# DECUS U.S. CHAPTER SIGs NEWSLETTERS

1961–1986
25 Years
Leading Toward The Future

MAY 1986  Volume I, Number 9

# GENERAL TABLE OF CONTENTS

**SECTIONS**                                                                 **PAGE NO.**

## HOW TO SUBMIT AN ARTICLE .. ORDER FORMS SECTION

## QUESTIONNAIRE SECTION

# APL SIG

**Chairman (RSTS)**
Larry LeBlanc
Teletype Corp.
Elk Grove, IL 60007

**Library Coordinator**
Susan Abercrombie
Ventrex Laboratories
Portland, ME 04103

**Newsletter Editor (RT, TSX)**
Douglas Bohrer
Bohrer and Company
Wilmette, IL 60091

**Symposia Coordinator (RSX)**
Bob Awde Jr.
General Mills Inc.
Minneapolis, MN 55427

**Symposia Assistant**
Bob Van Keuren
UserWare International, Inc.
Escondido, CA 92025

**Standards Representative**
Dan Esbensen
Touch Technologies
Escondido, CA 92025

**VAX APL (Contact:)**
OPEN

**European Contact (TOPS)**
Jean-Pierre Barasz
BARTS
75008 Paris, France

**Digital Counterpart**
Dave Quigley
Digital Equipment Corporation
Nashua, NH 03062
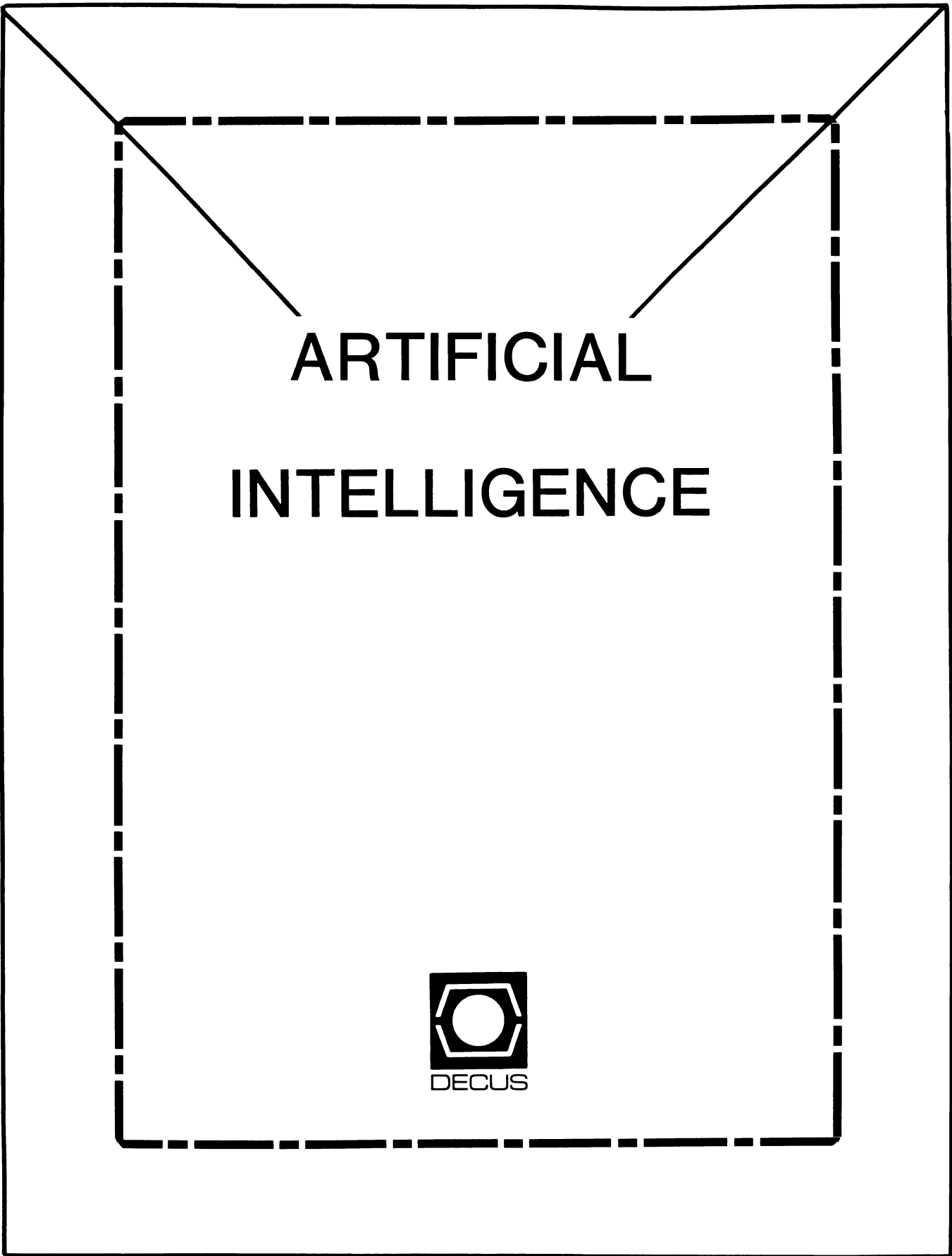
# ARTIFICIAL

# INTELLIGENCE

# ARTIFICIAL INTELLIGENCE SIC

**Chairperson**
Cheryl Jalbert
JCC
Granville, OH

**Ass't Chairman**
Don Rosenthal
Space Telescope Science Inst.
Baltimore, MD

**Symposium Coordinator**
David Slater
Institute for Defense Analysis
Alexandria, VA

**Ass't Symposium Coordinator**
**Session Note Editor**
Greg Parkinson
Cognitive Systems Inc.
New Haven, CT

**Newsletter Editor**
Terry Shannon
Digital Review
Boston, MA 02109

**Newsletter Publisher**
Bob Zeek
Pfizer Inc.
Groton, CT

**Membership Coordinator**
Pamela Vavra
KMS Fusion Inc.
Ann Arbor, MI

**PSS Scheduling**
Tom Viana

**Store & Buttons**
Sally Townsend
Inst. Defense Analysis
Alexandria, VA

**Quality Control Chair**
Dick Ciero
Harris Corp.
Palm Bay, FL

**Quality Control**
Carol Guyermelli

**Site Coordinator, Anaheim**
Chris Goodard

**Volunteer Coordinator, Anaheim**
Peter MacDonough
Tractor Inc.
California, MD

**Members-at-Large**
George Humfeld

Matt Matthews IV

**Evaluation Research Corp.**
King George, VA

**DEC Counterpart**
Art Beane
Digital Equipment Corporation
Hudson, MA

# BUSINESS

# APPLICATIONS

DECUS

# BUSINESS APPLICATIONS SIC STEERING COMMITTEE

**Chairman**
Stuart Lewis
Douglas Furniture
Bedford Park, IL

**Symposium Coordinator**
Steve Simek
IRT Corporation
San Diego, CA

**Asst. Symposium Coordinator**
Bobbie Wiley
CEI Perry Nuclear Power Plant
Euclid, OH

**LRP and Marketing Coordinator**
Arnold I. Epstein
D-M Computer Consultants
Rolling Meadows, IL

**Marketing Asst.**
George Dyer
Gallaudet College
Washington, DC

**Communications Representative**
OPEN

**Newsletter Editor**
Thomas Byrne
L. Karp and Sons
Elk Grove Village, IL

**Session Notes Editor**
Raymond Swartz
Goodyear Tire and Rubber Co.
Akron, OH

**Library Representative**
David Hittner
Projects Unlimited
Dayton, OH

**CL SIG Liaison**
Becky Burkes
Financial Insurance Consultants
Covington, LA

**DMS SIG Liaison**
Joe Sciuto
Army Research Institute
Alexandria, VA

**Members-at-Large**
Robert D. Lazenby
Dixie Beer Dist., Inc.
Louisville, KY

Robert Kayne
Gallaudet College
Washington, DC

Ray Evanson
Paragon Data Systems
Winona, MN

**Digital Counterparts**
Sue Yarger
Digital Equipment Corporation
Merrimack, NH

Ray Arsenault
Digital Equipment Corporation
Merrimack, NH

**SIC Mentor**
Bill Brindley
Networks SIG Chair

**SIC Review Committee**
Larry Jasmann
Leslie Maltz
Ted Bear
Jeff Killeen

CL SIG
SEAL of APPROVAL
Herschal

COBOL    BASIC    DIBOL    RPG

# COMMERCIAL LANGUAGES SIG

**Chairman**
Jim Wilson
Pfizer Inc. QC Div.
Terre Haute, IN

**Symposium Coordinator**
Ray Strackbein
Palm Desert, CA

**Library Coordinator**
Philip Hunt
System Industries
Milpitas, CA

**Session Note Editor**
Bob Van Keuren
Userware International, Inc.
Escondido, CA

**Newsletter Editor**
Ted A. Bear
Ramtek
Santa Clara, CA

**Ass't Newsletter Editors**
Beverly Welborne
LaPorte, IN

Kevin Cullen
VITA-Mix Corp.
Holmstead Falls, OH

Daniel Cook
Userware International, Inc.
Escondido, CA

**Basic Working Group Members**
Mark Hartman
Jadtec Computer Group
Orange, CA

Rocky Hayden
UserWare International Inc.
Escondido, CA

Bill Tabor
Computer Productss
Pompano Beach, FL

**Cobol Working Group Members**
Keith Batzel
Crowe, Chizek & Co.
South Bend, IN

Mary Anne Feerick
RDBS Inc.
Kernersville, NC

Bill Leroy
The Software House, Inc.
Atlanta, GA

Herbert J. Matthews IV
ManTech International Corp.
Alexandria, VA

Jim Welborne
Crowe Chizek & Co.
South Bend, IN

**DIBOL Working Group Members**
Neil Baldridge
CompuShare
Lubbock, TX

Becky Burkes
Financial Insurance Consultant
Covington, LA

Colin Chambers
Software Ireland Rep. Inc.
Portola Valley, CA

Mark Derrick
WAAY-TV
Huntsville, AL

Gary A.P. Kohls
Milwaukee, WI

Ken Lidster
Disc
Sacramento, CA

Kenneth M. Schilling
MCBA
Montrose, CA

Marty Schultz
Omtool Inc.
Tewksbury, MA

David L. Wyse
Professional Business Software
Dayton, OH

Marty Zergiebel
The Software Gallery
Brookfield, CT

**RPG Working Group Members**
Keith Batzel
Crowe Chizek & Co.
South Bend, IN

Ted Bear
Ramtek
Santa Clara, CA

**Digital Counterparts**
Tom Harris
Nashua, NH

Jim Totten
Nashua, NH

Joe Mulvey
Nashua, NH

Shirley Ann Stern
Nashua, NH

**Standards Representatives**
BASIC
Dan Esbensen
Touch Technologies, Inc.
Escondido, CA

COBOL
Bruce Gaarder
Macalester College
St. Paul, MN

DIBOL
Eli Szklanka
TEC
Newton, MA

ANSI X3J4 COBOL REPORT
for Dec-85 and Feb-86

by

Bruce L. Gaarder
Macalester College
1600 Grand Avenue
St. Paul, MN 55105

The ANSI COBOL committee has continued working towards adding new
features to the language after the September 1985 approval of
COBOL-85. In general, the agenda for each meeting is pretty close
to that of the last, so I have combined the last two meetings in
this report.

1.  We have processed the final interpretations of the 1974
    standard, and I edited the COBOL Interpretation Bulletin (CIB)
    containing them for publication this spring.

2.  We are already processing interpretations of the 1985 standard,
    most of which have been submitted by people close to the stand-
    ardization effort, either in the U.S. or in the ISO arena.

3.  Another item which has been on the agenda for some time is the
    proposed COBOL interface to the proposed ANSI Network Database
    Language (NDL). This is an unpopular item with most of the
    vendors, who believe that the NDL standard is years too late,
    and will not be implemented by many vendors, or used much if
    implemented. However, the government in the guise of the
    Federal Information Processing Standards (FIPS) organization
    plans to require such a COBOL interface and the ISO and ANSI
    have chartered the X3J4 committee to produce such a standard
    and we are probably the best people to do it. At the last
    meeting a resolution was passed that expressed the committee's
    sentiments regarding this standard and we are currently voting
    on sending the proposed standard to ANSI for public review.
    The general DECUS position on the NDL is neutral, so I voted
    to support the resolution saying that the NDL wasn't too
    useful (since it only expressed our sentiments) and I voted
    YES on the COBOL Interface to the NDL.

4.  A meeting of the ISO COBOL working group took place after the
    February meeting, and I will hear about the results next
    meeting. The ISO people are pressing for early revisions to
    the standard, but the review mechanism means that it will take
    a minimum of four years.

5.  We routinely hear liaison reports on what other committees are
    doing. The ones that currently are working on COBOL related
    things include: Information Resource Dictionary Systems
    (sort of like the CDD and a code management system combined);
    Screen Management; Programming Language Bindings; database;
    and graphics.

6. The list of items currently being researched as possible upwardly compatible additions to the current standard include:

    function facility;
    validate facility;
    user-defined collating sequence for indexed files;
    the VALUE ... REPEATED clause;
    the VALUE ... WHEN SET TO FALSE clause and SET TO FALSE;
    the DEPENDING ON data item in the RECORD clause may be
        qualified;
    in-line PERFORM and EXIT PERFORM;
    the GLOBAL clause with LINE-COUNTER and LINEAGE-COUNTER.

These items will be discussed at length, and there are some others which may be brought up later. I presented a session at spring DECUS about these items, and you are invited to let me know what you think of them. Send me your name and address if you want a copy of the paper that I presented at DECUS.

# DAARC

# DAARC

**Chairman**
James Deck
Inland Steel Research Lab
East Chicago, IN

**Symposium Coordinator**
Mack Overton
FDA
Chicago, IL

**Newsletter Editor**
Ellen Reilly
William H. Rorer
Ft. Washington, PA

**DEC Counterpart**
Nancy Kilty
Digital Equipment Corporation
Marlboro, MA

**Hardware & Interfacing**
Peter Clout
Los Alamos National Lab
Los Alamos, NM

**Math Statistics & Analysis**
Herbert J. Gould
C.C.F.A. University of Illinois Med Center
Chicago, IL

**Process Control - Industrial Automation**
Bill Tippie
Kinetic Systems Corp.
Lockport, IL

**RS-1**
George Winkler
CPC International
Argo IL

# DMS SIG
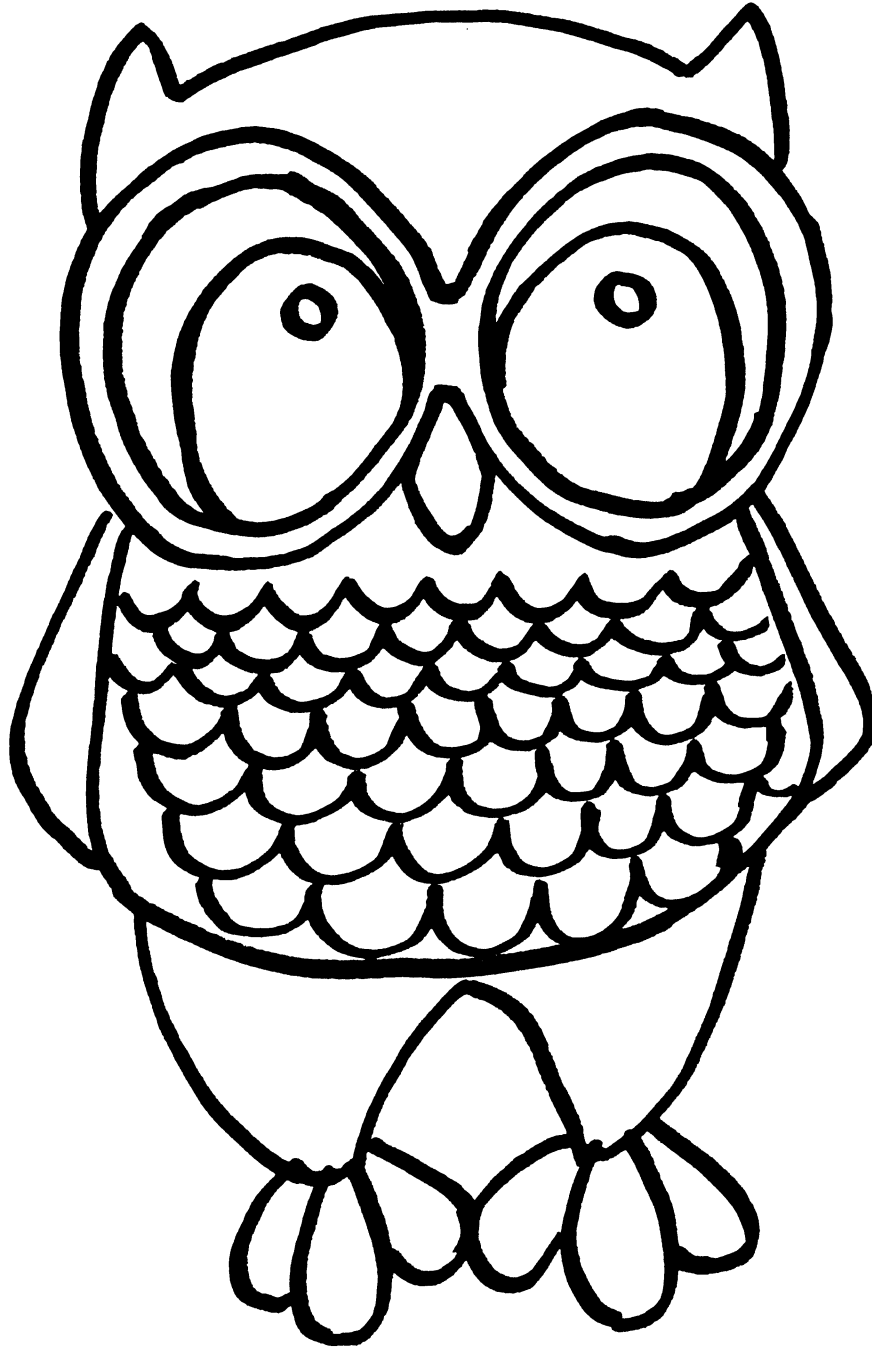


# JOIN THE WISE

Data Management Systems SIG Steering Committee
May 1, 1986
--------------------------------

SIG Chairman:
Joseph F. Sciuto
Army Research Institute
Alexandria, VA
(202) 274-9420

Comptroller:
Alan Schultz
Land Bank National DP Center
Omaha, NE
(402) 397-5040

Symposium Coordinator:
Keith Hare
JCC
Granville, OH
(614) 587-0157

Symposium Coordinator:
Barbara Mann
TRW
Redondo Beach, CA
(213) 532-2211

Communications Committee Rep
   Newsletter Editor:
J. G. Russell Poisson
SEED Software Corporation
Alexandria, VA
(800) 428-9400

Session Note Editor:


Mark Morgan
Farm Credit Banks
Springfield, MA
(413) 732-9721

Membership Coordinator:
VACANT

MIS Working Group Coordinator (Past SIG Chairman):
Steve Pacheco
Ship Analytics
North Stonington, CT
(203) 535-3092

MIS Working Group (Past SIG Chairman):
Sandy Krueger
Key Financial Systems, Inc.
Pine Brook, NJ
(201) 299-6600

Working Group Coordinator/
   Database Working Group:
Jim Perkins
PSC, Inc.
Shelburne, VT
(802) 863-8825

Forms Working Group:
Debbie Kennedy
Land Bank National DP Center
Omaha, NE
(402) 397-5040

Non-Digital Working Group:
Doug Dickey
GTE Government Systems
Rockville, MD
(301) 294-8400

RMS Working Group Coordinator:
Allen Jay Bennett
Lear Siegler Apistan
Grand Rapids, MI
(616) 451-6429

Pre-Symposium Seminar Coordinator/
   Black Book:
David B. Turner
Korn/Kerry International
Los Angeles, CA
(408) 945-9600

ANSI Standards Coordinator:
Herman "Spence" Spencer
Army Research Institute
Alexandria, VA
(202) 294-9420

Member-At-Large:
Larry W. Hicks

Relational Database Services
Kernersville, NC
(919) 996-4882

Member-At-Large:
Richard Arndt
Cognos Corporation
Houston, TX
(713) 690-1105

AI SIC Liaison:
David Slater
Institute for Defense Analysis
Alexandria, VA
(703) 845-2200

Datatrieve Liaison:
John Schutt
J. R. Simplot Company
Boise, ID
(208) 336-2110

DEC Counterpart:
Wendy Herman
Digital Equipment Corporation
Nashua, NH
(603) 881-2494

Contributions
_____

Contributions to the newsletter can be sent to either of
the following addresses:

Editor, DMS SIG Newsletter      Russ Poisson
c/o DECUS U.S.  Chapter         DMS SIG Newsletter Editor
219 Boston Post Road, BP02      SEED Software Corporation
Marlboro, MA 01752              2121 Eisenhower Avenue
                                Alexandria, VA 22314

Letters and articles for publication are requested from members
of the SIG.  They may include helpful hints, inquiries to other
users, reports on SIG business, summaries of SPR's submitted to
DEC, etc. Machine readable input is highly desirable.
Submitters should keep in mind the DECUS policy on commercialism.

_____
                      DMS SIG Product Overview
_____

Within the DMS SIG, there are a set of products known as the VAX
Information Architecture.  These products consist of VAX ACMS, VAX TDMS,
VAX CDD, VAX DBMS and VAX Rdb/VMS.  Although not part of the DMS SIG, VAX
Datatrieve is also a member of the VAX Information Architecture.  This
article will focus on the former five products and will not include
Datatrieve.  These products together give MIS and Data Processing
departments the capability of providing commercial applications for single
nodes as well as distributed and VAXcluster systems.

Also available from Digital are three packages of these products.  These
packages (known as VAXinfo I, II and III) provide combinations of the VAX
Information Architecture products on single pieces of media with single
installation procedures for each package.  This provides convenience and
ease-of-installation for these products.

VAX ACMS
----------

VAX ACMS, in conjunction with other VAX Information Architecture products,
is an application development system designed for creating transaction
processing applications on the VAX/VMS and MicroVMS Operating Systems.
These applications can be distributed, so that a user on one node can
select a task from an application existing on another node.

Applications developed using VAX ACMS can be used in:

- A single node environment.

- A VAXcluster environment with terminal users distributed across the
  VAXcluster.

- A local-area or wide-area network environment with terminal users
  distributed across the network(s).

- A combination of these environments.

The tools provided by VAX ACMS can simplify the development and maintenance
of distributed applications.  They can also provide a highly efficient
run-time environment for distributed applications.

VAX TDMS
--------

VAX TDMS is a product designed for the implementation of interactive,
forms-intensive applications running on the VAX/VMS Operating System, and
run-time only applications for MicroVAX/MicroVMS systems.  As a terminal
subsystem, VAX TDMS can reduce the application development and maintenance
effort by replacing application program logic, specific to terminal
interactions, with definitions that are external to the program.

VAX TDMS applications range from database inquiry/response/update to
real-time uses such as periodic display of an industrial process.  VAX TDMS
is typically used as a terminal subsystem in terminal data management
applications such as inventory control, manufacturing, distribution and
other form-intensive applications.

VAX CDD
-------

VAX CDD provides a single logical data dictionary for a VAX, MicroVAX or
VAXstation system, containing definitions for VAX ACMS, VAX Datatrieve, VAX
DBMS, VAX Rdb/VMS, VAX TDMS, VAX COBOL,VAX BASIC, VAX DIBOL, VAX FORTRAN,
VAX PASCAL, VAX DEC/MMS, VAX C, VAX PL/I, VAX RPG II, VAX SORT, and VAX
DECreporter.

The logical dictionary is built from one or more physical dictionary files.
This allows users to access one cluster-wide dictionary as well as

individual dictionaries per processor.  Users can now take full advantage
of VAX/VMS clusters with regard to dictionary usage.  Within this
dictionary, the CDD maintains a hierarchical directory of the user's data
descriptions; multiple versions are supported.

The CDD provides security based on its hierarchical directory.  Each
directory and data description can have its own Access Control List.  Each
entry in the Access Control List specifies a class of user and the type of
access allowed to the directory or data description.  A user class is
defined by selecting the UIC, user name, password, and terminal class
parameters.

VAX DBMS
--------

VAX DBMS is a multiuser, general purpose CODASYL-compliant database
management system that runs under the VAX/VMS and Micro/VMS Operating
Systems.  VAX DBMS is used to access and administer databases ranging in
complexity from simple hierarchies to complex networks with multilevel
relationships.  It supports full concurrent access in a multiuser
environment without compromising the integrity and security of the user's
databases.

VAX DBMS is based upon the March 1981 Working Document of the ANSI Data Definition Language Committee.

## VAX Rdb/VMS
----------

VAX Rdb/VMS is a full function relational database management system designed for the VAX/VMS and MicroVMS Operating Systems.  It is intended for general-purpose, multi-user, centralized or distributed applications. VAX Rdb/VMS supports a complete set of utilities and precompilers that enable users to maintain and manipulate databases.

VAX Rdb/VMS implements the DIGITAL Standard Relational Interface (DSRI). DSRI is an architecture for relational database management systems as well as a standard calling mechanism that can be used for database creation and population.  DSRI allows applications running on any VAX or MicroVAX node in a DECnet network to access all other DSRI compliant databases in the network.

---------------------------------------------------------------------------

### ONE MAN'S EXPERIENCE WITH RDB AND SMARTSTAR

by Tom Colati
System Analyst
Mobil Oil - Toxicology Div.
Princeton NJ 08540
(609) 737-5679

This is the first in a series of articles that I plan to submit to the newsletter outlining various things about the SMARTSTAR and RDB interface.

These articles are NOT, repeat NOT, intended to be a sales pitch for SMARTSTAR. It is intended as a means to pass on information that I have discovered and think is helpful about the SMARTSTAR/Rdb interface . This information was obtained (mostly through long hours and a lot of sweat) from reading the documentation, experimentation and conversations with DEC and STI. I would very much like to be informed if anything I say is off base. This article is geared for those who have SMARTSTAR and are familiar with its components.

## Background
----------

I am presently involved in redesigning the database for the Toxicology division of Mobil. I chose Rdb as the database manager mainly because it was here when I arrived ( I think it was tacked on some other large software order back in the days of the BIG SPENDING SPREE), so I thought I would put it to some use. I quickly discovered that there was no front-end to Rdb and when I asked DEC about this, they replied "Of course there is a frontend to Rdb, it's Datatrieve" . Well, after almost choking to death over THAT statement and realizing that Datatrieve and Managers mix as well as oil and water, I immediately set out to find a suitable frontend for Rdb. I decided on SMARTSTAR from Signal Technology Inc. (STI) from Goleta Ca.  I am sure there are other frontends that can be used for Rdb (TDMS ?, FMS ? or even FORTRAN ugh!) and even other database packages that are better (or worse), none-the-less , I  chose SMARTSTAR. Our present system configuration is Rdb V2.0, DTR V3.3, Smartstar V4.0 running on V4.3 on an VAX 8600, 11/750 cluster (at the time of this writing – March) with 3 common disks (RA81) with 8 meg of memory on the 750 and 16 meg on the 8600.

## Background of STI
----------------

STI is a small company  (about 50 people ) located in CA. They happen to have the right product at the right time and their sales took off. When they brought out the RDB interface ,I don't think they initially realized how popular it would be. They went from a small company with a nice product with  a small niche in the DEC world into the big leagues of software and along with it have come some growing pains . The company's thinking was not geared to servicing the needs of many installations as well as large installations that  demand and exce quality service from a software vendor. They are now in the process of restructering their thinking and things have started to improve. At this point , technical support is an area that is lacking the ability to handle the larger user base. While the support staff is very helpful, their depth of knowledge of the products is limited (because they are generally new to the company) . Also the scope of the questions are becoming more technical . Users are starting to ask questions beyond the basic "HOW TO" questions to the "WHY DOES THIS" and "WHICH IS FASTER" approach. Although this is a weak point in the company this area has improved drastically in the last month and it seems that it can only get better each day.

## Overview of software
--------------------

SMARTSTAR is divided into several components

SMARTDESIGN –  This module is used to create the ADF (application Definition File) . The ADF holds information such as field names, lengths, datatypes, indicies, screen formatting

SMARTQUERY –  Used to query the database through the ADF

SMARTREPORT – Used to produce reports from an ADF

SMARTCALL - The programming interface between the ADF and your program.
It is the way to get information to and from the ADF and
your program

OPAL - The programming interface to the backend database. Allows
interfacing to Rdb

REQUEST - integrated report and query system (that was taken from the
product description) . It is basically Datatrieve-like commands
(though a bit more powerful, especially in the report writer)
that can be called from OPAL. It is the way to control the
backend work

We currently have all the components except REQUEST, but we can use request
calls via OPAL. We just can't go interactively into REQUEST. We don't find
this a problem since SMARTREPORT and Datatrieve should take care of our
report writing needs.


Here is a crude map ( and one man's idea ) of how the components fit together

```
              SMARTQUERY      SMARTDESIGN      SMARTREPORT
                  |               |                |
                  |               |                |
                  |               V                |
                  ----------->ADF <---------------
                  |                | |
                  |---------------| |
                  V                 V
              SMARTCALL------------->DAR
                  |                  |
                  |                  |
                  |                  v
                  ------------------->OPAL
                                     |
                                     |
                                     v
                              ---------------
              Datatrieve ---------> | DSRI - RDB |
                              ---------------
```
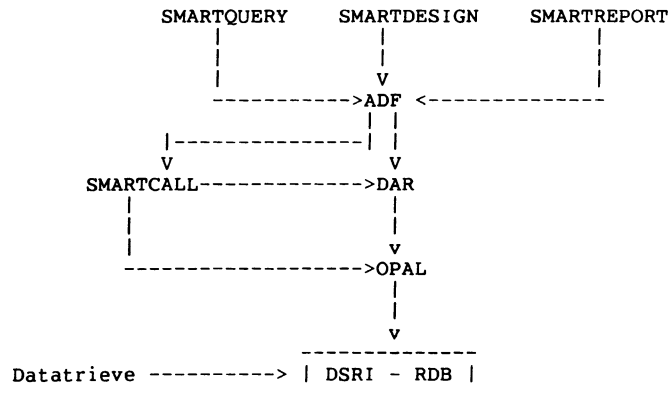
Interface
--------
There are several good maps in the manual that show how all the components fit
together. I have attempted to consolidate them here. The backend database is
/RDB, for RMS datafiles , other maps might apply


First it might be useful to define some definitions

ADF - Application definition File - This is a file that holds such information
as field names, lengths, datatypes, screen formatting

FORM BUFFER - Another name for the ADF (this is not technically correct but
it will be acceptable for now). Anything on the screen is in the
form buffer and anything in the form buffer is on the screen,

DAR - Data Action Routine - this is the interface to the backend database and

handles storing, deleting , retrieving and modifying data. It contains
OPAL routine calls. You have the ability to write your own DAR (which
we do quite a bit) or alter the DAR's supplied. Throughout the
remainder of this article I will refer to the DAR but I am really
refering to the function SC_OBACT that is used as the DAR for Rdb
databases.


There are 4 different ways you can control  a SMARTSTAR application

1. Total control of the application by SMARTSTAR. There is no host programming
required. Just define the ADF via SMARTDESIGN and invoke via SQUERY. All the
backend database work is taken care of by SMARTSTAR. This is fine for
applications that do not require extensive dataentry checking or use the same
ADF to do all the query's . I typically use this mode for all our read only
access into the database or for simple RMS applications.

```
              | ADF |<-- USER
              -------
                 |
                 v
        | Form buffer |<----> DAR <--------> OPAL <-----> Database
        ---------------
```
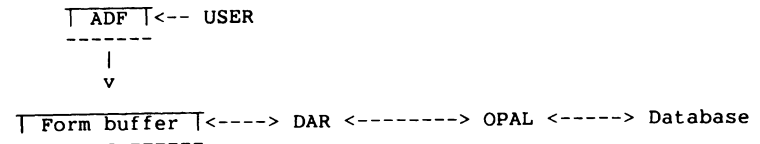

2. Partial control by SMARTSTAR and partial control by your host  program. The
ADF  can trigger outside programs to run  (via KAR,UAR) . We sometimes use this
to validate a field or  to move the contents of one field to another field.

```
              | ADF |<-- USER
              -------
                 |
                 v
        | Form buffer |<----> DAR <--------> OPAL <-----> Database
        ---------------
                 ^
                 |--------> user program
```
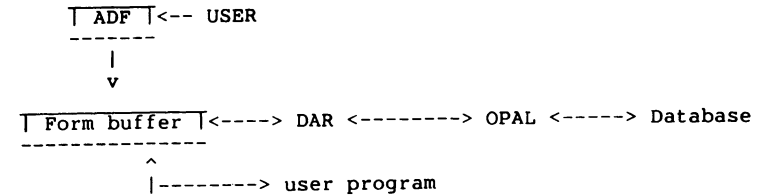
3. Total control of the application by your program. Here you are controlling
the sequence of events, that is, which ADF's are called, in what order, when to
call SQUERY. Generally, you are controlling the frontend processing, that is
you may be bouncing between several ADF's , menu screens etc. but are still
leaving the backend work to the DAR. An example

1. Call ADF-1 which asks the user his name and password
2. Your program verifies the response
3. Call ADF-2 which might be a menu screen listing several
options for the user.
4. You may call one of several ADF's depending on the user
response
5. Call SQUERY (from within your program). At this point
you have given control  SQUERY and can do nothing
from your program until the user exits SQUERY.

```
                                            USER
                                             |
                                             v
  | Form buffer |<----> SMARTCALL <--------> application program
  ---------------
          ^
          |
          v
  SMARTCALL <----> DAR (SC_OBACT) <---> OPAL  <---> DATABASE
```
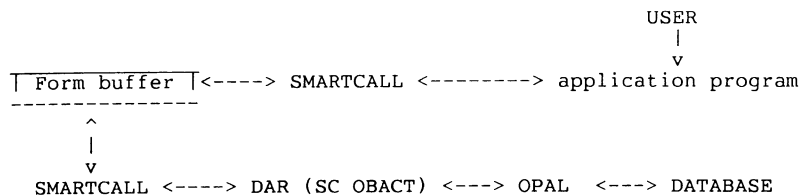
4. The 4th mode of programming control  REAL total control over the
application. This is similar to type 3 above except your program is controlling
the backend database work through calls to REQUEST via OPAL. This may sound
like alot of additional work for the programmer and defeat the purpose of your
4GL product but it  offers advantages , such as : speed; more control over the
backend processing ; trapping error codes; multiple transaction rollback.

```
                                            USER
                                             |
                                             v
  | Form buffer |<----> smartcall <--------> application program
  ---------------                                  ^
                       ----------------------|
                       |           |
                       v           v
  SMARTCALL <----> DAR (SC_OBACT) <---> OPAL  <---> DATABASE
```
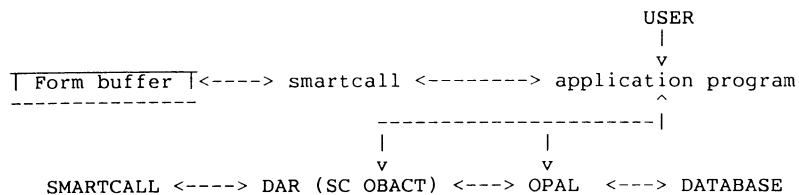
Here you can call the DAR from your application program and still let the
DAR do the work or you can use OPAL routines and directly control the backend
processing. In some cases,our  programs need to obtain information from other
relations before an actual store takes place. For example, we have an inventory
of chemicals , well if the user plans to dispense 50 gm of a chemical, I need
to check if we have 50gm of the chemical in the inventory, So I need to do
some intermediate processing before the actual append takes place.  I need to
do an initial retrieve first, inorder to perform this check. So my program is
directly calling OPAL to do the retrieve.

In the initial version of our database application, I let the DAR handle all the
database work, but I found that I was using many different ADF's to store the
data.

It actually is INEFFICIENT to  use several different ADF's to perform query
operations to the database due to what the DAR has to do. When in SQUERY  and
you issue a query command ( such as a retrieve or append) the first thing the
DAR does is look at the ADF that initiated the command, it then compares it to
the ADF that initiated the last query to the DAR. If it is the same ADF then
the query command is processed . If it is a different ADF then a call to OB_OPEN
is made. OB_OPEN allocates the internal buffer space that the DAR needs to
process the query. (ex. for a retrieve the DAR needs a place to temp. store the
information before sending it to the form buffer) These buffers remain in
effect until the next call to OB_OPEN and since OB_OPEN is called whenever a
different ADF calls the DAR, and if you use several ADF's to query the database
you have to wait for this allocation of buffer space each time. What the
OB_OPEN is doing when it creates the internal buffer is that it reads the ADF
and obtains all the fields, lengths, datatypes, and field names and allocates
space for each field. It is logical that a call to OB_OPEN is needed whenever

the DAR is called from a  different ADF since each ADF may have different
fields. What I do not understand is why OB_OPEN has to create a new internal
buffer each time the same ADF is referenced

    Here is a example

    1. The current ADF is ADF-1 and I issue a retrieve - A call to OB_OPEN is
       made
    2. I then issue additional query's - Since the ADF did not change between
       calls to the DAR an OB_OPEN call is not performed
    3. I change  to ADF-2 and issue another query - Since the ADF
       has changed another call to OB_OPEN is made for ADF-2
    4. I then return to ADF-1  and issue another query - Since the last
       query was through ADF-2 yet another call to OB_OPEN is made
      etc. etc.

I wonder why the buffer for ADF-1 has to be recreated, I would think the DAR
could access the buffer that was previously created for ADF-1

These opens are a bit time consuming. it is acceptable if you plan to use the
same ADF for many queries (like a data entry screen) By performing the query
independent of the ADF there is no need to allocate this internal buffer space
and thus eliminate this open problem. (NOTE you do have to open the database
initially in your program) There is a disadvantage and that is that your
program must now have to do some of the work that the DAR was taking care of.
Your program must be able to handle the flow of information from the ADF to the
backend and from the backend to the ADF. I typically used this approach for 2
situations. First, for appending data to the the backend. In this case, I know
that the flow of data is only going to go one way and my program does not have
to take into account all the other query options. When appending data my
program does not have to explicitly allocate buffer space for all the fields in
the ADF. There are SMARTCALL routines that I can use to take the data from the
ADF and pipe it directly into the OPAL query. The other instance for using this
technique is when I just want to know if a record exits ( do we have a certain
chemical) , Since I don't need to  access the information in the record my
program does not need to allocate space inorder to store the data that is being
retrieved. I issue the OPAL retrieve and trap the status code, since my program
does not allocate space for the information coming from the backend the
information just goes to the bit bucket.


Transaction management
----------------------
Rollback of single transactions is supported when using the DAR to perform the
backend work. Rollback of multiple transactions CAN NOT BE DONE THROUGH THE DAR
!!!!!. (The DAR supplied) A commit is performed after each successful operation
to the database. Rollback of multiple transaction is support through REQUEST
via OPAL. This is the main reason why much of our backend processing is done
outside the means of the conventional DAR.

When you let the ADF-DAR do the query to the database and the query fails for
some reason (a constraint fails) a rollback is issued, in effect, a rollback of
a start_trans with one transaction. If you have an ADF that accesses say 3 RDB
tables

Note a side bar - The DAR SC_OBACT only accesses the first table in ADF - you
must alter the SC_OBACT to perform operations on all the tables in the ADF
and all 3 tables should be updated or none are to be updated. If you alter the
DAR to perform updates on multiple tables it performs a commit after each

successful operation. So if update on table 1 is successful a commit is
performed. If update to table 2 is unsuccessful a rollback of update 2 is
performed but update 1 has already been committed. (Of course you can make a
status check on update 2 that if it fails not to perform update 3). By using
calls to request via opal we can issue a SMARTSTAR version of an RDB
START_TRANS.

Authors Note:
-------------
I have tried to pass on some things that I have come across and hope that they
will be  helpful to others. My intent was not to present an overview of the
product but present some deeper idea's as to what is going on. I am sure there
are other ways to do some of the same type of things that I have shown and I
would appreciate any other idea's anybody else has. In the next submission I
will provide some general observations and simple examples and talk about
BINDING (what it is , how to get around it and how to make it as generic as
possible in your program. I also did not deal with performance, since I have
not quite gotten that far in my application yet, but hopefully by next
issue I will be prepared to discuss that issue.

Feel free to give me a call if you have any questions about anything in this
article or want to talk in general - I am always happy to take a break from
banging on  my terminals' keys.

                                                        TOM


--------------------------------------------------------------------------------
A Hoot from the Editor
--------------------------------------------------------------------------------
I would like to express my sincere gratitude to Tom for his contribution to
this Newsletter, and to alert DMS SIG Members that Tom will be making a
presentation on this topic at the Spring DECUS U.S. SYMPOSIUM in Dallas,
the week of April 28-May 2, 1986. Thanks again, Tom.
--------------------------------------------------------------------------------

DTR

# DTR/4GL Special Interest Group - Officers

| | | | | | |
|---|---|---|---|---|---|
| SIG Chairman | Joe H. Gallagher | Research Medical Center | 2316 East Meyer Blvd. | Kansas City, MO 64132 | 816-276-4235 |
| Past SIG Chairmen | Larry Jasmann | U. S. Coast Guard | 10067 Marshall Pond Rd. | Burke, VA 22015 | 202-426-2344 |
| **Communications** | | | | | |
| Committee Rep. | Elaine McWilliams | Boeing Company | P. O. Box 24346 M/S 9c55 | Seattle, WA 98124 | 206-773-0102 |
| Editor | Donald Stern | Warner Lambert Company | 10 Webster Road | Milford, CT 06460 | 203-878-9351 |
| Production Editor | Steve Cordiviola | Kentucky Geological Survey | 311 Breckinridge Hall | Lexington, KY 40506 | 606-257-5863 |
| Artist | Bart Z. Lederman | | 2572 E. 22nd St. | Brooklyn, NY 11235 | 718-743-9593 |
| **Symposia** | | | | | |
| Symposia Committee Rep. | Chris Wool | E.I. duPont Eng. Dept. | Louviers Bldg. | Wilmington, DE 19898 | 302-366-4610 |
| Symposia Assitant | Diane Pinney | Computer Sciences Corp. | 443 Inyokern Road | Ridgecrest, CA 93555 | 619-446-6585x284 |
| Pre-Symposia Seminars | Dana Schwartz | | 15719 Millbrook Lane | Laurel, MD 20707 | 301-859-6277 |
| Symposium Notes | Elaine McWilliams | Boeing Company | P.O. Box 24346 M/S 9c55 | Seattle, WA 98124 | 206-773-0102 |
| **Product Development** | | | | | |
| Development Coordinator | Bart Z. Lederman | | 2572 E. 22nd St. | Brooklyn, NY 11235 | 718-743-9593 |
| New Product Focal Point | Diane Pinney | Computer Science Corp. | 443 Inyokern Road | Ridgecrest, CA 93555 | 619-446-6585x284 |
| DEC Counterparts | | | | | |
| Datatrieve | Andy Schneider | Digital Equipment Corp. | 110 SpitBrook ZK02-2/N59 | Naushua, NH 03062 | |
| TEAMDATA, RALLY | Basil Harris | Digital Equipment Corp. | 110 SpitBrook ZK02-2/N59 | Naushua, NH 03062 | |
| Library Committee Rep. | Bart Z. Lederman | | 2572 E. 22nd St. | Brooklyn, NY 11235 | 718-743-9593 |
| Volunteer Coordinator | Larry Jasmann | U. S. Coast Guard | 10067 Marshall Pond Rd. | Burke, VA 22015 | 202-426-2344 |
| Campground | Bert Roseberry | U. S. Coast Guard | 500 Camp Street | New Orleans, LA 70130 | 504-589-4934 |
| Special Effects | Alex L. Lamb | U. S. Army | 32 Pearce Ave | Eatontown, NJ 07724 | 201-532-3318x1843 |
| DMS SIG Liason | Phil Naecker | J M Montgomery Engineers | 250 N. Madison Ave | Pasadena, CA 91101 | 818-796-9141 |

## DTR/4GL Masters List

| | | | | | |
|---|---|---|---|---|---|
| RSX,P/OS,VMS | Bart Z. Lederman | | 2572 E. 22nd Street | Brooklyn, NY 11235 | 718-743-9593 |
| PRO,VMS | Joe H. Gallagher | 4GL Solutions | 10308 Metcalf Suite 109 | Overland Park, KS 66212 | 913-894-9550 |
| VMS | Joe Kelly | WymAn Gordon Co. | Worchester Road | North Grafton, MA 01536 | 617-839-4411x5480 |
| VMS | Larry Jasmann | U. S. Coast Guard | 10067 Marshall Pond Rd. | Burke, VA 20015 | 202-426-2344 |
| VMS | James Swanger | G. D. Searle & Co. | 4901 Searle Parkway | Skokie, IL 60077 | 312-982-7430 |
| VMS | Chris Wool | E. I. duPont, Engineering Dept. | Louviers Bldg. | Wilmington, DE 19898 | 302-366-4610 |
| VMS | Dick Azzi | Motorola SPS | Box 2953 MS: P116 | Phoenix, AZ 85062 | 602-244-4316 |
| VMS | Chris Heinz | American Board of Family Practice | 2228 Younge Drive | Lexington, KY 40505 | 606-269-5626 |
| VMS | Bert Roseberry | U. S. Coast Guard | 500 Camp Street | New Orleans, LA 70130 | 504-589-4934 |

## Contributions

Contributions for the newsletter can be sent to either of the following addresses:

Editor, DATATRIEVE Newsletter
DECUS U.S. Chapter
219 Boston Post Road, BP02
Marlboro, MA 01752

Donald E. Stern, Jr.
Warner Lambert Company
10 Webster Road
Milford, CT 06460

Letters and articles for publication are requested from members of the SIG. They may include helpful hints, inquiries to other users, reports on SIG business, summaries of SPRs submitted to Digital or other information for members of the DATATRIEVE SIG. Machine readable input is highly desirable and machine-to-machine transfer of material is preferred, but most anything legible will be considered. However, this newsletter is not a forum for job and/or head hunting, nor is commercialism appropriate.

## Table of Contents

## About the Cover

This month, the Wombat begins to look at some new friends. DEC has formally announced 2 new products, VAX TEAMDATA and VAX RALLY, which will be represented in DECUS, by the DTR/4GL SIG.

It will be interesting to see what mascots, if any, the Wombat chooses as companions....

## Chairman's Corner

The time has come to pass leadership of the Datatrieve/4GL SIG to Joe Gallagher. When I sat down to write this, my last Chairman's Corner as Chair of the SIG, it took a moment of concentrated thought to go back to the spring of 1982 when I took over the SIG from Chuck Watson. I was mildly shocked to discover that four years had passed. It just doesn't seem possible.

In 1982, VAX-Datatrieve was beginning to really gather momentum, and the SIG was in transition from a small, casual organization for a fairly minor layered product, to a SIG dealing with a product which was (and still is) seriously used at many, many DEC sites. At the present time, Datatrieve is second only to FORTRAN in sales of layered software products.

In 1986, the SIG is again in transition, this time to a more general focus on fourth generation data management products. It is proper and fitting that someone else get a chance to make this work and enable the SIG to make a quantum leap in the process of providing information to DECUS members. Joe Gallagher can and will do this.

I intend to remain active in the SIG and in DECUS. You will see me at symposia and my name will stay in the Wombat Examiner. Hopefully, I will have more time to contribute in a technical way to the SIG. I also intend to stay active in DECUS leadership.

Joe, I wish you the best of luck and all of my support. I hope that you get as much inspiration and personal reward from this job as I have had. I also look forward to working with you in the SIG.

Larry Jasmann
Chairman, Datatrieve/4GL SIG

## From the Editor's Pen

Donald E. Stern, Jr., Warner Lambert Company, Milford, CT

As you read this, the Dallas Symposium has probably been concluded and we're well into the planning stages for the San Francisco Symposium in October. Since this newsletter went to publication back in March, we obviously can't bring you news of the Symposium.

