PLUS V2.0     >>>      SEP-82
-D ANSWER SESSION     >>>  RSX-11M/M PLUS QUESTION AN-   JUN-81
>>> HOME DIRECTORIES FOR  RSX-11M/M+                     JUL-82
                     >>>  RSX-11S IBV11 SUPPORT          APR-81
           >>> HP3000 TO  RSX/IAS COMMUNICATIONS         FEB-81
                     >>>  RSX/IAS SIG TAPE ABSTRACTS     SEP-81
                     >>>  RSX/IAS SIG TAPE ABSTRACTS     JAN-82
                     >>>  RSX/IAS SIG TAPE ABSTRACTS     NOV-81
                     >>>  RSX/IAS SIG ELECTION           JUN-81
                     >>>  RSX/IAS SIG TAPE ABSTRACTS     OCT-81
                     >>>  RSX/IAS SIG TAPE ABSTRACTS     AUG-81
-ROUP 1979 MENU...   >>>  RSX/IAS SPECIAL INTEREST G     AUG-80
                     >>>  RSX/IAS STEERING COMMITTEE     JAN-82
                     >>>  RSX/IAS WHO'S WHO              OCT-83
-BLEM                >>>  RSX011M-PLUS V2.0 BOOT PRO-    SEP-82
          >>> ROADMAP OF  RSXMC.MAC                      OCT-80
>>> FLX DESTROYS CERTAIN  RT-11 DIRECTORIES             NOV-80
              >>> BOOTING  RT-11 FROM RSX-11M           APR-81
-N EXPERIENCES       >>>  RT-11 TO RSX-11M CONVERSIO-    JUL-82
-11/23               >>>  RUNNING IAS V3.1 ON A LSI--    MAY-81
 PDP 11/44           >>>  RUNNING RSX-11D V6.2 ON A      JUN-81
    >>> DIGITAL STANDARD  RUNOFF                         JAN-81
-OARD          >>> HAMMA  RUNS FOR DECUS EXECUTIVE B-    MAR-82
>>> FORTRAN IV PLUS V3.0  RUNTIME SYSTEM ERROR          SEP-80
-11M V4.0            >>>  RX02/RX03 PROBLEMS IN RSX--    SEP-82
       >>> BUG FIXES FOR  SAMPLE ACP                    AUG-82
    >>> AN EXPOSITION ON  SAV IN RSX-11M-PLUS V2.0      SEP-82
-TEN RSX-11M TIMESHARING  SCHEDULER  >>> A USER WRIT-    FEB-82
 MULTI-TASKER INDEX WORD  SEARCH GAME  >>> COMPLETE     OCT-83
-1M-PLUS >>> MAKE USE OF  SECONDARY POOL UNDER RSX-1-    APR-82
```

```
-11M QUESTION AND ANSWER  SECTION          >>> RSX--  FEB-82
>> ANAHEIM PRE-SYMBOSIUM  SEMINARS                    AUG-82
>> TRAINING/PRE-SYMPOSIA  SEMINARS                    MAR-82
-ER SESSION >>> CUSTOMER  SERVICES QUESTION AND ANSW-  JUL-81
>> BEST OF THE SIG TAPES  SESSION                     AUG-82
-LUS QUESTION AND ANSWER  SESSION    >>> RSX-11M/M P-  JUN-81
 IAS QUESTION AND ANSWER  SESSION              >>>    AUG-82
       >>> RSX-11M MAGIC  SESSION                     JUN-81
-AS QUESTIONS AND ANSWER  SESSION             >>> I-  JUN-81
 IAS QUESTION AND ANSWER  SESSION              >>>    JAN-82
-CES QUESTION AND ANSWER  SESSION >>> CUSTOMER SERVI-  JUL-81
                    >>>  SETTING EDI V3.0 DEFAULTS    SEP-82
 2 AND 3            >>>  SHORTCUTTING SYSGEN PHASE    AUG-82
     >>> HOW THE RSX-11M  SHUFFLER WORKS              APR-82
  >>> THE HISTORY OF THE  SIG                         OCT-83
             >>> RSX/IAS  SIG ELECTION                JUN-81
                    >>>  SIG HONOREES                 OCT-83
                    >>>  SIG LEADERSHIP CHANGES       MAY-82
          >>> DEDUS/RSX  SIG LIBRARY NEWS             MAY-82
          >>> DECUS/RSX  SIG LIBRARY NEWS             JUL-82
          >>> DECUS RSX  SIG LIBRARY NEWS             AUG-82
           >>> 1981 RSX  SIG MENU RESULTS             MAY-82