We, in the DTR/4GL Steering Committee, need and welcome your feedback. Consistent with this need, a "Letters to the Editor" section will be added to the newsletter so your feedback can be shared with other DECUS members. Let's get a dialog going.

At DECWorld, in Boston last February, Digital announced its two new software products, VAX TEAMDATA and VAX RALLY. It is hoped that, as these products are put to use in the real world, we can bring you high quality user written articles as has been the tradition with this newsletter. In addition, articles by people using other 4GL products are welcomed and will be considered for publication as well. Naturally, we continue to seek submissions dealing "Good Ole Datatrieve."

New with this issue of the Wombat Examiner and 4GL Dispatch is the Wombat Wizard column. The Wizard solicits queries great and small and will help you solve your DTR/4GL problems. The Wizard can be reached c/o the Newsletter Editor. In addition, we're considering a regular feature similar in nature to the VAX SIG's SIR (System Improvement Request). This would provide a forum for the readership to request DTR/4GL product improvements. Your responses will be collected, statistically summarized, and forwarded to the appropriate DEC Counterpart. What do you think?

---

*** Ask the Wombat Wizard ***

---

Each month the Wombat Wizard will answer user's questions regarding the use of Datatrieve. To make this a regular feature of the Wombat Examiner, the Wizard needs your help! Send him your sick, your slow, and your unfriendly Datatrieve problems, and the Wizard will provide wisdom and guidance in this column every month. No question is too silly, no problem too hard, no subject too sensitive! In the tradition of Dear Abby and the Playboy Advisor, no reasonable question will go unanswered. Each month, the most interesting and instructive questions and problems will be published in this column.

To send in questions or problems to the Wombat Wizard, use the form at the back of the Newsletter or just send a letter to me c/o the Newsletter Editor. (Please note that the Editor only collects the material. He is not the Wizard.) Be sure to include your phone number and DECUS membership number. Also, please follow these simple guidelines:

1. If you are trying to demonstrate a method or a concept, please simplify the procedures, records, and other information to the shortest form possible. Avoid long procedures where only a small portion of the procedure is required to demonstrate the concept.

2. Annotate your attachments. Simple comments or handwritten notes ("Everything worked until I added this statement.") go a long way toward identifying the problem.

3. Keep an exact copy of what you send. And number the pages on both copies. But send everything that is related to your question, even remotely.

4. Wombat Wizard is not the Telephone Support Center, nor is it part of DEC's Software Performance Reporting (SPR) system. Our goal is to answer "how to" or "how come" questions in an informative and instructive fashion - not to be a clearinghouse for software performance problems.

5. If you would like a direct response or would like your materials returned, please don't forget to include a stamped, self-addressed envelope large enough to hold the materials you send.

---

## INTRODUCING VAX TEAMDATA AND VAX RALLY
### NEW 4TH GENERATION SOLUTIONS FROM DIGITAL

Basil Harris, Jr. Product Manager, VAX TEAMDATA and VAX RALLY
Digital Equipment Corp. 110 Spit Brook Rd. Nashua, N.H. 03062

---

## 1.0 EXECUTIVE SUMMARY

Both in the office and in the data processing department, decision makers are looking for ways to put the right information into the hands of the right people at the right times. Application backlogs are currently clogging corporate MIS shops, while personal computer users are finding that their micro packages are not sufficient to solve many of the information problems they encounter. Managers are asking questions like:

  o   How do I get the best information to my users?

  o   How do I provide ways for my users to share and consolidate information, whether it's in a spreadsheet, on a graph, part of a memo or report, or in a database?

  o   Most of all, how do I maintain control over the entire information management process?

Users need the integration and ease-of-use features of a personal productivity package combined with the power and flexibility associated with mini- and mainframe-based 4th Generation products, all as part of a single solution. VAX TEAMDATA and VAX RALLY provide that single solution.

TEAMDATA is a sophisticated, yet easy-to-use, information management and decision support system targeted at the inexperienced computer user. Consisting of integrated database management, spreadsheet, graphics, query and report writing capabilities, TEAMDATA puts corporate, departmental and personal data at the user's fingertips. Much of TEAMDATA's power and flexibility comes from giving VAX users easy access to existing data and applications, including local and remote spreadsheets and data-

bases, and integrating with DIGITAL's comprehensive office automation package, ALL-IN-1.

RALLY, a companion product to TEAMDATA, is a powerful application generation system aimed at the more experienced computer user. RALLY is a complete application generation environment, providing integrated tools for the creation and maintenance of relational databases, forms, reports, menus and HELP text. Designed to handle simple to moderately complex commercial applications, like personnel tracking and financial planning systems, RALLY also includes a subset of functionality aimed at getting the less experienced application designer started. Once created, RALLY applications can be run standalone, from a TEAMDATA directory, or directly from an ALL-IN-1 menu.

Together, TEAMDATA and RALLY offer the VAX user a total information management and decision support package, providing:

o   Easy access to information from many sources;

o   Easy ways to consolidate and share information;

o   Easy tools for creating and maintaining complex applications;

o   Easy integration with existing data and applications.

## 2.0  VAX TEAMDATA

### 2.1  Overview

TEAMDATA provides powerful, yet easy to use, information management capabilities to those who need to use data in their work, but who don't want to do "data processing." TEAMDATA lets users easily store and modify data, using a screen-oriented, text-editing style, in a powerful relational database management system, VAX Rdb/VMS.

TEAMDATA helps users share data, stored locally and remotely, in a variety of storage formats. TEAMDATA users can access data stored in Rdb/VMS databases by TEAMDATA or by other Rdb/VMS applications. Through optional use of VAX DATATRIEVE, TEAMDATA allows read-only access to data stored in VAX DBMS databases or RMS files which have been defined as DATATRIEVE domains.

TEAMDATA allows users to manipulate their personal data, as well as shared or remote databases, in simple tables, spreadsheets, reports, and graphs, and to perform complex queries and data reduction operations.

TEAMDATA users can perform their tasks by selecting items from strip menus, or using the command language, or a combination of the two.

TEAMDATA can be used with ALL-IN-1 as an information management option on an ALL-IN-1 menu, or can be installed outside of ALL-IN-1. In addition, TEAMDATA provides a run-time environment for applications generated using VAX RALLY.

## 2.2  FEATURES

TEAMDATA offers a package of tightly integrated functions that provide:

o   Tables (Database)

TEAMDATA users can create and maintain data in tables for private or shared use. Using a text processing style that is consistent with WPS-PLUS function keys, users can "edit" screens of data without having to learn a complex data manipulation language. Data can be stored, sorted, updated, queried, and reported using menu selections, or commands, or a combination of the two.

Table features include:

-   Ability to display data in tabular form on the screen;

-   Ability to create tables and determine display format for new data;

-   Editing of data, including easy data entry, update, and deletion in word processing style;

-   Query, extract, and report by menu-driven record selection;

-   Transparent fetching of data from local and remote TEAMDATA and Rdb/VMS databases, and, through VAX DATATRIEVE, RMS files and DBMS databases;

-   Simple report generation using data from single or multiple data sources; ability to subset and sort data, establish control breaks, and save/print the results;

-   Joins between tables, with the ability to store the results in a new table;

-   Sorting of data by one or more fields;

-   Aggregate calculations, including TOTAL, COUNT, AVERAGE, MAX, AND MIN on a whole table or by groups (control breaks);

-   Computed fields, where one field is calculated in terms of one or more fields in the same row.

o   Spreadsheets

TEAMDATA uses compatible menus and screen-editing styles for the creation and modifications of both spreadsheets and tables.

Spreadsheet features include:

- Screen editing of spreadsheet data and formulas;

- Size of spreadsheet: up to 676 columns. Number of rows limited only by available disk space;

- Import and export of data in common spreadsheet formats like DIF, SYLK, WKS and DECalc (available in V1.X);

- Temporary or permanent creation of new cells/rows/ columns after initial spreadsheet creation;

- Automatic recalculation of cells by default;

- Column widths individually adjustable;

- Insert, replicate (copy), move, and delete rows and columns;

- Financial functions such as PRESENT VALUE, FUTURE VALUE, NET PRESENT VALUE, DEPRECIATED VALUE;

- Mathematical functions such as LOG, SQRT, ABS, AND FACTORIAL;

- Statistical functions such as SUM, AVERAGE, MINIMUM, MAXIMUM, SUM OF SQUARES, VARIANCE, STANDARD DEVIA- TION, AND COUNT;

- Ability to print spreadsheet formulas instead of data values;

- Conditional expressions (IF/THEN), with nesting and Boolean comparisons;

- ERROR capability: IF cell x "error" condition THEN substitute cell Y (or value Y) ELSE cell (or value) Z.

o Business graphics

Users can create BAR, MULTI-BAR, STACKED BAR, LINE, MULTI- LINE, PIE, OR SCATTER graphs of data from within tables or spreadsheets. Graphs can be displayed, printed, or saved.

o Folders

Users can organize tables, spreadsheets, and applications in folders. Folders can be nested hierarchically.

Folders contain a directory of the user's own private in- formation or can include public information for team or group use.

o Integration with ALL-IN-1

TEAMDATA V1.0 provides integration with ALL-IN-1 in the following areas:

- Function keys consistent with ALL-IN-1 and WPS-PLUS;

- Support of ALL-IN-1 Interrupt facilities;

- Ability to paste TEAMDATA data as text, into ALL-IN- 1 documents or mail (via the ALL-IN-1 scratch pad);

- Support for all printers that ALL-IN-1 supports (graphics available on SIXEL devices only).

TEAMDATA can be run from an ALL-IN-1 menu, but need not be.

o Run-time environment for VAX RALLY applications

TEAMDATA provides a run-time environment for RALLY-gener- ated applications (RGA). Application developers can create custom applications that include forms, reports, menus, application-specific user assistance and complex logic. These applications can then be included on a TEAMDATA di- rectory and run by the end-user. Refer to the VAX RALLY V1.0 Software Product Description for more information about this product.

o Ease of use features

TEAMDATA has many features designed to enhance ease of use:

- Comprehensive, context-sensitive online HELP;

- Integrated online tutorials that let users learn key concepts and functions while actually working with the TEAMDATA software;

- Ability to save sequences of keystrokes in named procedures for automating repetitive tasks;

- Ability to modify both system and user defaults;

- Ability to toggle between multiple active tasks.

o Access to existing data and applications  .

TEAMDATA allows users to access data in a variety of loca- tions: In tables and spreadsheets that other TEAMDATA users created; In Rdb/VMS databases; and in VAX DBMS databases and RMS files (read only, through the optional use of VAX DATATRIEVE). Users can also issue DCL commands and run DCL command procedures from within TEAMDATA.

o Other system-wide features

   - Cutting and pasting all or part of a table to a
     spreadsheet, a spreadsheet to a table, a table to a
     table, and a spreadsheet to a spreadsheet

   - Date arithmetic;

   - User-defined formatting of data;

   - Can be invoked from any menu system capable of run-
     ning a standard VMS image and relinquishing control
     of the screen to that image.

## 2.3 Benefits

o Easy yet very powerful data management and decision sup-
  port.

  TEAMDATA lets non-computer professionals build and maintain
  personal and shared databases (tables). TEAMDATA's screen-
  oriented, text editing style of data entry and modification
  makes these tasks easy for someone unfamiliar with data
  definition and data modification languages.

o Access to applications and distributed data

  TEAMDATA can access and combine local and remote data
  stored in Rdb/VMS databases and VAX DATATRIEVE domains.
  This, combined with TEAMDATA's facilities for easily query-
  ing data, makes TEAMDATA an ideal "front end" to distribu-
  ted data.

  TEAMDATA's environment can be tailored for individual work
  situations, with the addition of existing or new applica-
  tions. Users can select from TEAMDATA's "main menu" DCL
  procedures, applications generated by RALLY, or any other
  application such as a COBOL program or DATATRIEVE proce-
  dures. Additionally, access to the DCL dollar-sign prompt
  can be denied to TEAMDATA users, so users can be limited to
  the TEAMDATA environment with or without ALL-IN-1.

o Shared and private information

  TEAMDATA makes it easy to share data, whether it is stored
  locally or remotely, in TEAMDATA tables or spreadsheets, or
  in DBMS or RMS. Public "folders" are used to contain in-
  formation for sharing with others. Data may also be "made
  known" to individual users without inclusion in the public
  folder.

  TEAMDATA's strong data access capabilities and the ease of
  sharing data make it an attractive alternative (or supple-
  ment) to a pc-based system.

o Support of multiple spreadsheet formats (available in V1.X)

  By allowing users to import and export data in different
  spreadsheet formats, TEAMDATA provides an extremely power-
  ful and flexible tool for doing departmental roll-ups and
  data consolidation. Users can share and analyze data from
  many different sources, both DEC and non-DEC.

o Integrated functions and integrated software

  TEAMDATA also lets non-computer professionals design so-
  phisticated electronic spreadsheets, as well as generate
  simple reports and business graphics, all with one style of
  user interface.

o Integration with office automation

  Although TEAMDATA can run without ALL-IN-1, users may wish
  to run TEAMDATA from ALL-IN-1 menus. TEAMDATA adds data
  management and decision support to ALL-IN-1's traditional
  OA tools (word processing, calendar management-meetings,
  appointments, to-do lists and reminders, mail, calculator,
  phone directories and file and folder management).

  TEAMDATA's key usage is consistent with WPS-PLUS and with
  ALL-IN-1 function keys, so office users familiar with those
  products should quickly feel at home in the TEAMDATA en-
  vironment.

  TEAMDATA and ALL-IN-1 provide a complete solution for of-
  fice automation and office information systems.

o Part of VAX Information Architecture

  TEAMDATA extends the VAX Information Architecture to end
  users, providing a "friendly front end" to data stored in
  Rdb/VMS, DBMS, and RMS.

  TEAMDATA participates in the VAX Information Architecture
  by using the DIGITAL Standard Relational Interface (DSRI)
  to data and the software bus. As components/products are
  added to the VAX Information Architecture family, with
  interfaces to the software bus and DSRI, TEAMDATA will be
  able to take advantage of their functionality.

o Easy to use and to learn

  TEAMDATA provides two methods of issuing commands to the
  system: a series of strip menus, and a command language.
  New users will likely use the strip menus. More experienced
  users will appreciate the language (which is the strip menu
  choices chained together). Both new and experienced users
  will appreciate the ease of moving from one command method
  to the other: no special key or other indication of "mode"
  is necessary.

  The online, context-sensitive HELP and tutorials further
  contribute to the product's ease of use.

## 2.4 Examples Of Use

Because TEAMDATA is so easy to use, we expect people who haven't used computers to use TEAMDATA for a variety of applications. For instance:

o A secretary may use TEAMDATA to keep records of expenditures. She can easily set up her database, without knowing anything about databases. She can produce graphs and reports of the expenditures by category over time.

o A manager might use the expenditure records in a spreadsheet as she plans the next year's budget. She may include in her spreadsheet, revenue predictions that are stored in an external database (such as Rdb/VMS, DATATRIEVE, or RMS).

o Both of these users may use TEAMDATA from an ALL-IN-1 menu. The secretary finds the WPS-PLUS-style editing of TEAMDATA makes information management a simple and familiar task. And both users like the ability to paste TEAMDATA data into ALL-IN-1 documents for printing or mailing.

TEAMDATA will also be attractive to people who are more experienced computer users. These people are likely to use TEAMDATA in these ways:

o An information center consultant sets up an information database using Rdb/VMS. She allows several people to have read access to the data. She tells these people how to access the database and they then treat the relations in the database as TEAMDATA tables, using the query and report capabilities of TEAMDATA. Some people also copy data into spreadsheets for "what if" analysis.

o The information center consultant may also give a few people write access to some relations in the Rdb/VMS database. These people see the Rdb relations as TEAMDATA tables and update the data using the simple text-editing style of TEAMDATA, rather than using a complex data manipulation language.

o Read access can also be provided to data stored in RMS, VAX DBMS, and even in RMS on PDP-11's, through the use of VAX DATATRIEVE domains. The TEAMDATA user supplies the name of a DATATRIEVE domain and its Common Data Dictionary (CDD) path name only once. After that, the TEAMDATA directory shows that domain as a table that the user opens as any other TEAMDATA table.

o An information center consultant may use RALLY to produce an application, with TEAMDATA-style menus, that has complex reports and data entry forms. The application is "installed" in the TEAMDATA environment and the TEAMDATA user runs the application from within TEAMDATA.

## 3.0 VAX RALLY

### 3.1 Overview

VAX RALLY provides a powerful 4th Generation environment for developing applications. As an integrated package, RALLY provides the tools necessary to construct databases, screen forms, reports, menus, and online HELP for the application user.

Using a menu interface, the RALLY application developer can create an application from scratch or can take advantage of the extensive defaulting capabilities of RALLY. Once applications are created they can be:

o Run directly from within RALLY;

o Installed in a VAX TEAMDATA directory;

o Accessed directly from an ALL-IN-1 menu;

o Run from DCL.

RALLY can be used to do such things:

o Set up and maintain a complete departmental budgeting system;

o Create a low- to mid-range order entry system;

o Access corporate sales information for departmental reporting and analysis.

### 3.2 Features

VAX RALLY offers the application developer the following features and capabilities:

o Application generation tools

Built on a relational database management system (VAX Rdb/VMS), RALLY provides a 4th Generation environment for constructing applications. RALLY tools include database creation facilities, reports, menus and a built-in procedural language (ADL). Generated applications can run in one or more user-controlled windows on the terminal screen.

RALLY applications can be run at VMS command level, from a VAX TEAMDATA directory, or, with some intervention by a more experienced ALL-IN-1 user, directly from an ALL-IN-1 menu. In addition, application data can come from multiple data sources, including local and remote Rdb/VMS databases, TEAMDATA tables, and RMS files and DBMS databases that have been defined as VAX DATATRIEVE domains.

o   "Simple applications simply" capabilities

RALLY provides tools for less experienced users to get
started designing and building their applications.
Business systems analysts can take advantage of the
extensive defaulting capabilities of the product to
design the databases, forms, reports, and menus that make
up, for example, a simple data entry application. As the
user gains more experience, such simple applications can be
expanded and modified.

o   Database creation/maintenance features

RALLY provides a menu-driven database facility which allows
for the creation and maintenance of Rdb/VMS databases in a
style consistent with the other RALLY application genera-
tion facilities. This facility allows the definer to:

   -   define databases, relations, fields and indexes;

   -   modify definitions of relations and fields;

   -   delete definitions of relations, fields and indexes;

   -   add comments to relation, field and index
       definitions.

o   Unified forms and reports

The centerpiece of VAX RALLY is its integrated
form/report processor. One RALLY subsystem supports
both forms and reports. If the application developer so
allows, anything that can be done with a form can also be
done with a report, and vice versa. Depending on how the
person defining the application specifies access to a form/
report (i.e.: input only, output only, query, or browse/up-
date/delete), users interact with an object as a form, a
report, a template for queries, or an updateable report.

RALLY will automatically generate a form/report using
the information from a database description. The
definer can then use a full-screen editor to customize
the form/report. RALLY form/reports can include
scrolled regions, field-level validation mechanisms,
video highlighting, complex completion handlers, and can
span multiple screens.

Besides allowing users to construct complex reports with
control breaks, totaling and subtotaling, multiple data
source access, and flexible formatting, the RALLY form/re-
port processor lets the user actually manipulate the under-
lying data source(s) dynamically through the screen form/
report. This means that if the application definer so
allows, a user can edit a form/report, "browse" through it
(horizontally or vertically), change values, add/delete
records and then either save the changes or quit from the
report. Changes made to computed fields are reflected
instantly on the screen.

o   Menu creation

Application logic is typically specified through menus.
RALLY provides a menu generation facility that lets appli-
cation definers construct user menus in a variety of sty-
les. VAX TEAMDATA-style strip menus are the default style
for simple, defaulted applications. However, other menu
formats can also be built.

o   Built-In Application Definition Language (ADL)

RALLY includes an application definition language which is
designed to be used for such things as special field vali-
dation operations, special-purpose arithmetic computations,
batch updates, and unusual database management interac-
tions. Similar to PASCAL, ADL is intended to supplement an
application composed primarily of RALLY entities (for exam-
ple, forms, reports, menus) with computations and flow-of-
control.

o   Access to existing applications

While most RALLY application requirements will be satisfied
using the tools provided with the product, application
developers with special requirements can construct applica-
tions that call external routines (for example, a COBOL
program).

o   Optional integration with ALL-IN-1

RALLY provides integration with ALL-IN-1 in the following
areas:

   -   Function keys are (optionally) similar to ALL-IN-1
       and WPS+;

   -   If a RALLY Generated Application is run under TEAM-
       DATA, support for ALL-IN-1 Interrupt facilities is
       provided by TEAMDATA.

RALLY Generated Applications can be run from either an ALL-
IN-1 menu or a TEAMDATA directory. They can also be run
directly from DCL provided the user has purchased either
VAX RALLY or VAX TEAMDATA for the system where the applica-
tion is run.

o   Comprehensive on-line assistance

At every stage in the development process the definer can
get context-sensitive assistance on particular areas he or
she is working on. RALLY continually makes the definer
aware of appropriate values for fields, options, and vari-
ables.

Also, RALLY allows definers to include complete HELP infor-
mation in a generated application.

o   Other features

    -   date arithmetic;

    -   password security for entire applications or se-
        lected application parts (for example, forms or
        reports);

    -   single-user "protected write" or multi-user "shared
        write" access to Rdb/VMS databases;

    -   ability to save keystroke sequence in named macro
        files;

    -   toggle task capabilities;

    -   function keys similar to ALL-IN-1/WPS+, easily con-
        verted to EDT, or customizable to user's preference.

## 3.3  Benefits

RALLY provides many benefits to the application designer,
including:

o   Full application generation capabilities in a single, inte-
    grated package;

o   4th Generation technology running across VAX systems, from
    MicroVAX II up through 8800 clusters;

o   Easy-to-use tools for creating simple applications, as well
    as powerful procedural and non-procedural constructs for
    designing complex applications;

o   Integration with existing VAX Information Architecture and
    Office Automation products, including ALL-IN-1, VAX DATA-
    TRIEVE, VAX TEAMDATA, VAX languages, and VAX Rdb/VMS;

o   Access to existing applications and data across a network
    of VAXes.

## 3.4  Examples Of Use

RALLY is both powerful and flexible, allowing for the creation of
many different kinds of applications for many different purposes.
For example, a single, menu-based RALLY application could be used
in a sales office for the following tasks:

o   A sales manager might periodically print a report on all
    the outstanding orders for the customers for each salesper-
    son, with comparisons to the the projected sales for each
    product. She might then browse through the report on the
    screen (a unique and very powerful RALLY feature) and per-
    form ad hoc queries using the "query by example" tool;

o   A salesperson could read and print reports on monthly
    sales, comparing projected with actual results. He could
    also browse through the report and occasionally update
    customer information, order status, and so on;

o   An order entry person could enter new orders as they came
    in. Complex data validation and security mechanisms could
    be built into the application for such things as checking
    part number validity, ensuring that a customer record exis-
    ted in the data base, and denying application access to all
    but certain users;

o   Someone with the appropriate privileges could, on a peri-
    odic basis, perform maintenance activities such as entering
    actual expenses, and updating customer and product informa-
    tion.

## 4.0  QUESTIONS AND ANSWERS

VAX TEAMDATA/VAX RALLY

1.  What are the software prerequisites for TEAMDATA and RALLY?

    Users need VMS V4.3 or later and VAX Rdb/VMS V2.0 installed
    on their systems before installing TEAMDATA or RALLY. With
    regard to Rdb, only the runtime kit is required for running
    either product.

2.  Are TEAMDATA and RALLY separate products?

    They are companion products, designed to work together.
    Each can, however, be used by itself.

3.  What do you mean when you say TEAMDATA and RALLY are "com-
    panion products"?

    The combination of RALLY and TEAMDATA provide an extremely
    competitive and complete 4GL offering on VAX, spanning
    information management and application development from the
    end user through the sophisticated DP professional.

    They can be thought of as companion products for the fol-
    lowing reasons:

    .   Both are based on Rdb/VMS, and because of this...

    .   ...TEAMDATA data can be "grown" into data sources
        for RALLY-generated applications (RGA), and conver-
        sely...

    .   ...TEAMDATA can be used as an ad hoc query and re-
        porting tool for production (e.g., RALLY) databases.

    .   RGAs can be run from a TEAMDATA directory;

    .   Both TEAMDATA and RGAs can be run directly from an
        A1 all-in-1 menu;

    .   Conformance with WPS+ key mappings is a target for
        both products.

4. How will the runtime system for RALLY be packaged?

TEAMDATA will include a RALLY runtime system, so that one will be able to run RALLY applications from TEAMDATA. Of course, RALLY itself will include both the development and the runtime systems. A customer who wants to develop applications on Machine A and distribute the applications to users on Machines B, C, and D will probably buy the following licenses:

   o RALLY license; for the development system on Machine A

   o TEAMDATA licenses--for the runtime systems on Machines B, C, and D.

5. Do TEAMDATA and RALLY run on anything other than VMS?

Version 1 of both products will run on VMS and MicroVMS, across the entire range of VAX processors, with the exception of the VAX-11/725, from the 8800 down to the Micro VAX II. We are investigating other operating systems and hardware for future versions.

6. Why don't you have direct access to RMS in TEAMDATA and RALLY?

We do provide read-only access to RMS data through DATATRIEVE. The TEAMDATA or RALLY user specifies a DATATRIEVE domain and its CDD path as a data source. The TEAMDATA or RALLY user does NOT see DATATRIEVE prompts or have to know DATATRIEVE syntax. The TEAMDATA user has to name the DATATRIEVE domain only once; after that, the named domain appears in the TEAMDATA user's directory and the user opens it as any other data table.

We want to encourage people to use Rdb/VMS databases--either through storing their new data there or loading RMS files into Rdb/VMS. Rdb/VMS provides flexibility, protection of integrity, and other database features not available through RMS.

We are, however, investigating the advantages of providing direct RMS support for TEAMDATA and RALLY in future versions of those products.

7. Do TEAMDATA and RALLY use the Common Data Dictionary?

Only through their optional use of DATATRIEVE, which accesses the CDD (See preceding question for details on the use of DATATRIEVE). TEAMDATA and RALLY store metadata in Rdb/VMS system relations, as well as in their private dictionaries/directories. Rdb/VMS V2.0 includes a feature to add information in Rdb/VMS system relations to the CDD.

8. Does RALLY have access to TEAMDATA data?

Yes. Both RALLY and TEAMDATA are built on Rdb/VMS and store their data in Rdb relations, so data created by either product (as well as by Rdb/VMS) is available to the other.

For instance, a TEAMDATA user can "make known" an Rdb/VMS relation, treating it then as a TEAMDATA data table, while the RALLY application can reference it as it would any other Rdb/VMS relation. Thus, TEAMDATA can be a "friendly front end" to the Rdb/VMS data and RALLY can include in applications Rdb/VMS data that was created in a variety of ways; through TEAMDATA, RALLY, or Rdb/VMS products with user interfaces appropriate for a range of users.

VAX TEAMDATA

9. How will TEAMDATA Version 1 be integrated with ALL-IN-1?

TEAMDATA is integrated with ALL-IN-1 in the following ways for Version 1.0:

   - TEAMDATA function keys will be consistent with the BOS-E Office Products (ALL-IN-1, WPS-PLUS, etc.) User Interface Strategy Committee's keys.

   - TEAMDATA will support ALL-IN-1's interrupt menu.

   - It will be possible to paste TEAMDATA data (as text) into ALL-IN-1 documents or mail (via the scratch pad).

   - TEAMDATA will support all printers that ALL-IN-1 supports. Of course, graphics can be printed only on graphics printers. The user will be able to specify a default printer queue.

   - TEAMDATA will pass NEW MAIL notices on to the user in a nondisruptive manner.

   - TEAMDATA will provide mailable nonrevisable (except as text) files.

   - It will be possible to deny the TEAMDATA user privilege for DCL access.

   - TEAMDATA will run on all VAXes, from MicroVAX II on up.

10. How is locking handled for TEAMDATA data?

TEAMDATA allows concurrent access to TEAMDATA data tables, but does not allow concurrent update. Thus, if, for example one user is updating a data table, another user may open the data table and make queries on the data. The second user will receive a warning on opening the data table, informing him that he has read-only access. When

the second user tries to edit a field (modify data), he will receive an error message, saying that the data table cannot be modified at the moment.

11. Can I turn database journaling on and off in TEAMDATA?

    Not in Version 1 of TEAMDATA. We are looking at this for a later version of TEAMDATA.

12. If I don't have RDO bundled in TEAMDATA, how do I do back-up to move to a different machine?

    You don't need RDO to move an Rdb/VMS database to a different machine: you can use VMS COPY. Thus, you can move (through copying) your personal database of data tables.

    If you want to do a backup, but not move to a different machine, you can use VMS Backup.

    When Rdb/VMS makes version changes that change the on-disk format, TEAMDATA will do the backup and restore to convert to the new format.

13. What are the security mechanisms in TEAMDATA?

    Security in TEAMDATA provides protection from accidental, but not from malicious, misuse. TEAMDATA Version 1 provides for private data (only owner has access) and for public access (owner can place data in a "public folder," which allows world access).

    People with needs for greater security should use RDO on a database, or a relation in a database, to grant or deny access rights to specific users. TEAMDATA does not override Rdb/VMS security.

    We expect that we will provide more levels of protection within TEAMDATA in versions after Version 1. We have asked Rdb to provide field-level protection and expect to use that feature, when it is available. We welcome any suggestions for easy-to-use, and useful, security.

14. What are the data validation mechanisms available to the TEAMDATA user?

    TEAMDATA data validation is limited to data type checking, done when the user tries to exit from a field.

    Users who need more sophisticated data validation (such as range checking) can use RALLY, developing a data entry form for a data table or relation.

15. What sort of performance can I expect from TEAMDATA?

    We are currently in the process of doing single- and multi-user testing for TEAMDATA. The results of that testing are not yet available. However, based on preliminary data, we are confident that we will meet the perfor-

mance constraints set for the product, and that TEAMDATA users will be satisfied with the product's performance.

VAX RALLY

16. What is the target audience for RALLY?

    RALLY is a sophisticated application development system aimed primarily at the more advanced user. By virtue of its simple database creation tools and form/report/menu defaulting characteristics, however, RALLY allows sophisticated end users (for example, a Business Systems Analyst) to get started with the product and design simple, menu oriented applications.

    In order to be truly productive with RALLY and take maximum advantage of its capabilities, such users will either have to become more technical or turn their applications over to a more sophisticated user (for example, an Information Center Consultant or DP professional) for expansion and enhancement.

17. How is a BSA defined?

    A Business Systems Analyst (BSA), the entry level RALLY user, can be characterized as follows:

    . Has as his/her primary job responsibility understanding business-related problems and defining solutions for same (for example, knows the cost center budgeting process thoroughly and is chartered with building high level models to automate the process).

    . Has a data processing background (for example, has taken programming courses, was/is a DATATRIEVE user, understands basic computer concepts like database, file, record, field, etc., perhaps even did some programming in a former job, and so on); however, is not currently a programmer by profession.

    . Is not daunted by technical documentation, new learning, 4th Generation Tools; willing and able (given time, training and some guidance) to become a more technical RALLY user.

18. What makes the BSA an appropriate target user group?

    Because the BSA has a sound understanding of business problems and also a background in, and tolerance for, data processing tools and solutions, it is reasonable to believe that this person will want to help design and lay out the basics of an application. For example, this person may know what kind of database is appropriate to his/her application, may know what the forms and reports should look like, and therefore may want to pull together a

working prototype, complete with simple menu, that could later be expanded on by a more experienced user.

RALLY (and TEAMDATA) are perceived as ideal tools for an Information Center environment where many different kinds of users are encouraged to use 4GL tools to solve their own problems. According to the literature, ICs include a range of tools for such purposes, from the PC based spreadsheet and DB products to the full-blown 4GLs like EXPRESS and FOCUS. Clearly, the designation "end user" is inappropriate when applied to the person who has graduated from using spreadsheets for solving simple problems to designing complete, multi-user applications using FOCUS, RALLY and the like.

19. How will RALLY perform relative to COBOL? DATATRIEVE?

We are currently in the process of designing a test bed for RALLY. We have no final information at this time as to how RALLY will perform relative to COBOL or DATATRIEVE. However, it is a constraint for Version 1 that RALLY perform significantly better than DATATRIEVE in applications where comparison of the two products is appropriate. Additionally, it is a goal for Version 1 that performance of a RALLY application approach the performance of a comparable 3GL (e.g. COBOL) application.

RALLY is, in many respects, entirely different conceptually and architecturally from DATATRIEVE and COBOL. It is truly a 4th Generation System, integrating forms, reports, menus, database, messages and procedural code into a unified application generation environment. RALLY includes features that have no counterpart in either DATATRIEVE or COBOL. One such feature is the ability to create dynamic, editable reports. If the application definer chooses to do so, s/he can create a screen report for the user that can be edited and used to actually update the database. The user can move around in the report, adding and deleting fields and records, moving data, performing queries, editing text, and then actually commit the resulting changes in the underlying database upon exit. Such a feature presents to both the definer and the user unique opportunities for data entry, modification and presentation. However, the benefits are not easily assessed in terms of general performance, in comparison with products that don't include such functionality.

We are, therefore, trying to construct tests that measure both traditional features common to RALLY, DATATRIEVE, and COBOL (for example, data entry screens, form-to-record mappings, printed reports) as well as the unique features of RALLY. The results of this testing will be made available prior to product announcement.

20. Don't we have "Yet Another Forms" product with RALLY?

RALLY provides a forms function that is integrated with other application development functions. RALLY forms are not designed to be called and used in non-RALLY applications. However, we are currently investigating the feasibility of integrating FMS, TDMS, and RALLY forms.

21. How does RALLY handle concurrent access to data?

RALLY makes use of the underlying database capabilities of VAX Rdb/VMS to handle concurrent database access situations. This is one of the major advantages of building RALLY on a full-function relational database management system. The RALLY user enjoys all the advantages provided by such a system, including journaling, roll forward and roll back, and data security.

22. Are RALLY applications interpreted or compiled?

RALLY applications combine both interpreted and compiled components at runtime. RALLY does not generate source code. Also it is not an interpreter in the traditional sense, i.e., it does not generate a set of op-codes that are interpreted by a language processor at runtime. Rather, it is built around a unique, "data-structure-driven" architecture. This means that RALLY generates a set of data structures (records) which are treated at runtime much as any good program would treat its controlling structures.

All RALLY data structures are stored in a file called an AFILE, one AFILE(s) per application. All structure fields (i.e. features) which are not used are compressed and occupy virtually no space in the AFILE. At runtime, AFILE structures are materialized from the AFILE, kept in memory while they are used, and then discarded.

23. What validation capabilities are available in RALLY?

RALLY provides standard data validation mechanisms for forms, reports and database fields. Form fields, for example, can include range check validation or arbitrary validation through routines created by the application definer using RALLY's procedural language (ADL). The definer can also create a list of valid entries for one or more fields on a form. Users would then select from these values. RALLY would return an error for invalid entries.

24. What security capabilities does RALLY provide?

RALLY provides two levels of security for objects and applications:

    1. Passwords on objects: RALLY allows the definer to associate a password with one or more objects (for example, forms or reports) in an application.

    2. AFILE security: because RALLY applications are stored in VMS files (called AFILEs), definers can set protections on applications using VMS SET PROTECTION and ACL facilities.

Additional data security can be provided through the RDO utilities in Rdb/VMS.

25. Can RALLY access 3GL? How is this done?

Yes. RALLY provides the application definer with the ability to call 3GL code from a RALLY application. The definer specifies a RALLY action (known as an external program link) that identifies certain options and parameters that are to be used in hooking up with the user's program. He then can CALL or GOTO that action anywhere he would invoke any other action. Once called, the user program has control until it finishes, and once it returns, the user code invocation item is taken off the calling task's execution stack. The external program runs as part of the RALLY application sharable image, not as a separate image.

26. How does RALLY report writing functionality stack up against DATATRIEVE?

RALLY provides more powerful and flexible report writing capabilities than DATATRIEVE. For example, RALLY reports can be displayed in multiple terminal windows, can contain highly complex formats and can optionally be designed as editable.

While DATATRIEVE includes a report writing language, RALLY reports are built using a form-driven "dialogue" with the definer, and are typically part of a larger, menu-driven application. The RALLY report writer can be used to construct complex insurance form and cross tabs reports, constructs which DATATRIEVE does not handle easily, if at all.

---

## VAX/VMS ReGIS to SIXEL Conversion

Donald E. Stern, Jr., Warner Lambert Company, Milford, CT 06460

---

At our facility, we are deeply committed to the development of a coherent CIM (Computer Integrated Manufacturing) program. Many pieces of the CIM puzzle are in place, from real time data collection stations on the shop floor to electronic exchange of information at the top management level. In order to accomplish this with a minimum of staff, extensive use of Datatrieve and DECGraph has been made to process the data collected in the factory and present it in a condensed form to management. Quite often, the reports reaching management take the form of ReGIS plots which are created in batch processes during off hours. While most of the staff members are equipped with VT240 terminals, there is a need to produce hardcopies of the graphs for those staff members not equipped with graphics terminals.

While Datatrieve provides a mechanism to obtain hardcopies of ReGIS plots, it is limited. It requires the use of a printer attached to the printer port of the terminal displaying the ReGIS image. For example, the following Datatrieve commands will produce a plot on an attached printer.

    DTR> Ready yachts
    DTR> Plot x_y loa,price of yachts
    DTR> Plot hardcopy

When the "PLOT HARDCOPY" command is issued, the terminal will send data out through the printer port of the terminal to an attached printer. Several DEC printers, including the LA50, LA100, and LN03 are compatable with the data transmitted. This data is in the form of SIXEL graphics code. Each character of the code, in the range of 077 to 176 octal, represents a set of 6 vertically aligned pixels on the printed page. Thus, the ASCII characters ? through ~ are used to represent the SIXEL image.

The particular pixels "illuminated" for a given set are determined according to the following algorithm. The value 077 octal is subtracted from the SIXEL code and the resultant bit pattern determines which pixels are "on" and which are "off." The least significant bit (LSB) represents the topmost pixel while the most significant bit represents the bottom pixel in the set. For example, suppose the sixel codes "L?Gp" were received. The octal values of these characters are 114, 077, 107 and 160, respectively. The following table illustrates the translation.

| Octal Value | Binary Value | Translation |
|---|---|---|
| 114 | 001 001 100 | 000 001 101 |
| 077 | 000 111 111 | 000 000 000 |
| 107 | 001 000 111 | 000 001 000 |
| 160 | 001 110 000 | 000 110 001 |

The resultant three sets of six pixels are shown below, where "x" represents an "on" pixel and "o" represents an "off" pixel.

| Sixel Code: | | L | ? | G | p |
|---|---|---|---|---|---|
| Top Pixel Bit | 0 (LSB) | x | o | o | x |
| | 1 | o | o | o | o |
| | 2 | x | o | o | o |
| | 3 | x | o | x | o |
| | 4 | o | o | o | x |
| | 5 (MSB) | o | o | o | x |

In addition to the printable SIXEL codes, the ASCII characters "!" (041 octal), "$" (044 octal), and "-" (055 octal) are used as graphics control characters. The "-" character is the code for a new graphics line. When the printer intercepts this code, the printer returns the carriage and indexes the paper. The "$" character causes the carriage to be returned but no indexing takes place. The "!" character is used to introduce a repeat sequence. A repeat sequence takes the following form:

    !<numeric><print character>

where <numeric> is a string of numeric ASCII characters (060 - 071 octal) evaluated as a decimal number and which indicate the number of times to repeat the <print character>. As stated earlier, <print character> is in the octal range of 077 to 176. The following illustrates this function:

| Sequence | Result |
|----------|--------|
| !47~ | Prints 47 adjacent sets of completely filled vertically aligned pixels |
| !10? | Prints 10 adjacent sets of graphics spaces |
| !37@ | Prints 37 adjacent sets of patterns in which only the top pixel is filled. |

In order to use the Datatrieve "PLOT HARDCOPY" command, it is necessary to have a suitable printer attached to the terminal creating the ReGIS image. Since not all terminals can reasonably be expected to have an attached printer, one might ask whether graphics hardcopy can be redirected to a printer available system wide. As has been pointed out in a previous issue of the Wombat Examiner (1), one method for converting a ReGIS file to a SIXEL file suitable for printing, is to use DECSlide. The following set of commands illustrates the process.

```
$ DTR
DTR> Ready yachts
DTR> Plot x_y loa,price of yachts on YACHTS.SLO
DTR> Exit
$ SLIDE/noint/sixel=YACHTS.SLS YACHTS.SLO
$ PRINT YACHTS.SLS
```

In this example, Datatrieve creates the ReGIS file YACHTS.SLO and DECSlide processes that file, creating a SIXEL file YACHTS.SLS which can be printed on any printer cabable of handling SIXEL graphics. The problems with this approach are:

a.) A separate DECSlide license is required on all VAXes running DECSlide.
b.) Its cost isn't justified by merely its ability to create SIXEL files.
c.) IT DOESN'T WORK IN BATCH MODE; DECSlide will only perform the SIXEL conversion on an interactive terminal. (The /NOINTERACTIVE qualifier merely tells the DECSlide program that a ReGIS file already exists.)

While DECGraph permits the creation of SIXEL files directly, there exists the restriction that SIXEL conversions can only be run from an interactive process.

Since, at our site, there was a need to create these SIXEL files during off hours and because leaving interactive processes logged in was unacceptable from a system security standpoint, an effort was made to overcome the limitations imposed by DECSlide and DECGraph to produce the printable graphs. After discussing the problem with several people, the program SIXEL was created. The approach taken in SIXEL was to send the particular escape se-

quence to the graphics terminal instructing it to redirect graphics output from the printer port to the host. SIXEL was written in VAX FORTRAN and the source code is appended to this article.

SIXEL can be run either interactivly or in batch. It was written for use with the VT240 series of graphics terminals. The SIXEL program requires the use of a graphics terminal specified by the logical name DISPLAY$DEVICE. If no translation exists for this logical name, the program defaults to TT:. (Note: The process running SIXEL must be able to allocate DISPLAY$DEVICE. This means that suitable protection will have to be placed on the device or the process must be sufficiently privileged.)

A file called SIXEL.CLD, containing the following lines, should be placed in SYS$MANAGER.

```
DEFINE VERB SIXEL
IMAGE "APPLICATIONS$DISK:CONVERT_TO_SIXEL"
PARAMETER P1
        LABEL = REGIS_FILE
        VALUE (TYPE = $FILE, REQUIRED)
        PROMPT = "ReGIS Input File"
PARAMETER P2
        LABEL = SIXEL_FILE
        VALUE (DEFAULT = "SIXEL.SLS")
QUALIFIER LOCAL
        NONNEGATABLE
        PLACEMENT = GLOBAL
QUALIFIER RESET
        NONNEGATABLE
        PLACEMENT = GLOBAL
QUALIFIER GRAPH_TYPE
        VALUE (TYPE=GRAPH_KEYWORDS)
        NONNEGATABLE
        PLACEMENT = GLOBAL
DISALLOW SIXEL_FILE AND LOCAL
DEFINE TYPE GRAPH_KEYWORDS
        KEYWORD COMPRESSED
        KEYWORD EXPANDED
        KEYWORD ROTATED
```

The following line should be added to the system wide login command file.

```
$ SET COMMAND SYS$MANAGER:SIXEL
```

Alternatively, the SIXEL command could be permanently added to the DCL command tables or some set of alternate command tables.

To use SIXEL, a ReGIS file must already exist. Taking the YACHTS.SLO file created above, several possibilities exist for using SIXEL. In order to create a SIXEL file using the default graphics setup of a VT240, the following command would be used.

```
$ SIXEL YACHTS.SLO YACHTS.SLS
```

To create a rotated SIXEL file, the command

$ SIXEL/GRAPH_TYPE=ROTATED YACHTS.SLO YACHTS.SLS

could be used. Similarly, the /GRAPH_TYPE=EXPANDED and the
/GRAPH_TYPE=COMPRESSED qualifiers will explicitly instruct the
VT240 to produce expanded graphs and normal compressed graphics
output respectively. The /RESET qualifier is used to reset the
VT240 to compressed mode when finished. If the /LOCAL switch is
used, then the second parameter is disallowed and SIXEL output is
directed to the printer port of the graphics terminal.

Unlike DECSlide, SIXEL can be run in batch mode or from a non-
graphics terminal. As long as DISPLAY$DEVICE translates to a
ReGIS terminal (and it is not currently allocated to another
process), SIXEL can be used. For example, the following could
be run from a batch process.

```
$! Procedure Make_sixel.com
$!
$! First create a ReGIS file
$ DTR
        Ready yachts
        Plot X_Y loa,price of yachts on yachts.slo
        Exit
$!
$ DEFINE DISPLAY$DEVICE _TTA6:        !TTA6 is a VT240
$ SIXEL/GRAPH=EXPANDED YACHTS.SLO MY_GRAPH.SIX
$ PRINT MY_GRAPH.SIX
```

As stated earlier, SIXEL was developed for use with VT240 termi-
nals but some reduced functionality can be expected from other
ReGIS devices.

### Acknowledgment

The author would like to express gratitude to Bart Z. Lederman
for his many suggestions leading to the creation of this work
and for testing and criticizing the completed program.

### References

1.) Harris, Basil, Wombat Examiner, Vol. 6, No. 1, p. 16, 1985.

2.) "VT240 Series Owners Manual", Digital Equipment Corp.,
Maynard, MA, EK-VT240-UG-001, 1983

3.) "LA50 Printer Programmers Reference Manual", Digital
Equipment Corp., Maynard, MA, EK-0LA50-RM-002, 1983.

4.) "VAX/VMS V4.2 Documentation Set", Digital Equipment Corp.,
Maynard, MA, QL001-GZ-V4.2/QQ, 1985.

5.) "LN03 Programmer Reference Manual", Digital Equipment Corp.,
Maynard, MA, EK-0LN03-RM-001, 1985.

---

**PROGRAM CONVERT_TO_SIXEL**

```
C   Version:    V1.0
C
C   Functional Description:
C
C   This program will convert a ReGIS graphics file (such as
C   one produced by Datatrieve) and produce a SIXEL graphics
C   file which is suitable for printing on an appropriate
C   printer (LA50, LA100, LN03, etc.). It was written mainly
C   to get around the need for DECSlide to create SIXEL files
C   since a.) DECSlide isn't cheap and b.) it cannot be used
C   in batch mode to create the SIXEL files.
C
C   Since the SIXEL conversion is a hardware function of a
C   graphics terminal, the program requires the use of a REGIS
C   terminal. This device is defined by the logical
C   DISPLAY$DEVICE. For an interactive terminal, therefore,
C   a definitions such as the following should be made.
C
C       $ DEFINE DISPLAY$DEVICE TT:
C
C   If there is no logical translation for DISPLAY$DEVICE, the
C   program performs the above definition on the users behalf.
C
C   In addition, DISPLAY$DEVICE should be set to REGIS; the
C   program will abort if this characteristics is not set.
C
C   The specified ReGIS file is opened and displayed on the
C   screen of DISPLAY$DEVICE. The required sequences are sent
C   to this device which set graphics to host, enter REGIS,
C   and initiate hardcopy output.
C
C   The appropriate terminal characteristics are set and
C   successive QIO's are issued to read the SIXEL data which
C   is returned a byte at a time. The end of the conversion
C   process is signaled by receipt of the byte after the third
C   <ESC> character is detected. The SIXEL data is collected
C   in a 255 character buffer which is written to disk when
C   full. The SIXEL file, therefore, can be accessed by a
C   standard text editor if necessary.
C
C   Input to this program comes via the command line
C   interpreter. Specifically, the RTL routines CLI$GET_VALUE
C   and CLI$PRESENT are used. The command line definition
C   file SIXEL.CLD contains the necessary information to
C   define a command called SIXEL. The following command line
C   should be placed in the system wide login command file:
C
C       $ SET COMMAND <directory>SIXEL
C
C   where <directory> is the path to the SIXEL.CLD file. The
C   following command syntax is used to activate the program:
C
C
C   $ SIXEL/GRAPH_TYPE=<option>  regis-file-spec  sixel-file-spec
C   or
C   $ SIXEL/GRAPH_TYPE=<option>/LOCAL/RESET  regis-file-spec
```

```
C     where /GRAPH_TYPE, /RESET and /LOCAL are optional qualifiers.
C     Valid values for the /GRAPH_TYPE qualifier are EXPANDED, ROTATED,
C     and COMPRESSED.  If the EXPANDED option is selected, the SIXEL
C     output produced will be approximately 12x8 inches.  The ROTATED
C     option will produce output approximately 8x12 inches and rotated
C     90 degrees; this will fit on standard 8 1/2 x 11 paper such as
C     that typically loaded into an LA50 printer.  Finally, the
C     COMPRESSED option will produce a graph approximately 6x3 inches.
C
C     If the /LOCAL qualifier is present, SIXEL output is directed to a
C     printer attached to DISPLAY$DEVICE.  If this qualifier is
C     specified no sixel-file-spec is allowed.
C
C     If the /RESET qualifier is present, DISPLAY$DEVICE will be set to
C     COMPRESSED mode before exiting otherwise it will be left in the
C     mode specified by the /GRAPH_TYPE option.
C
C
C     CREATED 1986,     WARNER-LAMBERT COMPANY
C                       CONSUMER HEALTH PRODUCTS GROUP
C                       MILFORD CONN 06460
C
C     Author:                 Creation date:
C
C     Donald E. Stern, Jr.  -   February 4, 1986
C
C
C     Modification History:
C
C     Ver.     Author      Date     Reason for change
C
C
C     Execution Enviroment
C
C          VAX/VMS V4.2 or greater
C
C     Functions and Subroutines Called:
C
C     GET_TERMINAL_CHARACTERISTICS    - Internally developed routine to
C                                        get terminal characteristics
C     SET_TERMINAL_CHARACTERISTICS    - Internally developed routine to
C                                        set terminal characteristics
C     SYS$QIOW                        - Synchronous queued I/O system
C                                        service
C     SYS$ASSIGN                      - System service to assign a
C                                        channel number to a device
C     CLI$PRESENT                     - RTL CLI argument detector
C     CLI$GET_VALUE                   - RTL CLI argument fetcher
C     LIB$SYS_TRNLOG                  - RTL routine to translate
C                                        logical names
C     LIB$SET_LOGICAL                 - RTL routine to create a logical
C                                        name
C     LIB$GETDVI                      - RTL routine to get device
C                                        information
C
C     SYS$QIO Structures
```

DTR-29

```
      INTEGER*2    INPUT_CHAN         !I/O Channel
      INTEGER      CODE,              !Type of I/O operation
     1             INPUT_BUFFER_SIZE, !Input buffer size
     2             INPUT_SIZE         !Size of input as read

      PARAMETER    (INPUT_BUFFER_SIZE=1)

C     Terminal Characteristics Buffer

      BYTE         CLASS,
     1             TYPE
      INTEGER*2    WIDTH
      INTEGER*4    BASIC,
     1             EXTENDED
      INTEGER*4    OLD_BASIC,         !Old basic terminal characteristics
     1             OLD_EXTENDED       !Old extended terminal characteristics

C     I/O Operations Definitions

      INCLUDE      '($IODEF)'

C     Terminal I/O Operation Modifiers

      INCLUDE      '($TTDEF)'
      INCLUDE      '($TT2DEF)'

C     System Service Definitions

      INCLUDE      '($SSDEF)'

C     Device Information Definitions

      INCLUDE      '($DVIDEF)'
      INCLUDE      '($DCDEF)'

C     QIO Status Block
      STRUCTURE   /IOSTAT_BLOCK/
                  INTEGER*2    IOSTAT,        !Return Status
     1                         TERM_OFFSET,   !Location of Line terminator
     2                         TERMINATOR,    !Value of terminator
     3                         TERM_SIZE      !Size of terminator
      END STRUCTURE
      RECORD /IOSTAT_BLOCK/ IOSB

C     Subprograms

      INTEGER*4    SYS$ASSIGN,
     1             SYS$QIOW,
     2             CLI$PRESENT,
     3             CLI$GET_VALUE,
     4             LIB$SYS_TRNLOG,
     5             LIB$SET_LOGICAL,
     6             LIB$GETDVI

      INTEGER*4    STATUS             !Return status

      INTEGER*4    DEVICE_CLASS       !Device class returned by
                                      !     LIB$GETDVI
```

DTR-30

```fortran
      BYTE          ENTER_REGIS(4)        !Escape sequence to enter ReGIS
                                          !  mode
      BYTE          EXIT_REGIS(2)         !Escape sequence to exit ReGIS
      BYTE          LOCK_KEYBOARD(4)      !Escape sequence to lock keyboard
      BYTE          UNLOCK_KEYBOARD(4)    !Escape sequence to unlock keybrd
      BYTE          GRAPHICS_TO_HOST(5)   !Esc sequence for graphics to hst
      BYTE          GRAPHICS_TO_PRINTER(5)   !Sequence for graph to prntr
      BYTE          EXPANDED_PRINT(6)     !Esc sequence for 2X wide  graph
      BYTE          ROTATED_PRINT(6)      !Esc sequence for rotated graph
      BYTE          COMPRESSED_PRINT(6)   !Sequence for compressed print

      INTEGER*2     NR,                   !Number of characters in REGIS_FILE
     1              NS,                   !Number of characters in SIXEL_FILE
     2              NG,                   !Number of characters in QUALIFIER
     3              NT                    !Number of characters in TRANSLATION

      CHARACTER*2   EXIT_GRAPHICS         !Same as EXIT_REGIS
      CHARACTER*10    GRAPH_TYPE          !Command line qualifier
      CHARACTER*13   MAKE_HARDCOPY        !ReGIS instruction to make
                                          !  graphics hardcopy
      CHARACTER*255  REGIS_FILE           !ReGIS source file
      CHARACTER*255  SIXEL_FILE           !SIXEL destination file
      CHARACTER*255  TRANSLATION          !Logical name translation
      CHARACTER*256  BUFF                 !Buffer for ReGIS & SIXEL
                                          !  data

      LOGICAL       QUIT

      EQUIVALENCE (EXIT_REGIS(1),EXIT_GRAPHICS)

      DATA ENTER_REGIS /27,80,49,112/          ! <ESC>P1p
      DATA EXIT_REGIS /27,92/                   ! <ESC>\
      DATA LOCK_KEYBOARD /27,91,50,104/         ! <ESC>[2h
      DATA UNLOCK_KEYBOARD /27,91,50,108/       ! <ESC>[21
      DATA GRAPHICS_TO_HOST /27,91,63,50,105/   ! <ESC>[?2i
      DATA GRAPHICS_TO_PRINTER /27,91,63,48,105/ ! <ESC>[?0i
      DATA EXPANDED_PRINT /27,91,63,52,51,104/  ! <ESC>[?43h
      DATA ROTATED_PRINT /27,91,63,52,55,104/   ! <ESC>[?47h
      DATA COMPRESSED_PRINT /27,91,63,52,51,108/ ! <ESC>[?431
      DATA MAKE_HARDCOPY /'S(H(P[50,0]))'/

C     See if there is a logical translation for DISPLAY$DEVICE and, if
C     not, assign 'TT:' to it.

      STATUS = LIB$SYS_TRNLOG('DISPLAY$DEVICE',NT,TRANSLATION,,,)

      IF (STATUS .EQ. SS$_NOTRAN)
     1     STATUS = LIB$SET_LOGICAL('DISPLAY$DEVICE','TT:',,,)

      IF (.NOT. STATUS)
     1     STOP 'Error translating or defining DISPLAY$DEVICE'

C     Make sure that DISPLAY$DEVICE is a terminal device

      CODE = DVI$_DEVCLASS
      STATUS = LIB$GETDVI (CODE,,'DISPLAY$DEVICE',DEVICE_CLASS,,)
```

```fortran
      IF (STATUS .EQ. SS$_NOSUCHDEV)
     1     STOP 'Error - DISPLAY$DEVICE is unknown'

      IF (.NOT. STATUS)STOP 'Error getting device information'

      IF (DEVICE_CLASS .NE. DC$_TERM)
     1     STOP 'Error - DISPLAY$DEVICE is not a terminal'

C     Assign a channel number to DISPLAY$DEVICE

C     Get a Channel No. for  DISPLAY$DEVICE

      STATUS = SYS$ASSIGN('DISPLAY$DEVICE',INPUT_CHAN,,)

C     Stop on Errors

      IF (.NOT. STATUS)
     1     STOP 'Error assigning channel to DISPLAY$DEVICE'

C     Get the existing terminal characteristics and save them

      CALL GET_TERMINAL_CHARACTERISTICS (INPUT_CHAN,CLASS,
     1     TYPE,WIDTH,BASIC,EXTENDED)
      OLD_BASIC    = BASIC
      OLD_EXTENDED = EXTENDED

C     If the terminal does not support ReGIS stop with an error message

      IF((EXTENDED .AND. TT2$M_REGIS) .EQ. 0)
     1     STOP ' DISPLAY$DEVICE must be set to support ReGIS'

C     Use the command line interpreter facility to get inputs

      STATUS = CLI$GET_VALUE('REGIS_FILE',REGIS_FILE,NR)
      IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

      IF(CLI$PRESENT('SIXEL_FILE'))
     1     STATUS = CLI$GET_VALUE('SIXEL_FILE',SIXEL_FILE,NS)

      IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

      IF (CLI$PRESENT('GRAPH_TYPE'))
     1     STATUS = CLI$GET_VALUE('GRAPH_TYPE',GRAPH_TYPE,NG)

      IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

C     Open the input file

      OPEN (UNIT=1,NAME=REGIS_FILE(1:NR),
     1     ACCESS='SEQUENTIAL',STATUS='OLD',
     2     READONLY,FORM='FORMATTED',RECORDTYPE='VARIABLE',
     3     CARRIAGECONTROL='LIST',ERR=1100)

C     Associate LUN 7 with DISPLAY$DEVICE and allow for large records

      OPEN (UNIT=7,NAME='DISPLAY$DEVICE',RECL=1024,
     1     STATUS='OLD',ERR=1200)
```

```
C      Lock the keyboard to avoid extraneous input during conversion

       WRITE(7,18) LOCK_KEYBOARD
18     FORMAT('$',4A1)

C      Read the ReGIS code and display the plot on the screen of LUN 7

20     READ(1,25,END=100,ERR=1300) NC,BUFF(1:NC)
25     FORMAT(Q,A)
       WRITE(7,27) BUFF(1:NC)
27     FORMAT('+',A)
       GOTO 20                        !Loop on READ 'til EOF

100    CLOSE (UNIT=1)                 !Close the ReGIS file

C      Open a new file to contain the SIXEL information

       OPEN  (UNIT=1,NAME=SIXEL_FILE(1:NS),
1            ACCESS='SEQUENTIAL',STATUS='NEW',
2            FORM='UNFORMATTED',RECORDTYPE='VARIABLE',
3            CARRIAGECONTROL='LIST',ERR=1400)


C      Set the required terminal characteristics

       BASIC = BASIC .OR.
1              TT$M_HOSTSYNC .OR.       !Host sync. (CTRL-S on
                                        ! typeahead full)
2              TT$M_NOBRDCST .OR.       !No broadcast messages
3              TT$M_NOECHO   .OR.       !No echo
4              TT$M_TTSYNC             !Terminal/Host sync

       BASIC = IBCLR(BASIC,TT$V_WRAP)      !No wrap
       BASIC = IBCLR(BASIC,TT$V_EIGHTBIT)  !Use 7 bit codes
       EXTENDED = EXTENDED .OR. TT2$M_PASTHRU  .OR.  !Pass all data
1              TT2$M_XON                    !Resume output on SET

       CALL SET_TERMINAL_CHARACTERISTICS (INPUT_CHAN,CLASS,
1          TYPE,WIDTH,BASIC,EXTENDED)
C      Send the appropriate escape sequence to DISPLAY$DEVICE to set it
C      up to print graphics

       IF(GRAPH_TYPE(1:NG) .EQ. 'EXPANDED') THEN
              WRITE(7,102)EXPANDED_PRINT
              MAKE_HARDCOPY(7:7)='0'
       ENDIF

       IF(GRAPH_TYPE(1:NG) .EQ. 'ROTATED') THEN
              WRITE(7,102)ROTATED_PRINT
              MAKE_HARDCOPY(7:7)='0'
       ENDIF

       IF(GRAPH_TYPE(1:NG) .EQ.
1           'COMPRESSED') WRITE(7,102)COMPRESSED_PRINT
101    FORMAT('+',2A1,5A1,4A1,A13,$)
102    FORMAT('+',6A1,$)
103    FORMAT('+',2A1,4A1,A13,$)
```

```
104    FORMAT('+',2A1,4A1,$)

       IF (CLI$PRESENT('LOCAL'))THEN    !Hardcopy to attached printer
              WRITE(7,103)EXIT_REGIS,ENTER_REGIS,MAKE_HARDCOPY
              WRITE(7,104)EXIT_REGIS,UNLOCK_KEYBOARD
       ELSE
              WRITE(7,101)EXIT_REGIS,GRAPHICS_TO_HOST,
1                 ENTER_REGIS,MAKE_HARDCOPY

              N    = 1                  !Position in BUFF
              NESC = 0                  !Number of <ESC>'s detected
              QUIT = .FALSE.            !Quit code
              CODE = IO$_READVBLK       !I/O code - Read virtual
                                        !   block

200           STATUS = SYS$QIOW(,
1                   %VAL(INPUT_CHAN),     !DISPLAY$DEVICE
2                   %VAL(CODE),           !Read virtual block
3                   IOSB,,,               !I/O Status
4                   %REF(BUFF(N:N)),      !Input Buffer
5                   %VAL(INPUT_BUFFER_SIZE),,,,)    !Buffer Size

              IF (.NOT. STATUS) GOTO 1000
              IF (.NOT. IOSB.IOSTAT) GOTO 1000
              IF (QUIT) GOTO 250

              IF(BUFF(N:N) .EQ. CHAR(27))NESC=NESC+1    !Check for <ESC>
              IF(NESC .EQ. 3)QUIT = .TRUE.              !Esc on the byte
                                                        ! after the 3rd
                                                        !      <ESC>

              IF (N .EQ. 255) THEN
                     WRITE(1)BUFF(1:255)
                     N=0
              ENDIF

              N = N+1
              GOTO 200

250           WRITE(1)BUFF(1:N)
              CLOSE(UNIT=1)

              WRITE(7,252) EXIT_GRAPHICS, UNLOCK_KEYBOARD,
1                   GRAPHICS_TO_PRINTER
252           FORMAT('+',A2,4A1,5A1)
       ENDIF

C      Reset terminal characteristics

       CALL SET_TERMINAL_CHARACTERISTICS (INPUT_CHAN,CLASS,
1          TYPE,WIDTH,OLD_BASIC,OLD_EXTENDED)
       IF(CLI$PRESENT('RESET')) WRITE(7,102)COMPRESSED_PRINT
       STOP

C      The following lines process errors
1000   WRITE(7,252)EXIT_GRAPHICS,UNLOCK_KEYBOARD,GRAPHICS_TO_PRINTER
       CALL SET_TERMINAL_CHARACTERISTICS (INPUT_CHAN,CLASS,
1          TYPE,WIDTH,OLD_BASIC,OLD_EXTENDED)
```

```
        STOP 'Error reading SIXEL' 1100 STOP 'Error opening ReGIS file'
1200    STOP 'Error opening LUN 7' 1300 STOP 'Error reading ReGIS data'
1400    STOP 'Error opening SIXEL file'
        END



        Subroutine GET_TERMINAL_CHARACTERISTICS (CHANNEL,
1           CLASS, TYPE, WIDTH,
2           BASIC, EXTENDED)

C       Title: Get Terminal Characteristics
C
C       Version: 1.0
C
C       Functional Description:
C           Terminal characteristics are determined and appropriate
C           bits are set in the longwords BASIC and EXTENDED.
C           Reference the Terminal Driver section of the VAX/VMS
C           I/O Reference manual.
C
C       CREATED 1986,    WARNER-LAMBERT COMPANY
C                        CONSUMER HEALTH PRODUCTS GROUP
C                        MILFORD CONN 06460
C
C       Author: Donald E. Stern, Jr.      Creation date: 16-Jan-1986
C
C
C
C       Modification History:
C
C       Ver.        Author      Date      Reason for change
C
C
C       Execution Enviroment:
C           VAX/VMS
C
C       Functions and Subroutines Called:
C           SYS$QIOW  - Reference System Services Manual
C           LIB$SIGNAL
C

        INTEGER*2   CHANNEL         !Channel of terminal to set
        BYTE        CLASS,
1           TYPE
        INTEGER*2   WIDTH
        INTEGER*4   BASIC,
1           EXTENDED

C       I/O Operations Definitions

        INCLUDE         '($IODEF)'

C       QIO Status Block

        STRUCTURE    /IOSTAT_BLOCK/
            INTEGER*2 IOSTAT          !Return Status
            BYTE      TRANSMIT,
```

```
1                   RECIEVE,
2                   CRFILL,
3                   LFFILL,
4                   PARITY,
5                   ZERO
        END STRUCTURE
        RECORD /IOSTAT_BLOCK/ IOSB

C       Characteristics buffer

        STRUCTURE /CHARACTERISTICS/
            BYTE      CLASS,
1                     TYPE
            INTEGER*2 WIDTH
            INTEGER*4 BASIC
            INTEGER*4 EXTENDED
        END STRUCTURE
        RECORD /CHARACTERISTICS/ CHARBUF

C       Subprograms

        INTEGER*4   SYS$QIOW

        INTEGER*4   STATUS           !Return status

C       Get the current terminal characteristics

        STATUS = SYS$QIOW(,
1               %VAL(CHANNEL),        !I/O Channel Number
2               %VAL(IO$_SENSEMODE),     !Sensemode Function Code
3               IOSB,,,              !I/O Status Block
4               CHARBUF,             !Characteristics buffer
5               %VAL(12),,,,)        !Buffer Size

        IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))
        IF (.NOT. IOSB.IOSTAT) CALL LIB$SIGNAL(%VAL(IOSB.IOSTAT))

C       Return the terminal characteristics to the calling program

        CLASS    = CHARBUF.CLASS
        TYPE     = CHARBUF.TYPE
        WIDTH    = CHARBUF.WIDTH
        BASIC    = CHARBUF.BASIC
        EXTENDED = CHARBUF.EXTENDED

        RETURN
        END
```

```
      Subroutine SET_TERMINAL_CHARACTERISTICS (CHANNEL,
     1        CLASS, TYPE, WIDTH,
     2        BASIC, EXTENDED)

C     Title: Set Terminal Characteristics
C
C     Version: 1.0
C
C     Functional Description:
C         Terminal characteristics are set depending upon which
C         bits are set in the longwords BASIC and EXTENDED.
C         Reference the Terminal Driver section of the VAX/VMS
C         I/O Reference manual.
C
C     CREATED 1986,    WARNER-LAMBERT COMPANY
C                      CONSUMER HEALTH PRODUCTS GROUP
C                      MILFORD CONN 06460
C
C     Author: Donald E. Stern, Jr.        Creation date: 16-Jan-1986
C
C
C     Modification History:
C
C     Ver.        Author        Date      Reason for change
C
C
C     Execution Enviroment:
C         VAX/VMS
C
C     Functions and Subroutines Called:
C         SYS$QIOW  - Reference System Services Manual
C         LIB$SIGNAL
C
C     I/O Operations Definitions

      INCLUDE          '($IODEF)'

C     QIO Status Block

      STRUCTURE    /IOSTAT_BLOCK/
           INTEGER*2 IOSTAT          !Return Status
           BYTE        TRANSMIT,
     1                 RECIEVE,
     2                 CRFILL,
     3                 LFFILL,
     4                 PARITY,
     5                 ZERO
      END STRUCTURE
      RECORD /IOSTAT_BLOCK/ IOSB

C     Characteristics buffer passed to the QIOW system service

      STRUCTURE /CHARACTERISTICS/
           BYTE        CLASS,
     1                 TYPE
           INTEGER*2 WIDTH
           INTEGER*4 BASIC
           INTEGER*4 EXTENDED
```

```
      END STRUCTURE
      RECORD /CHARACTERISTICS/ CHARBUF

C     Characteristics arguments passed to this subroutine

      INTEGER*2    CHANNEL        Channel of terminal to set
      BYTE         CLASS,
     1             TYPE
      INTEGER*2    WIDTH
      INTEGER*4    BASIC,
     1             EXTENDED

C     Subprograms

      INTEGER*4    SYS$QIOW

      INTEGER*4    STATUS              !Return status

C     Set the appropriate characteristics

      CHARBUF.CLASS    = CLASS
      CHARBUF.TYPE     = TYPE
      CHARBUF.WIDTH    = WIDTH
      CHARBUF.BASIC    = BASIC
      CHARBUF.EXTENDED = EXTENDED

C     Set the new terminal characteristics

      STATUS = SYS$QIOW(,
     1          %VAL(CHANNEL),        !I/O Channel Number
     2          %VAL(IO$_SETMODE),    !Sensemode Function Code
     3          IOSB,,,               !I/O Status Block
     4          CHARBUF,              !Characteristics buffer
     5          %VAL(12),,,,)         !Buffer Size

      IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))
      IF (.NOT. IOSB.IOSTAT) CALL LIB$SIGNAL(%VAL(IOSB.IOSTAT))

      RETURN
      END
```

# The GREAT WOMBAT
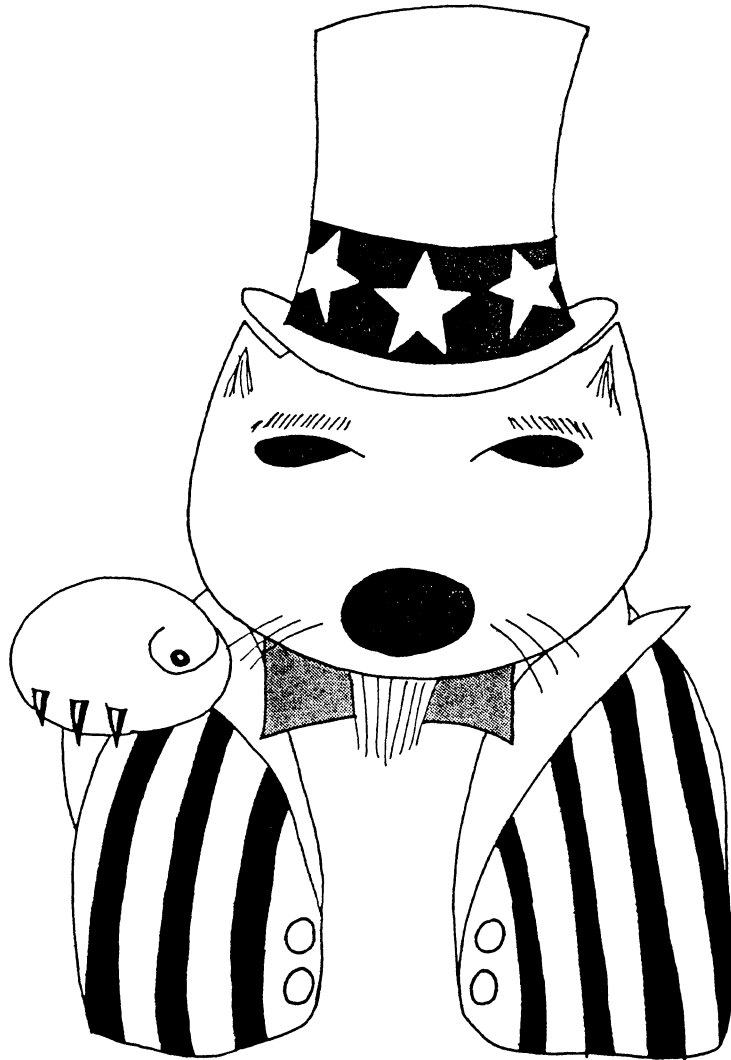


# Wants You !

To participate in the DTR/4GL SIG by submitting articles to the
newsletter, volunteering at symposia, or being a Datatrieve
Master.  Contact a SIG Officer for details.

# EDUSIG

**Chairman**
Robert A. Shive, Jr.
Millsaps College
Jackson, MS

**Symposium Coordinator**
Sue Bates
Northwestern Michigan College
Traverse City, MI

**Communications Committee Representative**
Robert W. McCarley
Millsaps College
Jackson, MS

**Newsletter Editor**
Fred Bell
Taft College
Taft, CA

**PSS Coordinator VAX Systems SIG Liaison**
Donald C. Fuhr
Tuskegee Institute
Tuskegee Institute, AL

**Administrative Applications Coordinator**
Dave Cothrun
Taft College
Taft, CA

**Courseware Coordinator**
Mary Jac Reed
Off Comp Based Instruction
Newark, DE

**DEC Counterpart**
Gary Finerty
Digital Equipment Corporation
Marlboro, MA

EDU-i

EDU
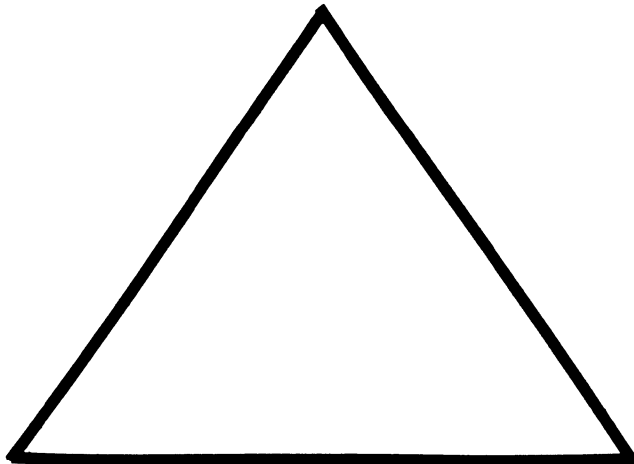
# A MAJOR ANNOUNCEMENT
## ACADEMIC SOFTWARE CLEARINGHOUSE

The following article outlines conditions for the newly
announced CLEARINGHOUSE FOR ACADEMIC SOFTWARE that will be
operated by Iowa State University. Establishing such a
clearinghouse has been a major goal of EDUSIG for over five
years, and the announcement is greeted with a great deal of
enthusiasm.

The Clearinghouse has published three documents, the Owner's
Guide, the Owner's Agreement and the Software Submittal Form.

EDUSIG feels the Owner's Guide outlines the important
aspects of the Clearinghouse, and has published the current
version in this issue. For the other documents, and further
correspondence:

> The Clearinghouse for Academic Software
> The Computation Center
> 104 Computer Science Building
> Ames, Iowa 50011
> (515) 294-0323

Digital Equipment Corporation, through the EDU group, has
funded dozens of institutions through several grant programs
to help colleges and universities to begin to produce quality
educational courseware. The results of such grants will soon
be found through the Iowa State non-profit Clearinghouse.

Since the Clearinghouse Owner's Guide is several pages, the
EDUSIG editors may well have used up page allotments for
several issues. New submissions are still accepted, however.

> Fred Bell -- David Cothrun, Editors
> Taft College, Taft, CA 93268

THE CLEARINGHOUSE FOR ACADEMIC SOFTWARE

OWNER'S GUIDE

(Revised February 4, 1986)

I. Introduction

The Clearinghouse for Academic Software is a not-for-profit
distribution center for higher education software and courseware
operated by Iowa State University and funded, in part, by a
grant from Digital Equipment Corporation.

The Clearinghouse was established in recognition of the need for
schools, universities, and colleges to share and benefit from the
innovative software developments at other institutions, to reduce
unprofitable duplication of effort, to have economical access to
existing software, and to encourage software developers to continue
creating high quality educational software and courseware.

The Clearinghouse inventory of software includes instructional,
research, administrative, and other software useful in education
and designed for use on Digital's VAX(tm) computer systems.  Software
is distributed to educational institutions at moderate prices and
royalties are paid to owners of software.

By providing software and courseware at moderate prices, the
Clearinghouse promotes the exchange of information, ideas, and
innovative educational tools among professionals in higher education.
And by rewarding developers with royalties, the Clearinghouse
recognizes their efforts and encourages the development of more and
better academic software and courseware.


II. Eligible Software

The Clearinghouse encourages owners to submit their software and
courseware packages to the Clearinghouse for distribution.  Any
software which meets the following descriptions is potentially
suitable for inclusion in the Clearinghouse catalog.

The software must:

   1.  be useful in the area of education.  Specifically,
       the software should be related to instruction, research,
       or administration, or to more than one of these areas.
       Software, such as utilities which enhance the operation
       of other software or hardware used in higher education would
       also be eligible.

   2.  operate in Digital's VAX(tm) computer environment.  Software
       packages for microcomputers, other mainframe computer systems,
       or other Digital hardware products are ineligible.

   3.  be saleable for a minimum price of $250.00.  Related software
       may be grouped and submitted as one package, when, if
       separately, they would not be saleable for the minimum price.

   4.  perform as described by the owner, have been tested and revised

sufficiently to reduce most programming bugs, and be documented with a user's manual. It is highly recommended that the software have been used by the owner or at the owner's institution for a period of time sufficient to establish its educational usefulness and freedom from errors and inefficiencies.

III.  Highlights of Contract Terms and Clearinghouse Policies

These highlights of contract terms and Clearinghouse policies are meant to briefly inform readers of the general nature of the relationships between software owners and the Clearinghouse. Please read the Software Owner's Agreement carefully to understand the full extent of the rights and responsibilities of both the owner and the Clearinghouse.

A.  Copyrights and Licenses

Owner - Ownership rights to the software remain with the owner of the package. The owner has the responsibility to complete any legal requirements necessary to secure those rights including in particular where the author of the software is not the owner.

It is suggested that owners clarify their rights and responsibilities to their institutions regarding the development and sale of software by discussing the Clearinghouse Software Owner's Contract with the appropriate administrator. Special arrangements for royalty distribution can be made (See Pricing and Royalties).

The owner decides whether source code for the package should be distributed to purchasers. The pro's and con's of this issue are discussed in Section V of this document.

The owner grants to the Clearinghouse a non-exclusive, royalty-free license to use, market, copy, have copied, and distribute the software package through sublicenses issued to purchasers. The owner also grants ISU a royalty-free license to use the software package with the same privileges and limitations of other purchasers of the software.

The owner grants Digital Equipment Corporation a royalty-free license to use and copy the software package for demonstration purposes.

Clearinghouse - The Clearinghouse will assure that the owner copyright notice on the software or documentaion provided to the Clearinghouse is included on all electronic and printed media which is distributed to purchasers.

Source code for the program will be used only internally by the Clearinghouse if the owner does not want it released to purchasers. The source code will be used by the Clearinghouse staff to help purchasers load the package on their own systems and to help the owner with revisions.

All purchasers of the software will be issued site licenses. For most purchasers that will be a single campus; a campus being defined as a single geographic entity. For other purchasers, the site would be defined by the specific

geographic location in which the software is loaded. The Clearinghouse reserves the right to define the site.

Purchasers will receive a single copies of the software package and all associated documentation. Source code will be distributed if it has been released by the owner. Purchasers will be allowed to make as many copies of the software and documentation as they need for use at that site only.

B. Performance and Support of the Software Package

Owner - The owner is responsible to the users for the performance of the software package. Questions from purchasers regarding the performance of a software package will be referred to the owner. It is highly recommended, therefore, that only carefully developed, thoroughly tested, and adequately documented software be submitted.

The importance of the good user documentation cannot be overstated; owners will reduce the number of user inquiries by providing users with a thorough user's guide and/or reference manual. Although it is not required, releasing source code to users may also reduce owner support by enabling users to answer more questions on their own.

Previous experience of owners has shown that users' questions are infrequent and generally concern three issues:
1. Users are having difficulty with the operation of the software.
2. Users disagree with the content of the software.
3. Users have recommendations for improvements in the software.

To prevent abuse of owner support, purchasers will be required to designate an individual who will be responsible for communicating any questions from that site to the owner.

Clearinghouse - Although the Clearinghouse will not guarantee that a particular software package will load on a given VAX(tm) configuration, the Clearinghouse staff will provide reasonable technical support to the purchasers of software in an attempt to work out any difficulties in loading the software. The extent of technical support provided to purchasers will be at the discretion of the Clearinghouse.

Purchasers will be expected to order only those packages for which they have the appropriate equipment; the Clearinghouse will not modify software packages to run on configurations other than those specified in the software description.

The Clearinghouse will allow purchasers a brief approval period during which they may test the software package. If they are unable to load the software or are unsatisfied with the package, they will be allowed to return it to the Clearinghouse for full credit.

C. Revisions

Owner – The owner will be expected to submit revisions or updates
   to the Clearinghouse in the following circumstances:

   a. The owner revises or updates the package for
      his/her own purposes.

   b. Digital Equipment Corporation issues a new
      version of the operating system, development tools,
      or hardware which makes the owner's software
      inoperable.

   c. The Clearinghouse determines that there are
      sufficient deficiencies in the current version
      of the software package to make it unmarketable.

In the case of the latter two situations, the owner would
be given 90 days to make the necessary revisions.  Software
packages deemed by the Clearinghouse to be unmarketable will
be excluded from distribution until the revisions have been
made and accepted.

Clearinghouse – The Clearinghouse will provide the owner with
   notice and, where possible, provide suggestions for any
   revisions required in the latter two situations.  The extent
   of such technical support will be at the discretion of the
   Clearinghouse.

The revised version will be tested and upon acceptance be
placed in the catalog for distribution.  Previous purchasers
of the software will be allowed to purchase a new copy
or instructions for revision for the cost of the tape plus
copying and shipping charges.  No additional purchase price
will be charged and no royalties will be paid to the owner
for revisions distributed to existing licensees of the
software.

D. Pricing and Royalties

Owner – The owner establishes the list price of the software
   within guidelines set by the Clearinghouse.  These guidelines
   and suggestions for setting prices are discussed in
   Section IV of the Owner's Guide.

Owners will receive 25% of the sale price for each of their
software packages.  Degree granting educational
institutions will be charged the list price; other purchasers
will be charged 200% of the list price.  In addition, all
purchasers will be charged a standard fee to cover the
direct costs of materials, copying, and postage; this
fee will not be subject to royalties.

Owners may designate that all royalties be paid to their
institution for appropriate distribution according to the
internal agreement between their institution and themselves.
Simply provide the institution's name, address, and tax

identification number on the Software Submittal Form.

Any arrangements for royalty distribution between the owner and parties other than their institution are the responsibility of the owner.

Clearinghouse – The Clearinghouse will accrue royalties throughout the calendar year.

Within 90 days of the close of the year, the Clearinghouse will issue checks for all royalties accrued along with reports of all sales. Any taxes or withholding required by state or federal government will be deducted.

E.  Promotion and Distribution

Owner – The owner's part in promotion is in writing an accurate description and providing adequate user documentation for a well-designed and well-written software package. This will contribute to sales directly from the Clearinghouse catalog and in sales resulting from word-of-mouth recommendations.

Owners are encouraged to promote their own software among professionals in their area and to refer potential users to the Clearinghouse for further information.

Clearinghouse – The Clearinghouse will work closely with Digital Equipment Corporation marketing to reach the potential users of the software. Clearinghouse announcements and catalogs of available software will be mailed to VAX(tm) education customers including professors, department heads, academic deans, and computer center personnel.

Clearinghouse announcements will also be made in professional journals and newsletters and at professional conferences. Articles focusing on particulary innovative and useful software will be submitted to EDU, Digital's education magazine, and to other relevant journals.

The Clearinghouse will also develop "interest lists" of potential purchasers of software in specific academic disciplines.

Both an online catalog and an online previewing system are planned for the future. The online catalog will allow potential purchasers to dial up and browse through the most current listings of available software. Where feasible, an online previewing system will permit purchasers to run and evaluate software prior to placing their order.

Reviews of software by users will be added to the catalog as they become available.

IV.  Suggestions for Setting Software Prices

A.  Guidelines

The purpose and spirit of the Clearinghouse is to encourage the

exchange of educational software at moderate prices. It is intended
to operate on a not-for-profit basis, using revenues to pay
royalties to software owners and to cover the expenses of
Clearinghouse operation. To accomplish these objectives, the
following guidelines and limitations have been established
regarding the pricing of software and the payment of royalties:

1. The minimum list price for any software package is $250.00.
2. The owner will receive royalties equal to 25% of the sale
   price of each unit of his/her software package sold.
3. Degree granting educational institutions will pay the list
   price for software; all other users will pay 200% of the
   list price.
4. The Clearinghouse reserves the right to increase the price
   established by the owner if it becomes necessary in order
   to generate sufficient revenue to operate the Clearinghouse.
5. All purchasers will receive site licenses to the software.

These terms are part of the Software Owner's Agreement entered into
by all contributors to The Clearinghouse.

B.  Suggested Procedure to Determine List Price

The following suggestions are presented to help owners determine
the selling price of their software. These suggestions are not
necessarily comprehensive and were derived to meet the very special
nature of the Clearinghouse.

Generally the price of software can be determined by considering:

1. Total costs to develop the software.

   -In a for-profit situation, the revenue generated by the
    software's sale should exceed all manpower, equipment/
    overhead, and marketing costs. In distributing software
    through the Clearinghouse, the owner should consider
    that only a portion of these costs may be recovered.

   -The features of the program itself combined with the
    owner's proficiency in programming academic software will
    determine the program's manpower cost. One means of
    assessing this factor is to estimate the total number of
    hours spent in coding, debugging, and documenting the
    program, and the estimated value per hour of the owner's
    time.

2. Total available market

   -Estimate the potential market based on the number of
    educational institutions which might find the software
    package useful and which also offer VAX(tm) based
    computing.

   -Professional academic societies sometimes have
    information about the number of schools offering courses
    or conducting reseach in particular disciplines. It can
    be assumed that a very large percentage of engineering
    and science students have access to VAX(tm) computers, as
    do a very high percentage of university students. Liberal
    arts students and students at smaller 4 and 2 year schools

are less likely to have access to VAX(tm) computers.

3. Competition from similar software or other materials.

   -Adjust the the potential sales volume figures to reflect
   the impact of competing products, both in software and
   hard copy form. In addition to considering similar
   software, consider other teaching methods such as hard
   copy text and course notes as these will limit the
   potential market for an educational software program.

   -Consider how long it would take for the program to be
   integrated into instructional, research or
   administrative activities. Very generally, programs
   which take a great deal of time to integrate sell fewer
   copies than those which can be integrated quickly.

4. Amount of reasonably expectable return.

   -Estimate the royalty necessary to breakeven for
   development costs by dividing total development costs
   by potential sales volume. Multiply this figure by four
   to obtain the estimated sales price.

   -Example:

   Manpower costs:        $20,000

   Potential sales:       100 packages

   $20,000 / 100 packages = $200/package royalty

   $200/package X 4 = $800 sales price per package

   -Look at this price realistically. Adjust the price of the
   program to reflect the price sensitivity of educational
   customers. Keep in mind that recovering all development
   costs is an unrealistic expectation unless those costs are
   fairly low.

The Clearinghouse may notify you if the price of your software
differs significantly from other similar software being distributed.

V. Distributing Source Code

The decision to distribute source code to purchasers is left to the
owner of the software package. Source code is required to be
submitted to the Clearinghouse for its own internal use.

There are several positive reasons why the source code should be
provided to purchasers. These all relate to support of the software
package. Because the owner is required to provide telephone
support to users these points may be very important.

   -User can customize the software package to meet own needs.
   -User can fix bugs without involving the owner.
   -User feels independent of owner; won't worry about owner

"disappearing" leaving the user with a software package that cannot be revised, debugged, or customized.

On the other hand, the owner may not want purchasers to have access to the source code to prevent inappropriate use of the ideas and concepts contained in the source code.

-User can make changes and present the work as his own.
-User can make changes for the worse and present the work as the original owner's.
-User would not find source code helpful due to its complexity.
-User could make changes that render the software inoperable and then ask for the owner's assistance.


VI.  Owner Documentation Guidelines

User documentation is required for all software packages submitted to the Clearinghouse.  The documentation is divided into three parts: the USER'S GUIDE and/or REFERENCE MANUAL, the SOURCE CODE, and the TAPE DIRECTORY.


A. USER'S GUIDE and/or REFERENCE MANUAL

The first part of the documentation, the USER's GUIDE or REFERENCE MANUAL, is very flexible in format.  It should however, contain all of the information needed by a purchaser to effectively use the software package.  This is what is commonly referred to as "documentation".  Included should be technical information, supplemental materials, special instructions required and, in the case of instructional programs, recommended instructional strategies.

The following are suggestions for topics that may be included in the USER'S GUIDE and/or REFERENCE MANUAL:

INSTRUCTIONAL USES - For instructional submissions suggest how the lesson should be integrated into the instructional process.

TECHNICAL DESCRIPTION - Identify formulas, principles and methodology underlying the software.  Where appropriate, provide interfacing information, error messages, definition of terms and functions, and other technical information needed to facilitate use of the software.

RECORD-KEEPING DETAILS - Provide specific information regarding student response records, lesson management information, and other types of data the software collects or maintains, and any reports that can be generated.

SUPPLEMENTARY MATERIALS -  Identify reference materials such as textbooks, articles and manuals to be used with this software. Worksheets, study guides and other materials to be distributed with the software should be appended.

Owners should add any additional information to the documentation that would help a user to more efficiently utilize the software.

B.  SOURCE CODE

Source code must be submitted to the Clearinghouse on the software tape. The code should be commented sufficiently to allow the reader to understand the flow to the code.

C. TAPE DIRECTORY

A directory of all files on the software tape should be submitted to the Clearinghouse in hard copy form.

VII. Checklist for submitting software to the Clearinghouse

All materials should be sent in a single package to:

> The Clearinghouse for Academic Software
> Computation Center
> 104 Computer Science Building
> Iowa State University
> Ames, Iowa
> 50011

The following materials must be received before the Clearinghouse will begin processing your submission.

_____ SOFTWARE OWNER'S AGREEMENT – Two signed and completed copies of the contract should be sent. One executed copy will be returned to you.

_____ SOFTWARE SUBMITTAL FORM – Send one completed copy.

_____ USER's GUIDE and/or REFERENCE MANUAL – Send three 8 1/2 X 11 copies of a quality suitable for photoduplication.

_____ TAPE DIRECTORY – Send one printed copy.

_____ SOFTWARE TAPE – The tape should include:

> _____ All necessary program files.

> _____ Source code.

Upon receipt of these materials, the Clearinghouse will begin processing your software submission. We will check for completeness of materials and load and test your program. You will be promptly notified of acceptance of your software or of any problems that would prevent the Clearinghouse from accepting your software.

# GRAPHICS

# GRAPHICS

**Chairman**
William Kramer
University of Delaware
Newark, DE

**Symposium Coordinator**
Bijoy Misra
Smithsonian Institution
Cambridge, MA

**Newsletter Editor**
Michael P. Anton
Houston, TX

**Session Note Editor**
Mike McPherson
Michigan State University
East Lansing, MI

**Standards Coordinator**
Jim Flatten
Ames Lab
Ames, IA

**Library Committee**
James M. Turner
Saber Technology
San Jose, CA

**DEC Counterpart**
Susan Usilton
Digital Equipment Corporation
Nashua, NH

**Information Officer**
Mike York
Boeing Computer Services
Seattle, WA

**Human Interface Working Group Coordinator**
Tom Owens
Graphics Research Center
Baltimore, MD

**Engineering Working Group Coordinator**
Dana Smith
Wilmington, DE

# The GAPSIG Steering Committee Meeting

by Mike York

The SIG Steering Committee met on February 24th & 25th in Westborough to do general planning for the SIG, and to meet with graphics developers at the DIGITAL Spitbrook and Maynard facilities. Committee members attending the meeting were:

**Bill Kramer** *SIG Chair*
**Jim Flatten** *Standards Coordinator*
**Bijoy Misra** *Symposia Coordinator*
**Mike McPherson** *Library and Session Handouts Coordinator*
**Michael Anton** *Newsletter Editor and Communications Representative*
**Richard McCurdy** *Assistant Symposia Coordinator*
**Mike York** *Information Officer*
**Rick Berzle** *DEC Counterpart (2/25 only)*

The first item on the agenda was to examine the particular topics that needed to be discussed during the course of the meeting:

- *Setup for Dallas*
- *Suite/Campgrounds*
- *Relations with Digital*
- *The SIG's alignment/non-alignment with Digital's workstation product line*
- *Newsletter*
- *SIG Working Groups*
- *Plan for meeting with Digital the next day*

Each member of the Steering Committee then gave a report on what was happening in their functional area:

Bill Kramer/Chair: Bill discussed the SIG budget. The major items in the budget are:

- *Participation in the Human Interface Task Force*
- *Sending a person to SIGGRAPH*
- *Steering Committe meeting*
- *Symposia suite*
- *Four pre-symposia seminars*

- *Session Notes*
- *Standard Coordinator's participation in ANSI X3H3 meetings*
- *Store items*

Bill then reviewed the SIG operating procedures.

Mike York/SIG Information Officer: Mike reported that the SIG survey/database project is at a standstill. There are too many questions about how to set up the database. It was decided that the survey should be printed and made available at Dallas, and possibly published in DECUSCOPE. Decisions on setting up the database will depend on the response.

Michael Anton/Newsletter Editor: Michael discussed problems with getting submissions for the newsletter. Various suggestions were made by other members of the Steering Committee. It was decided that some mini-tutorials might be useful. Mike York volunteered to submit a mini-tutorial on fractals.

Jim Flatten/Standards Coordinator: Jim discussed his participation with the ANSI X3H3 committee. His report will be published in the newsletter. Jim also discussed the possibility of designating an alternate to attend the meetings. Michael Anton was appointed the alternate. Jim is also coordinating the GAPSIG GKS working group, which is putting together a public domain GKS library written in C. Jim reported that a lot of interest has been generated, and a mechanism is being developed to distribute the source code to those who are interested in the project. Details will be published in the newsletter. *(Ed. Note: See April issue for Jim's report and information on the GKS project.)*

Mike McPherson/Library & Handouts: Mike

reported that the handouts for Dallas are right on schedule. He said that Digital submitted very little, but he saw no easy way to rectify the situation. Mike then put on his other hat and said that library efforts were not progressing very rapidly. Many suggestions were made with a decision to solicit volunteers to review current offerings in the DECUS library pertaining to graphics. Their reviews will be published in the newsletter.

Bijoy Misra/Symposia Coordinator: Bijoy reported his on-going ("never-ending" is the correct term) activties with the Symposia Committee. The number and quality of GAPSIG sponsored symposia sessions has increased significantly. The Steering Committee thanked Bijoy for all of his help – he has been instrumental in helping the SIG build momentum. There was much talk concerning the GAPSIG presence at Symposia. It was decided that the GAPSIG will have both a suite and a campground at the Dallas Symposia, with graphics equipment in both. Bijoy is getting the equipment lined up. It was also decided that *Graphics Hardcopy Systems* would be the theme for the GAPSIG in San Francisco.

Bill Kramer then reported that he changed jobs and anticipates being very busy the next few months. He may not be able to make it to the Dallas Symposium. The steering committee decided that a Vice Chair position should be created to help Bill with his duties. It was moved and seconded that Jim Flatten to be the Vice Chair and the motion carried unanimously.

After lunch, the topic was volunteers and working groups. It was decided that a list of objectives should be established for working groups. The suggestions were:

- *Define the Working Group's goal*
- *Maintain a bibliography*
- *Submit newsletter articles (Working Group*

*reports, articles of technical interest* o..
*book reviews)*
- *Report at Symposia on activities*
- *Review DECUS library submissions pertaining to the Working Group*

There are currently two active working groups in the GAPSIG: The GKS Working Group and the Human Interface Working Group, both of which started at the last symposia in Anaheim. The formation of two more working groups was suggested:

- *Conversion/Translation Working Group*
- *PHIGS Working Group*

Volunteers will be solicited for these groups in Dallas.