           >>> RSX/IAS  SIG TAPE ABSTRACTS           NOV-81
  PRELIMINARY SPRING '82  SIG TAPE ABSTRACTS     >>>  JUL-82
          >>> RSX/IAS  SIG TAPE ABSTRACTS            JAN-82
          >>> RSX/IAS  SIG TAPE ABSTRACTS            OCT-81
          >>> RSX/IAS  SIG TAPE ABSTRACTS            AUG-81
          >>> RSX/IAS  SIG TAPE ABSTRACTS            SEP-81
         >>> FALL 1981  SIG TAPE DISTRIBUTION        APR-82
       >>> BEST OF THE  SIG TAPES SESSION            AUG-82
  ...COSMETIC MOD TO THE  SIMPLE PRINT SPOOLER       SEP-80
    >>> ADJUSTING MEMORY  SIZE FOR RMDEMO           SEP-82
       >>> REDUCING THE  SIZE OF A FORTRAN PROGRAM  FEB-82
-TEM BOOT           >>>  SLAVING THE CONSOLE ON SYS-  JUL-81
          >>> DSC/BRU  SLIDES                       AUG-80
       >>> AUTO-DIALER  SOFTWARE                    NOV-81
                    >>>  SOFTWARE CLINIC VOLUNTEERS  FEB-81
-TS                 >>>  SOFTWARE PERFORMANCE REPOR-  AUG-80
-TS                 >>>  SOFTWARE PREFORMANCE REPOR-  AUG-82
-TS                 >>>  SOFTWARE PERFORMANCE REPOR-  SEP-80
-TS                 >>>  SOFTWARE PERFORMANCE REPOR-  JUL-82
       >>> RSX-11M V4.0  SOFTWARE TOOL FIXES         SEP-82
              >>> BRU  SORTING BUG                   JUL-82
          >>> RSX-11M  SPEAKS DCL                   AUG-80
-9 MENU...    >>> RSX/IAS  SPECIAL INTEREST GROUP 197-  AUG-80
-ES-11 ON-DISK STRUCTURE  SPECIFICATION      >>> FIL-  APR-82
         >>> COMMON RSX  SPELLING ERRORS             MAY-82
                    >>>  SPM-11M DECUS OVERHEADS     AUG-81
-RAN ACCESS TO THE PRINT  SPOOLER           >>> FORT-  DEC-80
 MOD TO THE SIMPLE PRINT  SPOOLER       ...COSMETIC   SEP-80
 >>> NON-PHYSICAL DEVICE  SPOOLING                   JUL-81
 >>> DIGITAL RESPONDS TO  SPR RESOLUTION            MAY-82
                    >>>  SPR RESOLUTION             JUN-81
                    >>>  SPR SURVEY RESULTS         SEP-82
-CTS     >>> PRELIMINARY  SPRING '82 SIG TAPE ABSTRA-  JUL-82
```

The DeVIAS Letter

Section Two

From the Editor

I am continually amazed at the magnitude of the effort which can result from a "minor inconvenience" imposed by major suppliers of computer equipment.

For example, the necessity of retask-building to go from one release of an operating system to the next is obviously a huge effort on the part of the supplier. But what about the one thousand odd sites? Is it any less of an effort for a customer with over 500 unique programs, over 10,000 modules, almost 2,000 command files and over 3 million lines of code. My department is involved in such an effort in upgrading (finally) from RSX-11D version 6.2 to IAS V3.1.

We were forced to abandon the simultaneous upgrade from Fortran 44 to Fortran 77 due to dramatic differences in the OTS routines and other undocumented differences.

I hope to provide a blow by blow description of the resultant "smooth transition" this Fall (the project is scheduled to take six months for 12 sites (PDP 11/70's and 11/45's).

Is this a scheme to persuade PDP users to downgrade to a VAX to avoid retask-building between releases?

Contributions

The DeVIAS Letter needs contribution in order to continue as an effective medium for exchange of information regarding IAS.