The problem of recruiting volunteers was discussed. In the past, there has been difficulty in following up on volunteers after symposia. The Committee decided to create the position of Volunteer Coordinator to track volunteers and match them with jobs. Richard McCurdy was selected to fill the postion.

The rest of the day was spent discussing the future of the GAPSIG. The major topic was Digital's emerging workstation product line, and how this affects the GAPSIG. It was decided that the SIG needs to re-examine its role in DECUS because of the workstations, but before we can proceed, we need a statement of direction from Digital and also need to resolve issues of overlap with other SIGs.

The second day was spent meeting with developers of various Digital graphics products, including GKS, UIS, workstation hardware and workstation applications. Much of what was discussed at these meetings is bound by a non-disclosure contract and cannot be commented on at this time. However, these meetings were significant since the GAPSIG Steering Committee as a group had the opportunity to discuss current and future issues with Digital developers involved in graphics. The

## GAPSIG (Con't)

Committee hopes that this two-way dialogue between DECUS GAPSIG members and the DEC developers will continue. We thank Rick Berzle, our DEC counterpart, for the time and effort he put in to setting up these meetings.

In summary, the GAPSIG Steering Committee Meeting was quite successful. The GAPSIG has come a long way in the last year and continues to gain momentum. While there are still problems, the Steering Committee is confident that the SIG is in a strong position to deal with them, and it will continue to grow as a valuable information resource for its members.

## Another Chance!

Many of you at the symposium in Anaheim last December heard Dick Puk deliver our first keynote address: *Graphics After Standardization.* Dick gave an excellent talk and a copy of his vu-graphs are on the following pages. If you missed the symposium, or would like to hear it again, the cassette tape is available from:

> Eastern Audio Associates, Inc.
> 6330 Howard Lane
> Elkridge, Maryland 21227
> (301) 796-0040

The cost is $7.50 plus $2.50 postage and handling. Ask for tape G001 from the Fall '85 DECUS Symposium.

## "Be a part of the picture..."

*Michael Anton*

At the Fall'85 Symposium Dick Puk seemed a bit surprised at the large turnout for his graphics keynote. He observed that as the graphics industry started to mature, it "lost some of the vibrancy it once had". I also remember Carl Machover commenting that when Tektronix introduced the storage tube in the late sixties, "everyone jumped on the graphics bandwagon". Consequently, this era is used to date the "graphics pioneers" in SIGGRAPH.

Though I haven't been around as long as Dick or Carl, or many of the others that you may meet at SIGGRAPH and NCGA, I will agree that the graphics industry "isn't what it use to be". I use to spend a lot of time explaining "why"; now I spend most of my time teaching "how". The change is refreshing. Just like in the film clasic *2001*, graphics is "evolving into something completely different".

One of the very rewarding activities in graphics is meeting and working with the people, like Dick and Carl, who made the industry what it is. I encourage you to not only get involved in the DECUS Graphics Applications SIG, but also join SIGGRAPH, the ACM Special Interest Group in Graphics, and NCGA, the National Computer Graphics Association.

Courtesy of Digital's GKS Development Group, I have a few NCGA applications that entitle you to join at half the regular membership fee ($12.50). This includes a monthly subscription to *Computer Graphics Today*. I'll also be happy to send you information on SIGGRAPH membership. Write to:

> Michael Anton
> P.O. Box 591293
> Houston, Texas 77259-1293

**GRAPHICS AFTER STANDARDIZATION**

by
DR. RICHARD F. PUK
PUK CONSULTING SERVICES

### *PCS*

### *FORTHCOMING STANDARDS*

- GKS-3D
- PHIGS
- LANGUAGE BINDINGS
- CGI
- PDES
- BASIC
- WINDOW MANAGEMENT

### *PCS*

**CURRENT STANDARDS**

- NAPLPS
- IGES
- GKS
- CGM

### *PCS*

### *WHEN WILL ALL THIS HAPPEN?*

ABOUT FIVE YEARS FROM NOW

(MAYBE LONGER)

### *PCS*

## PREDICTIONS
### (FURTHER AWAY)

- APPLICATION-SPECIFIC CHIPS
- STANDARD CHIP-TO-CHIP INTERFACES
- INTEGRATION OF ARTIFICIAL INTELLIGENCE WITH GRAPHICS SUBSYSTEMS

*PCS*

## WHAT DOES THIS MEAN TO GRAPHICS INDUSTRY

- MANY ADVANCES IN GRAPHICS INDUSTRY WILL BE APPLICATION-DIRECTED BUT BUILT ON MODELS DERIVED FROM STANDARDS

*PCS*

## WHAT DOES THIS MEAN TO GRAPHICS INDUSTRY

- NEW ALGORITHMS WILL BE DIRECTED TOWARDS VLSI IMPLEMENTATION
- GRAPHICS USAGE WILL BECOME MORE OBJECT-ORIENTED

*PCS*

## WHAT DOES THIS MEAN TO GRAPHICS INDUSTRY

- GRAPHICS PROCESSING WILL BECOME MORE DISTRIBUTED

*PCS*

## WHAT TAKES IT SO LONG?

- OBTAINING CONSENSUS
- DEVELOPMENT WITHIN INTERNATIONAL ARENA
- VOLUNTARY PARTICIPATION
- RULES AND REGULATIONS

*PCS*

## WHAT HAPPENS AFTERWARDS?

- REVISIONS
- FOLLOW-ON STANDARDS
- RESCISSION

*PCS*

## INDUSTRY DIRECTIONS

- PERSONAL INTELLIGENCE
- MORE MEMORY
- FASTER MICROCOMPUTERS
- ROM-BASED SOFTWARE
- WINDOW-BASED USER INTERFACES
- COLOR RASTER GRAPHICS DISPLAYS

*PCS*

## PREDICTIONS
### (NOT SO FAR AWAY)

- GRAPHICS SYSTEMS ON A CHIP
- REAL-IMAGE INTEGRATION
- FLOATING POINT GRAPHICS
- ADAPTABLE ENVIRONMENTS
- HARDWARE HIDDEN-OBJECT REMOVAL

*PCS*

## FUTURE STANDARDS

- APPLICATION-ORIENTED
- GRAPHICS CHIP PIN-OUT
- GRAPHICS COMMUNICATION PROTOCOLS

## RECOMMENDATIONS

- BECOME KNOWLEDGEABLE ABOUT GRAPHICS STANDARDS
- SUPPORT YOUR STANDARDS REPRESENTATIVE
- LISTEN TO YOUR STANDARDS REPRESENTATIVE

*PCS*

*PCS*

## PROBLEM AREAS

- GRAPHICS DATA BASE MANAGEMENT
- PARALLEL AND DISTRIBUTED PROCESSING OF GRAPHICS DATA
- GRAPHICS DATA ORGANIZATION
- NEEDS OF APPLICATIONS USING AI TECHNOLOGY

## RECOMMENDATIONS

- BUILD NEW GRAPHICAL CAPABILITIES WITHIN THE GRAPHICS STANDARDS FRAMEWORK

*PCS*

*PCS*

# Human Interfaces
by
Dottie J. Elliott

Once upon a time, computers required specially equipped rooms and large facilities for operations. Few people were allowed inside these shrines to computational power. Those that programmed did so on punch cards that were submitted to a system operator. In return, sometime later they received a paper printout of the program's results or a core dump of their crash. Programmer's knew intimately of the inner workings of the computer. Assembler programming was common practice. The users ( i.e programmers) had extensive training and experience in programming, using, and understanding the computer system. It took a good deal of KNOWLEDGE and TRAINING to use the computer.

Today, a person walks off the street, inserts a plastic card into an automatic teller machine, presses a few buttons, gets some cash and walks away. He's just used a computer. Computer knowledge is not required. Training is not necessary. Computers are found on desktops of many non-computer professionals. They are found in primary school classes and in colleges in the English department. You can actually TOUCH a computer and interact with it. Programs are entered directly, compiled, tested and debugged immediately. Many programmers now only have a rudimentary knowledge of how the computer works. Programs are written in high-level languages. Good documentation receives as much emphasis as the actual programming.

Today, computers are used 'to get the job done'. In our fast-paced, time-constrained society, users want to spend most of their time doing productive work and very little time training on the computer and its software. The computer's internal workings are becoming increasingly complex while its user interface is decreasing in complexity. Today's watchwords are 'user-friendly', 'menus', 'windowing', and 'pointing devices'. Everyone wants access to the computer: restaurants, teachers, students, musicians, movie makers, writers, grocery stores, researchers, and accountants.

As the need for user-friendly interfaces has increased, an area of research has grown to study the interaction of humans and computers. The human-computer interface has been defined as 'everything that the user sees, touches, or otherwise senses and interacts with in a computer system' [Rubenstein and Hersh 1984]. As researchers, users, and programmers, we are asking the question: 'How can the interaction between the user and the

computer be improved? This important area of research will define the shape of computer systems and software in the future.

As computer professionals, we must have input into these designs. We will have to use these systems and train users on them. Systems that decrease training time and increase productivity are important and will have a great impact on our computer centers. The **Human Interface Working Group** has been formed to provide information and make recommendations in this area It is part of the Graphics Sig since the best interfaces will involve visual as well as written information. We hope to gather information on what types of interfaces are best. Is it the light pen, a mouse, or a touch screen? Are menus really useful or do they hamper the productivity of experienced users? Is windowing important and should it be multi-tasking? How will color play a role? We hope to influence the design of workstations and interfaces within DEC. We will keep members informed on ANSI standards being discussed in this area. We also hope to build an expert contact and referral service for members to use when trying to implement interface designs into their software and hardware systems.

As Coordinator, I am now in the process of establishing a member base and building an annotated bibliography of work being done in human interfaces. Each month I hope to have an article in this newsletter discussing important issues in human interfaces. At the Spring DECUS, I will present a session defining various human interface tools and how they can be implemented and used effectively. If you would like to be part of this working group, please contact me. We can and should have influence into the hardware and software designs of the future.

Dottie J. Elliott
Northrop Services, Inc.
P.O. Box 12313
Research Triangle Park, N.C. 27709
919/ 541-2541


References:

[Rubinstein and Hersh 1984]
Richard Rubinstein and Harry Hersh
*The Human Factor: Designing Computer Systems for People*
Burlington, MA: Digital Press, 1984.

# H M S

DECUS

# HMS

**Chairman VAX SIG Liaison**
Thomas J. Provost
MIT/LNS Bates Linac Facility
Middletown, MA

**Product Planning Coordinator**
George Hamma
Synergistic Technology
Cupertino, CA

**Symposium Seminar Coordinator**
Mike Allen
Lawrence Livermore National Labs
Livermore, CA

**Communications Coordinator**
John G. Hayes
Information Systems - S. Central Bell
Birmingham, AL

**Publications Coordinator (Editor)**
Bill K. Walker
Monsanto Research Corp.
Miamisburg, OH

**Session Notes**
**DAARC SIG Liaison**
Bill Tippie
Kinetic Systems Corp.
Lockport, IL

**Standards Coordinator**
**CAMAC Working Group Coordinator**
Peter Clout
Los Alamos National Lab
Los Alamos, NM

**LUG Coordinator**
Gregg Giesler
Los Alamos Science Lab
Los Alamos, NM

**Pre-Symposium Seminar Coordinator**
Mike Allen
Lawrence Livermore National Labs
Livermore, CA

**TOEM (Chips % Boards)**
Jack J. Peterson
Horizon Data Systems
Richmond, VA

**HHK (Hardware Hints & Kinks)**
Wayne Kesling
Monsanto Research Corp.
Miamisburg, OH

**UNIBUS Hardware**
Ron Bogue
LIV Aerospace & Defense Co.
Dallas, TX

**Performance Measurement Coordinator**
William Wallace
600 W. Washington St.
Peoria, IL

**CAMAC Coordinator**
Peter Clout
Los Alamos National Lab
Los Alamos, NM

**CSS Coordinator**
Pratap Gohel
E.I. Dupont
Ingleside, TX

**Networks SIG Liaison**
Sandra Traylor
Target Systems
Yorba Linda, CA

**VAX SIG Liaison**
Dave Schmidt
5100 Centre Avenue
Pittsburgh, PA

**DAARC SIG Liaison**
Bill Tippie
Kinetic Systems Corp.
Lockport, IL

**UNISIG SIG Liaison**
Jim Livingston
1 Results Way
Cupertino, CA

**SITE SIG Liaison**
Emily Kitchen
A.H. Robbins Co.
Richmond, VA

**RT-11 SIG Liaison**
Gary Sallee
Sallee Software Consulting
Yorba Linda, CA

**RSX SIG Liaison**
Hans Jung
Associated Press
New York, NY

**Members-At-Large**
Mike Rembis
American Dade
Costa Mesa, CA

Hans Dahlke
Richland, WA

Jim Cutler
EDS Tower, 26533 Evergreen
Southfield, MI

**DEC Counterparts**
Terminals
Nina Abramson
Digital Equipment Corporation
Maynard, MA

TOEM (Chips & Boards)
Art Bigler
Digital Equipment Corporation
Marlboro, MA

Diagnostic
George D. Cooke
Digital Equipment Corporation
Maynard, MA

Storage
Marilyn Fedele
Digital Equipment Corporation
Maynard, MA

MSD (Micro Systems Development)
Roy Rodgers
Digital Equipment Corporation
Maynard, MA

Printer Products
Frank Orlando
Digital Equipment Corporation
Maynard, MA

DECUS Europe Liaison
Hans Zoller

# THE

# DeVIAS

# LETTER

# IAS SIG Steering Committee

Chairman
  Bob Curley
  Division of Medical Physics
  University of Pennsylvania
  Philadelphia, PA

WHIMS Commissioner
  Kathleen Anderson
  Eaton Information Management
      Systems Division
  Hampton, VA

Library Coordinator
  Bob Schuldt
  INCO Inc.
  McLean, VA

RSX Liaison
  Ray French
  Boeing Computer Services
  Seattle, WA

Member-at-Large
  Doug Reno
  Abbott Laboratories
  North Chicago, IL

DEC Counterpart
  Mike Reilly
  Digital Equipment Corp.
  Maynard, MA

Symposium Coordinator
  Skip Stanfield
  USAF
  Washington, DC

Librarian
  Mike Robitaille
  Grumman - CTEC, Inc.
  McLean, VA

DeVIAS Letter Editor
  John Roman
  McDonnell Douglas - Dept. N436
  600 McDonnell Blvd.
  Hazelwood, MO   63042

Member-at-Large
  Kerry Wyckoff
  LDS Church
  Salt Lake City, UT

DEC Counterpart
  Tim Leisman
  Digital Equipment Corp.
  Stow, MA

DEC Counterpart
  Bob Mack
  Digital Equipment Corp.
  Landover, MD

# Leverage
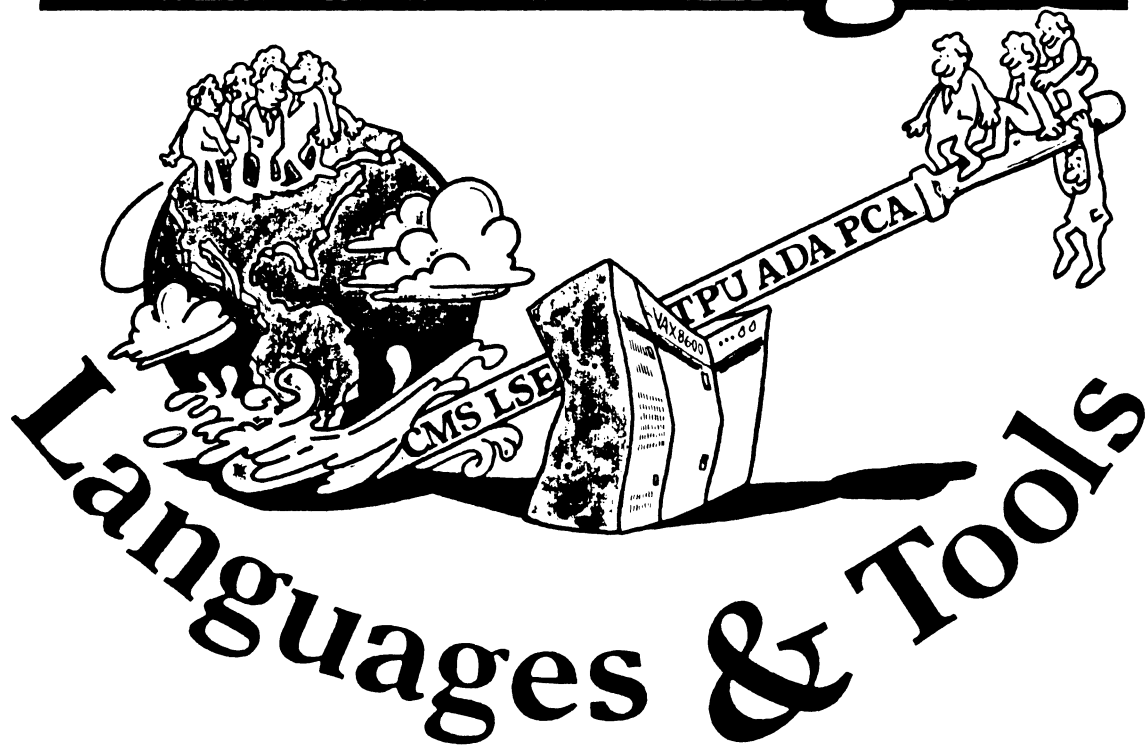


Languages & Tools

# STEERING COMMITTEE MEMBERS
# LANGUAGES AND TOOLS SIG

Susan Abercrombie
Ventrex Laboratories
217 Read Street
Portland, ME 04103

**Library Rep**
(207) 773-7231

Bob Awde
General Mills
9000 Plymouth Ave N
Minneapolis, MN 55427

**Steering Committee**
(612) 540-4432

Mark Bartelt
IISC - Research Development Ctr
555 University Avenue
Toronto, Ontario, Canada M5G 1X8

**UNISIG Interface**
(416)598-5955

Doug Bohrer
Bohrer & Co.
903 Ridge Rd. Suite 3
Wilmette, IL 60091

**Publications Committee**
(312) 251-9449

Gordon Brimble
Bldg 180 Labs Area
Defence Research Centre
Box 2151 GPO
Adelaide, S.A. Australia 5001

**Australian L&T Interface**
(61)(8)259-6119 (office)

Barb Chase
Hughes Aircraft
P O Box 92426
Bldg R1 MSC327
Los Angeles, CA 90009

**Human Interfaces Coordinator**
(213) 606-1601 (work)

Earl Cory
Cory Computer Systems
366 North Nueve Court
Camarillo, CA 93010

**Symposium Coordinator**
(818) 706-5385 (work)

Jack Davis
Philips Home Interactive Systems
1111 North Shore Drive
Knoxville, TN 37919

**Modula II Coordinator**
(615) 558-5206 (work)
(615) 588-5800 (switchbd)

Celeste LaRock
110 Spit Brook Rd ZK02-3/Q08
Nashua, NH 03062

**DEC Counterpart**

Jim Livingston
Measurex Corporation
1 Results Way
Cupertino, CA 95014

**Past Chair**
(408) 255-1500 X4468

Dave Martin
Hughes Aircraft Company
PO Box 92426
Bldg R1, MS C320
Los Angeles, CA 90009

**Tape Librarian**
**STUG Interface**
(213) 648-9927

Shava Nerad
Systems Alternatives
43 State St
Montpelier, VT 05602

**Steering Committee**
(802) 229-0823

Al Rizzuto
EMC Control, Inc
PO Box 242
Cockeysville, MD 21030

**Wishlist Coordinator**
(301) 628-8167 (work)
(717) 456-5014 (recorder)

Don Rosenthal
Space Telescope Science Institute
Homewood Campus
Baltimore, MD 21218

**LISP/AI Coordinator**
(301) 338-4844 (work)

Tony Scandora
Argonne National Laboratory
CMT 205
Argonne, Illinois 60439

**RSX Interface**
(312) 972-7541

Bill Segal
Digital Equipment Corp.
110 Spit Brook Rd.
ZK01
Nashua, NH 03062

**Counterpart Emeritus**
(603) 881-1263

Jim Flatten
Ames Lab
304 Metallurgy
Ames, IA 50001

GAPSIG Interface
(515) 294-4823 (work)

Alan Folsom, Jr.
Fischer & Porter Co.
E. County Line Rd.
Warminster, PA 18974

Newsletter Editor
(215) 674-7154 (work)

Bob Gable
Lear Siegler, Instrument Division
4141 Eastern SE MS 121
Grand Rapids, MI 49508

Ada Coordinator
(616) 241-8273

Dorothy Geiger
Wollongong Logistics Group
49 Showers Drive # 451
Mountain View, CA 94040

Intersig Coordinator
(415) 962-7160

Bernd Gliss
Max-Planck-Institute
Heisenbergstraße 1
7000 Stuttgart 80, W. Germany

European Methods, L&T Interface
(711) 686-0251 (work)

Keith Hare
JCC
PO Box 381
128 West Broadway
Granville, Ohio 43023

DMS & DTR Liaison
(614) 587-0157 (work)

Howard Holcombe
RCA
Front & Cooper St.
Camden, NJ 08055

DEC Personnel Coordinator
(609) 338-4946 (work)

Kathy Hornbach
Lear Siegler/Instrument Division
4141 Eastern SE MS 121
Grand Rapids, MI 49508

Chair
Productivity Tools Coordinator
Pre-Symposium Seminar
    Coordinator
(616) 241-8800

Mark Katz
GTE Govt Systems
100 First Ave.
Waltham, MA 02154

Session Notes Editor
(617) 466-3437

Kathy Tamer
Rockwell International
1840 Nasa Rd./MS ZC01
Houston TX 77058

Ada Packages Project
(713) 333-0827 (work)

Pat VanMunn
Measurex Inc.
One Results Way
Cupertino, CA 95014

Methods Coordinator
PSS Committee Representative
(408) 255-1500

Jay Wiley
Bechtel Power Corp
12400 East Imperial Highway
Norwalk, CA 90650

Standards Coordinator
Fortran Coordinator
(213) 807-4016 (work)

JR Westmoreland
Custom Software Products
6748 Acoma Rd
Midvale, UT 84047

Assistant to the Chair
C Coordinator
Commercialism Task Force
TEX/LaTEX Coordinator
(801) 535-4784 (work)

Melodee Westmoreland
Custom Software Products
6748 Acoma Rd
Midvale, UT 84047

Recording Secretary
(801) 533-2350 (work)

Sam Whidden
American Mathematical Society
201 Charles St
PO Box 6248
Providence, RI 02940

36 bit Coordinator
Store Liaison
(401) 272-9500 (work)

Ed Whipple
Lawrence Berkeley Labs
University of California
Berkeley, CA 94720

CMS/MMS Coordinator
Session Chair Coordinator
(415) 486-7167 (work)

Louise Wholey
Measurex Corp
One Results Way
Cupertino, CA 95014

VMS Interface
(408) 255-1500 X4452 (work)

Jim Wilson
QZ Division
PO Box 88
Terre Haute, IN 47808

Commercial Languages Interface
(812) 299-2121 X271 (work)

## Table of Contents

## Editor's Notes

As I write this, it is Easter Sunday, the only day I can find to
work on the newsletter. As usual, the office here is in an
uproar as we rush to meet a deadline, we are regularly getting
panicky calls from other groups within the company, and I'm still
not sure if I will or will not be able to make the Dallas Sympo-
sium. Ah, the joys of this industry!

Acutally, things may be looking up. The upper level management
is convinced we need to modernize our software development facil-
ity, and, for what may be the first time, we have actually in-
cluded in the budget allocations for software tools. I'm sure
many of you have gone through this process also, perhaps you can
write and tell me how you managed to justify (or not) expendi-
tures in this area.

This issue of "Leverage" has a number of articles which should be
of interest. I would like to apologize to Nabajyoti Barkakati,
the author of one of those articles, because his submission seems
to have gotten lost in the shuffle over the last few months.
Sorry I couldn't get it published earlier.

Also, Richard K. Wallace, of Los Alamos National Labs, has sub-
mitted a very nice article on Choosing a Document Formatting Sys-
tem. I am very grateful to both Mr. Wallace and the U.S. Depart-
ment of Energy, for the submission. It is very gratifying to re-
ceive an article of this nature, especially when it comes unsoli-
cited.

Finally, I have a couple submissions from DEC. These include the
release status of a variety of DEC products of interest, and a
series of notes from a presentation entitled "What will Your
Favorite Language Resemble on 1990?". This is a good concise
synopsis of current standards activities.

As ususal, I'm going to end with a plea for submissions. Please
considered it pled (pleaded?). If you can't or won't make sub-
missions, please let me know what you would like to see. Some-
times I feel like I'm sending newsletters to the DECUS office,
and they're deep-sixing them. I know that's not true though,
since eventually I get about ten copies myself. This is a
volunteer, user oriented society, despite the attempts of recent
years to make it more slick and "professional". Without your
help, these newsletters will be of no good to anyone.

With all that said, I hope you all have a glorious spring. If the
weather here is any forecast, we're in for some good days ahead.
If you made it to Dallas, I hope I met you there. If you did and
I didn't, please send me your notes!

SUPPORTING A VARIETY OF TERMINALS WITHIN YOUR FORTRAN 77 APPLICATIONS

By Nabajyoti Barkakati*

Most FORTRAN programmers would welcome the ability to handle alphanumeric and graphics displays for any arbitrary terminal within their applications. This is most appealing when the I/O involves functions such as clearing the screen, positioning the cursor, displaying a message in reverse-video etc. -- in short, in any application requiring effective man/machine communications.

In the Berkely Unix environment the database facility called "termcap" provides this capability by storing the special programming sequences for various terminals in a text file. As it stands now, termcap does not support graphics displays fully.

The FORTRAN 77 programs presented here show how you can have your own termcap (we call it TERM.DEF) file and provide alphanumeric as well as graphic display support for various terminals.

### THE TERM.DEF FILE

Display terminals can be programmed to perform tasks such as: clear the screen, position the cursor at a specified location, turn on or off video attributes (e.g., reverse-video, bold, underline, blink) etc. Each of these actions is performed by sending a sequence of special ASCII characters called

* Nabajyoti Barkakati is a Senior Engineer with Systems Engineering, Inc.,

7833 Walker Drive, Suite 308, Greenbelt, Maryland 20770.

escape sequences to the terminal. For some terminals, the escape sequences conform to the standards set by ANSI, while for many others they are specific to each manufacturer. So it is quite troublesome to provide support for a variety of terminals. We get around this problem by storing these sequences for all terminals of interest in the TERM.DEF file. The unprintable characters are stored in a coded form, e.g., the ESCAPE key is represented as "\E". This allows us to prepare the file with the system editor (which in many cases may not permit us to enter unprintable characters in a file). In the application program the user enters the terminal name and the program reads in the appropriate sequences into a special internal data structure. Usually we need three types of variables for this: integer, logical and character string variables. The integers store such values as maximum number of lines and columns, resolution in graphics mode etc. The logical variables tell us whether the terminal has graphics capabilities, or whether it can do certain things, e.g., display in bold or reverse video, etc. The actual escape sequences are stored in the character strings. When integer parameters needed by a sequence (e.g., line number and column number for cursor positioning), the position where they should appear are marked by a \? in the TERM.DEF file. This is translated to the the ASCII NUL (or 0) character in the internal data. Routines that read these strings fill in the appropriate parameter values.

A sample entry: Here is an example showing the entry for a VISUAL 550 graphics terminal in the TERM.DEF file. We use the Visual 550 because it works like a VT100 in the alphanumeric mode, and it also has graphics capabilities. Comments are preceded by a # followed by a space.

```
ALIAS VISUAL V550 VIS550          # Stores the various names by which this
#                                   terminal is called.
DESCR 'VISUAL 550 GRAPHICS TERMINAL' # Descriptive name
LINES 33 COLUMN 80  SCREEN 2      # Maximum number of lines and columns
#                                   This terminal has 2 screens.
#
#  Next comes the cursor movement sequences
#      \E means ESCAPE
#      \? means a parameter
#
  CURUP '\E[\?A'  CURDN '\E[\?B'  CURRT '\E[\?C'  CURLFT '\E[\?D'
# Cursor up       cursor down     cursor right    cursor left
#
   CURPOS '\E[\?;\?H'   SETSCR '\E[\?;\?r'    CLRSCR '\E[2J'
# Cursor  positioning   Set scrolling region   Clear the screen
#
BOLD UNDRLN BLINK REVERS    # Logical variables that tell us whether
#                             terminal can have display in bold, underline,
#                             blink or reverse video.
#
  setnul '\E[Om'      SETBOL '\E[1m'    setund '\E[4m'
# video attributes off  turn on bold    turn on underlining
#
  setbli '\E[5m'   setrev '\E[7m'      resett '\Ec'
# turn on blink    turn on reverse-video   reset the terminal
#
GRAPHX  MXSCRX 767  MXSCRY 584    # terminal has graphics capabilities
#                                   with resolution 767x584 dots
GRERAS '\E\F'   # erase graphics
BACKAN '\C'     # back to alphanumeric mode
YFIRST          # y-coordinate goes first
GRHEAD '\G'     # precede vector plot sequences with \G = 29 (Decimal)
GRTAIL '\U'     # end plot sequences with \U = 31 (Decimal)

HIDEV 32        # to prepare coordinates divide by 32 (decimal)
#                 to get the high-order bits
MASK 31         # bit-wise AND with 31 (decimal) to get low-order
#                 bits
HIYOR 32        # high-order y-coord is bit-wise OR'ed with 32 (decimal)
LOYOR 96        # low-order y-coord is OR'ed with 96.
HIXOR 32        # corresponding sequences for high-order bits
LOXOR 64        # and low-order bits of the x-coordinate.

END             # End of definitions for this entry.
```

Note that tabs or blanks separate the entries in the file. Each attribute name is upto 6 characters long and the character strings are enclosed in single quotes. Unlike the Unix termcap file, our TERM.DEF file is easy to understand and update. Also, we are only demonstrating our idea with a file having relatively few generic entries; if your terminal has many more capabilities you could add to this list with attribute names of your own choosing.

## USING THE TERMINAL CHARACTERISTICS

We have set up the internal data structure in such a way that in the programs when we have to refer to, say, the maximum number of lines for the current terminal we would write IVALU(LINES), similarly, we can turn on reverse video by writing to the terminal the string CVALU(I) of length CLEN(SETREV) with the index I taking values starting at CSTRT(SETREV). That is, if you want to access a character string parameter you do so by using the name CVALU(attribute_name) where attribute_name is its six character name in the TERM.DEF file. A better way to use this information is to write subroutines to perform specific tasks, e.g., preparing a string with various video attributes (see MAKSTR below) clearing the screen (see subroutine ERASA), etc.