Contributions may be submitted in any form you wish. Originals on 8½ x 11 paper are preferred. However, even photocopies of relevant match-book covers would be appreciated.

Send all contributions to:

Ontario Hydro
700 University Avenue
Toronto, Ontario

Department of Radiation Therapy
University of Pennsylvania
Room 410 - 133 South 36th Street
Philadelphia, Pennsylvania 19104
6 January 1984

Dear IAS SIG Member,

Happy New Year. There's a very good chance that it will be a great year for IAS fans. Anyone running IAS on a Pro 350? I have only heard on one person trying. How about on a Micro-11?

Mr. Hudson, Data and Research Services, in England has noted a bug in using large FORTRAN direct access files. He traced it to a difference in the way an 11/44 executes a DIV instruction compared with an 11/60. He was running his application on an 11/60 and the same code failed on the 11/44. Both processors set the V bit on overflow, but the 11/60 leaves the original arguments unchanged, the 11/44 doesn't. The IAS code misses somewhere. The suggested workaround: Use the RSX SYSLIB to build the task, RSX never uses EIS. This paragraph is my memory of the event last month and the accuracy is not guarranteed. I mention it to alert those of you having similiar problems to call me.

Which brings up a sore point. My mailing address. If you wish to send me mail with a high probability of delivery use my PO Box:

R.F. Curley
P.O. Box 322
Flourtown, PA 19031-0322

If you send a package by UPS the PO Box won't work. Use the address of my office at the top of this letter. U.S. Mail packages to the above address cause me problems if there is no one in the office to receive them (once in a while). The Post Office leaves a little yellow slip and I must pick it up at the main Post Office, a parking and access problem. Do Not Ever send mail to me at the old address of 3400 Spruce Street. I still hear stories of returned and missing mail.

And when you call, please use 215-662-3083 (my office). My secretary, Linda Michael, is very good at taking messages. Please tell her what you want/need from me. It is unnecessarily expensive for me to call you back just to find out that I need to do some homework first, then call you back again. Thanks.

While I'm picking bones, please drop me a note when you find an an interesting or irritating feature of IAS. Especially it you've found a workaround. I'd like to share it. Also, please send me a copy of your SPR's. I've started sending them in again with very rewarding results (that I can't spell yet). Please use your SPR's when you have a problem or bug. It is VERY helpful to all of us. And, COMPLAIN when you are not satisfied with the answer.

A recent contribution to my blackboard cartoons (by a student who appreciated good humor):

$$\frac{2(NaCl\ aq)}{(C\text{-}C\text{-}C\text{-}C\text{-}C\text{-}C\text{-}C)}$$

What is it? The answer: Saline, Saline, over the seven C's No fair saying bad words until you've made a contribution!

As usual we need material for the newsletter. I understand how hard is it to write an article. When you write a subroutine that does something interesting or potentially useful, send me a paper copy. If it really is too murky, I'll talk to you about it. But many of you write subroutines that do neat stuff. Even I do! Which reminds me, next issue I've got one to include. And making a contribution as a subroutine is a very precise way of doing it, even if the English part is sketchy, the computer part is exactly right. And who cares if we understand it, just as long as our PDP-11 does! Please send me some goodies, as John Drummond says, Anything that is printable!

Happy New Year,
Robert F. Curley

Mr. Drummond

Ontario Hydro
700 University Avenue
Toronto, Ontario
CANADA, M5G 1X6

—

uw brief van	uw referentie	onze referentie	Heerlen,
			21th december 1983

Dear Mr. Drummond,

We here with like to inform you about the latest SPR's we have sent to DEC. Some of these problems have been fixed at DSM and the adapted sources are sent with the SPR's.

The nature of the problems is such that they should be solved in the forthcoming release of IAS V3.2.

Sincerely,

J.L.C. Plasman
Afd. Systeem Technieken DSM
Postbus 600
6160 MJ GELEEN

digital

SOFTWARE PERFORMANCE REPORT

FIELD NO.:	CORPORATE SPR NO.:

534253

TO SET UP FOR PROPER ALIGNMENT, START AT MARK BELOW.