Although we are not going to illustrate it here you could also write your own generic plotting routines that would work on a variety of terminals. The program listings appear below. A sample TERM.DEF file is also included. The author welcomes comments about implementing the programs. Before executing the program, remember to define TERM$DEF as the directory where your TERM.DEF file resides.

```
C*******************************************************************************
C
C         This program illustrates the use of information from the TERM.DEF
C         file to manage screen display.
C
C*******************************************************************************
          CHARACTER * 6 TNAME, BTNAME(6)*1
          CHARACTER * 80  PBUF(80)*1
          EQUIVALENCE (TNAME,BTNAME(1))
          INCLUDE 'TRMCOM.FOR'

          WRITE(IOUT,6000)
 6000     FORMAT(' ENTER 6 CHARACTER NAME FOR YOUR TERMINAL:',$)
    1     READ(IIN,5000,ERR=1)TNAME
 5000     FORMAT(A6)

          IF(TNAME.EQ.'      ')GO TO 1
          CALL SETTRM(TNAME,IERR)
          IF(IERR.NE.0)STOP'*** ERROR IN TERMINAL SETUP ***'
C
C  CLEAR THE SCREEN
C
          CALL ERASA
C
C  AT LINE=10 COLUMN=20 DISPLAY THE TERMINAL NAME IN REVERSE VIDEO.
C
          NHERE=0
C  POSITION CURSOR
          CALL MAKSTR(CURPOS,10,20,I3,I4,I5,PBUF,NHERE)
C  TURN ON REVERSE-VIDEO
          CALL MAKSTR(SETREV,I1,I2,I3,I4,I5,PBUF,NHERE)
C  PUT TERMINAL NAME IN THE BUFFER AND UPDATE BUFFER LENGTH
          DO I=1,6
            NHERE=NHERE+I
            PBUF(NHERE)=BTNAME(I)
          END DO
C  TURN OFF REVERSE-VIDEO
          CALL MAKSTR(SETNUL,I1,I2,I3,I4,I5,PBUF,NHERE)
C  OUTPUT THE STRING TO THE TERMINAL
          CALL PUTSTR(PBUF,NHERE)
          END
```

```
C*******************************************************************************
C      SUBROUTINE: MAKSTR
C
C      Puts terminal control string into a buffer.
C      Uses parameter values where necessary.
C
C      Inputs:  STRID         -- Integer attribute ID (same name as the
C                                 one used in the TERM.DEF file
C               PAR1 thru PAR5 -- Integer parameters that may be
C                                 required by some attributes.
C               PBUF          -- Character * 1 array that holds the
C                                 string being prepared.
C               NHERE         -- Current length of the string
C
C      Output: PBUF with updated NHERE.
C
C*******************************************************************************
          SUBROUTINE MAKSTR(STRID,PAR1,PAR2,PAR3,PAR4,PAR5,PBUF,NHERE)
          INTEGER STRID,P1,P2,P3,
     *    PARPUT,P(5),PAR1,PAR2,PAR3,PAR4,PAR5
C
C  PARAMETERS PAR1,..., PAR5 ARE IGNORED IF NOT NEEDED
C
          CHARACTER * 1 PBUF(80)
          INCLUDE 'TRMCOM.FOR'
C
C THE NUMBER OF PARAMETERS NEEDED BY THIS ATTRIBUTE IS STORED
C IN THE INTEGER VARIABLE "CPARAM(STRID)"
C
          IF(CPARAM(STRID).NE.0)GO TO 200
C
C  NO PARAMETER CASE
C
          DO 100 I=NHERE,NHERE+CLEN(STRID)-1
          J=CSTRT(STRID)+I-NHERE
  100     PBUF(I+1)=CVALU(J)
          NHERE=NHERE+CLEN(STRID)
          RETURN


  200     CONTINUE
C
C  NEED PARAMETERS
C
          P(1)=PAR1
          P(2)=PAR2
          P(3)=PAR3
          P(4)=PAR4
          P(5)=PAR5
          I=NHERE
          J=CSTRT(STRID)-1
          JLAST=J+CLEN(STRID)
          PARPUT=0
```

```
       1   I=I+1
           J=J+1
           NHERE=NHERE+1

           IF(CVALU(J).EQ.NUL)THEN
C
C  WE'LL PUT A PARAMETER HERE
C
              PARPUT=PARPUT+1
              IF(PARPUT.GT.5)THEN
                I=I-1
                NHERE=NHERE-1
                GO TO 201
              ENDIF

              IF(.NOT.LVALU(CODELC))THEN
C LINE AND COLUMN NUMBERS ARE DIGITS IN ASCII
C GET THE DIGITS FOR THE PARAMETER VALUE
                CALL CONV(P(PARPUT),P1,P2,P3)
                IF(P3.EQ.0)THEN
                   IF(P2.EQ.0)THEN
                     PBUF(I)=ALPHA(P1+1)
                     GO TO 201
                   ENDIF
                   IF(P2.NE.0)THEN
                     PBUF(I)=ALPHA(P2+1)
                     I=I+1
                     NHERE=NHERE+1
                     PBUF(I)=ALPHA(P1+1)
                     GO TO 201
                   ENDIF
                ELSE
                   PBUF(I)=ALPHA(P3+1)
                   I=I+1
                   NHERE=NHERE+1
                   PBUF(I)=ALPHA(P2+1)
                   I=I+1
                   NHERE=NHERE+1
                   PBUF(I)=ALPHA(P1+1)
                   GO TO 201
                ENDIF
              ELSE
C  LINE AND COLUMN NUMBERS ARE IN CODED FORM
                PBUF(I)=CHAR(P(PARPUT)+IVALU(OFFSET))
                I=I+1
                NHERE=NHERE+1
                GO TO 201
              ENDIF
      201     IF(J.LT.JLAST)GO TO 1
              RETURN
           ENDIF
C
C  NO PARAMETER -- STRAIGHT COPY
```

```
C
           PBUF(I)=CVALU(J)
           IF(J.LT.JLAST)GO TO 1

           RETURN
           END
```

```
C*****************************************************************
C       SUBROUTINE: ERASA
C
C       Erase alphanumeric screen
C
C
C*****************************************************************
        SUBROUTINE ERASA
        CHARACTER * 1 PBUF(80)
        INTEGER STRID
        INCLUDE 'TRMCOM.FOR'
C
C  CONSTRUCT STRING TO BE OUTPUT
C
        NHERE=O
        CALL MAKSTR(CLRSCR,I1,I2,I3,I4,I5,PBUF,NHERE)
C
C  SEND STRING TO TERMINAL
C
        CALL PUTSTR(PBUF,NHERE)
        RETURN
        END
```

```
C*****************************************************************
C       SUBROUTINE: PUTSTR
C
C       Displays a string of characters at the terminal.
C
C
C*****************************************************************
        SUBROUTINE PUTSTR(STR,LEN)
        INCLUDE 'TRMCOM.FOR'
        CHARACTER * 1 STR(LEN)
C*****************************************************************
        WRITE(IOUT,6000)(STR(I),I=1,LEN)
6000    FORMAT(1X,80A1,$)
        RETURN
        END
```

```
C**********************************************************************
C       SUBROUTINE: CONV
C
C       Separates the integer INTIN into its decimal digits. Assumes that
C       0 < INITIN < 999.
C
C**********************************************************************
        SUBROUTINE CONV(INTIN,DIGIT1,DIGIT2,DIGIT3)
        INTEGER INTIN,DIGIT1,DIGIT2,DIGIT3
C**********************************************************************
        DIGIT3=INTIN/100
        INTIN=INTIN-100*DIGIT3
        DIGIT2=INTIN/10
        DIGIT1=INTIN-10*DIGIT2
        RETURN
        END
```

```
C       SUBROUTINE: SETTRM
C
C       This subroutine reads the terminal characteristics from a
C       file named "TERM$DEF:TERM.DEF"
C
        SUBROUTINE SETTRM(TNAME,IERR)

        INCLUDE 'TRMCOM.FOR'

        INTEGER TUNIT
        CHARACTER * 6 TNAME,TMPNM,BTMPNM(6)*1
        EQUIVALENCE (TMPNM,BTMPNM(1))
        CHARACTER *10 TYP
        CHARACTER *40 TFILE
        DATA TUNIT/1/
        DATA TFILE,TYP/'TERM$DEF:TERM.DEF','OLD'/
C
C INITIALIZE EVERYTHING
C
        CALL INITRM
        IERR=0
C
C CONVERT TERMINAL NAME TO UPPERCASE
C
        TMPNM(1:6)=' '
        TMPNM(1:6)=TNAME(1:6)
        CALL UCASE(BTMPNM,6)
        TNAME(1:6)=TMPNM(1:6)
C
C       OPEN THE TERMINAL DEFINITION FILE
C       (OPEN FOR READONLY)
C
        OPEN(UNIT=TUNIT,FILE=TFILE,STATUS=TYP,ERR=999,
     *       READONLY)
C
C LOCATE THE ENTRY FOR THIS TERMINAL
C
        CALL TLOCAT(TUNIT,TNAME)
        IF(TERR.EQ.3)THEN
        CALL TRMERR(TERR,ERUNIT)
          TERR=4
          CALL TRMERR(TERR,ERUNIT)
          RETURN
        ENDIF
C
C TERMINAL NAME LOCATED
C LOAD CHARACTERISTICS TABLE
C
        CALL TSLOAD(TUNIT)
        CLOSE(TUNIT)
        GO TO 900

999     TERR=1
```

```
              CALL TRMERR(TERR,ERUNIT)

  C  RETURN ERROR CODE IF EVERYTHING WASN'T OK
    900   IF(TERR.NE.0)IERR=TERR
          RETURN
          END
```

```
              SUBROUTINE INITRM
  C
  C  INITIALIZE ALL VARIABLES
  C
          INCLUDE 'TRMCOM.FOR'
  C
  C  DATA STATEMENTS -----
  C
          DATA IIN,IOUT,ERUNIT/5,6,6/
          DATA BLSYM/'    '/

          DATA CPARAM,CSTRT/MAXPNT*0,MAXPNT*0/

          DATA CSYMS/
       *  'DESCR ','CURUP ','CURDN ','CURRT ','CURLFT','CURPOS',
       *  'SETSCR','CLRSCR','SETBOL','SETNUL','SETUND','SETBLI',
       *  'SETREV','RESETT','GRHEAD','GRTAIL','GRERAS','BACKAN',
       *  NFRCS*'    '/
  C
          DATA ISYMS/
       *  'LINES ','COLUMN','SCREEN','MXSCRX','MXSCRY','HIDEV ',
       *  'MASK  ','HIYOR ','LOYOR ','HIXOR ','LOXOR ','OFFSET',
       *  NFRIS*'    '/
  C
          DATA LSYMS/
       *  'GRAPHX','BOLD  ','UNDRLN','BLINK ','REVERS','YFIRST',
       *  'CODELC',
       *  NFRLS*'    '/
  C
          DATA NUL,TAB/0,9/
          DATA STYP/MAXPNT*'C',MAXPNT*'I',MAXPNT*'L'/
  C
  C     CHARACTER SET
  C
  C           0      10     20     30     40     50
  C
  C     0     0      A      K      U      COLON :  LESS  <
  C     1     1      B      L      V      PLUS  +  GREAT >
  C     2     2      C      M      W      MINUS -  POUND #
  C     3     3      D      N      X      STAR  *
  C     4     4      E      O      Y      SLASH /
  C     5     5      F      P      Z      BSLASH \
  C     6     6      G      Q      BLANK  EQUAL =
  C     7     7      H      R      LPAREN (  DOT  .
```

```
C    8     8     I        S  RPAREN  )  COMMA  ,

C    9     9     J        T  SEMI    ;  QUOTE  '

C

      DATA BSLTBL/10*' ',
     *  2*' ',24,' ',27,12,29,3*' ',
     *  7*' ',30,2*' ',
     *  31,4*' ',26,4*' ',
     *  5*' ','\',4*' ',
     *  3*' '/
      DATA ALPHA /'0','1','2','3','4','5','6','7','8','9',

     $    'A','B','C','D','E','F','G','H','I','J',

     $    'K','L','M','N','O','P','Q','R','S','T',

     $    'U','V','W','X','Y','Z',' ','(',')',';',

     $    ':','+','-','*','/','\','=','.',',','''',
     $    '<','>','#'/
C

C    ALTERNATE CHARACTER SET

C

      DATA ALPHB /'0','1','2','3','4','5','6','7','8','9',

     $    'a','b','c','d','e','f','g','h','i','j',

     $    'k','l','m','n','o','p','q','r','s','t',

     $    'u','v','w','x','y','z',' ','(',')',';',

     $    '|','+','-','*','/','$','=','.',',','"',

     $    '[',']','#'/
C

C***********************************************************************
C
C  SET ERROR INDICATORS TO ZERO
C

      TERR=0
C
C  INITIALIZE THE VARIABLES
C
      DO I=1,MAXPNT
        IVALU(I)=0
```

```
        LVALU(I)=.FALSE.
      END DO

      DO I=1,MAXCHR
        CVALU(I)=' '
      END DO
C
C  INITIALIZE POINTER ARRAYS.
C
      DO 100 I=1,MAXPNT
      CPNTR(I)=I
      IPNTR(I)=I
      LPNTR(I)=I
100   CONTINUE

      EOL=.FALSE.
      MODSYM=.TRUE.
      CPNCUR=1

      RETURN
      END
```

```
C****************************************************************************
C       SUBROUTINE: TLOCAT
C
C       This routine locates a terminal name in the TERM.DEF file.
C       Once found, the characteristics of the terminal are read in.
C
C****************************************************************************
        SUBROUTINE TLOCAT(UNIT,TNAME)
        INTEGER UNIT
        CHARACTER * 6 TNAME, ALIAS
        DATA ALIAS/'ALIAS '/
        INCLUDE 'TRMCOM.FOR'
C****************************************************************************
C       GET A LINE FROM THE FILE
C
   1    CALL GETLIN(UNIT)
        IF(TERR.EQ.3)RETURN
C
C  GET A SYMBOL
C
        CALL GETSYM
        IF(EOL)GO TO 1
        IF(SYMBOL.EQ.ALIAS)THEN
   2      CALL GETSYM
          IF(EOL)GO TO 1
          IF(SYMBOL.EQ.TNAME)THEN
C  FINISH READING THAT LINE
  11          CALL GETSYM
              IF(.NOT.EOL)GO TO 11
              RETURN
          ENDIF
          GO TO 2
        ENDIF
        IF(SYMBOL.NE.ALIAS)GO TO 1
        RETURN
        END
```

```
C****************************************************************************
C       SUBROUTINE: TSLOAD
C
C       This routine loads the symbol table for terminal set up.
C       The symbol table appears at the top of TERM.DEF.
C
C****************************************************************************
        SUBROUTINE TSLOAD(UNIT)
        INTEGER UNIT
        CHARACTER * 6 END
        DATA END/'END   '/
        INCLUDE 'TRMCOM.FOR'
C****************************************************************************
C       GET A LINE FROM THE FILE
C
   1    CALL GETLIN(UNIT)
        IF(TERR.EQ.3)RETURN
C
C  ELSE, PROCESS THE LINE
C
   2    IF(MODSYM)THEN
            SYMPNT=1
            CALL GETSYM
            IF(EOL)GO TO 1
            IF(SYMBOL.EQ.END)GO TO 900
            CALL SRCHTB
C  CHECK FOR ERROR
            IF(TERR.EQ.5)RETURN
C
C  ELSE SYMBOL IS OK, GO TO GET VALUE
C
            MODSYM=.FALSE.
            GO TO 2
        ENDIF
C
C  PROCESS A VALUE
C
        IF(.NOT.MODSYM)THEN
C
C  NO NEED TO GET VALUE IF SYMBOL IS LOGICAL
C
            IF(STYP(SYMPNT).EQ.'L')THEN
              NPNTV=SPNTR(SYMPNT)
              LVALU(NPNTV)=.TRUE.
              MODSYM=.TRUE.
              GO TO 2
            ENDIF
C
C  SYMBOL IS EITHER CHARACTER STRING OR HAS INTEGER VALUE
C
            CALL GETVAL
            IF(EOL)GO TO 1
```

```
              CALL INTLDV
              MODSYM=.TRUE.
              GO TO 2
          ENDIF
C
   900    RETURN
          END
```

```
C*********************************************************************
C        SUBROUTINE: SRCHTB
C
C        This routine searches for a symbol in SYMTBL and if found,
C        returns the location of that symbol in the table.
C
C*********************************************************************
          SUBROUTINE SRCHTB
          INCLUDE 'TRMCOM.FOR'
C*********************************************************************
          SYMPNT=1
          LAST=3*MAXPNT
   1      CONTINUE
          IF(SYMTBL(SYMPNT).EQ.BLSYM)THEN
             SYMPNT=SYMPNT+1
             IF(SYMPNT.GT.LAST)GO TO 990
             GO TO 1
          ENDIF
          IF(SYMTBL(SYMPNT).EQ.SYMBOL)THEN
C  SYMBOL FOUND -- RETURN
             RETURN
          ELSE
C  GO ON SEARCHING
             SYMPNT=SYMPNT+1
             IF(SYMPNT.GT.LAST)GO TO 990
             GO TO 1
          ENDIF
   990    TERR=5
          CALL TRMERR(TERR,ERUNIT)
          RETURN
          END
```

```
C*********************************************************************
C       SUBROUTINE: GETLIN
C
C       This routine reads in a line from the specied unit number.
C       It then sets the pointer at the beginning and returns.
C
C*********************************************************************
        SUBROUTINE GETLIN(UNIT)
        INTEGER UNIT
        INCLUDE 'TRMCOM.FOR'
C*********************************************************************
C
C  SET EOL FALSE AND THEN READ A LINE
C
        EOL=.FALSE.
        LINE(1:80)=ALPHA(37)
     1  READ(UNIT,1000,END=998,ERR=999)LINE
 1000   FORMAT(A80)
C
C  SKIP LINES STARING WITH '#'
C
        IF(BLINE(1).EQ.ALPHA(53))GO TO 1
        NCHAR=LENGTH(BLINE,80)
        LPOINT=1
        RETURN
C
C       ERRORS
C
  999   TERR=2
        CALL TRMERR(TERR,ERUNIT)
        RETURN
  998   TERR=3
        CALL TRMERR(TERR,ERUNIT)
        RETURN
        END
```

```
C*********************************************************************
C       SUBROUTINE: GETSYM
C
C       One symbol is separated from the line. The routine starts at
C       the current pointer and searches for the next separator (blank
C       or tab).
C
C*********************************************************************
        SUBROUTINE GETSYM
        INCLUDE 'TRMCOM.FOR'
C*********************************************************************
        EOL=.FALSE.
        IF(LPOINT.GT.NCHAR)THEN
          EOL=.TRUE.
          RETURN
        ENDIF
C
C  ACCUMULATE SYMBOL UNTIL NEXT BLANK --
C  (SYMBOLS ARE ONLY 6 CHARS LONG)
C
        SYMBOL(1:6)=' '
        IPOINT=1
     1  IF(IPOINT.LE.6)BSYMB(IPOINT)=BLINE(LPOINT)
        IPOINT=IPOINT+1
        LPOINT=LPOINT+1
        IF(LPOINT.GT.NCHAR)GO TO 3
        IF(BLINE(LPOINT).NE.' '.AND.
     *     BLINE(LPOINT).NE.TAB)GO TO 1
C
C  POSITION POINTER AT NEXT NON-BLANK CHARACTER
C
     5  LPOINT=LPOINT+1
        IF(LPOINT.GT.NCHAR)GO TO 3
        IF(BLINE(LPOINT).EQ.' '.OR.
     *     BLINE(LPOINT).EQ.TAB)GO TO 5
C
     3  CONTINUE
C
C  CONVERT SYMBOL TO UPPER CASE
C
        CALL UCASE(BSYMB,6)

        RETURN
        END
```

```
C******************************************************************
C       SUBROUTINE: GETVAL
C
C       The value of a symbol is read from the terminal characteristics
C       file. The value is either integer or character string.
C
C******************************************************************
        SUBROUTINE GETVAL
        INCLUDE 'TRMCOM.FOR'
C******************************************************************
        EOL=.FALSE.
        IF(LPOINT.GT.NCHAR)THEN
          EOL=.TRUE.
          RETURN
        ENDIF
C
C  ACCUMULATE SYMBOL UNTIL NEXT BLANK -- ALPHA(37)
C  (VALUES ARE UPTO 80 CHARS LONG)
C
        VALUE(1:80)=' '
        NCHINV=0


        IF(BLINE(LPOINT).EQ.ALPHA(50))THEN
C STRING WITH ' '
  1       LPOINT=LPOINT+1
          IF(LPOINT.GT.NCHAR)GO TO 3
          IF(BLINE(LPOINT).EQ.ALPHA(50))GO TO 5
          NCHINV=NCHINV+1
          IF(NCHINV.LE.80)BVALU(NCHINV)=BLINE(LPOINT)
          GO TO 1

        ENDIF

        IF(BLINE(LPOINT).EQ.ALPHB(50))THEN
C STRING WITH " "

  11      LPOINT=LPOINT+1
          IF(LPOINT.GT.NCHAR)GO TO 3
          IF(BLINE(LPOINT).EQ.ALPHB(50))GO TO 5
          NCHINV=NCHINV+1
          IF(NCHINV.LE.80)BVALU(NCHINV)=BLINE(LPOINT)
          GO TO 11

        ENDIF

C NUMBERS
  111   IF(LPOINT.GT.NCHAR)GO TO 3
        NCHINV=NCHINV+1
        IF(NCHINV.LE.80)BVALU(NCHINV)=BLINE(LPOINT)
        LPOINT=LPOINT+1
        IF(
     *    BLINE(LPOINT).NE.' '.AND.
```

```
     *    BLINE(LPOINT).NE.TAB)THEN
          GO TO 111
        ELSE
          GO TO 5
        ENDIF


  5     LPOINT=LPOINT+1
C
C POSITION POINTER AT NEXT NON-BLANK CHARACTER
C
        IF(LPOINT.GT.NCHAR)GO TO 3
        IF(BLINE(LPOINT).EQ.' '.OR.
     *    BLINE(LPOINT).EQ.TAB)GO TO 5

  3     CONTINUE
C
C DONE!
C
        RETURN
        END
```

```
C**********************************************************************
C       SUBROUTINE: INTLDV
C
C       This routine interprets and loads the value of a symbol
C       into the appropriate place in a table.
C
C**********************************************************************
        SUBROUTINE INTLDV
        INCLUDE 'TRMCOM.FOR'
        CHARACTER * 80 VALLD
        CHARACTER * 1 BVALLD(80)
C**********************************************************************
C
C    VALUE IS IN CHARACTER CONSTANT VALUE WITH NCHINV CHARACTERS.
C      CHARACTER STRING IF LEADING CHAR IS ' OR ".
C
        NPNTV=SPNTR(SYMPNT)
        IVP=0
        ILD=0
C
C *** FOLLOWING IF BLOCK IS FOR LOADING CHAR STRING ***
C
        IF(STYP(SYMPNT).EQ.'C')THEN
C
C    INTERPRET CHARACTERS PRECEDED BY A \
C
   1      IVP=IVP+1
          IF(BVALU(IVP).EQ.ALPHA(46))THEN
C
C    INTERPRET \? TO MEAN PARAMETER
C
            IVP=IVP+1
            IF(BVALU(IVP).EQ.'?')THEN
C
C    INCREASE PARAMETER COUNT AND PUT 'NUL' AT THAT CHAR. POSITION
C
              CPARAM(NPNTV)=CPARAM(NPNTV)+1
              ILD=ILD+1
              BVALLD(ILD)=NUL
              IF(IVP.GE.NCHINV)GO TO 2
              GO TO 1
            ENDIF
C
C    ELSE LOOK UP IN BSLTBL
C
            DO 100 K=1,53
            IF(BVALU(IVP).EQ.ALPHA(K))GO TO 105
  100       CONTINUE
C
C    UNDEFINED 'BACKSLASH' SEQUENCE
C
            TERR=6
```

```
            CALL TRMERR(TERR,ERUNIT)
  105       ILD=ILD+1
            BVALLD(ILD)=BSLTBL(K)
          IF(IVP.GE.NCHINV)GO TO 2
          GO TO 1
          ENDIF
C
C
C    REGULAR CHARACTER STRING -- JUST LOAD IT IN
C
          ILD=ILD+1
          BVALLD(ILD)=BVALU(IVP)
          IF(IVP.GE.NCHINV)GO TO 2
          GO TO 1


   2      CONTINUE
C
C    COPY THE STRING INTO THE ARRAY CVALU
C
          CLEN(NPNTV)=ILD
          CSTRT(NPNTV)=CPNCUR
          I=1
   3      CVALU(CPNCUR)=BVALLD(I)
          CPNCUR=CPNCUR+1
          I=I+1
          IF(I.GT.ILD)GO TO 900
          GO TO 3
C
        ENDIF
C
C *** FOLLOWING IF BLOCK IS FOR LOADING INTEGER VALUE ***
C
        IF(STYP(SYMPNT).EQ.'I')THEN
C
C    THEN WE HAVE AN INTEGER TO LOAD
C
          IVALU(NPNTV)=0
   5      IVP=IVP+1
          DO 200 K=1,10
          IF(BVALU(IVP).EQ.ALPHA(K))GO TO 205
  200     CONTINUE
C    NON-INTEGER VALUE WHERE INTEGER EXPECTED -- TERR=7
          WRITE(ERUNIT,6000)IVP,BVALU(IVP)
 6000     FORMAT(1X,'CHARACTER NO. ',I2,'=',A1,'  ...')
          TERR=7
          CALL TRMERR(TERR,ERUNIT)
          RETURN
C
  205     IVALU(NPNTV)=10*IVALU(NPNTV)+(K-1)
          IF(IVP.GE.NCHINV)GO TO 900
C
```

```
C  ELSE CONTINUE TRANSLATION
C
         GO TO 5
      ENDIF
C
  900 RETURN
      END
```

```
C*********************************************************************
C       INTEGER FUNCTION: LENGTH
C
C       This function gives the length of a line of characters. It
C       strips off all blanks at the beginning and the end of the line.
C
C*********************************************************************
      INTEGER FUNCTION LENGTH(BCLIN,MAXCHR)
      CHARACTER *1 BCLIN(MAXCHR)
      INCLUDE 'TRMCOM.FOR'
C*********************************************************************
      LAST=MAXCHR

      DO 10 J=1,LAST
      IF(BCLIN(J).NE.' '.AND.BCLIN(J).NE.TAB)GO TO 15
   10 CONTINUE
      LENGTH=0
      RETURN

   15 DO 20 I=1,LAST
      K=LAST-I+1
      IF(BCLIN(K).NE.' '.AND.BCLIN(K).NE.TAB)GO TO 25
   20 CONTINUE
      LENGTH=0
      RETURN

   25 LENGTH=K-J+1
      DO 30 I=J,K
      L=I-J+1
   30 BCLIN(L)=BCLIN(I)
      RETURN
      END
```

```
C****************************************************************
C       SUBROUTINE: UCASE
C
C       This routine converts a line of characters to uppercase.
C       It can handle lines upto 80 characters long.
C
C****************************************************************
        SUBROUTINE UCASE(BCLINE,LEN)
        INCLUDE 'TRMCOM.FOR'
        CHARACTER* 1 BCLINE(LEN)
C****************************************************************
        DO 10 I=1,LEN
        DO 10 J=11,36
C
C ALPHB(11) THRU ALPHB(36) STORES THE LOWERCASE ALPHABETS
C CORRESPONDING LOCATIONS IN ALPHA HAS THE UPERCASE.
C
C ** THIS SCHEME OF CONVERSION TO UPPERCASE WORKS FOR ANY
C    CHARACTER CODE (ASCII, EBCDIC ..) **
C
C CONVERT TO UPPERCASE
C
C
        IF(BCLINE(I).EQ.ALPHB(J))BCLINE(I)=ALPHA(J)
   10   CONTINUE
        RETURN
        END
```

```
C       SUBROUTINE: TRMERR
C
C       This subroutine prints out messages if errors occur during
C       input/output operations at the terminal.
C
        SUBROUTINE TRMERR(ERCODE,ERUNIT)

        INTEGER ERUNIT,ERCODE
        LOGICAL EFATAL(7)
C
        CHARACTER * 80 TERMSG(7)
        DATA TERMSG/
     1  'ERROR IN OPENING FILE',
     +  'ERROR DURING READ.',
     +  'END DURING READ.',
     +  'UNKNOWN TERMINAL NAME.',
     5  'UNKNOWN SYMBOL IN TERMINAL DEFINITION FILE.',
     +  'NO DEFINITION FOR SYMBOL BEGINNING WITH A "\".',
     +  'NON-NUMERAL WHERE NUMERAL EXPECTED.'/
        DATA EFATAL
     +  /.FALSE.,.TRUE.,.TRUE.,.FALSE.,.FALSE.,.FALSE.,.FALSE./
C           1       2       3       4       5       6       7
C****************************************************************
        IF(ERCODE.LE.0.OR.ERCODE.GT.7)RETURN
        WRITE(ERUNIT,5000)TERMSG(ERCODE)
   5000 FORMAT(1X,A80)
        IF(EFATAL(ERCODE))STOP '** FATAL ERROR **'
        RETURN
        END
```

```fortran
C       INCLUDE FILE: TRMCOM.FOR
C
C       This file contains the common blocks and variables for the
C       display management routines.
C
C *** PARAMETERS -----------------------------------------------------
C
        PARAMETER (MAXPNT=30)
        PARAMETER (NFRCS=MAXPNT-18)
        PARAMETER (NFRIS=MAXPNT-12)
        PARAMETER (NFRLS=MAXPNT-7)
        PARAMETER (MAXCHR=6*MAXPNT)
C
C *** CHARACTERS -----------------------------------------------------
C
        CHARACTER * 80 LINE,VALUE
        CHARACTER * 6 SYMBOL,SYMTBL,CSYMS(MAXPNT),
     *  ISYMS(MAXPNT),LSYMS(MAXPNT),BLSYM
        CHARACTER * 1 BLINE(80),ALPHA,ALPHB,TAB,
     *               BSYMB(6),STYP,CVALU,
     *               BVALU(80),NUL,BSLTBL
C
C *** INTEGER VARIABLES ----------------------------------------------
C
        INTEGER NCHAR,LPOINT
        INTEGER ERUNIT, TERR, SYMPNT
        INTEGER IVALU,IPNTR(MAXPNT),CPNTR(MAXPNT),LPNTR(MAXPNT)
        INTEGER CLEN,SPNTR,CPARAM,CPNCUR,CSTRT
C
C THE FOLLOWING ARE POINTERS FOR CHARACTER STRINGS
C
        INTEGER DESCR,CURUP,CURDN,CURRT,
     *  CURLFT,CURMOV,CURPOS,SETSCR,CLRSCR,
     *  SETBOL,SETUND,SETBLI,SETREV,RESETT,
     *  GRHEAD,GRTAIL,GRERAS,BACKAN,SETNUL
C
C POINTERS FOR LOGICAL VARIABLES
C
        INTEGER GRAPHX,BOLD,UNDRLN,BLINK,REVERS,YFIRST,CODELC
C
C POINTERS FOR INTEGER NUMBERS
C
        INTEGER LINES,COLUMN,SCREEN,MXSCRX,MXSCRY,HIDEV,MASK,HIYOR,
     *  LOYOR,HIXOR,LOXOR,OFFSET
C
C *** LOGICAL VARIABLES ----------------------------------------------
C
        LOGICAL * 1 EOL,MODSYM,LVALU
C
C *** EQUIVALENCES ---------------------------------------------------
C
        EQUIVALENCE (LINE,BLINE(1))
```

```fortran
        EQUIVALENCE (SYMBOL,BSYMB(1)),(VALUE,BVALU(1))

        EQUIVALENCE (DESCR,CPNTR(1)),(CURUP,CPNTR(2)),
     *  (CURDN,CPNTR(3)),(CURRT,CPNTR(4)),(CURLFT,CPNTR(5)),
     *  (CURPOS,CPNTR(6)),(SETSCR,CPNTR(7)),(CLRSCR,CPNTR(8)),
     *  (SETBOL,CPNTR(9)),(CPNTR(10),SETNUL),(SETUND,CPNTR(11)),
     *  (CPNTR(12),SETBLI),(CPNTR(13),SETREV),(RESETT,CPNTR(14)),
     *  (CPNTR(15),GRHEAD),(CPNTR(16),GRTAIL),(GRERAS,CPNTR(17)),
     *  (CPNTR(18),BACKAN)
C
        EQUIVALENCE (LINES,IPNTR(1)),(COLUMN,IPNTR(2)),
     *  (SCREEN,IPNTR(3)),(MXSCRX,IPNTR(4)),(MXSCRY,IPNTR(5)),
     *  (HIDEV,IPNTR(6)),(MASK,IPNTR(7)),(HIYOR,IPNTR(8)),
     *  (LOYOR,IPNTR(9)),(HIXOR,IPNTR(10)),(LOXOR,IPNTR(11)),
     *  (OFFSET,IPNTR(12))
C
        EQUIVALENCE (LPNTR(1),GRAPHX),(LPNTR(2),BOLD),
     *  (LPNTR(3),UNDRLN),(LPNTR(4),BLINK),(LPNTR(5),REVERS),
     *  (LPNTR(6),YFIRST),(LPNTR(7),CODELC)
C
        EQUIVALENCE (SPNTR(1),CPNTR(1)),
     *              (SPNTR(MAXPNT+1),IPNTR(1)),
     *              (SPNTR(2*MAXPNT+1),LPNTR(1))
C
        EQUIVALENCE (SYMTBL(1),CSYMS(1)),
     *              (SYMTBL(MAXPNT+1),ISYMS(1)),
     *              (SYMTBL(2*MAXPNT+1),LSYMS(1))
C
C *** ALL COMMONS ----------------------------------------------------
C
        COMMON/SYMS/IVALU(MAXPNT),
     *  CLEN(MAXPNT),SPNTR(3*MAXPNT),CPARAM(MAXPNT),
     *  CSTRT(MAXPNT), BSLTBL(53),STYP(3*MAXPNT),SYMTBL(3*MAXPNT),
     *  CVALU(MAXCHR),ALPHA(53),ALPHB(53)

        COMMON/UNUMS/IIN,IOUT,ERUNIT
        COMMON/LINP/LINE,NCHAR,LPOINT,SYMBOL,SYMPNT,
     *  VALUE,NCHINV,CPNCUR
        COMMON/CHARS/BLSYM,NUL,TAB
        COMMON/LOGIC/TRMRDY,MODSYM,EOL,LVALU(MAXPNT)
```

```
# This file contains the terminal characteristics.
# Blank lines are skipped.
#
# Designed by:  Naba Barkakati
#
#
ALIAS VISUAL V550 VIS550
# Added 10/29/84
DESCR 'VISUAL 550 GRAPHICS TERMINAL'
        LINES 33        COLUMN 80       SCREEN 2
        CURUP '\E[\?A'  CURDN '\E[\?B'  CURRT '\E[\?C'
        CURLFT '\E[\?D' CURPOS '\E[\?;\?H'
        SETSCR '\E[\?;\?r' CLRSCR '\E[2J'
        BOLD    UNDRLN  BLINK   REVERS
        SETBOL '\E[1m'  setnul '\E[0m'  setund '\E[4m'
        setbli '\E[5m'  setrev '\E[7m'  resett '\Ec'
 graphx
        MXSCRX 767      MXSCRY 584      YFIRST
        GRHEAD '\G'     GRTAIL '\U'     HIDEV 32        MASK 31
        HIYOR 32        LOYOR 96        HIXOR 32        LOXOR 64
        GRERAS '\E\F'   BACKAN '\C'
END
#
# Definitions for VT100 terminal
#
ALIAS VT100 ANSI
# Added 12/14/84
DESCR 'DEC VT100 TERMINAL'
        LINES 24        COLUMN 80       SCREEN 1
        CURUP '\E[\?A'  CURDN '\E[\?B'  CURRT '\E[\?C'
        CURLFT '\E[\?D' CURPOS '\E[\?;\?H'
        SETSCR '\E[\?;\?r' CLRSCR '\E[2J'
        BOLD    UNDRLN  BLINK   REVERS
        SETBOL '\E[1m'  setnul '\E[0m'  setund '\E[4m'
        setbli '\E[5m'  setrev '\E[7m'  resett '\Ec'
END
#
# Definitions for VT52 terminal
#
ALIAS VT52 VT_52
# Added 12/14/84
DESCR 'DEC VT52 TERMINAL'
LINES 24        COLUMN 80       SCREEN 1
CURUP '\EA'     CURDN '\EB'     CURRT '\EC'
CURLFT '\ED'    CURPOS '\EY\?\?'
SETSCR '\E<\E[\?;\?r\E[?21' CLRSCR '\EH\EJ'
BOLD    UNDRLN  BLINK   REVERS  CODELC  OFFSET 31
SETBOL '\E<\E[1m\E[?21' setnul '\E<\E[0m\E[?21' setund '\E<\E[4m\E[?21'
setbli '\E<\E[5m\E[?21' setrev '\E<\E[7m\E[?21' resett '\E<\Ec\E[?21'
END
# Characteristics of Televideo 950 terminal
# (Not fully tested)
# Added 3/27/85
```

```
ALIAS TV950 TELEVID TVIDEO
DESCR 'TELEVIDEO 950 TERMINAL'
LINES 24 COLUMN 80 SCREEN 1
CURUP '\Ej'  CURPOS '\E=\?\?'
CODELC  OFFSET 31
# \R translates to 30(Decimal)
clrscr '\R\EY'
undrln          blink           revers
setund '\EG8'  setbli '\EG2'  setrev '\EG4'
setnul '\EG0'
END
```

# CHOOSING A DOCUMENT-FORMATTING SYSTEM

Richard K. Wallace
Los Alamos National Laboratory

February 3, 1986

## ABSTRACT

After surveying available tools for formatting large computer code manuals, we chose the TEX system, to be initially implemented on VAX 11/780 and 8600 computers. We also recognised that a "What You See Is What You Get" word processor offers sufficient capabilities for small (5 – 10 page) reports and manuals, and recommended that WordMARC be considered for formatting in those situations.

---

## 1  BACKGROUND

Los Alamos National Laboratory is a federally funded applied research laboratory managed by the University of California for the U.S. Department of Energy under contract W-7405-ENG- 36. The Laboratory engages primarily in energy, national defense, and accelerator/nuclear physics research. It employs about 7800 people and is divided organisationally into 43 Divisions. This paper discusses criteria used by the Applied Theoretical Physics Division (X Division) to select a document formatting system. X Division consists of about 260 employees, more than 200 of whom have doctorates in physics-related disciplines and all of whom have extensive interactive computing experience.

The major Laboratory computing center, managed by C Division, is the Central Computing Facility, which contains 7 Cray supercomputers, 8 large CDC computers, and 10 DEC VAXs, with a total computing capacity equivalent to 20 Cray-1 supercomputers. In addition, nearly 100 Distributed Processors, all VAX 11/780, 785, or 8600s, are scattered over 43 square miles, linked by DEC-Net and managed by the individual divisions. Owing to the defense work, the computing resources are divided into classification partitions, each completely separate (no communication channels) from all other partitions.

## 2  PURPOSE

In August 1984, we formed a Committee to recommend a replacement for the then-current computerised documentation tools (TRIX/RED, REDPP), which would be unavailable after removal of the Laboratory's secure CDC 7600. Recent turnover in the code user groups emphasised the lack of *current, comprehensive* documentation (user and physics manuals) for the major X-Division production codes. This lack of documentation increases the training time required for new users and code developers and hinders efficient code use by them and by experienced users. The existing code manuals must be continually revised and expanded as the codes rapidly evolve.

We have therefore surveyed the field of document production in search of a modern, efficient, long-term document-formatting system that will satisfy our need for producing thorough, clear, current documentation as simply as possible. The system development was coordinated with C Division to reduce duplication of effort and prevent future compatability problems.

## 3  SUMMARY

We recommended that TeX be used for formatting X-Division code manuals. Although the Division should not *require* the use of TeX, that tool should be seriously considered for any major documentation effort. We recognize that WordMARC may offer sufficient formatting capabilities for small (5-10 page) reports and manuals and should be considered for those applications.

To obtain the full benefit of the TeX documentation system, the following hardware was recommended:

- A high speed (at least 24 pages/min) laser printer.

- An upgrade for one of our two VAX 11/780s to a DEC 8600 to provide greater responsiveness, larger CPU capacity, and improved availability of full screen text editors. Even if TeX became available on CTSS (the Cray operating system), the local VAXs could be heavily used for text entry and WordMARC applications.

- A low-cost (under $3000) laser printer that can produce local (in office) TeX output; possible candidates include the DEC LN03 and the HP Laser-Jet.

- Workstations with a preview capability for frequent TeX users.

C Division was strongly encouraged to provide the following software support:

- A CTSS (Cray) implementation of TeX; this is in progress.

- Simple lineprinter/ASCII output from standard TeX DVI files; rudimentary package is now in use.

- Central Computing Facility output capable of producing 5000 formatted pages/day.

- A method to merge TeX text with graphics files that are in the unique Los Alamos Common Graphics System metafile format.

- Conversion programs for TROFF, TRIX/RED, and VMS WordMARC.

- Classified consulting services on TeX.

- "Writer's Workbench"-type software (such as a spelling checker) for TeX files.

## 4  JUSTIFICATION - Requirements

The selection of TeX for the X-Division formatting system was based on its satisfaction of the following unique X-Division requirements. The system should

1. be easily portable to new operating systems, minimizing future translations such as must now be done for the large number of LTSS (CDC 7600 operating system) TRIX/RED files. The system should also be widely used outside of DOE to increase the support for and knowledge about it,

2. be declarative (using predefined structures for headers/footers, sections, paragraph indentations, examples, etc.) rather than procedural (requiring the author to define page layout during text–, or content–, entry). This requirement allows a few experienced people to maintain the detailed page layout macros, whereas casual users simply enter text,

3. easily accept mathematical equations and format them with as little user assistance as possible,

4. be capable of merging text with computer-generated graphics,

5. have automatic Table of Contents generation,

6. have automatic Index generation,

7. provide for nested tables,

8. have a source file format that facilitates macro construction to support detailed page layout macros, translation macros (from previous systems and into future systems), and text unformatting macros (to easily allow incorporation of arbitrary machine-readable text),

9. allow text input from any ASCII terminal (including Tektronix 4000 and 4100 series),

10. be accessible transparently from CTSS to eliminate user investment in learning a different operating system or accessing special hardware (most users work exclusively on the Cray CTSS systems rather than on VAXs),

11. produce simple ASCII text output for online help files from the same source file that produces fully formatted documents,

12. allow comments in the source file,

13. facilitate page layout changes or even allow determination of the layout after text entry,

14. symbolically reference equation, figure, section, and page numbers, and

15. allow "interactive" execution to provide error diagnostics and allow recovery from minor source file errors.

# 5   COMPARISONS

The major software for code documentation that begins to address the requirements listed above is the following:

Interleaf

Advantages:

1. Interactive "What you see is what you get" (WYSIWYG) system. This can be much easier and faster to use than a batch formatter for small files.

2. Instant feedback (screen shows all page layouts, fonts, text sizes, pagination, etc.).

Disadvantages:

1. No symbolic equation entry. Equations must be entered with a graphics package that draws each individual symbol or character on the page.

2. No symbolic referencing of equation numbers, sections, etc.

3. Operates only on SUN, APOLLO, and VAXStation II workstations.

4. Cost is $12,000 per workstation node, which is prohibitively expensive.

Interleaf was the most capable WYSIWYG formatting system on the market. It would unquestionably be the most productive system to have for a single user. However, the lack of symbolic mathematical entry and the unavailability

for a timesharing system are fatal flaws for our purposes. The $12,000 per node price, coupled with the price of providing SUN-class workstations to everyone contributing text, is prohibitive. In addition, no SUN-class workstation has been approved for classified processing.

WordMARC, Version 5 ("Composer")

Advantages:

1. WYSIWYG system that is much easier and faster to use than a batch formatter for nonequation typing of small files.

2. Instant feedback of text and general page layout.

3. Preserves author's meaning (equations displayed on first typing).

Disadvantages:

1. Procedural; no declarative format.

2. Cannot easily change existing document format.

3. No comments allowed in source file.

4. VT100 emulation terminal required (for example, no Tektronix 4014).

5. Response slows to unacceptable times with large documents and many simultaneous users. Response time is more critical for completely interactive systems. The continuous formatting increases the CPU load compared with that of a batch formatter.

6. Less involvement allowed to professional editors/designers.

7. Limited (and in some cases insufficient) mathematical capabilities.

8. No proportionally spaced laser printer output.

The disadvantages indicate that WordMARC may be ideal for formatting memos and short reports but would be inadequate for very large manuals. Although WordMARC (from Marc Software) was specifically compared here, the disadvantages are similar for other WYSIWYG systems, such as MASS-11. They all generally require VT-100 emulation capability, are generally procedural (requiring some author involvement in page layout), are difficult to use for changing page layout retroactively, and require interactive computer response time. However, screen editors in such WYSIWYG systems could be used to prepare the ascii input files for a batch editor, such as TeX or TROFF.

We found no WYSIWYG systems with all the capabilities listed under "Requirements" above. However, two batch formatting systems in common use (TROFF and TeX) could satisfy nearly all of our requirements, and their respective advantages are listed below. C Division has decided to support both TROFF and TeX as Laboratory document production systems.

## TROFF with EQN and TBL

1. May be easier to learn than standard TeX (but not significantly easier than LaTeX).

2. Better table generation capability than standard TeX.

3. Writer's Workbench editorial software available.

## TeX

1. Arbitrary length command names (TROFF restricts commands to less than 2 characters).

2. More portable than TROFF (TeX is available in generic Pascal and C, whereas TROFF is tied intimately to the UNIX operating system).

3. Los Alamos Common Graphic System/TeX interface exists for QMS laser printers, so merging text and graphics is a reality.

4. Slightly more control over output appearance.

5. More widely available screen preview systems (including SUN, APOLLO, IBM AT, Apple Macintosh, and Tektronix 4014).

6. TROFF requires the UNIX operating system, which is currently unacceptable for classified computing.

Points 2 and 6 above are sufficiently serious that we consider TROFF an unacceptable solution. TeX is therefore the most appropriate choice for an X-Division formatter.

# 6  CONCLUSIONS

We chose TeX as our standard document formatting system, largely because of its great portability compared to TROFF. For small memos and reports, many secretaries still use WordMARC. Since we reached our decision, several other divisions at the Laboratory have begun using TeX, and the the official publication division (which uses an APS-5 phototypesetter for high-quality output) is committed to switching completely to TeX. The Laboratory is moving to standardize on Postscript (from Adobe Systems) as a common text/graphics device independent file structure, and we are now obtaining hardware and software to allow TeX output through Postscript devices. In addition, the Laboratory has just moved to support LaTeX (a TeX macro package) as the standard version of TeX. We currently use LaTeX on SUN, APOLLO, VAXStation II workstations, IBM XT, AT, Apple Macintosh, VAX/VMS, and VAX/UNIX, and have contracted for an implementation on CTSS.

# 7  FURTHER INFORMATION

- TeX: TeX Users Group, P.O. Box 594, Providence, RI 02901.

- LaTeX: TeX macro package developed by Leslie Lamport (now at DEC). For information, contact the reference under "TeX".

- TeX on workstations, and output to Postscript devices: Textset Inc., 4116 4th. St., P.O. Box 7993, Ann Arbor, MI 48107. (313) 996- 3566.

- TeX on IBM XT/AT: PCTeX Inc., 20 Sunnyside, Suite H, Mill Valley, CA 94941, (415) 388-8853, or MicroTeX, Addison-Wesley Publishing Co., Educational Media Systems Division, Reading, MA 01867. (617) 944-3700, ext. 2677.

- WordMARC: Marc Software International, 260 Sheridan Ave, Suite 200, Palo Alto, CA 94306. (415) 326-1971.

- Interleaf: Interleaf Inc., 1100 Massachusetts Ave., Cambridge, MA 02138. (617) 497-5570.

- MASS-11: Microsystems Engineering Corp., 2040 Hassal Road, Hoffman Estates, IL 60195.

- TROFF: UNIX System manual, Bell Laboratories or Computer Science Division, University of California, Berkeley, CA 94720.

This report was formatted with the TeX macro package "LaTeX", Version 2.06a,

**VAX (tm) Ada (r)**

        Current Version: V1.2
        V1.2 Started Shipments: February, 1986
        Major Features of V1.0:

                -Full ANSI Language
                -Production quality
                -Highly integrated into VAX/VMS Environment
                -Multi-language capabilities
                -Comprehensive diagnostics
                -U.S. Government validated
                -Full symbolic debugging support
                -VAX Language-Sensitive Editor support

(tm) VAX is a trademark of Digital Equipment Corporation
(r) Ada is a registered trademark of the U.S. Government (Ada Joint
 Program Office)

**VAXELN Ada**

        Current Version:  V1.0
        To Start Shipments:  April, 1986
        Major Features:

                -Compatible with VAX Ada
                -Retargetable to real-time/embedded environment
                -Remote debugger
                -Tailorable run-time environment
                -Run-time library retargetted from VAX/VMS to VAXELN
                -Package of interfaces to VAXELN services

**VAX APL**

        Current Version: V2.0
        V2.0 Start Shipments: February, 1986
        Major Features:

                -Performance Improvements
                -APL can call other VAX languages which adhere
                      to the VMS calling standard
                -Multi-key ISAM

**VAX Basic**

        Current Version:  V2.4
        V2.4 Started Shipments: September, 1985
        Major Features:

                -VAX Language-Sensitive Editor support
                -CDD support
                -Contains compile-time directives
                -Provides structured programming constructs
                -Conforms to ANSI Minimal Basic

and printed on a (300 dpi) QMS 800 laser printer.

Author's Address:

Richard K. Wallace
X-7, MS B257
Los Alamos, NM 87544

VAX Bliss

    Current Version:  V4.2
    V4.2 Start Shipments:  February, 1986
    Major Features:
        -Ease of use
        -/Check qualifiers
        -/Cross reference switch
        -VAX Language-Sensitive Editor support

VAX C

    Current Version:  V2.1
    V2.1 Start Shipments:  August, 1985
    Major Features:
        -Full Debug support
        -CDD support
        -VAX Language-Sensitive Editor support
        -Improved run-time routines for UN*X compatibility
        -Shareable run-time library

VAX Cobol

    Current Version:  V3.2
    V3.2 Start Shipments: June, 1985
    Major Features:
        -VAX Language-Sensitive Editor support
        -Screen handling extensions
        -Extended DML

VAX Fortran

    Current Version:  V4.4
    V4.4 Start Shipments: February, 1986
    Major Features:
        -VAX Language-Sensitive Editor support
        -Global optimizations
        -CDD support
        -Records

VAX Pascal

    Current Version:  V3.2
    V3.2 Start Shipments:  December, 1985
    Major Features:
        -Performance/Runtime Optimizations
        -CDD Support
        -VAX Language-Sensitive Editor Support
        -Compatibility support for VAELN Pascal
        -Source Line Debugging

VAX PL/I

    Current Version: V2.4
    V2.4 Start Shipments: March, 1986
    Major Features:
        -VAX Language-Sensitive Editor support
        -CDD support
        -Compile-time pre-processor

VAX RPG II

    Current Version: V2.0
    V2.0 Start shipping: December, 1985
    Major Features:
        -Conforms and is an extended implementation of the
           IBM RPGII defacto standard
        -Fast compile and runtime performance
        -Full screen editor
        -Compatible with IBM implementations on
           Systems 3, 34, and 38
        -CDD support
        -Increased IBM compatibility compared with V1.0
        -New data structures in V2.0
        -Syntax editing capabilities added to RPG
           editor in V2.0

VAX Performance and Coverage Analyzer

    Current Version: V1.1
    V1.1 began shipping: December, 1985

    Major features:
        -Helps to find execution bottlenecks in application
           programs
        -Provides test coverage analysis to determine which
           lines of an application are executed by a given
           set of test programs
        -Has an interface to the VAX DEC/Test Manager

VAX DEC/CMS

Current version: V2.1
V2.1 began shipping: February, 1986

Major Features of V2.0 are:
-a callable interface
-new security features
-significantly improved performance
-groups for the easy organization of related
files

o CMS offers functionality similar to SCCS (Source Code
Control System) on UN*X

VAX DEC/MMS

Current version: V2.1
V2.1 began shipping:  December, 1985
Major features of V2.0 are:

-support for CDD
-support for TDMS
-support for FMS

o MMS offers functionality similar to MAKE on UN*X


VAX DEC/Shell

Current Version:  V1.1
V1.1 began shipping:  February, 1986

Major features of the V1.0 DEC/Shell include:

-an alternate command line interpreter
-the script language
-a set of commonly used UN*X utilities


o DEC/Shell is based on the UN*X V7 Bourne Shell


VAX DEC/Test Manager

Current Version:  V2.0
V2.0 began shipping: February, 1986

Major features:
-Ability to test interactive applications on a
character cell terminal
-Increased integration with VAX DEC/CMS (can store
tests in a CMS library for Test Manager
retrieval)
-Performance Improvements

VAX Language-Sensitive Editor:

Current Version: V1.2
V1.2 began shipping: February, 1986

Major Features:

-Supports Ada(r), Basic, Bliss, C, Cobol, Fortran,
Pascal, Pl/I
-Edit, compile, review, and correct compilation
errors withing a single editing session
-Speeds up source code entry using formatted
language-specific source code templates
-Provides for interactive editing capabilities
during a debugging (VAX Debug) or performance
analysis session (VAX Performance and Coverage
Analyzer)
-User tailorable and user extensible
-Extensive on-line help for supported VAX languages


VAX Scan

Current Version:  V1.0
V1.0 began shipping: November, 1985

Major features:
-A complete VAX programming language used to create
programs that deal with pattern matching and
text transformation
-Used for creating translators, preprocessors,
filters, and parsers
-To build tools for converting data from other
vendor's computing equipment
-Finds and replaces text in files

```
===========================================
```

```
                What Will Your
                   Favorite
                Language Resemble
                    In 1990 ?

                Fall '85 DECUS


         o  APL
         o  BASIC
         o  COBOL
         o  C
         o  Pascal
         o  Ada (R)
         o  PL/I
         o  FORTRAN


    (R) Ada is a Registered Trademark of the U.S. Government,
        Joint Ada Program Office
```

-----------------------------------------------

Standards Organizations

o   ANSI - The American National Standards Institute

o   ISO - The International Standards Organization

o   ECMA - European Computer Manufacturers Association

-----------------------------------------------

To increase DECUS participation in standards, contact:

L&T SIG Standards Coordinator
Jay Wiley
Bechtel Power Corp.
12400 E. Imperial Highway
Norwalk, CA  90650

```
===============================================
```

                                APL

o   standards bodies:

    X3J10 in ANSI
    TC97/SC22 WG3 working group in ISO

o   controversial topics:

    file systems - current draft standard doesn't mention them;
            some members want too much and other want none at all
    report formatting - same as file systems
    nested arrays - recent topic, some disagreement on which
            things to add
    screen management - micros can do it in APL so why can't
            mainframes

o   where to write:

    Garth Foster, acting chairman
    Syracuse University
    111 Link Hall
    Syracuse, NY  13244-1240

```
==================================================
              BASIC

      Standards Committees

              ANSI X3J2

              ISO/TC97/SC22/WG8

              ECMA TC21

              EWICS TC2
--------------------------------------------------

      Current and Future Directions

              Minimal BASIC 1978

              BASIC 1986
                  currently at X3 level

              Additional Enhancement Modules
                  in progress
--------------------------------------------------

      For additional information, write to:

              Jim Totton, Chair X3J2
              Digital Equipment Corp.
              ZKO2-3/K06
              110 Spit Brook Rd.
              Nashua, NH  03062
```

```
==================================================
              COBOL

      Standards Committees

              ANSI X3J4

              CODASYL

              ISO/TC97/SC22/WG4
--------------------------------------------------

      Current and Future Directions

              COBOL 1985
                  approved in September, 1985

              Currently Considering New Directions
```

====================================================

C

ANSI X3J11 C Language Standards Committee

> Proposed standard to be voted out to formal
> public review in March, 1986
>
> Publication of final standard expected
> circa March, 1987
>
> Plans to propose their standard as an ISO
> standard when complete

Other efforts

> IEEE P1003 Portable Operating System (UNIX*)
> Standard
>
> /usr/group UNIX* run time library standards
> working groups

* UNIX is a trademark of AT&T Bell Laboratories

----------------------------------------------

Major features of X3J11 C Standard

o Function prototypes

> - Specify formal parameter types
> - May cause argument conversion

o Preprocessor changes

> - "string-ization" and token concatenation

o Expression evaluation rules

> - Value-preserving vs. unsigned-preserving

o Deletion of obsolete features

> - Structure/union operands of . and ->
>   must be of proper type
>
> - String literals may not be modified,
>   and need not be stored distinctly

====================================================

Pascal

ISO/7185-1983  submitted by British Standards Institution

ANSI/IEEE770X3.97-1983 submitted jointly by X3 and IEEE

----------------------------------------------

Pascal
ANSI Differences From ISO

No conformant array feature

Parameter evaluation order for READ, WRITE, PACK and UNPACK

Relationship between EOLN marker and character set

Allowable Extensions

Corrections of several errors

----------------------------------------------

Extended Pascal Development

Joint effort by X3J9 and IEEE 770 Joint Pascal Committee (JPC)

Parallel effort by British Standards Institution (BSI)

Coordination by ISO/TC 97/SC 22/WG 4 (International Pascal
        Working Group)

---

Candidate Extensions to Pascal

Variable Length Strings
Separate Compilation and Modularity
Direct Access Files
Binding and Unbinding

Schema Types
Dynamic Schema Types

OTHERWISE Clause in CASE Statement
OTHERWISE Clause in Variant Records
Ranges in CASE Statement Constant Lists
Ranges in Case Variant Constant Lists

Relaxation of Order of Declaration
Constant Expressions
Structured Value Constructors
Value Initialization
Structured Function Results
Underscore in Identifiers
EXTEND File Handling Procedure
Standard Numeric Input
Zero TotalWidth
Negative TotalWidth
Inverse ORD/RELPOS
Date and Time
READSTR and WRITESTR Procedures
CARD Function
Set Symetric Difference Operator
FOR-Each Statement
Function-Result Variable
Type Inquiry
Non-Decimal Numbers

---

Candidate Extensions to Pascal
(that may be controversial)

Separate Compilation and Modularity

Schemata and Dynamic Schemata

Binding and Unbinding

=================================================

Ada (R)

Standards bodies:

  AJPO remains the designated ANSI authority

  ISO  TC97/SC22/WG9
        Currently in draft proposal
        status.  Full approval as an
        international standard may
        occur by the end of 1986.

(R) Ada is a registered trademark of the U.S. Government,
    Ada Joint Program Office

---

WHERE THE Ada STANDARD MIGHT GO


Supplementing tasking with a means
of low-level synchronization
(e.g. semaphores)

Tasking for distributed computers
(i.e. without common memory)

Pragma name standardization

Semantics of multiple program
libraries

Exporting Ada subsystems with
internal name suppression

---

WHERE TO VOICE OPINIONS

Ada Language Maintenance Committee
  Dr. John Goodenough, Chairperson
  SofTech
  460 Totten Pond Road
  Waltham, MA 02154

Ada Joint Program Office
  OUSDRE(R&AT)
  Washington, D.C. 20301

ACM Ada Letters
  Various addresses listed in front
  cover

=================================================

PL/I

PL/I Standard maintained by X3J1 (ANSI)

FULL LANGUAGE STANDARD          1976

GENERAL PURPOSE SUBSET          1981

REAL TIME MULTITASKING          RIP 1984

FUTURE FULL LANGUAGE            RIP

    o  X3J1 doing away with Full Language

    o  Lack of vendor support or user interest

    o  Incompatible with "common practice"

-----------------------------------------

FUTURE SUBSET of PL/I

PUBLIC COMMENT 1986

    o  "ONLY" PL/I

    o  New features from Full Language, "common
       practice", and thin air

       (including REFER attribute, condition prefixes, programmer-
        named conditions, many built-in functions)

    o  Only very modest "new language", nothing very
       exciting

       (including UNION structures; named computational constants,
        extensive compile-time arithmetic; SELECT, UNTIL,
        and LEAVE for structured control; "Packages";
        many string-handling functions)

    o  Some 1981 Subset features removed from new Subset

       (including DEFINED variables, LABEL variables and
        parameters)

---------------------------------------------

To express opinions or obtain more information, contact

Dr. John Klensin
Chairman, X3J1
Massachusetts Institute of Technology
Room 20A-226
77 Massachusetts Avenue
Cambridge, MA   02139

```
=============================================
                FORTRAN

EXISTING STANDARDS


    FORTRAN  IV    1966, superceded by

    FORTRAN  77    1978  Full language and subset

    FORTRAN STANDARD maintained by X3J3 (ANSI)


---------------------------------------------

            FUTURE:  FORTRAN 8X


Upward Compatibility from FORTRAN 77?

Draft FORTRAN 8X says:

       "All conforming FORTRAN 77 programs are conforming
        FORTRAN 8X programs"

DEC comment (1984)

       "Must be able to add new features to old programs."

       X3J3 has adopted this requirement,
        except for new source form


    Some new source form features:
      (Not DEC "free form" compatible)

          Significant blanks
                  No blanks inside names, constants
                  Must be used to separate keywords, names, and
                          constants not otherwise delimited

          For instance, these are not allowed:
                  GOTO42
                  X = 1 000 000.D0

       No column 6 rules
       Long lines (132 columns), no column 72 rule
       Alphanumeric labels
       Continuation marking at end of previous line (&)
```

```
-------------------------------------------------
                FORTRAN 8X


Obsolete Features  (X3J3 term: "Deprecated Features")

Many commonly used FORTRAN features are "replaced" with "better"
        features, mostly borrowed from more modern languages.

"Candidates for removal from subsequent revisions of FORTRAN"

Examples:

"Obsolete Feature"                Proposed Replacement

Old source form                   New form
Assumed size arrays               Assumed shape arrays
Statement functions               Internal Procedures
COMMON statement                  Global data in modules
EQUIVALENCE statement             Dynamic arrays, variant structures
DATA statement                    INITIAL attribute
ENTRY statement                   Internal Procedures in Modules
DOUBLE PRECISION type             Arbitrary precision floating point
PAUSE statement                   READ statement


Concerns

       Is replacing old features with new something users
               want to do?  Is it worh the cost of recoding?

       New features should be added compatibly with old ones.

       Many new features borrowed from new languages less popular
               than FORTRAN;  are they really better?

       Conversion cost (Time, Money, Performance)
```

------------------------------------------------

FORTRAN 8X


Several new features are needed and desirable
Examples:
        Most array extensions
        Data structures

Concerns
        Compatibility
        Obsolete features
        Performance
        Subset

------------------------------------------------

FORTRAN 8X


Performance

Some good news, some bad news

Good News

        Extensions for array processing will improve run time
            performance for array operations, especially on
            array processors and vector processors.  User does
            "vectorization", rather than the compiler.


Bad News

        Some features (including some array features) may cause
        performance degradation when used with FORTRAN 77 programs.

        Examples:

                Assumed shape arrays (hidden parameter passing)
                Dynamic arrays (Heap storage management)


        Large language will degrade compile speed

        Examples:

                Two source forms
                Arbitrary precision and range
                Module processing
                User defined overloaded operators

------------------------------------------------

FORTRAN 8X


Subset


        No subset defined in FORTRAN 8X

        Since FORTRAN 8X will replace FORTRAN 77,
                the FORTRAN 77 subset will lapse.

        Many users dislike subset implementations, feel that
                this will force full implementations


Concerns

        FORTRAN 8X is much larger language than FORTRAN 77, so
                subset implementations will abound.

        Unless a subset standard is established, users will suffer
                from incompatible subset implementations.

        Many vendors will never implement full language (e.g. PL/I)

        Even vendors that eventually offer full language will
                initially offer subset.

DECUS

LARGE SYSTEMS SIG DECUS U.S. CHAPTER
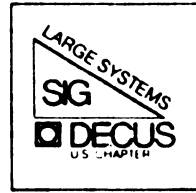
# DIGITAL EQUIPMENT COMPUTER USERS SOCIETY
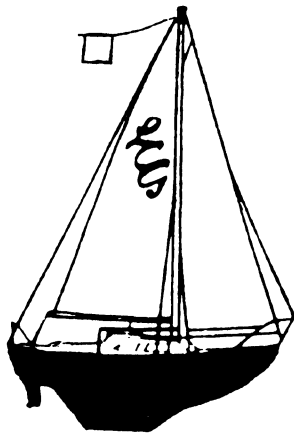
AT LARGE

LS

## The Newsletter of the Large Systems SIG

# SIG STEERING COMMITTEE

SIG Chairperson
    Leslie Maltz
    Stevens Institute of Technology
    Computer Center
    Hoboken, NJ   07030
    (201)420-5478; BITNET:LMALTZ@SITVBX; ARPANET:SIT.MALTZ@CU20B


Symposium Coordinator
    Robert C. McQueen
    Stevens Institute of Technology
    Computer Center
    Hoboken, NJ   07030
    (201)420-5454; BITNET:RMCQUEEN@SITVXB; ARPANET:SIT.MCQUEEN@CU20B


Newsletter Editor
    Michael D. Joy
    The First Church of Christ, Scientist
    Christian Science Center A30
    Boston, MA   02115
    (617)262-2300 x3903


Menu Coordinator
    Charles R.T. Bacon
    National Institutes of Health
    Building 12 B  Room 2N207
    Bethesda, MD   20205
    (301)496-4823


Hardware Coordinator
    Clive Dawson
    Microelectronics & Computer Technology Corp.
    9430 Research Blvd.; Echelon Bldg. #1, Suite 200
    Austin, TX   78759
    (512)343-0860; ARPANET/CSNET:CLIVE@MCC


Languages Coordinator
    David Edwards
    DPEX, Inc.
    240 Pamela Drive, Suite #1
    Mountain View, CA   94040
    (415)965-3739


TOPS-20 Coordinator
    Pete Galvin
    University of Texas at Austin
    Computation Center
    Austin, TX   78712
    (512)471-3241


Networks Coordinator
    Richard Janick
    Abbott Laboratories
    AP14, D-0048
    Abbott Park, IL   60064
    (312)937-4305


Systems Software Coordinator
    Betsey Ramsey
    American Mathematical Society
    P.O. Box 6248
    Providence, RI   02940
    (410)272-9500 x295


Special Projects Coordinator
    Osman Ahmad
    Assiociation of American Railroads
    3140 S. Federal St.; Technical Center, Research & Test Dept.
    Chicago, IL   60616
    (312)567-3627


DEC Counterparts
    Dave Braithwaite
    Digital Equipment Corporation
    Marlboro, MA

    Jack Buckley
    Digital Equipment Corporation
    Marlboro, MA

    Reed Powell
    Digital Equipment Corporation
    Marlboro, MA

# CHAIRPERSON'S ARTICLE

Chairperson's Article

This is a rather unique experience - writing about events as if they had already occurred. Specifically I am referring to the Spring Symposium. In order to meet publication deadlines, I will look into a virtual crystal ball to predict some of the outcomes of the symposium.

The Large Systems SIG has now entered a new phase in its life. With the acceptance of highend systems and clusters in the mainstream of our sites, the SIG sponsored several sessions oriented toward issues associated with operating and managing large systems of many flavors. Your comments on these new activities is welcomed, particularly in the form of correspondence with this newsletter. Suggestions for future sessions and activities to meet evolving needs would likewise be appreciated.

We are involved in two large scale data collection activities this spring. One is the SIG MENU, our traditional feedback mechanism that is used to identify our priorities and concerns. The other is a survey intended to provide data to be used in formulating long range plans for the SIG. Your thoughts and responses to both activities are important, and we urge you to respond promptly. Once tabulated and analyzed, the results will be shared with all.

Lastly, as we continue along our evolutionary path, we need your help in supporting an expanded array of activities. We need volunteers from both the newly defined highend environment as well as our traditional base to help support the SIG's efforts. Let's hear from you!

# DOCTOR TOPS

Dear Doctor Tops,
        Could you PLEASE, Oh PLEASE find a way for me to have my VAX VMS users select their account string at LOGIN time? I am spending too much time in AUTHORIZE doing MODIFY. I cannot take this any longer. If you cannot help, I'll just do without accounting on my VAX.
                        Battered Manager


Dear Battered Manager,
        The solution you desire is rather simple, but deserves a word of caution; Whenever the 4.x releases of VMS software come out, you MUST re-link these programs and re-install them (with CMKRNL privs) using INSTALL/COMMAND_MODE. The same programs will work under VMS 3.7 with a small amount of work. Beware the 8 character limitation on the size of the account string, and the format of the account validation file. Enclosed are the programs for BATCH and INTERACTIVE mode, as well as the SYSLOGIN.COM and SYSTARTUP.COM files.

                        Dr. Tops


```
SAMPLE00        !Note Overhead accounts are restricted-- change the code
SAMPLE01        !in the fortran file for your own site
TESTINGX
FOOBAR20
T--8600
VERYMESS
XYZIMBS
```

```
$ compile selbchbwa,setacct
$ link/notraceback selbchbwa,setacct,sys$system:sys.stb
$ copy selbchbwa.exe vms4:[sys0.hacks]
$ set prot=(w:e) sys$system:selbchbwa.exe
$ install replace sys:selbchbwa
```

```fortran
        program selbchbwa
c
c       this program validates the charge number given by a user
c       at LOGIN time or when an account is changed needs CMKRNL
c       and BYPASS privs from INSTALL utility *BATCH ONLY*
c
        logical lerror
        character*8 name,upname
        integer setacct
        open(unit=1,name='syaccount.dat',status='old',err=10,
     1  readonly,shared)
        read(1,101,end=2,err=2)i,name
101     format(1x,q,1a8)
        if(i.le.0) then
           type *,'%Account: Null account name in SYACCOUNT.DAT'
           goto 10
           endif
        call upcase(name,upname)
        if(upname.eq.'LOGOUT  ')goto 11
        if(upname.eq.'ABORT   ')goto 11
        if(ipname.eq.'QUIT    ')goto 11
        if(upname.eq.'STOP    ')goto 11
        lerror=.false.
        if(upname.eq.'OVERHEAD')goto 3
        call findacct(upname,lerror)
        if(lerror) goto 2
3       continue
        ierr=setacct(upname)     !set account
        if (ierr.ne.1) type *,'%Account: account information not changed'
        call exit
2       continue
        type *,'%Account: Illegal account name in SYACCOUNT.DAT'
        type *,'Bad data: ',upname
10      continue
11      continue
        call sys$delprc(,)       !kill myself
        end
```

```fortran
        subroutine upcase(in,out)
c
c       this routine converts lowercase into uppercase
c
        character*8 in,out
        character*1 x
        do 1 i=1,8
        x=in(i:i)                !get a character
        if((x.ge.'a').and.(x.le.'z')) then
           x=char(ichar(x)-(ichar('a')-ichar('A')))
           endif
        out(i:i)=x
1       continue
        return
        end
```

```fortran
        subroutine findacct(acctname,flag)
c
c       this routine looks for an account name in bwa.bas
c       and returns true if found and access is allowed
c
        character*8 acctname,plopnurd
        logical flag
c
c       assume ok for now
c
        flag=.false.
c
c       find bwa.bas
c
        open(unit=1,name='sys$system:bwa.bas',access='sequential',
     1  carriagecontrol='none',status='old',err=99,
     2  readonly,shared)
c
c       read the file
c
        do 1 i=1,4096
        read(1,100,end=98,err=98)plopnurd
        if(plopnurd.eq.acctname)then
           close(unit=1)
           return
           endif
1       continue
100     format(1a8)
        flag=.true.
        close(unit=1)
        return
c
c       here on end or error reading bwa.bas
c
98      continue
        flag=.true.
        close(unit=1)
        return
c
c       here on no bwa.bas
c
99      continue
        type *,'%Account: sys$system:bwa.bas not found.'
        return                   !what can you do?
        end
```

```
$ compile selectbwa.for,setacct.mar
$ link/notraceback selectbwa,setacct,sys$system:sys.stb
$ copy selectbwa.exe sys$sysroot:[hacks]
$ set prot=(w:e) sys$system:selectbwa.exe
$ install replace sys:selectbwa
```

```fortran
        program selectbwa
c
c       this program validates the charge number given by a user
c       at LOGIN time or when an account is changed needs CMKRNL
c       and BYPASS privs from INSTALL utility
c
        logical lerror
        character*39 user_name
        character*8 name,upname,filname
        character*4 charge_ok
        integer setacct
        i=lib$get_symbol('User_name',user_name)
        i=lib$get_symbol('acct_string',name)
        open(unit=1,name='syaccount.dat',
     1  status='old',readonly,shared,err=55)
        read(1,102,end=55,err=55)j,filname
102     format(1x,q,1a8)
        close(unit=1)
        goto 56
55      continue
        j=0
        filname=' '
56      continue
c       do 10 irpt=1,5   !timeout loop
c       type 100
c100    format(/,' Charge number: ',$)
c       read(6,101,end=2,err=2)i,name
c101    format(q,1a8)
        i=1
        if(name.eq.'DEFAULT ') i=0
        if((i.le.0).and.(j.le.0)) then
            type *,'%Account: Null account names are not allowed'
            goto 10
        endif
        if(i.le.0) then
            i=j
            name=filname
            type *,'Defaulting to: ',name
        endif
```

                          LS-5

```fortran
        call upcase(name,upname)
        if((upname.eq.'HELP    ').or.(upname.eq.'?       '))then
            type *,'Type a charge number of the form xxxxxxx'
            type *,'Enter STOP or QUIT or HELP or an 8 character'
            type *,'Account string for billing'
            type *,'Enter ABORT to end this session'
            goto 10
        endif
        if(upname.eq.'LOGOUT  ')goto 11
        if(upname.eq.'ABORT   ')goto 11
        if(upname.eq.'QUIT    ')goto 11
        if(upname.eq.'STOP    ')goto 11
        lerror=.false.
c
c       ream non-overheads
c       ok system
c
        if(upname.eq.'OVERHEAD') then
            if(user_name(1:1).eq.'A') goto 2
            goto 3
        endif
        call findacct(upname,lerror)
        if(lerror) goto 2
3       continue
        ierr=setacct(upname)      !set account
        if (ierr.ne.1) type *,'%Account: account information not changed'
        open(unit=1,name='syaccount.dat',
     1  status='unknown',err=12)
        write(1,111)upname
111     format(1x,1a8)
        close(unit=1)
12      continue
        i=lib$set_symbol('charge_ok','TRUE')
        call exit
2       continue
        type *,'%Account: Invalid or improper account name'
10      continue
        call exit
11      continue
        call sys$delprc(,)        !kill myself
        end
```

```fortran
        subroutine upcase(in,out)
c
c       this routine converts lowercase into uppercase
c
        character*8 in,out
        character*1 x
        do 1 i=1,8
        x=in(i:i)                 !get a character
        if((x.ge.'a').and.(x.le.'z')) then
            x=char(ichar(x)-(ichar('a')-ichar('A')))
        endif
        out(i:i)=x
1       continue
        return
        end
```
                          LS-6

```fortran
      subroutine findacct(acctname,flag)
c
c     this routine looks for an account name in bwa.bas
c     and returns true if found and access is allowed
c
      character*8 acctname,plopnurd
      logical flag
c
c     assume ok for now
c
      flag=.false.
c     find bwa.bas
c
      open(unit=1,name='sys$system:bwa.bas',access='sequential',
     1 carriagecontrol='none',status='old',err=99,
     2 readonly,shared)
c
c     read the file
c
      do 1 i=1,4096
      read(1,100,end=98,err=98)plopnurd
      if(plopnurd.eq.acctname)then
         close(unit=1)
         return
         endif
1     continue
100   format(1a8)
      flag=.true.
      close(unit=1)
      return
c
c     here on end or error reading bwa.bas
c
98    continue
      flag=.true.
      close(unit=1)
      return
c
c     here on no bwa.bas
c
99    continue
      type *,'%Account: sys$system:bwa.bas not found.'
      return                   !what can you do?
      end
```

```dcl
$ set nocontrol=Y        !Disable ^y and ^c
$ set noon               !disable error stop
$ set noverify           !disable verify in batch
$ xf := ""               !process self
$ xu := "username"       !Ask for user name
$ xn = f$getjpi(xf,xu)   !and assign to XN
$ User_Name = xn         !Copy user name
$!
$! ask for and verify accounting info
$!
$ Num_tries = 6
$ Charge_ok = "FALSE"
$ if f$mode() .eqs. "BATCH" then $ run sys$system:selbchbwa !Batch mode
$ if f$mode() .eqs. "BATCH" then $ goto ask_valid
$ if f$mode() .eqs. "OTHER" then $ run sys$system:selbchbwa !Batch mode
$ if f$mode() .eqs. "OTHER" then $ goto ask_valid
$ write sys$output " "
get_acct:
$ Inquire acct_string "Charge Number"
$ if acct_string .eqs. "" then $ acct_string := DEFAULT
$ num_tries = num_tries - 1
$ if num_tries .gt. 0 then $ goto ask_acct
$ write sys$output "Call the system manager or operator"
$ logout/full
ask_acct:
$ run sys$system:selectbwa
$ if charge_ok .nes. "TRUE" then $ goto get_acct
ask_valid:
$!
$! user made it in ok
$!
$ set control=Y          !Enable ^Y again
$ set control=T          !Enable ^T stats
$!
$! type message of the day
$!
$ write sys$output       " "
$ type sys$system:notice.txt
$ exit
```

```
$ set noon
$!
$! install various well known programs
$!
$ install := $sys$system:install /command
$ install
    create sys$system:selectbwa    /priv=(bypass,cmkrnl)    !Accounting hack
    create sys$system:selbchbwa    /priv=(bypass,cmkrnl)    !Same hack batch
$!
$! list all active devices
$!
$ show device
$!
$! done
$!
$ exit
```

```
        .title  setacct set account id
        .subtitle Dr. Tops @VMSLand Center

        .ident  /1-001/

;       this routine changes the user account string only in P1 space
;       (Stolen from the PAGESWAPPER and duly modified for 4.x)
;       (Sorry, VAX types, but BYTE BACKWARDS spins my head)
;
;
;       requires CMKRNL priv to function
;
;       input: string descriptor pointed to by AP
;
;       output: the changed P1 and system space account string
;
        .library        /sys$library:lib.mlb/
        $jibdef         ;JIB symbols
        $pcbdef         ;proccess control blocks

        .psect  _lib_code, pic,usr,con,rel,lcl,shr,exe,rd,nowrt


;
;       starting address of procedure
;
        .entry  setacct, ^M<r3,r4,r5,r8>

        sub12   #12,sp                  ;allocate local storage
        moval   (sp),-(sp)              ;create string descriptor
        pushl   #8                      ;length is 12 bytes
        pushaq  (sp)                    ;move text in two chunks
        pushaq  @4(ap)                  ;second bite
        calls   #2,g^lib$scopy_dxdx     ;copy string into this space
        blbc    r0,10$                  ;check for error
        addl    #8,sp                   ;restore used space
        movab   (sp),r8                 ;put arg addr into r8
        $CMKRNL_S routin=writeuser      ;enter kernel mode
10$:    ret                             ;thats all folks

;
;       kernel mode routine
;
        .entry  writeuser,0             ;save nothing
        movc3   #8,(r8),ctl$t_account   ;write to P1 space
        movl    ctl$gl_pcb,r0           ;get pcb addr into r0
        movl    pcb$l_jib(r0),r1        ;get addr of jib
        movc3   #8,(r8),jib$t_account(r1) ;write name to system space
        movl    #1,r0                   ;return success
        ret                             ;thats all

        .end
```

# MUMPS

# MUMPS SIG STEERING COMMITTEE

**Chairman**
Mark Berryman
Plessey Peripheral Systems
Irvine, CA

**Symposium Coordinator**
Chris Richardson
Computer Sciences Corp.
Ridgecrest, CA

**Communications Rep.**
Mark Hyde
Advanced Computing Services
DeWitt, NY

**Newsletter Editor**
Janet Berryman
Plessey Systems, Inc.
Irvine, Ca

**VAX Liaison**
Coyett A.J. Dese
VA DM&S Verification & Dev. Ctr.
San Francisco, CA

**Digital Counterparts**
Beatrice Walther
Digital Equipment Corp.
Marlboro, MA

Diane Brown
Digital Equipment Corporation
Marlboro, MA

# NETwords

## The NETWORKS SIG Newsletter

| Application | | | |
|---|---|---|---|
| Presentation | | | |
| Session | | | |
| Transport | | | |
| Network | | | |
| Data Link | | | |
| Physical | | | |
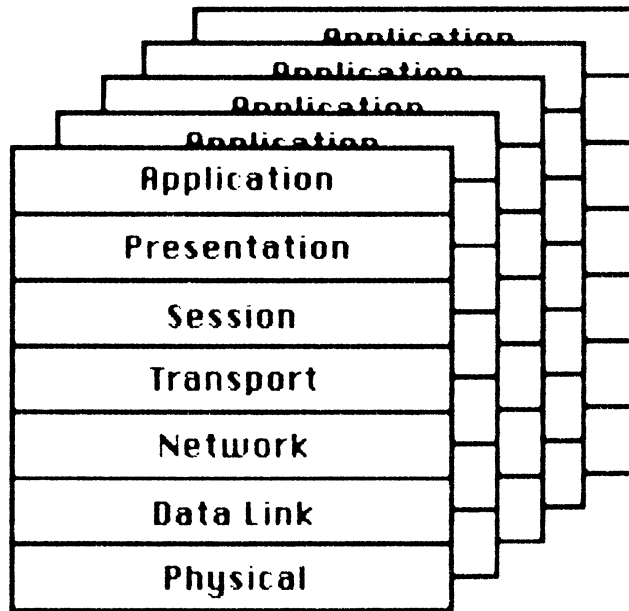
# Networks SIG Steering Committee

**Bill Brindley**
Chairman
Naval Security Group Command
(202) 282-0527

**Sandy Traylor**
Symposia Coordinator
Target Systems, Inc.
(714) 921-0112

**Jim Ebright**
Communications Coordinator
Software Results Corporation
(614) 421-2094

**Bill Hancock**
Technology/
  Standards Coordinator
(214) 495-7353

**Vickie Hancock**
Newsletter Editor
(214) 495-7353

**Carole Greenfield**
DEC Counterpart
Digital Equipment Corporation


The Networks Special Interest Group (SIG) is one of 25 SIG's within in Digital Equipment Computer User's Society (DECUS).  The main purpose of the Networks SIG is to promulgate information concerning the use, development, and standardization of network products that function or involve Digital Equipment Corporation systems.  Additional functions of the SIG include the coordination and scheduling of symposia sessions, providing methods for free-flow communications, publication of the Networks SIG newsletter NETWords, participation in domestic and international standards committees, input to Digital for new products and corrections to existing products, promotion of working groups for special network needs and topics, and many, many other functions.

The Networks SIG Steering Committee invites you to participate in the Networks SIG.  There are many ways that you can help the Networks SIG. Some of those include chairing sessions at symposium, participation in the various Networks SIG working groups, participation in special research projects, and others.  If you are interested in devoting your time and expertise, contact any of the steering committee members.

DECUS is run entirely by volunteer leadership.  Help us make DECUS and the Networks SIG better – take an active part in **your** SIG!

# OFFICE

# AUTOMATION

# OFFICE AUTOMATION SIG STEERING COMMITTEE

**Chairman**
Katherine 'Kit' Trimm
Pivotal, Inc.
Tucson, AZ

**Vice Chairman**
Ralph Bradshaw
Johnson and Johnson
Raritan, NJ

**Communications Committee Representative**
E. Catherine Ditamore
ARA Services
Philadelphia, PA

**Symposium Coordinator**
Mitch Brown
Gen Rad, Inc.
Waltham, MA

**Special Projects**
Gene LeClair
HQ Dept. of Army
Washington, DC

**BOF Coordinator**
Ray Kaplan
University of Arizona
Tucson, AZ

**Newsletter Editor**
Therese LeBlanc
Wheeling, IL

**Library**
Bob Hassinger
Liberty Mutual Research Center
Hopkington, MA

**Tape Copy Coordinator**
Randall Buck
Columbia Savings
Irvine, CA

**ALL-IN-1 Working Group**
Leon E. Ottley
Evans and Sutherland
Salt Lake City, UT

**Symposia Assistant**
Sal Gianni
Northeast Utilities
Hartford, CT

**Store Coordinator**
Mike Jackson
Air Force Operational
Test and Evaluation Center
Kirtland AFB, NM

**Personal Computer SIG Liaison**
Cheryl Johnson
Grinnell College
Grinnell, IA

**Networks SIG Liaison**
Gene LeClair
HQ Dept. of Army
Washington, DC

**DECUS Europe OA SIG**
Andreas Verbay
Telinco AG
Spiegelstrasse 20

**Digital Counterparts**
Les Agigian
Digital Equipment Corporation
Merrimack, NH

Geof Bock
Digital Equipment Corporation
Merrimack, NH

**Session Notes**
Martha Rudkin
GMF Robotics
Troy, MI

## IN THIS ISSUE. . .

## FROM THE EDITOR

We have three related 'technical' articles for you in this
issue...all by the same author!  If you have some feedback
or input for Bart on these subjects, please feel free to
share it with the rest of our readers by sending a copy to
me.

Since this issue of the newsletter follows so closely on the
heels of the Dallas Symposium, please stay tuned to the June
and July issues for post-Symposia information.  We will be
publishing the System Improvement Requests (SIR) that you
voted on in the March issue, and DEC's response to them.
Plus, we will have a new list of SIR's from Dallas for you
to vote on for the Fall Symposium.

And speaking of fall, the San Fancisco Symposium (October
6-10th) isn't far away.  If you are interested in presenting
a session, or being on a user panel, please contact:

      Mitch Brown - (617) 890-4900

before May 15th.


Regards,

*Therese*

Therese M. LeBlanc
275 London
Wheeling, IL  60090
(312) 459-1784

Preventing <u>spurious</u> <u>characters</u> <u>in</u> <u>documents</u>

<u>printed</u> <u>through</u> <u>a</u> <u>terminal</u> <u>printer</u> <u>port</u>

<u>using</u> <u>ALL-IN-1</u>

Bart Z. Lederman

Greenberg Brothers Partnerships
60 Madison Avenue Room 1101
New York, NY 10010

We are using WPS-PLUS within ALL-IN-1 (V2.0): we also happened to have an electronic typewriter which has an RS-232 interface that allows it to act as a printer, and a terminal (VT220) with a printer port. It appeared reasonable to connect them all together, as one of the destinations to which AI1 will print is a printer port on a terminal: AI1 is supposed to send the proper escape sequence to turn the printer port on and off to direct the text to the printer. We soon discovered, however, that spurious characters (always the same characters) would print at the beginning and end of every document.

I examined the script files for this function (they are in the directory OA$DO and have names like WPPPORT.SCP, WPPORTLA5.SCP, etc.), and found they sent escape sequences after turning on the printer port, and contained escape sequences that I could not find in any terminal manual, so I edited them: I made the sequence that turned on the printer port (<ESC>[5i) the last characters before the document is sent, and the sequence that turned off the printer port (<ESC>[4i) the first characters after the document is sent. For example, the relevant portion of the script file might look like this:

```
.TEXT 23,1,"<ESC>[6i<ESC>[5i"
GET OA$FUNCTION="COPY " #PRINT_OUTFILE " TT:"
.TEXT "<FF><ESC>[4i<ESC>[7i"
GET #PRINT_OPERATION_FAILED=""
```

This should have solved the problem; but although it got rid of the spurious characters at the beginning, it did not eliminate them at the end of the document. [For the benefit of those who have never edited a document with non-printing characters: the <ESC> and <FF> are not text as you see them here but one single character that EDT represents in this way. You can insert these characters in a file using the SPCINS key: press GOLD, the number of the character (27 for an <ESC> and 12 for a <FF>), GOLD, and SPCINS (the "3" key on the keypad if it hasn't been redefined). WPS-PLUS does not identify special characters, displaying all of them as "▓".

### The true cause of the problem.

At this point, I made several calls to the TSC in Atlanta. At first they said that they had heard of this problem, and suggested changes in the script files similar to my own. They then said that this solved the problem, and that they could not reproduce my reported problem. I persisted, however, and after several false starts I was eventually contacted by the local office. By the time this happened, I had run several experiments of my own using a Rainbow in terminal emulation: first as the printer to a VT220 printer port, then as the terminal itself. By capturing the text, I found that AI1 was indeed sending extra escape sequences at the end of the document before shutting off the printer port, and that there did not seem to be any way to make AI1 stop doing this. The escape sequences sent include such things as save and restore cursor position, which obviously have no relevance for a hard copy device. Atlanta probably tested only with DEC printers that were capable of ignoring these spurious escape sequences, but we were using a non-DEC printer. I should point out that the printer profile specified an LP11 type printer: this DEC device does not recognize escape sequences, so it too would have printed spurious characters, as would some other DEC devices, so the problem is not confined to those people using non-DEC devices.

### Finally, a solution.

I now had proof that AI1 was faulty (so what else is new?), but that didn't help me to make my printer function properly. Since I had continued to pester the TSC, they eventually referred it to the local (New York City) office, and I received a call from Warren Archer. He suggested the essentials of the alternative procedure given here; the details were worked out with me on my system and him on the other end of the telephone. The first step is to modify the printer port script or create a new one with the following change: the supplied scripts do a 'GET OA$FUNCTION="COPY "----' and will either have a '.TEXT "<ESC>[5i"' before it and another text sequence after it as shown in the above example or will use the AI1 commands 'PORT_ON' and 'PORT_OFF'. All of these commands will be replaced with a single command to invoke a DCL procedure which will turn the printer port on, send the text, and turn the printer port off. For example, the relevant portion of the script file might look like this:

```
DISPLAY Printing document . . .
GET OA$DCL= "@OA$LIB:PORT_C " #PRINT_OUTFILE
GET #PRINT_OPERATION_FAILED=""
```

You may put in whatever command file specification you wish, but OA$LIB is a reasonable place to put it. Note that there must be one blank space after the file specification and before the closing double quotes ("). You must now also create the corresponding command file, which in this example is PORT_C.COM:

```
$SET TERMINAL/FORM/NOWRAP
$WRITE SYS$OUTPUT "<ESC>[5i"
$TYPE/OUTPUT='F$LOGICAL( "SYS$OUTPUT") 'P1
$WRITE SYS$OUTPUT "<ESC>[4i"
$SET TERMINAL/NOFORM/WRAP
$EXIT
$!
$! Run output to AI1 terminal in lieu of WPP script
$! B. Z. Lederman as recommended by local office.
```

The terminal should be set for "/FORM" and "/NOWRAP" so it will
pass along all text without interpreting it. The next line
turns on the printer port. The third uses VMS to send the
document directly to the terminal rather than having AI1 do it:
this avoids the problem of AI1 inserting spurious escape
sequences. Note the logical function: when you are in AI1,
output goes to a mailbox and not directly to your terminal,
hence you must translate the logical assignment for SYS$OUTPUT.
The name of the file will be passed to the procedure as the
first parameter P1. There is no need to check to see if the
document exists as AI1 has already done this. Once the
document is sent, the printer port is turned off and the
terminal is reset.

All that remains is to enter AI1 from the system manager's
account and select the PRINTER profile. You can then either
change your definition of PORT to use your new script, or
define a new printer type such as PORT_C to use the new script
without changing the supplied scripts.

                    Unexpected benefits!

Even if you have DEC printers you may wish to use this
type of script anyway as it has one big advantage: you can
cancel a document being printed to your port with a Control-C.
When the print buffer runs down the procedure will exit and
return you to AI1. With the scripts supplied with AI1, once
you start to print a document on a printer port there is no way
out except for a privileged account to stop your process.

                    Re-Installing the TXL

                     for ALL-IN-1 V2.0


                      Bart Z. Lederman

              Greenberg Brothers Partnerships
               60 Madison Avenue Room 1101
                  New York, NY 10010


When trying a new script such as the printer port
replacement previously described, you may simply place it in
OA$LIB, and AI1 will find it. Once it is working, however,
there is a performance advantage to have it in the TXL. During
my testing I though everything had to be in the TXL so I was
doing a number of TXL re-installations, but found that the
script files didn't change. I also pursued this with the TSC,
and found that the manuals are wrong.

The correct command to remove a TXL while AI1 is active is
OA$TXL_REMOVE. You will not find this command in the manuals:
they tell you to do an OA$TXL_CLOSE. It may be a good idea to
close the TXL before removing it, but the close does not remove
it. You may then compile and install the TXL with the commands
shown on page 6-21 of the System Manager's Guide. You should
move your new scripts into OA$DO so they are all where AI1
expects them to be before compiling the TXL.

## Backwards output from ALL-IN-1?


Bart Z. Lederman

Greenberg Brothers Partnerships
60 Madison Avenue Room 1101
New York, NY 10010


In researching the printer port problem I came across a
very peculiar situation that I would like to pass on to the
user community for further research.

When text is sent to a device such as a terminal, each
line must be terminated with a carriage return (to move the
print head or cursor back to the left hand edge of the screen
or paper) and a line feed to move to the next line. With very,
VERY, **VERY** few exceptions, terminals used on DEC systems
expect both characters to be sent. (A DEC terminal would be
set otherwise only if it was connected to a non-DEC system.)
Because the print head on early mechanical printers (such as
telex machines) took some time to move back across the page, it
became the custom many decades ago to send the carriage return
first, then the line feed: that way, the print head could be
moving across the page while the paper moved up, and that gave
the print head time to return to the left hand edge before the
first character on the new line had to be printed. These early
printers had no buffers, so if the next character arrived too
soon it printed wherever the print head happened to be as it
moved back across the paper.

This sequence is heavily embedded in DEC hardware and
software: for example, most text files do not even have the $C_R$
(carriage return) and $L_F$ (line feed) characters in them:
instead, the file is marked with the "Carriage Control"
attribute, and a $C_R$-$L_F$ pair is assumed to exist between records:
the operating system then inserts the $C_R$ and $L_F$ when needed, such
as when a TYPE or PRINT operation is performed. Most DEC
editors will operate properly only on this type of file: EDT,
for example, will reject most files without this attribute as
the records will be 512 bytes (one block) long and EDT has a
limit of 255 characters.

I mention all of this because the text sent out to the
printer port by WPS-PLUS under AI1 has the sequence backwards:
it sends one or more line feeds followed by a single carriage
return. When I sent this text file up to the VAX to edit and
print it (and find out what was going wrong with the printer
port operation), I had a great deal of difficulty with it as
none of the DEC software would really handle it properly. All
of the software which would normally convert from one type of
text file to another expects a record to be terminated by $C_R$-$L_F$
and not by $L_F$-$C_R$: even TECO would not do the conversion
properly! I had to edit the file using the EDT script for TPU
in order to get the records down under 255 characters and

replace the internal $L_F$-$C_R$ to implied $C_R$-$L_F$, then edit with EDT to
see the actual embedded characters (TPU doesn't make the
non-printing characters visible as EDT does).

I'm not sure how much of a problem this is. I can think
of quite a few instances where outputting a document to the
printer port and capturing it on a system like a Rainbow, or
PRO, or anything else emulating a terminal, would be a very
handy way to transfer a document, especially to systems that
don't have other communications packages, or can't read WPS
files, or where trying to get Kermit or one of the commercial
communications packages to read a WPS document out of an AI1
folder would be too difficult. Unfortunately, if AI1 persists
in sending the carriage control sequence incorrectly, the
resulting document will not be usable.

I would be interested to know what the rest of the user
community thinks about this. If AI1 is in fact sending the
carriage control sequence out in the wrong order, I believe it
should be corrected, at least to be consistent with every other
piece of DEC hardware and software I've ever seen in the last
10 years; and so the output from AI1 can be captured and
processed. What do you think?

DECUS

# PERSONAL COMPUTER

# SIG NEWSLETTER
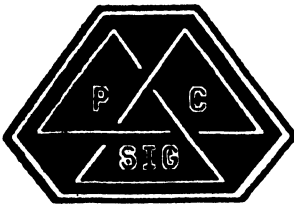
```
===========================================
```
**PERSONAL COMPUTER SIG STEERING COMMITTEE**
```
===========================================
```

Chairman

    Barbara Maaskant
    UT Health Science Center
    7703 Floyd Curl Drive
    San Antonio, TX  78284
    (512) 691-7351

Symposia Coordinator

    Rick Eliopoulis
    5258 Vickie Drive
    San Diego, CA  92109
    (619) 225-7867

DECmate Working Group Chairman

    Cheryl Johnson
    Grinnell College
    P. O. Box 805
    Grinnell, IA  50112-0810
    (515) 236-2570

Pro Working Group Chairman

    Thomas R. Hintz
    U. of Florida
    IFAS Computer Network
    1022 McCarty Hall
    Gainesville, FL 32611
    (904) 392-5181

Rainbow Working Group Chairman

    Lynn Jarrett
    Union Tribune Pub. Co.
    P. O. Box 191
    San Diego, CA  92108
    (619) 299-3131 (x1130)

Newsletter Editor/Comm Comm Rep.
    Caroline M. Mack
    9007 Mears Street
    Fairfax, VA  22031
    (703) 280-4404

Session Notes Editor

    Alan Bruns
    Allied Electronics
    401 E. 8th Street
    Fort Worth, TX  76102
    (817) 336-5401

National LUG Organization Rep

    Pierre Hahn
    SUNY  HSC-T10-02808101
    Stony Brook, NY  11794
    (516) 444-1362

Library Comm. Rep/PC Librarian

    Frederick (Fritz) Howard
    Eastman Kodak Company
    901 Elmgrove Raod D345-LP
    Rochester, NY  14650
    (716) 724-5331

Campground Coordinators

    Ron S. Hafner
    Hafner and Associates
    P. O. Box 2924
    Livermore, CA  94550
    (415) 449-4178

    Jim Wilson
    National Technical Institute for the Deaf
    Rochester Institute of Technology
    Box 9887
    Rochester, NY  14623
    (716) 475-6241

PC/RSX Liason

    Peter Flack
    Computer Sciences Corp.
    P. O. Box 12233
    Research Triangle Park, NC  27709
    (919) 541-4669

PC/Graphics Liason

    Dr. Khin Maung Yin
    Kent State University
    1411 Clarindon-Troy Road
    Burton, OH  44201
    (216) 951-1447

Pre-Symposia Session Coordinator

    Frederick G. Howard
    Eastman Kodak Company
    901 Elmgrove Road  D345-LP
    Rochester, NY  14650
    (716) 724-5331

Members-at-Large

    Jim Christine
    SPSS, Inc.
    1815 S. Cuyler
    Berwyn, IL  60402
    (312) 329-3580

    Russ Wertenberg
    Sandia National Labs
    Div 8352
    Livermore, CA  94550
    (415) 422-2663

    Michael Bowers
    University of California
    Animal Science Department
    Davis, CA  95616
    (916) 752-6136

Digital Counterparts

DECmate

    Ron Gemma
    Digital Equipment Corp.

PRO

    Lynn Olsen
    Digital Equipment Corporation
    146 Main Street
    ML21/2/U2
    Maynard, MA  01754

Rainbow

    Katrina Holman
    Digital Equipment Corp.
    LJ02/I3
    30 Porter Road
    Littleton, MA  01460
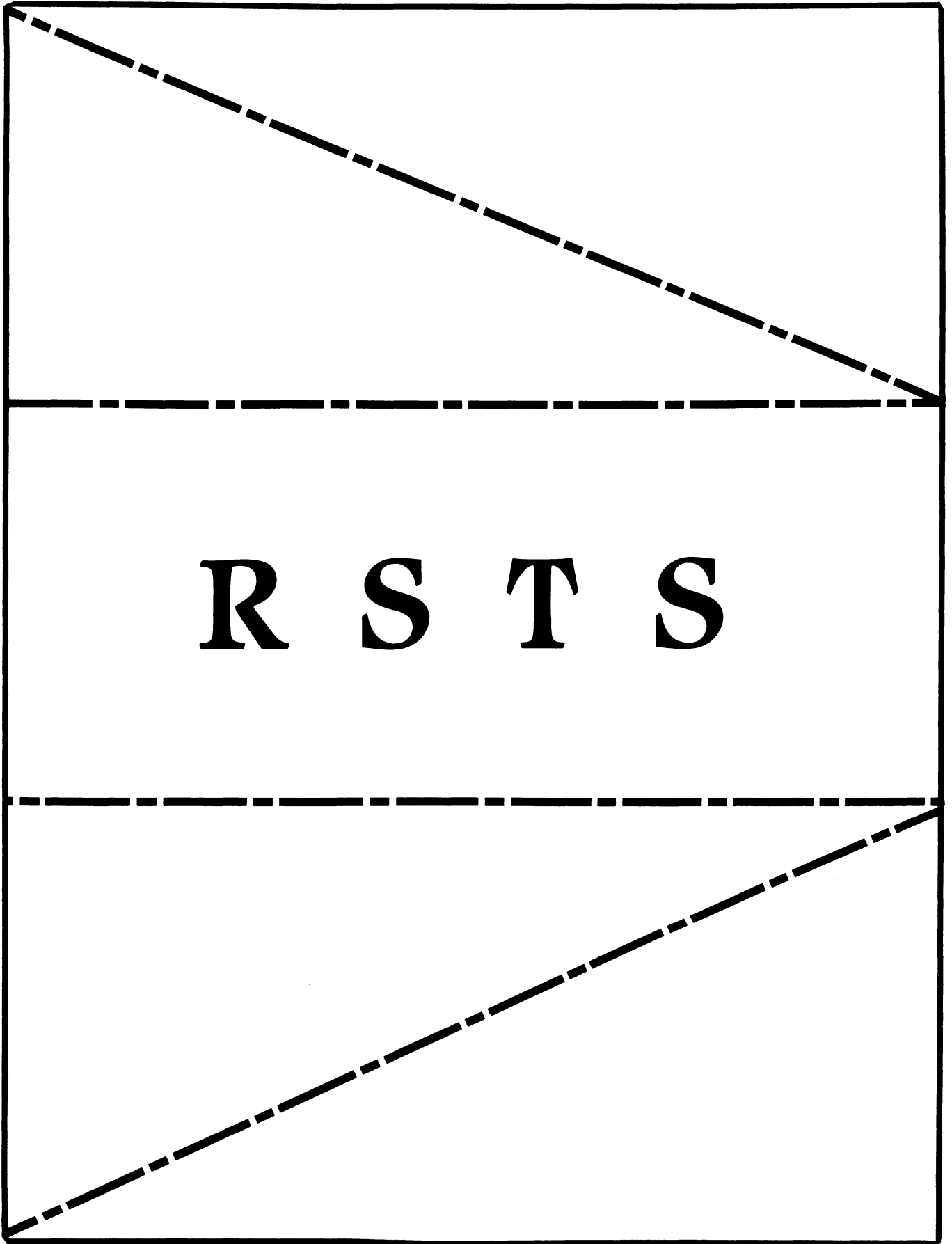
# RSTS

RST

# RSTS

**Chairman**
Charles Mustain
Stark County School System
Louisville, OH

**Symposium Coordinator**
Scott W. Pandorf
Kittle's Home Furnishings
Indianapolis, IN

**Assistant Symposium Coordinator**
Wef Fleischman
Software Techniques
Cypress, CA

**Newsletter Editor**

Open

**Library Representative**
Susan Abercrombie
Ventrex Laboratories Inc.
Portland, ME

**DEC Counterpart**
Joel Arker
Digital Equipment Corporation
Merrimack, NH

**Pre-Symposium Seminar Coordinator**
Bruce Gaarder
Macalester College
St. Paul, MN

**Wish Lists Coordinator**
Neal E. Goldsmith
Software Techniques, Inc.
Cypress, CA

**Vice SIG Chairman**
**Wish Lists & Tape Copy Coordinator**
Philip Hunt
System Industries
Milpitas, CA

**EDUSIG Liaison**
George Wyncott
Purdue University Computing Center
W. Lafayette, IN

**RSTS Product Planning Coordinator**
Errol E. Ethier
Shrewsbury, MA

**Members-At-Large**
Ed Beadel
Instructional Computer Center
Oswego, NY

Scott Daily
Great Lakes Chemical Corp.
W. Lafayette, IN

Mark Gilmore
Cal State University
Long Beach, CA

Mark Hartman
Jadtec Computer Group
Orange, CA

Jeff Killeen
Information Design & Management
Hopedale, MA

Newton J. Munson
Rochester Institute of Technology
Rochester, NY

VERSION 9.1 BUG SWATTED

Thanks go to Mark Hartman for sharing this one-bit patch for
Version 9.1. The fix will no doubt be incorporated in the next
release, but in the meantime, this has already proven crucial to
our shop's 9.1 production machine. (Ed.)


This patch re-enables privilege retention when using the .CCL EMT
from a user program.(Allows a non-privileged user to issue a DCL
command from inside a program.)

```
!
!  Patch to reenable non-privileged commands of the form:
!
!      T$ = SYS(CHR$(14%) + "$ dcl command")
!
! which were failing under RSTS/E V9.1
!
File to patch?
Module name? EMT
Base address? FSS@OVR
Offset address? 4056
 Base    Offset  Old     New?
125352  004056  020000  ? 120000
125352  004060  004767  ? ^C
```

Brian Nelson
Computer Services
University of Toledo
2801 West Bancroft
Toledo, OH 43606

In 1981 I wrote what was then called SHDVR, a DYNPRI that was set up as
a psuedo driver running in Kernel mode. This was done to reduce the
extremely high overhead incurred running a DYNPRI in user mode with the
attendant price of using executive directives such as UU.SYS and .PEEK.
The placement of SHDVR in the monitor also meant that (1) the system
was now unsupported and (2) every new version of RSTS required a
recheck of the code and modification of the sysgen command file.

With version 9, however, one can have a low overhead DYNPRI without
putting it into the monitor by mapping the read/write data area of the
executive with PLAS directives. This means that a user mode process can
access any executive data structures without extra overhead. A new
version of DYNPRI nows takes advantage of that feature. It will be on
the SPRING 1986 Decus tape, as well as being available via Kermit dialup
from the VAX 11/785 at the University of Toledo.

A complete Kermit distribution is also maintained here as is Kermit-11,
which this author wrote and supports. If you want this DYNPRI code
now,or simply want a newer Kermit, try:

            (419) 537-4401 or 537-4411

    When online, autobaud with carriage return(s),
    then at the Enter Class prompt, type: VX785A. You
    will then get a message, and you will then need
    to type a couple more carriage returns to get VMS
    to autobaud. At that point you log into user
    KERMIT password KERMIT. You will be in menus, and
    once you start the VMS Kermit server you can:

    Kermit-11>GET RSTS_FILES:DYN.MAC

As a closing note, be sure to catch the Kermit session in Dallas on
Tuesday April 29 at 10:00 am.


    Editor: For those of you who have no dial-out or no KERMIT, here
            is a listing of Brian's Dynpri.