PAGE _____ OF _____

ERATING SYSTEM	VERSION	SYSTEM PROGRAM OR DOCUMENT TITLE	VERSION OR DOCUMENT PART NO.	DATE
AS	3.1	DK handler	IDENT /039/	9.12.83

NAME: J.L.C. Plasman
FIRM: DSM Limburg
afd. Systeem Technieken
DDRESS: Postbus 600 6160 MJ GELEEN
ST. NO.:

DEC OFFICE AND CONTACT PERSON

DO YOU HAVE SOURCES? YES [X] NO [ ]

REPORT TYPE/PRIORITY

- [X] PROBLEM/ERROR
- [ ] SUGGESTED ENHANCEMENT
- [ ] OTHER

1. [X] HEAVY SYSTEM IMPACT
2. [ ] MODERATE SYSTEM IMPACT
3. [ ] MINOR SYSTEM IMPACT
4. [ ] NO SIGNIFICANT IMPACT
5. [ ] DOCUMENTATION/SUGGESTION

BMITTED BY: . Plasman PHONE: 04494 -66776

CAN THE PROBLEM BE REPRODUCED AT WILL? YES [X] NO [ ]

ATTACHMENTS

G TAPE [ ] FLOPPY DISKS [ ] LISTING [X] DECTAPE [ ]

HER:

COULD THIS SPR HAVE BEEN PREVENTED BY BETTER OR MORE DOCUMENTATION? PLEASE EXPLAIN IN PROVIDED SPACE BELOW. YES [ ] NO [ ]

TYPE	SERIAL NO.	MEMORY SIZE	DISTRIBUTION MEDIUM	SYSTEM DEVICE	DO NOT PUBLISH
DP11/##44	19665	256K	RK05	RK07	[ ]

roblem description

Handler could hang on 11/44 doing a set characteristics function whilst seek in progress on another drive.

Handler could hang on 11/44 doing a seek on the same drive as for which a XFR has come ready, but has not yet passed the IO DONE routine. The seek is now inhibited for thés drive until the XFR has been fully completed.

ORT NAME	MNT. CAT.	MNT. GRP.	XFER GRP.	PL	PRB. TYPE

E RECEIVED (MAIL)	DATE TO MAINTAINER	XFER DATE	LOGGED ON

E RECEIVED (ASG)	DATE RECEIVED FROM MAINTAINER	DATE ANSWERED	LOGGED OFF

digital

SOFTWARE PERFORMANCE REPORT

SERVICE NO. 2M3 | CORPORATE SER NO.: PROTOTYPE

481963

TO SET UP FOR PROPER ALIGNMENT, START AT MARK BELOW.

PAGE _____ OF _____

OPERATING SYSTEM	VERSION	SYSTEM PROGRAM OR DOCUMENT TITLE	VERSION OR DOCUMENT PART NO.	DATE
RSX	3.1	F11ACP	00322	9.12.83

NAME: J.L.C. Plasmeijer
FIRM: DSM Limburg
Afd. Systeem Technieken

ADDRESS: Postbus 600 6160 MJ GELEEN

COST NO.:

DEC OFFICE

DO YOU HAVE SOURCES? YES [X] NO [ ]

REPORT TYPE/PRIORITY

- [X] PROBLEM/ERROR
- [ ] SUGGESTED ENHANCEMENT
- [ ] OTHER

1. [X] HEAVY SYSTEM IMPACT
2. [ ] MODERATE SYSTEM IMPACT
3. [ ] MINOR SYSTEM IMPACT
4. [ ] NO SIGNIFICANT IMPACT
5. [ ] DOCUMENTATION/SUGGESTION

SUBMITTED BY: J. Plasmeijer PHONE: 04494 - 66776

CAN THE PROBLEM BE REPRODUCED AT WILL? YES [X] NO [ ]

ATTACHMENTS
MAG TAPE [ ] FLOPPY DISKS [ ] LISTING [ ] DECTAPE [ ]
OTHER:

COULD THIS SPR HAVE BEEN PREVENTED BY BETTER OR MORE DOCUMENTATION? PLEASE EXPLAIN IN PROVIDED SPACE BELOW. YES [ ] NO [ ]

CPU TYPE	SERIAL NO.	MEMORY SIZE	DISTRIBUTION MEDIUM	SYSTEM DEVICE	DO NOT PUBLISH
PDP11/44	19665	256 K	RK05	RK07	[ ]

Problem description

The ACP task examines at several places taskheaders without inhibiting shuffling. This problem has already caused several system crashes.