        .title  dynpri
        .ident  /9.2.01/

        .include        /SY:[1,2]COMMON.MAC/
        .include        /SY:[1,2]KERNEL.MAC/
        .psect
        .enabl  gbl


    ;       This was, under RSTS/E v7 and 8, a pseudo device driver For

```
;    V9, we will simply map 0-117777 of  the  exec  to  get  fast
;    access to the job data.
;
;    This  is  tested under 9.1 and 9.2; for 9.1 you MUST set the
;    symbol R$$9.1 equal to 1 and rebuild the image; if  9.2  you
;    MUST  set  the  symbol R$$9.1 to zero and rebuild the image.
;    This is due to the executive data structures  that  moved  a
;    bit.
;
;
;    R$$9.1  ==       0                ; 9.2-06
;
;
;    It  COULD  need reassembly and task build for later versions
;    of RSTS/E if data structures  change.  It  could  even  stop
;    functioning  under  a  later release of RSTS/E. This warning
;    is here because this code maps executive data space.
;
;    Building:       $ mac dyn
;                    $ tkb dyn=dyn
;                    $ run dyn
;                    DYN>GO
;
;    Other commands: SET BURST1 n     Run burst for priority -8
;                    SET BURST2 n     Run burst for priority -16
;                    SET BURST3 n     Run burst for priority -24 and -32
;                    SET SLEEP n      Set scan interval
;                    KILLDET          Kills detached jobs with group > 10
;                    KILLHIB          Same as KILLDET but iif hibernating
;
;    The default:    SET BURST1 6
;                    SET BURST2 5
;                    SET BURST3 4
;                    SET SLEEP 2
;
;    The KILLxxx code is called once every five minutes
;
;    copyright (C) 1981 1982 1986 Brian Nelson
;
```
```
;    .sbttl  more comments and things
;
;    edits:
;
;    date        ver  who      why
;
;    xx-apr-81   01   bdn      initial coding
;    15-mar-82   02   bdn      Reduce df$lev to 1
;                             Check for odd addresses in case we interupted
;                             'ttijob' in the middle of job creation.
;                             Save jobtbl to reduce conflicts with 'ttijob'
;                             to nill.
;    03-jun-82   03   bdn      Force priv jobs to go down to lowest
;                             Skip push to -32 if no cputime @ -24
;
;    31-oct-82   04   bdn      allow bubble up to -16 if @ -24 and no cpu
;
;    18-nov-82   05   bdn      skip bubble up to -16 if load is too high
;                             (as i write this, my job count is 59)
;                             also raise priority if swapped, runnable
;                             and the job has been at -32 for a long
;                             time (computed from jobcount)
;
;    26-Feb-86   06   bdn      No longer a driver, map the exec.
;    27-Feb-86        BDN      Command interface.
;    01-Mar-86        BDN      9.2 Field Test support
;
;
;    Brian Nelson
;    Computer Services
;    University of Toledo
;    2801 West Bancroft
;    Toledo, Ohio 43606
;
;    (419) 537-2841
;    (419) 537-2511
;
;    This  software  is  furnished in an as-is condition, with no
;    committments of support or updates. This  software  may  NOT
;    be  sold for profit nor can it be included in any package to
;    be sold for  profit  without  the  written  consent  of  the
;    author.  This  software  may  be  used only within the above
;    conditions of use. The information in  contained  herein  is
;    subject to change or revision at any time without notice.
```

```
        .sbttl  more info                              .mcall  gtsk$s  ,qiow$s ,exit$s ,exst$s

;       Data space APR usage:                          STRING  =       0
;                                                       NUMBER  =       1
;       0       Maps image data
;       1-5     Maps executive low core                .save
;       6       Maps Job Control Region (9.2 and later) .psect  comdat  ,rw,d,lcl,rel,con
;       7       Unused                         comhea: .blkw   1
;                                              comdat:
;       This image can be linked for I/D support if you CPU supports  .restore
;       such. It may be the case in the future that this mode is the
;       only supported mode.                           $comlas =       comhea          ; Generate first link
;
;                                                       .macro  cmddef  command,dispat,nargs,type=0
;       Changing parameters
;                                                       .save                           ; save current psect context
;                                                       .psect  comdat  ,rw,d,lcl,rel,con
;       CYCLES  is  the  number  of  scan  periods  to  wait  before  $$ = .           ; Save current location
;       dropping the priority of a runnable job from -16 to  -24.  A  .asciz /command/ ; Insert command text
;       reasonable  value  is  one  (1)  or  two  (2). This value is  .even            ; Always do this
;       tightly linked to the scan interval.           $coml   =               ; save current pc in 'comdat'
;                                                       .word   0               ; link to next is zero
;       LEVEL is the minimum amount of cpu time  (in  1/10  seconds)  .word   $$      ; Insert text address
;       that  the  job must get before it's priority will be dropped  .word   dispat  ; Insert dispatch routine address
;       from -8 to -16. A resonable value is one (1) or two (2).      .word   nargs
;                                                       .word   type
;       SLEEP is the value that is placed  in  the  timeout.  Values  .even            ; must do
;       of  two  or three seconds are reasonable. Anything less than  $compc = .       ; save the current pc
;       that (ie, 1 second) and dynpri may be unable to  get  enough  . = $comlast     ; backup to link word from previous
;       useful  information  about  the  jobs  on the system to make  .word   $coml    ; insert address of new entry
;       any kind of intelligent priority choices.      . = $compc       ; restore correct pc
;                                                       $comlast = $coml ; lastlink = current_entry
;       The default values are DF$LEV, DF$CYC and DF$SLP.            .restore         ; restore old psect context
;
;                                                       .endm                           ; thats it
;       Current Command interface:
;
;       DYN>GO                                          .macro  strcat  dst,src         ; Concatonate two asciz string
;       DYN>START                                       mov     src     ,-(sp)          ; Stuff source address
;       DYN>SET SLEEP N                                 mov     dst     ,-(sp)          ; Stuff destination
;       DYN>SET BURST1 N                                jsr     pc      ,strcat         ; Do it
;       DYN>SET BURST2 N                                .globl  strcat                  ; In case
;       DYN>SET BURST3 N                                .endm   strcat                  ; All done
;       DYN>KILL

                                                        .macro  strcmp  s1,s2           ; Compare two asciz strings
                                                        mov     s2      ,-(sp)
                                                        mov     s1      ,-(sp)
                                                        call    strcmp
                                                        .globl  strcmp
                                                        .endm   strcmp


                                                        .macro  strcpy  dst,src         ; Copy asciz string
                                                        mov     src     ,-(sp)
                                                        mov     dst     ,-(sp)
                                                        jsr     pc      ,strcpy
                                                        .globl  strcpy
                            RST-5                       .endm   strcpy        RST-6
```

```
.macro  moverr  e               ; Die on error function
movb    e       ,r0             ; If no error, all is well
beq     .+6                     ;
call    error                   ; Error, it's fatal
.endm   moverr                  ; Thats all.
```

```
.macro  mapjcr  job2,dst=r0     ; Insure JCR is currently mappea
mov     job2    ,-(sp)          ; Do it
call    mapjcr                  ; Simple
mov     r0      ,dst            ; Return address of it
.endm   mapjcr                  ; Done
```

```
.macro  getexec off,base,dst=r0 ; Get a word from executive mapped
mov     base    ,r0             ; data (ie, first 20KW).
.if nb, off                     ; Again, if offset is there, us
add     off     ,r0             ; it else ignore it
.endc                           ; Addressing for executive low
add     #20000  ,r0             ; memory is biased off of APR1
mov     (r0)    ,dst            ; thus we add 20000 into every
.endm   getexec                 ; address.
```

```
.macro  save    list            ; Register saves, inline code.
.if b , <list>                  ; If no passed arguments then
.ift                            ; recursivly call ourself to
save    <r0,r1,r2,r3,r4,r5>     ; generate the instructions.
.iff                            ; Otherwise run through the
.irp    x,<list>                ; argument list.
mov     x,-(sp)                 ; ....
        .endr                   ; .....
.endc                           ;
.endm   save                    ; Alldone
```

```
.macro  unsave  list            ; Opposite of SAVE
.if b , <list>.                 ; These two macros ARE order
.ift                            ; dependent, ie:
unsave  <r5,r4,r3,r2,r1,r0>     ;   SAVE <R1,R2>
.iff                            ;   .....
.irp    x,<list>                ;   UNSAVE <R2,R1>
mov     (sp)+,x                 ; Generate the code in-line
        .endr                   ;
.endc                           ;
.endm   unsave                  ; All done
```

```
.macro  clrfqb                  ; Clear the FIRQB out
call    $clrfq                  ; Do it
.endm   clrfqb                  ; Thats all

.macro  clrxrb                  ; Clear the XRB out
call    $clrxr                  ; ...
.endm   clrxrb                  ; All done


.macro  putdec  v               ; Print a number on the terminal
mov     r0      ,-(sp)          ; Save a register
mov     v       ,r0             ; Stuff passed value into it
call    putdec                  ; Do the work
mov     (sp)+   ,r0             ; Pop something
.endm   putdec                  ; and all done

.macro  putoct  v               ; Save as PUTDEC but in octal
mov     r0      ,-(sp)          ; format. Save a register
mov     v       ,r0             ; Load value
call    putoct                  ; Dump and exit
mov     (sp)+   ,r0             ; Pop the register
.endm   putoct                  ; Done


.macro  getlin  s               ; Read a line from the terminal
mov     s       ,-(sp)          ; Stuff string address
call    getlin                  ; Call someone to read the data
tst     (sp)+                   ; Pop arg
.endm   getlin                  ; Done

.macro  putbin  s,1             ; Dump (in pass-all mode) to
.if b   ,1                      ; the terminal. Can be invoked as:
.ift                            ;       PUTBIN #STRING
clr     -(sp)                   ; or:
.iff                            ;       PUTBIN #STRING,LENGTH
mov     1       ,-(sp)          ; Passed length, stuff it on stack
.endc                           ;
mov     s       ,-(sp)          ; Push string address onto stack
call    putbin                  ; Dump it
cmp     (sp)+   ,(sp)+          ; Pop parameter list
.endm   putbin                  ; Done

.macro  getval  s               ; Parse a asciz string thats
mov     s       ,r0             ; supposedly a number. GETVAL
call    getval                  ; returns 'C' set on error.
.endm   getval                  ; Done
```

```
BADDIR  =:      1
BADNAM  =:      2
INUSE   =:      3
NOROOM  =:      4
NOSUCH  =:      5
NODEVC  =:      6
NOTCLS  =:      7
NOTAVL  =:      10
NOTOPN  =:      11
PRVIOL  =:      12
EOF     =:      13
ABORT   =:      14
DATERR  =:      15
HNGDEV  =:      16
HNGTTY  =:      17
FIEXST  =:      20
DTOOOF  =:      21
BADFUO  =:      22
INTLCK  =:      23
WRGPAK  =:      24
NOTMNT  =:      25
PAKLCK  =:      26
BADCLU  =:      27
PRIVAT  =:      30
INTPAK  =:      31
BADPAK  =:      32
DETKEY  =:      33
CTRLCE  =:      34
SATTBD  =:      35
DEVNFS  =:      36
BADCNT  =:      37
NOBUFS  =:      40
B.4     =:      41
B.10    =:      42
B.250   =:      43
B.STAK  =:      44
B.SWAP  =:      45
B.PRTY  =:      46
MAGSEL  =:      47
MAGRLE  =:      50
NRRTS   =:      51
```

```
        .sbttl   the ddb defined

;        Some of this is historical, I am leaving it in place
;        so I won't have to change other code here.

        .psect   rwdata   ,rw,d,lcl,rel,con
        .dsect

ddidx:  .blkb                        ; driver index
ddsts:  .blkb                        ; status and access control byte
ddjbno: .blkb                        ; owner job number times 2 (0 if free)
ddunt:  .blkb                        ; device unit number
ddtime: .blkw                        ; time assigned or inited
ddcnt:  .blkw                        ; init count and assignment control
ddflag: .blkw                        ; device flags
ddlev:  .blkw                        ; cpu time level word
ddcyc:  .blkw
ddslp:  .blkw                        ; time to wait between scans
ddburl: .blkw                        ; burst for level 1 priority
ddbur2: .blkw                        ; burst for level 2 priority
ddbur3: .blkw                        ; burst for level 3 priority
ddpri:  .blkw                        ; save the job's priority
state:  .blkw
elptim: .blkw                        ; j2cpu-timevector
savtim: .blkw
ddcycv: .blkw    64.                 ; cycle count vector for all jobs
ddtimv: .blkw    64.                 ; lsw of total job time for all jobs
ddjobt: .blkw    64.                 ; save old jobtbl here
        .blkw    1                   ; reserved
ddend   =        .
ddsize  =        . - ddidx

        .psect   rwdata   ,rw,d,lcl,rel,con
ddb:    .blkb    ddsize+2
myjob2: .word    0                   ; My job numbers times two
mapsts: .word    0                   ; Status word
mapid:  .word    0                   ; Region ID of low memory mapping
apr6:   .word    0                   ; Current apr6 mapping address (in mmu's)


        MYJCR    =:      1030        ; My offset into APR6 mapping
        MMUJCR   =:      1032        ; Fixed location of JCR base in MMU's


jcrsiz: .word    0                   ; Size of a JCR
$$jcr6: .word    0                   ; MMU address of first JCR
jcmapid:.word    0                   ; Region ID of JCR mapping
tim.sh: .word    0                   ; Timeout value
$jobtbl:.word    0                   ; Save JOBTBL address here
$jbstat:.word    0                   ; Save JBSTAT address here
$jbwait:.word    0                   ; Save JBWAIT address here
$jobcnt:.word    0                   ; Save JOBCNT address here

cmdbuf: .blkb    80.                 ; Command buffer
argptr: .blkw    40.                 ; Command arg addresses
kbiost: .word    0,0,0,0             ; I/O Status block for read QIO's
kildet: .word    0                   ; If <> then purge jobs
kiltmo: .word    0                   ; If it's time to kill jobs
kilcnt: .word    0                   ; Ditto
```

```
hibreq: .word    0                   ; If <>, require killed jobs to be in HB

        .psect   code     ,ro,i,lcl,rel,con

df$cyc  =        1                   ; default for minimum cycles
df$lev  =        1                   ; default for minimum cpu time
df$slp  =        2                   ; default for scan sleep time
priorl  =        -8.                 ; normal job priority
prior2  =        -16.                ; and then some
prior3  =        -24.                ; even lower still
prior4  =        -32.                ; the pits
burstl  =        6.                  ; default runburst for priorl
burst2  =        5.                  ; default runburst for prior2
burst3  =        4.                  ; default runburst for prior3
waitb   =        32766.              ; mask for jbwait
```

```
        .sbttl  get started

        cmddef  START   ,cmdgo  ,0              ; Trivial
        cmddef  GO      ,cmdgo  ,0              ; Ditto...
        cmddef  KILLDET ,cmdkil ,0              ; For deleting jobs
        cmddef  KILLHIB ,cmdhib ,0              ; Delete iif HIBERNATING
        cmddef  SET     ,cmdset ,2,STRING       ; Setting parameters
        cmddef  BURST1  ,cmdbu1 ,1,NUMBER       ; Run burst setting
        cmddef  BURST2  ,cmdbu2 ,1,NUMBER       ; ....
        cmddef  BURST3  ,cmdbu3 ,1,NUMBER       ; ....
        cmddef  SLEEP   ,cmdsle ,1,NUMBER       ; Sleep time

start:: call    init                    ; Init the world
10$:    call    getcmd                  ; Get a command
        tst     r0                      ; Success?
        beq     10$                     ; No
        jsr     pc      ,@r0            ; Yes, do the command now
        br      10$                     ; And get another


        .enabl  lsb

cmdgo:: putbin  #200$                   ; Detach
        CLRFQB                          ; Insure cleared out
        movb    #UU.DET ,FIRQB+FQFUN    ; Do it
        .UUO                            ; ...
        mov     #5*60.  ,r1             ; intervals for waiting to kill
        clr     r0                      ; Setup for divide
        div     tim.sh  ,r0             ; Get interval count
        mov     r0      ,kiltmo         ; Save it
        mov     r0      ,kilcnt         ; Init counter
10$:    mov     tim.sh  ,XRB+0          ; Sleep
        .SLEEP                          ; Nap time
        call    tmo$sh                  ; Go do some work
        call    kill                    ; Perhaps delete jobs
        br      10$                     ; Some more

        .save
        .psect  rwdata  ,d
200$:   .asciz  /Detaching/<15><12><15><12>
        .even
        .restore
        .dsabl  lsb
```

```
        .sbttl  More commands

cmdkil: mov     sp      ,kildet         ; For deleting detached jobs
        return                          ; Exit

cmdhib: mov     sp      ,kildet
        mov     sp      ,hibreq
        return

        .enabl  lsb

cmdset: sub     #120    ,sp             ; Allocate a buffer
        mov     sp      ,r4             ; And a pointer for it
        clrb    @r4                     ; Null string
        mov     #argptr ,r5             ; Argument pointers
10$:    strcat  r4      ,(r5)+          ; Move first arg into CMDBUF
        tst     (r5)                    ; Move next arg address back
        beq     20$                     ; And concat the next arg
        strcat  r4      ,#200$          ; Stuff a space in there
        br      10$                     ; Next please
20$:    strcpy  #cmdbuf ,r4             ; Copy back into command buffer
        call    prsbuf                  ; Find the command
        tst     r0                      ; Success?
        beq     90$                     ; No
        cmp     10(r1)  ,#STRING        ; Should we look for a value
        beq     30$                     ; No
        getval  argptr                  ; Yes, do it please
        bcs     90$                     ; Oops
30$:    mov     4(r1)   ,r1             ; Go do the command
        jsr     pc      ,@r1            ; Call it
        br      100$                    ; Exit
90$:    putbin  #210$                   ; Error
        br      100$                    ; Exit now
100$:   add     #120    ,sp             ; Pop buffer and exit
        return

        .save
        .psect  rwdata  ,d
200$:   .asciz  / /
210$:   .asciz  /Bad SET command or bad number/<15><12>
        .even
        .restore
        .dsabl  lsb
        .enabl  lsb


cmdbu1: mov     #ddbur1 ,r1
        br      setpar
cmdbu2: mov     #ddbur2 ,r1
        br      setpar
cmdbu3: mov     #ddbur3 ,r1
        br      setpar
cmdsle: mov     #ddslp  ,r1
        br      setpar
setpar: getval  argptr
        bcs     90$
        mov     r0      ,ddb(r1)
        putbin  #210$
```

```
        putdec  ddb(rl)
        putbin  #220$
        br      100$
90$:    putbin  #200$
100$:   return

        .save
        .psect  rwdata  ,d
200$:   .asciz  /Bad number/<15><12>
210$:   .asciz  /Parameter set to /
220$:   .byte   15,12,0
        .even
        .restore
        .dsabl  lsb
```

```
        .sbttl  getcmd
        .enabl  lsb

getcmd: putbin  #200$                   ; A prompt
        mov     #cmdbuf ,r5             ; Point to a buffer
        getlin  r5                      ; Read something
        tst     r0                      ; Anything?
        beq     120$                    ; No, the read failed
prsbuf: mov     #argptr ,r4             ; Arg pointers
        mov     #cmdbuf ,r5             ; Point to a buffer
        clr     (r4)                    ; Init the field
        clr     r2                      ; Arg count
        clr     r3                      ; Flag
        tstb    @r5                     ; Read ok, read anything?
        beq     110$                    ; No
10$:    tstb    (r5)                    ; End of the line yet?
        beq     50$                     ; Yes, exit please
        mov     r3      ,(r4)           ; Stuff new address for args?
        beq     15$                     ; No
        tst     (r4)+                   ; Yes, and point to NEXT field
        clr     (r4)                    ; Insure NEXT is NULL
        clr     r3                      ; No longer need new address
        inc     r2                      ; Count args
15$:    cmpb    (r5)    ,#'A!40         ; Convert case?
        blo     20$                     ; No
        cmpb    (r5)    ,#'Z!40         ; Well?
        bhi     20$                     ; No
        bicb    #40     ,(r5)           ; Yep, make it upper case
20$:    cmpb    (r5)    ,#40            ; A space in command line?
        bne     30$                     ; No
        mov     r5      ,r3             ; Get set to insert next addr
        clrb    (r3)+                   ; Yes, turn it into a NULL
30$:    inc     r5                      ; Point to NEXT character
        br      10$                     ; Get next character

50$:    call    findcm                  ; Get command matched up
        tst     r0                      ; Success
        bne     100$                    ; Yes, exit
        putbin  #210$                   ; Nothing, error message time
        br      110$                    ; Exit

100$:   cmp     r2      ,6(rl)          ; Enough command line args?
        bge     105$                    ; Yes
        putbin  #220$                   ; No, error time.
        clr     r0                      ; Failure
105$:   return                          ; Bye

110$:   clr     r0                      ; No data, return failure
        return

120$:   exst$s  #EX$SUC

        .save
        .psect  rwdata  ,d
200$:   .asciz  /DYN>/
210$:   .asciz  /Unknown command/<15><12>
220$:   .asciz  /Insufficient arguments/<15><12>
        .even
        .restore
```

```
        .dsabl  lsb


findcm: mov     r2      ,-(sp)      ; Save this
        mov     comhea  ,r2         ; Get listhead for commands
10$:    strcmp  2(r2)   ,#cmdbuf    ; Find the command?
        tst     r0                  ; Well?
        beq     100$                ; Yes, exit with address in R0
        mov     (r2)    ,r2         ; No, get next command in the list
        bne     10$                 ; Something was there
        clr     r0                  ; Failure
        br      110$                ; Exit
100$:   mov     4(r2)   ,r0         ; Success, return dispatch addr
        mov     r2      ,r1         ; Return block address also
110$:   mov     (sp)+   ,r2         ; Pop it
        return                      ; Exit
```

```
        .sbttl  init code

init:   mov     #^R...  ,FIRQB+FQNAM1   ; Change our name
        mov     #^RDYN  ,FIRQB+FQNAM1+2 ; to ...dyn
        .NAME                           ; Trivial
        mov     #ddb    ,r1             ; /05/May as well leave the code alone
        mov     #ddcyc  ,r2             ; first, clear out accumulated
        mov     #ddend  ,r3             ; cycle count and total jobtime
        add     r1      ,r2             ; point to the cycles vector
        add     r1      ,r3             ; point to the end of the DDB
10$:    clr     (r2)+                   ; clear a word from ddb
        cmp     r2      ,r3             ; reach the end of it all ?
        blo     10$                     ; no
        mov     r1      ,r2             ; stuff the ddb address in
        add     #ddlev  ,r2             ; point to the area
        mov     #df$lev ,(r2)+          ; set default cputime minimum
        mov     #df$cyc ,(r2)+          ; set default cycle count.
        mov     #df$slp ,(r2)+          ; default sleep
        mov     #burst1 ,(r2)+          ; burst for level 1
        mov     #burst2 ,(r2)+          ; burst for level 2
        mov     #burst3 ,(r2)+          ; burst for level 3
        mov     #df$slp ,tim.sh         ; enable once a second timeouts

        movb    #UU.TB1 ,FIRQB+FQFUN    ; Get some needed base address
        .UUO                            ; ....
        movb    FIRQB+2 ,myjob2         ; Save my job number also
        mov     FIRQB+14,$jobtbl        ; Save base address of JOBTBL
        mov     FIRQB+16,$jbstat        ; Save base address of JBSTAT
        mov     FIRQB+20,$jbwait        ; Save base address of JBWAIT
        movb    #UU.TB2 ,FIRQB+FQFUN    ; More needed addresses
        .UUO                            ; ...
        mov     FIRQB+16,$jobcnt        ; Save base address of JOBCNT

        CLRFQB                          ; Clear FIRQB out please
        movb    #CRAFQ  ,FIRQB+4        ; Create address window
        movb    #1      ,FIRQB+7        ; Use APR 1
        mov     #40*24  ,FIRQB+12       ; Map 20 KW please
        .PLAS                           ; Do it
        moverr  FIRQB                   ; If failure then DIE
        movb    FIRQB+6 ,mapid          ; Save ID for the window
        CLRFQB                          ; Clear the firqb out please
        movb    #MAPFQ  ,FIRQB+4        ; Map the exec data structures
        movb    mapid   ,FIRQB+6        ; Window ID
        mov     #-4     ,FIRQB+14       ; Flag a map of physical memory
        mov     #40*24  ,FIRQB+20       ; 20 KW to map, starting at 0
        .PLAS                           ; and readonly access
        moverr  FIRQB                   ; If failure then DIE

        .IF EQ  ,R$$9.1                 ; If 9.2, we have to map the JCR
        .IFT
        CLRFQB                          ; Create address window for APR6,
        movb    #CRAFQ  ,FIRQB+4        ; which will be user to map JCR's
        movb    #6      ,FIRQB+7        ; Use APR 6
        mov     #40*4   ,FIRQB+12       ; Map 4 KW please
        .PLAS                           ; Do it
        moverr  FIRQB                   ; If failure then DIE
        movb    FIRQB+6 ,jcmapid        ; Save ID for the window
        moverr  FIRQB                   ; If failure then DIE
```

```
        getexec base=#MYJCR,dst=r1      ; Get my JCR APR6 mapping
        sub     #140000 ,r1             ; Compute JCRSIZ now
        clr     r0                      ; ...
        div     myjob2  ,r0             ; Done
        mov     r0      ,jcrsiz         ; Save JCR size please
        getexec base=#MMUJCR,dst=$$jcr6 ; Save base address (in MMU's) for JCR
        .ENDC                           ; for 9.2 support
100$:   return
```

```
                                        .sbttl  MAP a JCR entry


        ;       Passed: 2(sp)   job number times 2
        ;       Return: R0      Address, already biased for APR6 mapping
        ;
        ;       Code for determining offsets taken from GETJCR using DOB
        ;       Double check this code for new releases of RSTS/E
        ;
        ;       I'm not real sure, but I think that the value @1032 will
        ;       not always be the same as $$JCR6, as the value @1030 can
        ;       not exceed 147200,  thus I will refetch the JCR MMU base
        ;       address (in MMUs) after possibly recomputing the offset.
        ;       Will try to confirm from disassembly or (if lucky) DEC.


mapjcr::save    <r1,r2>                         ; Save registers
        mov     $$jcr6  ,r2                     ; Get MMU address of JCR region
        mov     2+4(sp) ,r1                     ; Get job number times 2
        mul     jcrsiz  ,r1                     ; Compute offset
10$:    cmp     r1      ,#17200                 ; See if it fits in 4KW mapping
        blo     20$                             ; It does
        add     #172    ,r2                     ; No, so move up a bit
        sub     #17200  ,r1                     ; The '17200' is from GETJCR
        br      10$                             ; Check again please

20$:    cmp     r2      ,apr6                   ; Is this mapping current?
        beq     100$                            ; Yes, no need to remap then.
        CLRFQB                                  ; Clear the firqb out please
        movb    #MAPFQ  ,FIRQB+4                ; Map the exec data structures
        movb    jcmapid ,FIRQB+6                ; Window ID
        mov     #-4     ,FIRQB+14               ; Flag a map of physical memory
        mov     r2      ,FIRQB+16               ; MMU address to map into
        mov     #40*4   ,FIRQB+20               ; 4KW window
        .PLAS                                   ; and readonly access
        moverr  FIRQB                           ; If failure then DIE
        mov     r2      ,apr6                   ; Save mapping context
100$:   mov     r1      ,r0                     ; Return JCR address in R0
        add     #140000 ,r0                     ; Done
        unsave  <r2,r1>                         ; Pop registers
        mov     (sp)+   ,(sp)                   ; Move return address up and exit
        return                                  ; And exit now


        .assume ddcyc   eq      <ddlev+2>
        .assume ddslp   eq      <ddlev+4>
        .assume ddbur1  eq      <ddlev+6>
        .assume ddbur2  eq      <ddbur1+2>
        .assume ddbur3  eq      <ddbur2+2>
```

```
        .sbttl  kill nonpriv detached jobs

;       I know that  we should base this on some job priv but
;       this program must never exceed 4KW. Also, we are still
;       using the old [1,*] convention.

        .iif ndf, MAXGRO        ,MAXGRO = 10

kill::  tst     kildet                  ; Delete things?
        beq     100$                    ; No
        dec     kilcnt                  ; Time to scan?
        bne     100$                    ; No
        mov     kiltmo  ,kilcnt         ; Yes, reset
        clr     r5                      ; Job number
10$:    inc     r5                      ; Next job please
        CLRFQB                          ; Insure cleared out
        movb    #UU.SYS ,FIRQB+FQFUN    ; Get some job data
        movb    r5      ,FIRQB+4        ; Insert JOB number
        .UUO                            ; Get the job data
        movb    FIRQB   ,r1             ; Success?
        bne     20$                     ; No, ignore it
        bitb    #200    ,FIRQB+5        ; Detached?
        beq     20$                     ; No
        cmpb    FIRQB+27,#MAXGRO        ; Allowed to be detached?
        blos    20$                     ; Yes, ignore
                                        ;
        tst     hibreq                  ; Require job to be hibernating?
        beq     15$                     ; No
        CLRFQB                          ; Insure cleared out
        movb    #UU.SYS ,FIRQB+FQFUN    ; Get some job data
        movb    r5      ,FIRQB+4        ; Insert JOB number
        incb    FIRQB+5                 ; Subfuction 1
        .UUO                            ; Get the job data
        movb    FIRQB   ,r1             ; Success?
        bne     20$                     ; No, ignore it
        tst     FIRQB+14                ; Is JBWAIT entry zero?
        bne     20$                     ; No, not hibernating then
                                        ;
15$:    CLRFQB                          ; Trash the firqb please
        movb    #UU.CHU ,FIRQB+FQFUN    ; Kill a job call
        movb    r5      ,FIRQB+4        ; Job number to delete
        movb    #377    ,FIRQB+35       ; Flag kill call
        .UUO                            ; Do it
20$:    cmpb    r1      ,#BADFUO        ; Time to leave?
        bne     10$                     ; No
100$:   return                          ; Exit
```

```
        .sbttl  timeouts

;       tmo$sh
;
;       r0      =       unit number times 2 (always 0)
;       r1      >       ddb/fcb


rot     =       180.                    ; patchable, 'rot' cycle count

tmo$sh::mov     #ddb    ,r1             ; Point to r/w data
        mov     ddslp(r1),tim.sh        ; reset the timeout please
        mov     r0      ,-(sp)          ; save a few registers
        mov     r2      ,-(sp)          ;
        mov     r3      ,-(sp)          ;
        mov     r4      ,-(sp)          ;
        mov     r5      ,-(sp)          ;
        clr     r5                      ; r5 is the current job number
10$:    add     #2      ,r5             ; next job please
        cmpb    r5      ,myjob2         ; Don't fix myself up please
        beq     10$                     ; Ignore
        getexec base=$jobtbl,off=r5,dst=r2
        beq     10$                     ; no job of this number active
        cmp     r2      ,#-1            ; is this the end of jobtbl ?
        beq     20$                     ; yes, so exit back to RSTS
        mov     r1      ,r3             ; check to see if this job is the
        add     r5      ,r3             ; same as one as last time to avoid
        cmp     ddjobt(r3),r2           ; potential confict with 'ttijob'
        bne     15$                     ; job creation in ttdvr. skip if ne
        mov     r2      ,-(sp)          ;
        call    dyn                     ; no, check this job out please
        mov     (sp)+   ,r2             ;
15$:    mov     r1      ,r3             ; save the jobs jdb address for the
        add     r5      ,r3             ; next time thru
        mov     r2      ,ddjobt(r3)     ; save jdb address
        br      10$                     ; next job please
20$:    mov     (sp)+   ,r5             ; pop the registers now please
        mov     (sp)+   ,r4             ;
        mov     (sp)+   ,r3             ;
        mov     (sp)+   ,r2             ;
        mov     (sp)+   ,r0             ;
        return                          ; for now
```

```
        .sbttl   check out the current job (r2 > jdb)

;       register usage
;
;       rl      >       ddb/fcb
;       r2      >       jdb+20000
;       r3      >       jdb2
;       r4      >       ddb + jobnumbertimes2
;       r5      =       job number*2


dyn:    add     #20000  ,r2             ; Correct for APR1 biasing
        bit     #JFNOPR ,JDFLG(r2)      ; Job not logged in ?
        bne     100$                    ; no, could have been 'ttijob'
        mov     rl      ,r4             ; point to the timvec offset by
        add     r5      ,r4             ; the job number times two.
        .IF NE  ,R$$9.1                 ; If 9.0 or 9.1, the JBD2 is
        .IFT                            ; in low core, else map the jcr
        mov     JDJDB2(r2),r3           ; address of job's JDB2 block
        add     #20000  ,r3             ; Correction for APR1 basing
        bit     #1      ,r3             ; avoid crash if 'ttijob' was
        bne     100$                    ; interupted in the middle.
        mov     J2CPU(r3),r0            ; compute elasped cpu time.
        .IFF                            ; This must be 9.2
        MAPJCR  job2=r5                 ; So thus map the user's JCR
        bit     #1      ,r0             ; Insure valid please
        bne     100$                    ; Not valid, then exit asap
        mov     JCCPU(r0),r0            ; Fetch the cpu time.
        .ENDC                           ;
        mov     r0      ,SAVTIM(rl)     ; Save it please
        sub     DDTIMV(r4),r0           ; subtract off time last used.
        mov     r0      ,ELPTIM(rl)     ; save elapsed cpu time now.
        .IF NE  ,R$$9.1                 ; Old RSTS/E today?
        .IFT                            ; Yes, priority is in the JDB
        clr     r0                      ; r3 now has pointer to jdb2
        bisb    JDPRI(r2),r0            ; the job's current priority
        .IFF                            ; No, we must map the JCR instead
        mov     rl      ,-(sp)          ; Save this
        MAPJCR  job2=r5,dst=rl          ; Map this jobs JCR please
        clr     r0                      ; r3 now has pointer to jdb2
        bisb    JCPRI(rl),r0            ; And extract priority now
        mov     (sp)+   ,rl             ; Pop rl
        .ENDC                           ;
        cmp     r0      ,#128.          ; would like this as a signed
        blt     10$                     ; integer please
        sub     #256.   ,r0             ; make it into -128....128
10$:    mov     r0      ,DDPRI(rl)      ; we may need this later
        bge     100$                    ; > -8 priority, so do nothing
        cmp     r0      ,#PRIOR4        ; at the lowest priority we
        blt     100$                    ; are looking at today ?
        bit     #4      ,r0             ; job has special run priority?
        bne     100$                    ; yep
        bit     #JFCC!JF2CC!JFSPCL,JDFLG(r2) ; a special condition exist
        beq     20$                     ; no
        jmp     d.rst                   ; yes, restore job to -08 prior

20$:    clr     -(sp)           RST-23; next figure out if stalled
```

```
        getexec base=$jbstat,off=r5,dst=-(sp) ; mov JBSTAT(job*2),-(sp)
        getexec base=$jbwait,off=r5,dst=-(sp) ; mov jbwait(job*2),-(sp)
        bit     (sp)+   ,(sp)+          ; Is the job stalled for something?
        bne     30$                     ; Yes
        com     (sp)                    ; No, so lag as being runnable
30$:    getexec base=$jbwait,off=r5     ; It's stalled, but why is it?
        bit     #WAITB  ,r0             ; ....
        bne     40$                     ; Stalled for non disk waits.
        clr     (sp)                    ; Flag as being stalled
40$:    mov     (sp)+   ,STATE(rl)      ; save the 'job stalled' flag
        br      d.chek                  ; ok, set new priority up please

100$:   return
```

```
        .sbttl   actually do the changes now.

;       r1       >       ddb / fcb
;       r2       >       jdb
;       r3       >       jdb2

d.chek: inc      DDCYCV(r4)                 ; at last
        mov      DDPRI(r1),r0               ; get the current job priority
        sub      #PRIOR4 ,r0                ; change -32...-8 to 0..24
        ash      #-3     ,r0                ; /8 to    0...3
        dec      r0                         ; to      -1...2
        dec      r0                         ; to      -2...1
        bgt      d.hip                      ; job is at the highest prior
        tst      STATE(r1)                  ; at a lower priority, is the
        bne      d.rst                      ; stalled (but not for FIP)
        tst      r0                         ; again, 0 if priority eq -16
        beq      d.midp                     ;
        blt      d.lowp                     ; again, lt 0 if prior lt -16

d.hip:  clr      DDCYCV(r4)                 ; clear out job's cycle count
        tst      STATE(r1)                  ; is the job stalled ?
        bne      skipjb                     ; yes, skip this one then.
        cmp      ELPTIM(r1),DDLEV(r1)       ; did they get the minimum time
        ble      skipjb                     ; no, skip the job then
        mov      #PRIOR2 ,-(sp)             ; yes, drop the job's priority
        mov      DDBUR2(r1),r0              ; and set the new run burst
        br       d.chan                     ; do it then

d.midp: cmp      DDCYCV(r4),DDCYC(r1)       ; lower priority, have we made
        ble      skipjb                     ; a reasonable (?) # of passes?
        mov      #PRIOR3 ,-(sp)             ; perhaps, so drop the priority
        mov      DDBUR3(r1),r0              ; some more and set run burst.
        br       d.chan                     ; and change it !

d.lowp: cmp      #PRIOR4 ,DDPRI(r1)         ; for very long cpu bound jobs.
        beq      20$                        ; priority is currently -32 ?
        mov      #PRIOR4 ,-(sp)             ; -24. assume move down to -32
        tst      ELPTIM(r1)                 ; any cpu time accumuated here?
        bne      10$                        ; yes, allow drop to -32 then
        mov      #PRIOR2 ,@sp               ; no, move up to -16 for a while

10$:    clr      r0                         ; no change for the run burst.
        br       d.chan                     ; and change it

20$:    tst      ELPTIM(r1)                 ; no cpu time increment yet ?
        bne      skipjb                     ; yes, do not bubble up 1 level.
        mov      DDPRI(r1),-(sp)            ; no, so set new prior eq old
        getexec  base=$jobcnt               ; Get current jobcount
        swab     r0                         ; Get count into low byte
        bic      #^C377  ,r0                ; Drop the garbage
        asl      r0                         ; how long to wait before we drop
        asl      r0                         ; the cpu time checking of a job
        asl      r0                         ; at priority -32
        cmp      DDCYCV(r4),r0              ; ok, how long at lowest prior?
        blos     30$                        ; ok so far
        .IF NE   ,R$$9.1                    ; If 9.1, swap location is in JDB
        .IFT                                ; otherwise its in JOB CONTROL REGION
```

```
        tstb     JDSWAP(r2)                 ; been below -8 for a while, are we
        .IFF                                ; Must be 9.2 or later
        MAPJCR   job2=r5                    ; Map the JCR please
        tstb     JCSWAP(r0)                 ; And see if they are swapped out.
        .ENDC                               ; for 9.1 vs 9.2
        beq      40$                        ; swapped. if so, avoid sched bug.
30$:    add      #8.     ,(sp)              ; priority plus 8
40$:    clr      r0                         ; no change in run burst though.
        br       d.chan                     ; and do it

d.rst:  mov      #PRIOR1 ,-(sp)             ; restore priority to sys def.
        mov      DDBUR1(r1),r0              ; same for the job's run burst.
        clr      DDCYCV(r4)                 ; and clear out cycle count.

d.chan: bic      #^C3     ,DDPRI(r1)        ; mask off exec's priority bits
        bisb     DDPRI(r1),(sp)             ; and stuff those bits in.
        CLRFQB                              ; Clear the FIRQB out please
        mov      #FIRQB+FQFUN,r2            ; Setup to change the job settings
        movb     #UU.PRI ,(r2)+             ; Stuff the subfunction code
        movb     r5      ,(r2)              ; Job number times 2
        asrb     (r2)+                      ; But we don't want job*2 here
        incb     (r2)+                      ; We want to change the priority
        movb     (sp)+   ,(r2)+             ; The new priority setup
        movb     r0      ,1(r2)             ; New run burst
        beq      10$                        ; Don't change it
        incb     (r2)                       ; Flag we want to change it
10$:    .UUO                                ; At last
        .br      skipjb

skipjb: mov      SAVTIM(r1),DDTIMV(r4)      ; save the current cpu time
        return                              ; at last
```

```
        .sbttl  odds and ends

$clrxr::save    <r0>                    ; Clear the XRB out
        mov     #xrb    ,r0             ; Point to it
10$:    clr     (r0)+                   ; Clear a word
        cmp     r0      ,#xrb+14        ; See if done
        blos    10$                     ; Not yet, try again
        unsave  <r0>                    ; Pop registers
        return                          ; and exit


$clrfq::save    <r0>
        mov     #FIRQB  ,r0
10$:    clr     (r0)+
        cmp     r0      ,#FIRQB+36
        blos    10$
        unsave  <r0>
        return

putoct: save    <r3>
        mov     #$cbomg ,r3
        br      pcom

putdec: save    <r3>
        mov     #$cbdmg ,r3

pcom:   save    <r1,r2>
        sub     #20     ,sp
        mov     r0      ,r1
        mov     sp      ,r0
        clr     r2
        jsr     pc      ,@r3
        mov     r0      ,r1
        mov     sp      ,r0
        sub     r0      ,r1
        qiow$s  #io.wal,#5,,,,,<r0,r1>
        add     #20     ,sp
        unsave  <r3,r2,r1>
        return

putbin: save    <r0,r1>
        mov     2+4(sp) ,r0
        mov     4+4(sp) ,r1
        bne     20$
        mov     r0      ,r1
10$:    tstb    (r1)+
        bne     10$
        sub     r0      ,r1
        dec     r1
20$:    qiow$s  #io.wal,#5,,,,,<r0,r1>  ; rsx and p/os
        unsave  <r1,r0>
        return

        .enabl  lsb

error:  putbin  #200$                   ; VERY crude error reporting
        putdec  r0
        putbin  #210$
```

RST-27

```
        exst$s  #EX$SEV

        .save
        .psect  rwdata  ,d
200$:   .asciz  /Exec directive returned error /
210$:   .byte   cr,lf,0
        .even
        .restore
        .dsabl  lsb
        .enabl  lsb


getlin: mov     2(sp)   ,r0             ; Read a command line
        QIOW$S  #io.rvb,#5,,,#kbiost,,<r0,#80.>
        cmpb    kbiost  ,#IS.SUC
        bne     100$
        add     kbiost+2,r0
        clrb    (r0)
        mov     #1      ,r0
        br      110$
100$:   movb    #32     ,(r0)+
        clrb    (r0)
        clr     r0
110$:   putbin  #200$
        return

        .save
        .psect  rwdata  ,d
200$:   .byte   12,0
        .restore
        .dsabl  lsb
```

RST-28

```
        .sbttl  strcat and strcpy

;       input:
;                0(sp)     return address
;                2(sp)     dst address
;                4(sp)     src address
;       output: r0        dest address


strcpy::save    <rl>                        ; save temp registers please
        mov     2+2(sp) ,r0                 ; destination address
        mov     2+4(sp) ,rl                 ; source .asciz address
10$:    movb    (rl)+   ,(r0)+              ; copy until a null
        bne     10$                         ; not done
        mov     2+2(sp) ,r0                 ; return the dst address
        unsave  <rl>                        ; pop rl and exit
        mov     (sp)    ,4(sp)              ; move return address up now
        cmp     (sp)+   ,(sp)+              ; pop junk and exit
        return


strcat::save    <rl>                        ; save temp registers please
        mov     2+2(sp) ,r0                 ; destination address
        mov     2+4(sp) ,rl                 ; source .asciz address
5$:     tstb    (r0)+                       ; look for the end of the dst string
        bne     5$                          ; not found yet
        dec     r0                          ; found it, fix the pointer
10$:    movb    (rl)+   ,(r0)+              ; copy until a null
        bne     10$                         ; not done
        mov     2+2(sp) ,r0                 ; return the dst address
        unsave  <rl>                        ; pop rl and exit
        mov     (sp)    ,4(sp)              ; move return address up now
        cmp     (sp)+   ,(sp)+              ; pop junk and exit
        return

strcmp::mov     2(sp)   ,r0                 ; Pick up 'a'
        mov     4(sp)   ,rl                 ; And 'b'
10$:    cmpb    (r0)+   ,(rl)               ; Are they the same
        bne     20$                         ; No
        tstb    (rl)+                       ; At the end of the string
        bne     10$                         ; No
        clr     r0                          ; Equal return
        br      100$

20$:    blo     30$                         ; Br if a<b
        mov     #1      ,r0                 ; A>b return
        br      100$

30$:    mov     #-1     ,r0                 ; A<b return
100$:   mov     (sp)    ,4(sp)              ; Move return address up now
        cmp     (sp)+   ,(sp)+              ; Pop junk and exit
        return
```

```
        .sbttl  more utility stuff

;       Passed: R0      string
;       Return: R0      vaule, C clear on success

getval: save    <rl,r2>                     ; Save temps
        clr     rl                          ; Init accum
5$:     cmpb    @r0     ,#40               ; Eat leading spaces
        bne     10$                         ; No
        inc     r0                          ; Next
        br      5$                          ; ...
10$:    clr     -(sp)                       ; Positive
        cmpb    @r0     ,#'-                ; Leading minus sign?
        bne     20$                         ; No
        com     (sp)                        ; Yes, flag
        inc     r0                          ; Next char now
20$:    tstb    @r0                         ; End of it all ?
        beq     100$                        ; Yes, exit
        cmpb    @r0     ,#'0                ; In range ?
        blo     90$                         ; No, thats an error
        cmpb    @r0     ,#'9                ; Keep looking
        bhi     90$                         ; Error again
        mul     #12     ,rl                 ; Ok, shift over old number
        bvs     90$                         ; Overflowed
        movb    (r0)+   ,r2                 ; Get number
        sub     #'0     ,r2                 ; Convert
        add     r2      ,rl                 ; And add the digit in
        bcc     20$                         ; Ok, get another digit

90$:    tst     (sp)+                       ; Pop sign indicator
        sec                                 ; Error exit
        br      120$                        ; Exit
100$:   mov     rl      ,r0                 ; Success, return result
        tst     (sp)+                       ; Negative??
        beq     110$                        ; No
        neg     r0                          ; Make < 0
110$:   clc                                 ; Clear for success
120$:   unsave  <r2,rl>                     ; Pop registers and exit
        return                              ; Bye


crlfou: putbin  #crlf                       ; Simple carriage return/line feed
        return                              ; Dumper

        .save                               ; Save current psect context
        .psect  rwdata  ,d                  ; Switch to data space psect
crlf:   .byte   15,12,0                     ; Data
        .even                               ; Nice to use word boundaries
        .restore                            ; Pop back to code psect


        .end    start
```

A small enhancement to V9 SYSTAT
===================================

By Mark Hartman, JadxDtec Computer Group, Orange CA  714 997 8927


By now many of you have installed RSTS V9 and have noticed the nice new VAX-
like print headers on spooled files.  "Wouldn't it be nice," many of you say,
"to have these mysterious names which print on the spool headers show up on
line so we can finally see just who 67,1 really is?"

Well, wait no longer, RSTS fans.  The following patch to SYSTAT will give you
this feature.  Just type this file into your system, APPEND it to the file
SOURCE$:SYSTAT.BAS, and compile with BASIC+, CSPCOM, or BASIC-PLUS-2.

A sample SYSTAT ($SHOW USERS) listing follows.