ALL SUBMISSIONS BECOME THE PROPERTY OF DIGITAL EQUIPMENT CORPORATION.

REPORT NAME	MNT. CAT.	MNT. GRP.	XFER GRP.	PL	PRB. TYPE

DATE RECEIVED (MAIL)	DATE TO MAINTAINER	XFER DATE	LOGGED ON

DATE RECEIVED (ASG)	DATE RECEIVED FROM MAINTAINER	DATE ANSWERED	LOGGED OFF

digital

SOFTWARE PERFORMANCE REPORT

FIELD NO.:	CORPORATE SPR NO.:

534350

TO SET UP FOR PROPER ALIGNMENT, START AT MARK BELOW.

PAGE _____ OF _____

ERATING SYSTEM	VERSION	SYSTEM PROGRAM OR DOCUMENT TITLE	VERSION OR DOCUMENT PART NO.	DATE
RSX	3.1	DM handler	IDENT /24/	9.12.83

NAME: J.L.C. Plasman
FIRM: DSM Limburg
Afd. Systeem Technieken

ADDRESS: Postbus 600 MK 6160 MJ GELEEN

CUST. NO.:

DEC OFFICE AND CONTACT PERSON

DO YOU HAVE SOURCES? YES [X] NO [ ]

REPORT TYPE/PRIORITY

- [X] PROBLEM/ERROR
- [ ] SUGGESTED ENHANCEMENT
- [ ] OTHER

1. [ ] HEAVY SYSTEM IMPACT
2. [X] MODERATE SYSTEM IMPACT
3. [ ] MINOR SYSTEM IMPACT
4. [ ] NO SIGNIFICANT IMPACT
5. [ ] DOCUMENTATION/SUGGESTION

SUBMITTED BY: J. Plasman
PHONE: 04494 - 66776

CAN THE PROBLEM BE REPRODUCED AT WILL? YES [ ] NO [ ]

ATTACHMENTS

MAG TAPE [ ] FLOPPY DISKS [ ] LISTING [X] DECTAPE [ ]

OTHER:

COULD THIS SPR HAVE BEEN PREVENTED BY BETTER OR MORE DOCUMENTATION? PLEASE EXPLAIN IN PROVIDED SPACE BELOW. YES [ ] NO [ ]

CPU TYPE	SERIAL NO.	MEMORY SIZE	DISTRIBUTION MEDIUM	SYSTEM DEVICE	DO NOT PUBLISH
PDP11/44	19665	256 K	RK05	RK07	[ ]

Problem description

Error correction is wrong if XFER is not a whole no. of blocks.

ALL SUBMISSIONS BECOME THE PROPERTY OF DIGITAL EQUIPMENT CORPORATION.

SHORT NAME	MNT. CAT.	MNT. GRP.	XFER GRP.	PL	PRB. TYPE

DATE RECEIVED (MAIL)	DATE TO MAINTAINER	XFER DATE	LOGGED ON

DATE RECEIVED (ASG)	DATE RECEIVED FROM MAINTAINER	DATE ANSWERED	LOGGED OFF

DECUS SUBSCRIPTION SERVICE
DIGITAL EQUIPMENT COMPUTER USERS SOCIETY
ONE IRON WAY, MRO2-1/C11
MARLBORO, MASSACHUSETTS 01752

Bulk Rate
U.S. Postage
PAID
Fitchburg, MA
Permit No.
21

MOVING OR REPLACING A DELEGATE?

Please notify us immediately to guarantee continuing receipt of DECUS literature. Allow up to six weeks for change to take effect.

( ) Change of Address
( ) Delegate Replacement

DECUS Membership No.: _____

Name: _____

Company: _____

Address: _____

_____

State/Country: _____

Zip/Postal Code: _____

Mail to: DECUS - ATT: Subscription Service
One Iron Way, MRO2-1/C11
Marlboro, Massachusetts 01752 USA

g label
is not
nt old
address here.
Include name of installation, company, university, etc.