==============================
```
10010    PRINT #0%,C$;CVT$$(RIGHT(SYS(CHR$(6%)+CHR$(9%)+CHR$(0%)),3%),4%); &
         " status at ";DATE$(0%);", ";TIME$(0%);" Up: "; &
         \ J%=PEEK(36%)/1000%+2% &
         \ J2%=(PEEK(512%)/1000% <> PEEK(36%)/1000%) &
         \ J2%=635%+((J%/4%*4%)=J%) IF J2% &
                \ T=((PEEK(512%)-PEEK(36%)-J2%)*1440.+ &
         PEEK(38%)-PEEK(514%))*60.+60.-(PEEK(516%) AND 255%) &
         \ GOSUB 14000 &
         \ PRINT #0%,S$ &
         \ J%=0% &
         \ PRINT #0%,C$; &
         "Job";TAB(6%);"Who";TAB(12%);"Name";TAB(26%);"Where";TAB(33%); &
         "What";TAB(41%);"Size";TAB(47%);"State";TAB(56%);"Run-Time"; &
         \ IF PRIV.TUNE% THEN &
                 PRINT #0%, TAB(66%);"Pri/RB";TAB(75%);"RTS" &
             ELSE   PRINT #0%, TAB(68%);"RTS" &
! PRINT THE SYSTEM HEADER INCLUDING THE CURRENT DATE AND &
! TIME (USING MINUTES TILL MIDNIGHT - SECONDS UNTIL NEXT MINUTE) &
! AND LENGTH OF TIME THE SYSTEM'S BEEN UP. &
! CHECK FOR CHANGE OF YEAR, ADJUST FOR LEAPYEAR. &
! NOTE: 635 (636 IF START DATE WAS A LEAPYEAR) IS NOT A TYPO: &
! E.G., 31-DEC-78 -> 01-JAN-79 &
!      DATE$(8365) -> DATE$(9001) --> 9001-8365=635+1 &
! PEEK(n)         CONTENTS &
! 36              IDATE = DATE SYSTEM LAST STARTED &
! 38              ITIME = TIME SYSTEM LAST STARTED &
! 512             DATE = CURRENT DATE &
! 514             TIME = CURRENT TIME &
! 516 (BYTE)      TIMSEC = SECONDS TO NEXT MINUTE &
! CLEAR THE JOB # VARIABLE. &
! PRINT THE JOB STATUS HEADING &
! INCLUDE PRIORITY/RUNBURST IF USER HAS SYSIO PRIVILEGE &

10060    C%=INSTR(1%,S$,",") &
         \ C%=3% IF C%=0% &
         \ PRINT #0%, S0$;SPACE$(5%-C%);S$; &
         \ PRINT #0%, SPACE$(8%-(LEN(S$)+(4%-C%))); &
         \ PRINT #0%, FNACCOUNT.NAME$(PROJ%,PROG%);" "; &

10065    S$="KB" &
```
                                    RST-31

```
\ IF INTFAC%=8% AND J2%<>-1% THEN &
        S$="PK" &
\       J2%=J2%-P3% &
\       C%=(PEEK(PEEK(P2%+J2%*2%)+2%) AND 255%)/2% &
\       IF C%<>0% THEN &
                S$="P"+NUM1$(J2%)+"J" &
\               J2%=C% &
! PRINT THE PPN OR TAG. &
! SET THE DEFAULT TO KB. &
! IF INTERFACE TYPE IS 8 AND NOT DETACHED THEN &
!       RESET TO 'PK' &
!       ADJUST UNIT # TO REFLECT ACTUAL PK #. &
!       LOOK AT THIS PK'S DDB AND GET THE DDJBNO. &
!       IF THE DEVICE IS NOT 'FREE' THEN &
!               SET UP 'P'#'J' STRING &
&
10070    S$=S$+NUM1$(J2%) &
\ S$=S$+"*" IF TTINTF% AND 16384% &
\ S$="Det" IF J2%=-1% &
\ PRINT #0%,TAB(26%);S$;TAB(33%);RAD$(PEEK(JDB2%+12%)); &
         RAD$(PEEK(JDB2%+14%));TAB(39%); &
         FNN$(4%+PRIV.TUNE%,(PEEK(JDB%+22%) AND 255%)); &
\ PRINT #0%,"/";FNN$(2%,(PEEK(JDB%+30%) AND 255%)); IF PRIV.TUNE% &
\ PRINT #0%,"K";TAB(47%); &
\ T%=PEEK(M%(15%)+J%) &
\ S$="RN" &
\ GOTO 10110 IF (T% AND PEEK(M%(13%)+J%))<>0% &
\ S$="RS" &
\ GOTO 10110 IF PEEK(JDB%+24%)=0% AND &
(PEEK(JDB%+22%) AND -256%)=(NOT 15359%) &
\ T%=T% AND (NOT 16384%) IF (T% AND (NOT 16384%))<>0% &
\ S$="BF" &
\ GOTO 10110 IF (T% AND 16384%)<>0% &
\ S$="SL" &
\ GOTO 10080 IF (T% AND 8192%)=0% &
\ S$="SR" IF (PEEK(JDB%+4%) AND 255%)=5% &
\ GOTO 10110 &
! PRINT KB INFO - ADD A * IF IT'S A DIAL-UP LINE (MODEM) &
! PRINT THE JOB NAME (J2NAME) AND THE CURRENT SIZE, &
! IN K (M.SIZE). &
! PRINT THE JOB'S MAXIMUM SIZE, IN K (JDSIZM) IF THE &
! USER'S PRIVILEGED. &
! T%=JBWAIT &
! IN GENERAL, SET S$('STATE')=STRING WHICH WILL BE PRINTED &
! IF THE TEST WHICH FOLLOWS IMMEDIATELY IS SATISFIED. &
!       ! POSSIBLE STATES            CONDITIONS &
!       RN                      (JBWAIT AND JBSTAT)<>0 AND &
!                               'ACTUAL' STATE IS NOT DESIRED. &
!       RS (RESIDENCY WAIT)     IF M.PHYA=0% AND &
!                               M.CTRL=LCK,SWP AND IN &
! (TURN OFF BUFFER WAIT BIT IF ANY OTHERS ARE ON. CHECK FOR THE &
! FOLLOWING STATES BY LOOKING AT JBWAIT.) &
!       BF (BUFFER WAIT)        IF BIT 14 IS ON &
!       SL (SLEEP WAIT)         IF BIT 13 IS ON. &
!       SR (RECEIVER WAIT)      IF BIT 13 IS ON AND ERROR CODE &
!                               IS 5 &

10130    PRINT #0%,S$; &
\ T0=PEEK(JDB2%+2%) &
```
                                    RST-32

```
\  T0=T0+65536. IF T0<0. &
\  T0=T0+65536.*(SWAP%(PEEK(JDB2%+10%)) AND 255%) &
\  T=INT(T0/10.) &
\  T0=T0-10.*T &
\  GOSUB 14000 &
\  PRINT #0%,FNS$(8%);".";CHR$(48%+T0); &
\  S$=NUM1$(PEEK(JDB%+28%)*256%/256%) &
\  PRINT #0%,FNS$(5%);"/";NUM1$(SWAP%(PEEK(JDB%+28%)) AND 255%); &
            IF PRIV.TUNE% &
\  T%=PEEK(JDB%+12%) &
\  PRINT #0%,TAB(65%-(PRIV.TUNE%*9%)-((NOT PRIV.TUNE%)*2%)); &
            RAD$(PEEK(T%+2%));RAD$(PEEK(T%+4%)) &
\  GOTO 10020 &
!  PRINT THE SWAPPING INFO. &
!  PRINT THE ACCUMULATED CPU TIME (J2CPU AND J2CPUM). &
!  GET THE PRIORITY FROM JDPRI. &
!  LEFT THEN RIGHT SHIFT IT TO GET THE PROPER SIGN. &
!  PRINT THE PRIORITY IF THE USER'S PRIVILEGED. &
!  PRINT THE RTS NAME THEN GET OUT &

15010   DEF* FNACCOUNT.NAME$(PROJ%, PROG%) &
\       ON ERROR GOTO 15020 &
\       Z$ = SPACE$(13%) &
\       LSET Z$ = "<No name>" &
\       LSET Z$ = CVT$$(MID(SYS(CHR$(6%)+CHR$(-25%)+CHR$(-1%)+CHR$(5%)+ &
                   CHR$(PROG%)+CHR$(PROJ%)+STRING$(16%,0%)+ &
                   "SY"+STRING$(2%,0%)),8%,13%),5%) &
                        UNLESS PROJ%=0% AND PROG%=0% &
\       GOTO 15030 &

15020   RESUME 15030 &
el
15030   FNACCOUNT.NAME$ = Z$ &
\       ON ERROR GOTO 19000 &
\       FNEND &
!       RETURN THE NAME OF THE ACCOUNT &

        ==============================


RSTS V9.0-14 Jadtec status at 11-Oct-85, 02:42 PM Up: 40:45:44

Job   Who    Name         Where   What    Size   State   Run-Time  Pri/RB   RTS
 9+   22,1   PanAm/C'trans  KB22   AUD    18/64K  TT          13.9   -8/18   ...RSX
10   150,1   Inventory      KB15   INCMAS  5/64K  KB           6.6   -8/18   BASIC
11+    1,10  Warren Nagler  KB32   UTLMGR 24/64K  RN          36.8  -15/18   ...RSX
12-    7,4   WP/S. Breese   KB35   ...WPS 22/64K  KB          55.7   -8/18   ...RSX
13+   94,1   JadAccounting  KB21   USMENU  8/64K  KB          12.8   -8/18   BASIC
14+    1,12  Mark Hartman   KB29   ...SYS 20/64K  RN Lck    1:08.8   -8/18   ...RSX
15-    7,21  WP/W. Nagler   KB33   ...WPS 22/64K  KB All      49.1   -8/18   ...RSX
17   151,1   Service Calls  KB37   JCSCDC 13/64K  KB           8.7   -8/18   BASIC
18-    7,7   WP/Strong      KB18   ...SAT 27/64K  RN          47.9   -8/18   ...RSX
19     1,9   Jack Lloyd     KB34   DCL     4/64K  ^C A09       6.0   -8/18   DCL


        ==============================
```

# RSX

# MULTI-TASKER



DECUS

# RSX

**Chairman**
Dan Eisner
Perkin-Elmer Corp.
Garden Grove, CA

**Vice-Chairperson**
Elizabeth Bailey
Tennessee Valley Authority
Muscle Shoals, AL

**Symposium Coordinator**
Rick Sharpe
Toledo Edison
Toledo, OH

**Pre-Symposium Seminar Coordinator**
Hans Jung
Associated Press
New York, NY

**Communications Committee Representative**
Allen Bennett
Lear Siegler Rapistan
Grand Rapids, MI

**Newsletter Editor**
Dominic J. DiNollo
Loral Electronics
Yonkers, NY

**Store Coordinator**
Bob Freeborn
Savin Corporation
Binghamton, NY

**Session Note Editor**
Burt Janz
Northern Telecom Inc.
Concord, NH

**Librarian**
Glenn Everhart
Mt. Holly, NJ

**Campground Coordinator**
Jerry Ethington
Prolifix Inc.
Frankfort, KY

**DEC Counterparts**
Tim Martin
Digital Equipment Corporation
Maynard, MA

Dick Day
Digital Equipment Corporation
Nashua, NH

Bruce Webster
Digital Equipment Corporation
Nashua, NH

**Working Group Coordinator**
Ed Cetron
Center for Biomedical Design
Salt Lake City, UT

**Working Group Chair**
Evan Kudlajev
Philadelphia Electric Co.
Philadelphia, PA

**RSX Group Chair Software Clinic Coord.**
Roy S. Maull
U.S. Air Force
Offutt AFB, NE

**Software Clinic Coordinator**
Bruce Zielinski
RCA
Moorestown, NJ

**Volunteer Coordinator**
Gary Maxwell
U.S. Geological Survey
Menlo Park, CA

**Multi-Processors Working Group Coordinator**
Bruce Mitchell
Machine Intelligence & Indus. Magic
Hudson, WI

**Networks Working Group Coordinator**
Mark Podany
Case Western Reserve University
Cleveland, OH

**SRD Working Group Coordinator**
Bob Turkelson
Goddard Space Flight Center
Greenbelt, MD

**Accounting & Performance Working Group Coord.**
Denny Walthers
American McGaw
Irvine, CA

**Menu Coordinator**
Ed Cetron
Center for Biomedical Design
Salt Lake City, UT

**Members-At-Large**
Jim McGlinchey
Warrenton, PA

Jim Neeland
Hughes Research Labs.
Malibu, CA

Anthony E. Scandora, Jr.
Argonne National Laboratory
Argonne, IL

Ralph Stamerjohn
Creve Coeur, MO

Jim Hopp
Carleton Financial Comp.
South Bend, IN

Table of Contents

From the Editor

How to Contribute to the Multitasker

The Multitasker publishes articles and notes on all topics   dealing
with  or relating to RSX based systems.  If you are doing something
new or innovative with RSX we would like to hear from you.

Please send all correspondence for publication in machine   readable
form.   A  list  of  acceptable media and formats follow.   If these
formats  are  a  problem  for  someone,  please  call.    Alternate
arrangements such as electronic transfer may be possible.

| | | |
|---|---|---|
| Magnetic Tape: | 1600/6250 BPI | BRU, PIP, FLX, VMS BACKUP |
| Floppy Disk: | 8 Inch | RX01/RX02 ODS-1 or ODS-2 |
| | 5 Inch | RX50 |
| TU58 Cart | | |
| TK50 Cart | | |

Please send all Multitasker submissions and correspondenc
to:

Dominic DiNollo
Loral Electronic Systems
Engineering Computer Center
Ridge Hill
Yonkers, New York 10710
(914) 968-2500 ext 2210

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

Dear Dominic:

I would like to know about the format of BRU tapes, in particular how end-of-volume is handled. We have only an 800 BPI tape drive and often must convert 1600 BPI BRU tape to 800 BPI. This usually requires finding someone with an 800/1600 tape drive plus a big empty disk and some spare time. However, we have access to an IBM mainframe with all types of tape drives which can make record-by-record copies of various density tapes. This method works all right until it has to split one input tape into two output tapes, and then it must know how BRU terminates each tape. My feeling is that it is much like an ANSI tape set.

I would appreciate hearing from anyone could suggest an algorithm for splitting BRU tapes. This may become more important in the future if symposium tapes start appearing at 6250 BPI.

David Villeneuve
Division of Physics M23A
National Research Council
Ottawa, Ontario K1A OR6

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

Dear Mr. DiNollo:

One "Macro Trick" has been overlooked for so long that it is nearly obsolete. To enter "Polish Mode", the threaded code version of the FORTRAN compiler generates:

```
        JSR    R4,$POLSH
        .WORD  RUNTIME-ADDRESS
        .
        .
where                                        .
        $POLSH TST    (SP)+
               JMP    @(R4)+
```

After nearly 15 years, someone might have noticed that $POLSH is not necessary because the following in-line sequence is better:

```
        MOV    (SP)+,R4
        JSR    R4,@(PC)+
        .WORD  ROUTINE-ADDRESS
```

(This is of course a tiny improvement, but it does reduce the population of runtime routines by one)

The addressing modes of the PDP-11 are so convenient for threaded code one might wonder whether Gordon Bell had the technique in mind when he implemented them. I have found a variety of "direct threaded code" to be useful at times. Letting R4 be the interpretive program counter, define:

```
        ENTER    =    4437        ;"JSR R4,@(PC)+"
        EXIT::  MOV    (SP)+,R4    ;Drop a level of code
                JMP    @(R4)+      ;and continue

        EMERGE:: RTS    R4
        NEXT     =    134         ;"JMP @(R4)+"
```

The usual kind of threaded code routines can be written by starting
with ENTER and ending with EXIT.

      TH1:    ENTER, SWAP, DUP, EXIT

"EMERGE" is a threaded code routine that returns  control  directly
after its use:

      ENTER

      ... Threaded Code Statements ...

      EMERGE

      ... MACRO Statements

      NEXT

Thus "EMERGE" serves the same  purpose  as  ".+2"  in  the  FORTRAN
threaded code, but using ENTER-EXIT and EMERGE-NEXT as defined here
allows nested calls to any depth.

The PDP-11 instruction set is so flexible that  it  allows  several
schemes for realizing threaded code.  I have not seen this one used
before, and it has the advantages of being short,  fast,  and  uses
only  one  register.   ENTER,  EMERGE,  and  NEXT  do not alter the
condition codes, and EXIT preserves the C-bit so indicators may  in
some cases be returned by routines very cheaply.

                Very Truly Yours,

        M.B Clausing
        Symbolic Systems
        Mason, Ohio 45050

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

Dear Dom:

I am currently maintaining a device polling  system  whose  various
functions  are  initiated  by  operator  entered  commands.   I  am
currently using DEC routines such as GMCL and  CSI  to  handle  the
parsing  of  the  command  lines.  I am interested in modifying the
software to use TPARS as the command parser.  The reason  for  this
is for easy migration to a MicroVAX II.

The examples in chapter 7 of the I/O Reference Manual are not  only
very confusing but are not applicable to our system.

I would appreciate any information you can provide as  to  where  I
can find real-time examples and/or references to RSX TPARS.

                Thank you,

        Domenic Cutillo
        Toys 'R' Us
        395 West Passaic St.
        Rochelle Park, N.J. 07622

Macro-11 Tricks


Terry Dutcher
Electronic Data Systems FC
1600 N. Beauregard Street
Alexandria, Virginia 22311


This note is in response to the Macro-11 trick that appeared in the
December 85 issue of the Multitasker.

Being an oldtime RSXer, I thought the method the author of WHEE
used to find the UCB address was a little harder than necessary.
There are 2 easier methods for finding it that come to mind. The
first is to assign a logical unit to the device you wish to make
priviledged and then extract the UCB from your task header. The
second is to get the TI: UCB from the task TCB. The code that
follows demonstrates both ways.

```
        .TITLE  SETPRV - Set a terminal to privileged
;
        .IDENT  /V1/
        .ENABLE LC
;+
;                   Copyright (C) 1985 Terry Dutcher
;
;================================================================;
;                                                                ;
;           THIS PROGRAM IS PROVIDED ON AN "AS IS" BASIS ONLY.   ;
;                                                                ;
; The author disclaims all warranties on the program, including without ;
; limitation, all implied warranties of merchantablity and fitness.     ;
;                                                                ;
; Full permission and consent is hereby given to reproduce, distribute, ;
; and publish and to permit others to reproduce in whole or in part, in ;
; any form and without restriction, this program and any information    ;
; relating thereto freely but not to sell commercially.                 ;
;                                                                ;
;================================================================;
;
; Purpose:
;
;       To set the privilege bit in a terminals UCB.
;
; Input:
;
;       jsr     pc,setprv
;
;       with the symbols TTLUN, TTNAM, and TTNMB defined as follows
```

```
;       TTLUN   - logical unit number to be assigned to the terminal
;       TTNAM   - ASCII device name i.e. TT, VT, HT etc.
;       TTNMB   - unit number of the device
;
; Output:
;
;       U2.PRV of the terminal U.CW2 is set
;
; Local Symbols:  none
;
; Local Macros:  none
;
; Local Data:  none
;
; Subroutines:  none
;
; Files:  n/a
;
; Algorithm:
;
;       assign a logical unit to the terminal that is to be set /priv
;       if assign successful then
;           switch to system state
;           pick up my task header address
;           extract the UCB address from my header
;           set the priviledge bit
;           return to user state
;       endif
;       return to caller
;
; Restrictions / Warnings:
;
;       This code assumes the task was built /PR:5.
;       The symbols TTLUN, TTNAM, and TTNMB are assumed to be defined.
;-
        .library        /lb:[1,1]exemc/
        .mcall  hdrdf$, ucbdf$
        hdrdf$                          ;define task header stuff
        ucbdf$                          ;define UCB stuff

        .mcall  alun$s

setprv::
        mov     r0,-(sp)                ;save r0 for latter
        alun$s  #ttlun,#ttnam,#ttnmb    ;assign the device to ttlun
        cmpb    $dsw,#is.suc            ;did we do it right ?
        bmi     10$                     ;if minus, no
        swstk$  10$                     ;;go to system state
        .ifdf   r$$mpl                  ;;is this M+ ?
        .ift                            ;;yes
        mov     $sahdb,kisar6           ;;map to secondary pool
        mov     $sahpt,r0               ;;get M+ task header
```

```
        .iff                                ;;no
        mov     $headr,r0                   ;;get M task header
        .endc
        mov     h.lun+<<ttlun-1>*4>(r0),r0  ;;get the UCB
        bis     #u2.prv,u.cw2(r0)           ;;set /priv=ttn:
        rts     pc                          ;;back to user state
10$:
        mov     (sp)+,r0                    ;restore r0
        rts     pc                          ;return to where ever
        .end
```

The advantage of this code is that you can set any terminal to a priviledged state. The disadvantage is that it uses a logical unit and you may not have one to spare. The next routine gets around that.

```
        .TITLE  SETPRV - Set TI: to priviledged state
;
        .IDENT  /V2/
        .ENABLE LC
;+
;               Copyright (C) 1985 Terry Dutcher
;
;==============================================================================;
;                                                                             ;
;          THIS PROGRAM IS PROVIDED ON AN "AS IS" BASIS ONLY.                 ;
;                                                                             ;
; The author disclaims  all warranties on the program, including without ;
; limitation, all implied warranties of merchantablity and fitness.          ;
;                                                                             ;
; Full permission and consent is hereby given to reproduce,  distribute, ;
; and publish and to permit others to reproduce in whole or in part, in  ;
; any form and without restriction, this program and any information         ;
; relating thereto freely but not to sell commercially.                      ;
;                                                                             ;
;==============================================================================;
;
; Purpose:
;
;       To set /priv=ti:
;
; Input:
;
;       jsr     pc,setprv
;
; Output:
;
;       Tasks TI: is made priviledged
;
; Local Symbols:  none
;
; Local Macros:  none
;
; Local Data:  none
;
; Subroutines:  none
;
; Files:  n/a
;
; Algorithm:
;
;       go to system state
;       pick up tasks TCB
;       get TI: UCB from TCB
;       set priv bit
;       return to user state
;       return to caller
;
; Restrictions / Warnings:
;
;       Assumes task was built /PR:5
;-
        .library        /lb:[1,1]exemc/
        .mcall  tcbdf$, ucbdf$
        tcbdf$                      ;define TCB stuff
        ucbdf$                      ;define UCB stuff

setprv::
        mov     r0,-(sp)            ;save r0
        swstk$  10$                 ;;go to system state
        mov     $tktcb,r0           ;;get my TCB
        mov     t.ucb(r0),r0        ;;get TI: ucb
        bis     #u2.prv,u.cw2(r0)   ;;set /priv=ti:
        rts     pc                  ;;back to user state
10$:
        mov     (sp)+,r0            ;restore r0
        rts     pc                  ;return to where ever
        .end
```

This sets a tasks TI: to priviledged state. The disadvantage is that it only works for the device that is TI:.

By the way, the method the author of WHEE used to find the UCB can be found in the MCR subroutine $FDUCB ([12,10]FNDUCB.MAC and [1,24]MCR.OLB). This routine may be used to find the UCB of any device in your system. If you don't want to retype WHEE from the article I encourage you to examine this code.

# HOW TO USE THE RSX

# FULL-DUPLEX TERMINAL DRIVER

# WITH ROGUE DEVICES

Dale R. Donchin

Digital Equipment Corporation

# MAXIMIZING PERFORMANCE I

### OPTIMAL SUBFUNCTIONS

READ PASS-ALL (TF.RAL)

WRITE PASS-ALL (TF.WAL)

READ AFTER PROMPT (TF.RPR)

### EFFICIENT USE OF ASTS

MINIMIZE THEIR USE

TF.NOT PREFERRED

### BUFFERING TECHNIQUES

MAKE TASK NON-CHECKPOINTABLE

USE LARGE BUFFERS

POSSIBLY DOUBLE-BUFFER

KEEP READ POSTED

# MAXIMIZING PERFORMANCE II

### RECOMMENDED INTERFACES

USE DH11 OR DHU11 OR DHV11

DON'T USE DL11 OR DLV11

DZ11 IS POOR FOR MODEM CONTROL

### TYPEAHEAD

ALWAYS ENABLE

SET MAXIMUM SIZE

USE AS 'SLOP' BUFFER

### EXAMPLES

IO.RPR WITH TF.XOF AND XON AS PART OF PROMPT

ATTACH WITH TF.NOT, MARK-TIME, READ TYPEAHEAD

ENABLE FULL-DUPLEX AND KEEP READ POSTED

TURN OFF INPUT CHARACTER ECHO

# MAXIMIZING PERFORMANCE III

### USE XOFF/XON SUPPORT

ENABLE TERMINAL AND HOST SYNCHRONIZATION

READ PASS-THROUGH (TF.RPT)

### OTHER CONSIDERATIONS

ALWAYS ATTACH OR SLAVE THE LINE

SLAVE LINE IN VMR UNTIL APPLICATION BEGINS

FULL-DUPLEX (IMPLIES NO ECHO AND NO TF.XOF)

KEEP TASK NON-CHECKPOINTABLE

# READ REQUEST

module   PARAMETERS VALIDATED
TTINI     I/O PACKET PLACED IN QUEUE
             TRY TO RETRIEVE A REQUEST ($GSPKT)
             GET A READ REQUEST


module   DETERMINE WHETHER TO BUFFER
TTRW      IF YES THEN ALLOCATE FIRST BUFFER
             (MORE BUFFERS ARE ALLOCATED AS NEEDED)
             PROCESS TYPEAHEAD


module   DEVICE DEPENDENT ROUTINES CALL ICHAR1
TTICH     EACH CHARACTER RECEIVED IS PROCESSED
             INPUT LOGIC DETECTS READ COMPLETION
             INPUT COMPLETE FORK REQUEST IS QUEUED


module   DETERMINE IF BUFFERING:
TTRW         IF YES THEN CALL $QUEBF
                IF NO THEN CALL $IOFIN


module   COME TO ROUTINE $FINBF IF $QUEBF CALLED
SYSXT    $FINBF COPIES DATA FROM TTDRV BUFFERS
             TTDRV CALLED TO DEALLOCATE ITS BUFFERS

# WRITE REQUEST

module   PARAMETERS VALIDATED
TTINI     I/O PACKET PLACED IN QUEUE
             TRY TO RETRIEVE A REQUEST ($GSPKT)
             GET A WRITE REQUEST


module   BEGIN BUFFERING
TTRW      CALL EXPCHR IN A LOOP
             PUT EACH CHARACTER EXPANSION IN BUFFER
             POSSIBLY CHECKPOINT TASK
             DEVICE DEPENDENT ROUTINE STARTS OUTPUT


module   EACH BUFFER IS OUTPUT IN TURN
TTODN    DEVICE DEPENDENT ROUTINE CALLS NXTOB
             OUTPUT NEXT BUFFER IN CHAIN, IF ANY
             IF END OF CHAIN, QUEUE A FORK REQUEST


module   EXEC CALLS TTDRV AT FORK ENTRY
TTRW      TTDRV NOTICES OUTPUT REQUEST DONE BIT
             FPORD CALLED THROUGH DISPATCH TABLE
             BUFFERS DEALLOCATED
             $IOFIN OR $QUEBF CALLED

# INPUT CHARACTER PROCESSING

```
┌─────────────────────────────────┐
│   DEVICE DEPENDENT ROUTINES     │
└─────────────────────────────────┘
                 │
                 ▼
┌─────────────────────────────────┐
│           ICHAR1::              │
└─────────────────────────────────┘
                 │
                 ▼
┌─────────────────────────────────┐
│            IPROC::              │ ◄──┐
└─────────────────────────────────┘    │
                 │                      │
                 │            ┌─────────────────┐
                 │            │    TYPEAHEAD    │
                 │            └─────────────────┘
                 ▼
┌─────────────────────────────────┐
│           ICHAR2::              │
└─────────────────────────────────┘
                 │
                 ▼
┌─────────────────────────────────┐
│            FPIRD::              │
└─────────────────────────────────┘
        │                   │
        ▼                   ▼
┌──────────────┐     ┌──────────────┐
│   $QUEBF::   │     │   $IOFIN::   │
└──────────────┘     └──────────────┘
        │                   │
        └─────────┬─────────┘
                  ▼
       ┌──────────────────┐
       │   TASK BUFFER    │
       └──────────────────┘
```

# OUTPUT CHARACTER PROCESSING

```
┌─────────────────────────────────┐
│          TASK BUFFER            │
└─────────────────────────────────┘
                 │
                 ▼
┌─────────────────────────────────┐
│           EXPCHR::              │
└─────────────────────────────────┘
                 │
                 ▼
┌─────────────────────────────────┐
│         TTDRV BUFFERS           │
└─────────────────────────────────┘
                 │
                 ▼
┌─────────────────────────────────┐
│   DEVICE DEPENDENT ROUTINES     │
└─────────────────────────────────┘
                 │
                 ▼
              OUTPUT
```
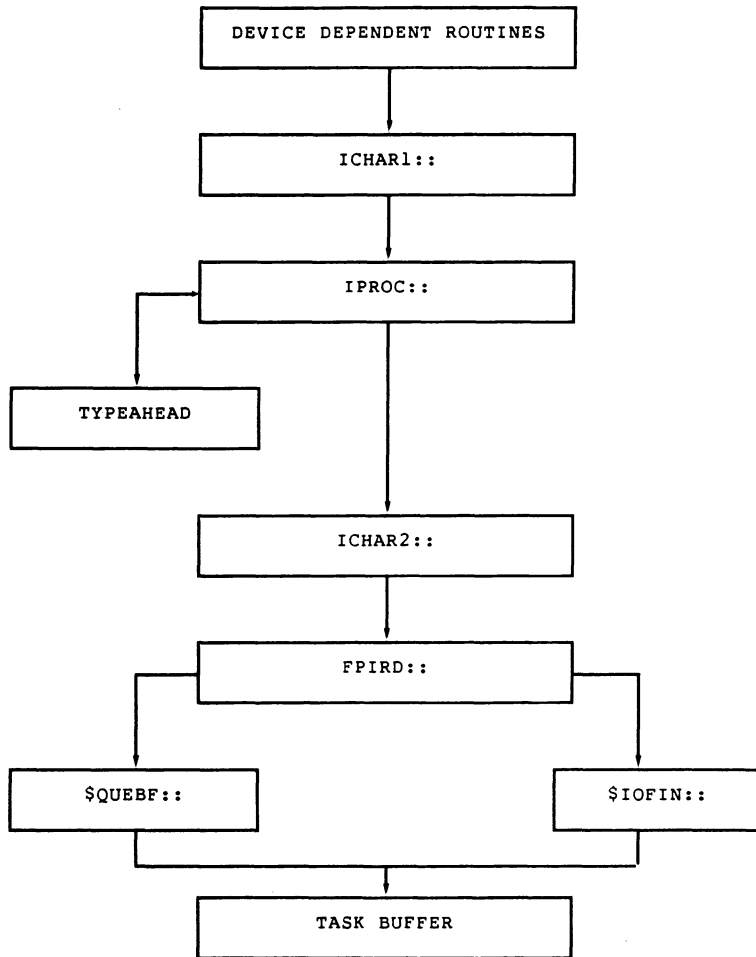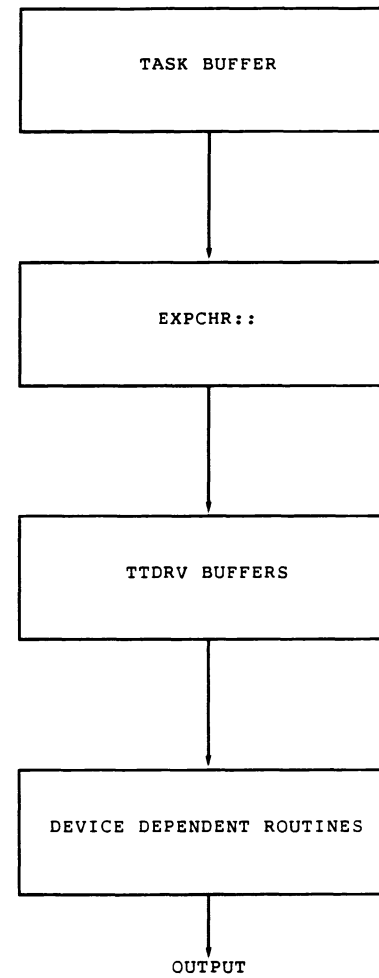
# INTERRUPT PROCESSING

INPUT                                    OUTPUT

```
                    ┌──────────┐
                    │   TTYH   │
                    └──────────┘
  ┌──────────┐      ┌──────────┐
  │  TTMIS   │      │  TTYHV   │
  └──────────┘      └──────────┘

                    ┌──────────┐
                    │   TTYZ   │      ┌──────────┐
                    └──────────┘      │  TTOIS   │
                    ┌──────────┐      └──────────┘
                    │   TTYL   │
                    └──────────┘

  ┌──────────┐    ECHO COMPLETION    ┌──────────┐
  │  TTICH   │─────────────────────│  TTODN   │
  └──────────┘                      └──────────┘

   READ COMPLETION            WRITE COMPLETION

  ┌──────────┐      ┌──────────┬──────────┐
  │  TTFP    │      │  FPIRD   │  FPORD   │
  └──────────┘      ├──────────┴──────────┤  TTRW
                    │ Call $IOFIN or $QUEBF│
Call $QASTT         └─────────────────────┘
                         ┌──────────┐
                         │  TTINI   │
                         └──────────┘
```

---
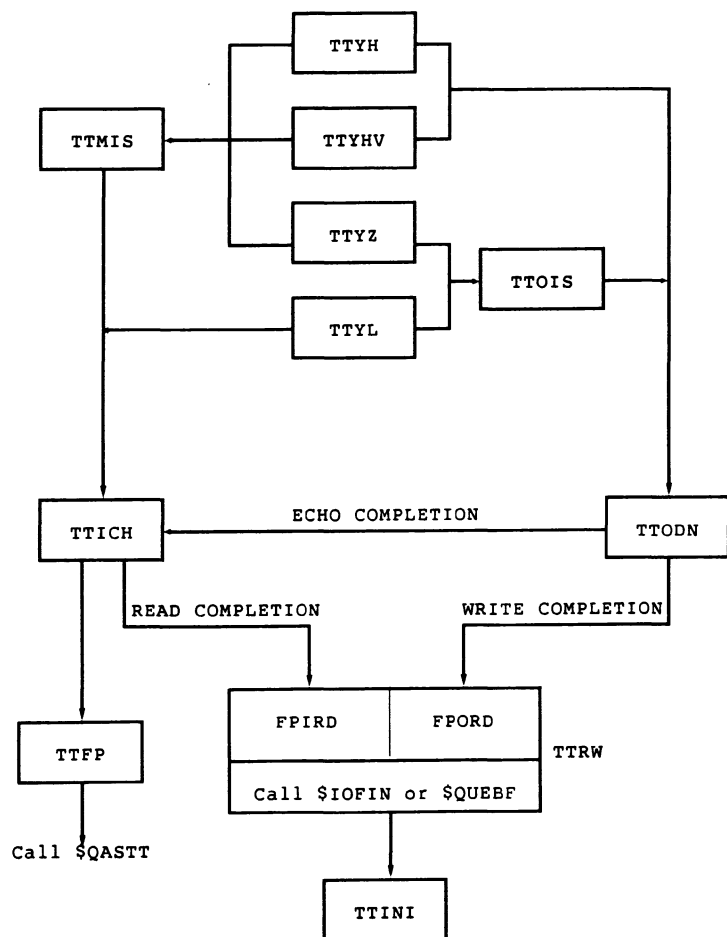
# DEVICE DEPENDENT ROUTINES

**CTRD - Call controller dependent routine.**

```
;
;+
; CTRD - CALL CONTROLLER DEPENDENT ROUTINE.
;
; INPUT:
;       R2      ROUTINE INDEX:
;               0 - START OUTPUT
;               2 - ABORT OUTPUT
;               4 - RESUME OUTPUT
;               6 - STOP OUTPUT
;               10 - POWER-UP (RSX-11M)
;               12 - MODEM TIMER (RSX-11M)
;               14 - GET/SET LINE PARAMETERS (RSX-11M)
;               10 - MODEM TIMER (RSX-11M-PLUS)
;               12 - CONTROLLER POWER-UP (RSX-11M-PLUS)
;               14 - UNIT POWER-UP (RSX-11M-PLUS)
;               16 - CONTROLLER ONLINE (RSX-11M-PLUS)
;               20 - CONTROLLER OFFLINE (RSX-11M-PLUS)
;               22 - UNIT ONLINE (RSX-11M-PLUS)
;               24 - UNIT OFFLINE (RSX-11M-PLUS)
;               26 - GET/SET LINE PARAMETERS (RSX-11M-PLUS)
;       R5      POINTER TO U.TSTA
;
; CALLS CONTROLLER DEPENDENT ROUTINE WITH:
;       R2      PHYSICAL UNIT NUMBER * 2
;               (ONLY IF MULTIPLEXERS IN SYSTEM)
;       R3      CSR ADDRESS
;       R4      UCBX ADDRESS
;       R5      POINTER TO U.TSTA
;
; REGISTERS ALTERED: R2,R3,R4
;-
;
```

# ADDING USER-WRITTEN CODE

### WHERE

ICHAR1, IPROC ROUTINES IN MODULE TTICH
TO CHANGE INPUT CHARACTER HANDLING

EXPCHR ROUTINE IN MODULE TTSUB
TO CHANGE OUTPUT CHARACTER PROCESSING

TTMAC PREFIX FILE TO CHANGE BUFFER LENGTH
AND HOST/TERMINAL SYNCHRONIZATION THRESHOLD

ADD MCGEN MACRO CALLS IN MODULE TTDAT
TO DEFINE OR CHANGE CHARACTERISTICS

MANS ROUTINE IN MODULE TTMOD
TO MODIFY BEHAVIOR WHEN ANSWERING A CALL

### HOW

ASSEMBLE EACH MODULE WITH TTMAC PREFIX FILE

REPLACE MODULE IN [1,24]TTDRVBLD.OLB FILE

REBUILD WITH [1,24] OR [200,200]TTDRVBLD.CMD

RE-VMR TO UNLOAD OLD AND LOAD MODIFIED DRIVER


*NOTE: YOU MUST REMOVE, RE-INSTALL AND FIX*
*THE TTCOM REGION (AND TTETC IN V3.0) ON AN*
*I/D-SPACE RSX-11M-PLUS SYSTEM DURING THE VMR STEP*

# ADDING NEW PORTS

### GENERAL IDEA

CLASS AND PORT SCHEME

CTRD ROUTINE PASSES CONTROL TO PORT

NEED CTB, KRB, DCB AND UCBS

### STEPS TO TAKE

1.  ADD TO IRP IN TTTBL AND TTDAT

2.  ADD SPEED TABLE IN TTDAT

3.  ADD TO LIBRARY AND BUILD FILES

### OR

SYSGEN IN UNSUPPORTED DJ11
INTERFACE BY DEFINING D$$J11

REPLACE TTYJ MODULE WITH YOUR OWN

# ANCILLARY CONTROL DRIVERS

### WHAT ARE THEY?

EXTERNAL SUBROUTINES CALLED BY THE DRIVER
AT VARIOUS SIGNIFICANT EXECUTION POINTS

THEY RECEIVE THE CURRENT DRIVER CONTEXT
WHICH THEY MAY ACCEPT OR ALTER

### WHAT IS THEIR APPLICATION?

MODIFICATION OF NORMAL CHARACTER PROCESSING

INTERRUPT-LEVEL ACCESS TO INPUT CHARACTERS

DRIVER CODE IN ALTERNATE ADDRESS SPACE

LOADABLE/UNLOADABLE DRIVER CODE PER TERMINAL

HOOK POINTS FOR MONITORING DRIVER EXECUTION

### HOW ARE THEY USED?

TWO PARTS - LOADER AND ACD ITSELF

QIO TO DRIVER ASSOCIATES AN ACD
WITH A GIVEN TERMINAL

ACD CALLED THROUGH A DISPATCH TABLE

# PROBLEMS AND SOLUTIONS

### GOTCHAS

NO RTS/CTS SUPPORT

NO BREAK SUPPORT

NO CONTROL OF DTR

NO 7-BIT CHARACTERS

### WORKAROUNDS

CHANGE THE PORT CODE

OR

USE A PRIVILEGED TASK

### ALTERNATIVES

CONNECT TO INTERRUPT

THIRD-PARTY DRIVER

```
+--------------------------------------------------+
|                                                  |
|        LEARNING FROM EXPERIENCE                  |
|                                                  |
|                                                  |
|                                                  |
|                                                  |
|            TALK TO ME                            |
|                                                  |
|                                                  |
|                                                  |
|  WHAT PROBLEMS HAVE YOU FOUND?                   |
|  _____                   |
|                                                  |
|                                                  |
|                                                  |
|  WHAT SUGGESTIONS DO YOU HAVE?                   |
|  _____                   |
|                                                  |
|                                                  |
|                                                  |
|     WHAT SUPPORT DO YOU NEED?                    |
|     _____                    |
|                                                  |
|                                                  |
|                                                  |
|       WHAT CAN YOU GIVE UP?                      |
|       _____                       |
|                                                  |
|                                                  |
+--------------------------------------------------+
              RSX-25
```

**digital**  **TTDRV For Non-Terminal Applications Questionnaire**
*(Sponsored by the RSX engineering group)*

Please help us provide adequate support of devices connected to an RSX system through a terminal interface by answering the following questions:

Name                                           Company
_____

Address                                        Telephone
_____

Brief description of application
_____

Please ✓ all that apply:

☐ I am currently using TTDRV for a non-terminal application
☐ I would like to do so but need more TTDRV features
☐ I use (plan to use) a non-DEC terminal interface
☐ I use (plan to use) a non-DEC terminal interface driver
☐ I give my permission for Digital to contact me for further information

Additional Comments:

Send response to:
    Dale Donchin
    Digital Equipment Corporation
    ZKO1-3/H1
    110 Spit Brook Road
    Nashua, NH 03062-2698

RSX-26

# The Kermit protocol and the PDP-11

Brian Nelson
University of Toledo
2801 West Bancroft Street
Toledo, Ohio 43606

Abstract

This article will describe the author's implementation of the Kermit file transfer protocol for the PDP-11 series under RSTS/E, RSX11M/M+, P/OS RT11 and TSX+. This protocol allows many (if not most) types of computer systems to effect, at minimum, error free file transfer with other systems and microcomputers over asynchronous lines.

The first obvious use of any program or protocol designed to accomplish file transfer is to be able to provide the ability to support file uploads and downloads from superminis such as the VAX and PDP-11 to remote personal computers, such as the PC and Rainbow. Since as of this date (03-Oct-1985) there are available over 160 versions of Kermit available for numerous micro, mini and mainframe configurations, Kermit addresses this need quite well.

Other uses of Kermit are quite numerous. I routinely use Kermit for transfering software developed for the PRO/350 on a RSTS/E 11/23+ host as well as using the PDP-11/44 and VAX 11/785 I run at the University of Toledo for dialing out to other systems, such as the LCG Tops 20 system and the LDP public domain library. Considering that there exists a Kermit for almost any DEC configuration one can even use Kermit as a poor man's Decnet. In my case, I have a DMF modem port from the 11/785 and a DZ11 port from the 11/44 connected to the Gandalf PACX front end switch, which allows me to connect either system to any of the other systems on the PACX, which includes an IBM 370 compatible system as well as connecting the VAX to the PDP-11.

With the knowledge that there are Kermit implementations for most personal computers in use it becomes apparent that the Kermit standard is well worth looking in to. A list of versions running on Digital hardware follows the article.

## The Kermit protocol

The Kermit protocol is designed to operate over normal asynchronous terminal lines. All data and commands are transferred with a packet oriented protocol, basically consisting of a start of packet character (normally SOH), followed by length, control, data and checksum fields. Communications is half duplex, in that for every packet sent, the sender must wait for either an acknowledgement packet (ACK) or a negative acknowledgement packet (NAK). Transmission is in ascii, with no requirements needed for the transmission of eight bit characters or control characters other than the choice of control-A for marking the start of a packet. All 'control' characters imbedded in the data are prefixed to convert them to printable characters, the same

applying to eight bit characters if required by the characteristics of the line. Since there are many different implementations of Kermit, the protocol provides a mechanism by which the capabilities of two connected Kermits can be negotiated to allow for differences in the level of protocol support. Examples of protocol features that not all Kermits understand include data compression and transfer of file attributes.

Rather than to go into more detail about the the Kermit protocol, the reader should consult the references listed at the end of this article.

## The PDP-11 Kermit-11 implementation

The author's version of Kermit-11 is written in Macro-11 and can run on RSTS/E, RSX11M, RSX11M Plus, P/OS and RT11. The RSTS and RSX file system interface is via RMS11 version 2, while the RT11 interface attempts to emulate the RMS11 subsystem. The choice of Macro-11 for the implementation language was made for several reasons, one being the availability of the assembler on all systems and another being speed and compactness of the code.

RMS11 was used for RSTS/E and RSX to provide a common i/o interface to the host file system. Additionally, Bob Denny of Alisa Systems further extended the RMS interface to support remote file access over DECNET with Kermit, allowing commands such as SEND NODENAME::[BRIAN.FUBAR]FILE.TYPE and other remote file accesses over DECNET. RMS11 version 2 also provides a very simple and powerfull means of doing wildcard searching, file renames and file deletion via the $PARSE, $SEARCH, $RENAME and $DELETE macros. Points against RMS basically amount to it's size, RMS is quite large even if overlayed. This is helped by using the segmented RMSRES available on RSTS/E and RSX11M Plus, though there is no remote file access for RMSRES in the current release of Kermit-11. The other objection to RMS will come from RSTS/E users, who are used to using files that normally lack file attributes. This is overcome by the ability of RMS v2 to create stream ascii files.

The RSTS/E Kermit, while it does 'run' under RSX emulation, does NOT use any RSX directives (apart from GTSK$S) to interface to the executive, as (one) the RSX directive emulation under RSTS/E is only a small subset of 'real' RSX and (two) there is no need to go though an additional layer of overhead to make RSTS/E map RSX calls to native calls. The 'multiple private delimiters' feature is used to avoid losing read pass all (binary) mode on read timeouts, as well as setting the link to '8-bit' mode to keep the terminal driver from stripping the high bit from data received.

The RSX11M/M+ and P/OS versions of Kermit-11, like the RSTS/E and RT versions, receive eight bit data assuming no parity is used. Where parity is a must, Kermit-11 has to use a prefixing scheme for eight bit binary data. Like the RSTS/E version, binary files are created as FIXED no carriage control files such as used for task images. Note that parity generation is done by software in Kermit-11. The P/OS version runs under control of DCL. The next release of Kermit-11, which will be 2.37, will include support for the PRO TMS (Telephone Management System) option.

The RT11 and TSX+ version of Kermit-11 maintains source module compatability with the RSTS/E and RSX versions. Each version of Kermit-11 has it's own source file to deal with the operating system, for RSX it is K11M41.MAC, for RSTS/E they are K11E80.MAC and K11S0S.MAC, and for RT11 they are called K11RT*.MAC. Apart from these specific files, all other source files are shared. The RT11 Kermit-11 can use either the version 5.x XL and XC handler for high throughput, or it can use multiple terminal service to do all its terminal i/o. This second option allows the use of any interface supported, including the PDT150 modem port, DL/DLV11's and DZ/DZV11's. The drawback is overhead, the RT11 MT service can't sustain a rate much past 1200 baud at most. This is not a problem for Kermit, however, due to it's half duplex nature and the fact that no packet received is ever longer than the ring buffer size. The only problem is in when Kermit-11 is running as a terminal emulator (the Kermit CONNECT command) where the data coming from the remote host can easily overrun the executive's buffer. A SET RT11 [NO]FLOW command was added to force Kermit-11 to send its own flow control to the host via XON and XOFF. TSX+ users can connect to CL: for dialing out, the exact means is documented in the Kermit-11 users guide. The disk i/o emulates the RSTS/E and RSX RMS11 version, and each executive directive has its error codes mapped into an unique global error code, with the symbolic names corresponding to the nearest RMS11 error name. Wildcarding is handled, of course, by non file-structured access to the directory on the desired volume, and supports full RT11 wildcard filenames.

Transmission of file attributes

One of the optional features of the Kermit protocol is the ATTRIBUTE packet. The attribute packets allow a Kermit program to send to a receiving Kermit information regarding the file organization, size, cluster/retrieval size, protection and so forth. There is even a system dependant attribute packet type that can be used to transfer things like the RMS11 IFAB (the RMS/FCS attributes). One of the things that two Kermits exchange before any file transfer is an information packet, this packet tells the receiving Kermit about itself. The last field in this packet, the CAPAS mask, tells Kermit if the other one can process attribute packets. If two Kermit-11's are communicating, they will find that each can do so, and the sender of a file will then send over attribute packets indicating the need (or lack of) for binary transmission, based on the file organization, filetype and protection code (for RSTS/E). If the sending Kermit-11 is running on RSTS/E, RSX11M/M+ or P/OS it will also send a copy of the RMS/FCS attributes so the received file will be identical (to FCS and/or RMS) to the copy on the sender's system. Since other implementations of Kermit may use this special system attribute packet, Kermit-11 always sends an attribute packet telling the receiver what hardware and operating system it is running on, and thus will only use such data if they are compatible. Of course, there will be times when a file may be binary and Kermit-11 can't tell so, many Kermit's have a SET FILE BINARY and SET FILE ASCII to allow the user to override defaults. Kermit-11 also has a SET

FILE AUTO/NOAUTO to disable it from trying to determine a file's binary status.

Future directions

With the advent of packet switched networks and satellite communications the Kermit protocol will likely be extended to increase efficiency over such links. The main problem is the half duplex nature of Kermit, the packet acknowledgements can take up to several seconds in transit thus drastically reducing the throughput. There are several possibilities under discussion and a standard should be appearing shortly.

Summary

 This article describes only the PDP-11 Kermit-11  implementation,
for further reading see:

Kermit: A File-transfer Protocol for Universities
Frank da Cruz and Bill Catchings
BYTE Magazine, June/July 1984

The Kermit Protocol Manual, version 5
Frank da Cruz    April 1984
Columbia University Center for Computing Activities

Information on obtaining Kermit:

KERMIT Distribution
Columbia University Center for Computing Activities
7th Floor, Watson Laboratory
612 West 115th Street
New York, N.Y.  10025

Kermit is also usually found on the Decus symposia SIG tapes.
Kermit-11 is available from DECUS as number 11-731

Digital hardware that Kermit is currently available for:

| Machine | Operating System | Program Language | Contributor |
|---------|------------------|------------------|-------------|
| DEC PDP-11 | MUMPS-11 | MUMPS-1982 | Cornell U |
| DEC PDP-11 | RSTS/E | Macro-11 | U of Toledo |
| DEC PDP-11 | RSX-11/M | Macro-11 | U of Toledo |
| DEC PDP-11 | RSX-11/M+ | Macro-11 | U of Toledo |
| DEC PDP-11 | RT-11 | Macro-11 | U of Toledo |
| DEC PDP-11 | RT-11 | OMSI Pascal | U of Toronto |
| DEC PDP-11 | TSX+ | Macro-11 | U of Toledo |
| DEC PDP-11 | Unix 2xBSD | C | Columbia U |
| DEC PDP-11, ... | Unix V7 | C | Columbia U |
| DEC PDP-8 | OS8, RTS8 | PAL-8 | R. Hippe |
| DEC Pro-3xx | P/OS | Bliss, Macro | Stevens I.T. |
| DEC Pro-3xx | P/OS | Macro-11 | U of Toledo |
| DEC Pro-3xx | Pro/RT | Macro-11 | U of Toledo |
| DEC Pro-3xx | Venix V1 | C | Columbia U |
| DEC Pro-3xx | Venix V2 | C | Columbia U |
| DEC Rainbow | CPM86 | ASM86 | Columbia U |
| DEC Rainbow | MS-DOS | MASM | Columbia U |
| DEC Rainbow | QNX 1.x | C | Merrell-Dow |
| DEC VAX | Ultrix-32 | C | Columbia U |
| DEC VAX | VMS | Bliss,Macro | Stevens I.T. |
| DEC VAX | VMS | C (VAX-11 C) | Columbia U |
| DEC VAX | VMS | Pascal | U of Toronto |
| DEC VAX, ... | Unix 4xBSD | C | Columbia U |
| DEC VT-180 Robin | CPM80 | Turbo Pascal | Jeff Duncan |
| DEC VT-180 Robin | CPM80 2.2 | M80,LASM | ACC |
| DECmate-II,III | CPM80 2.2 | M80,LASM | ACC |
| DECsystem-10 | TOPS-10 | Bliss, Macro | Stevens I.T. |
| DECSYSTEM-20 | TOPS-20 | MACRO-20 | Columbia U |
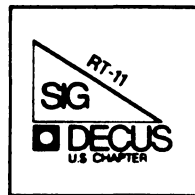
# RT-11 Mini-Tasker

RT

# RT SIG Steering Committee

SIG Chairman
  John T. Rasted
  JTR Associates
  Meriden, CT

Symposium Coordinator
  Ned Rhodes
  Software Systems Group
  Rockville, MD

Communications Committee Representative
FMS Contact
  Susan Rasted
  Software Dynamics, Inc.
  Wallingford, CT

Newsletter Editor
COBOL Contact
  Bill Leroy
  The Software House Inc.
  Atlanta, GA

Pre-Symposium Seminar Coordinator
Suite Manager
  Bruce Sidlinger
  Sidlinger Computer Corp.
  San Antonio, TX

Standards Coordinator
  Kenneth L. Aydlott
  Teledyne Hastings-Raydist
  Hampton, VA

Digital Counterpart
  Diana L. Miller
  Digital Equipment Corporation
  Maynard, MA

Tape Copy Generation Contact
  Ralston Barnard
  Sandia Laboratories
  Albuquerque, NM

APL Contact
  Doug Bohrer
  Bohrer & Company
  Wilmette, IL

MACRO Contact
  Nick Bourgeois
  NAB Software Services, Inc.
  Albuquerque, NM

Teco Contact
Product Planning Contact
  John Crowell
  Crowell Ltd.
  Los Alamos, NM

DECNET Contact
  Ken Demers
  Adaptive Automation
  New Haven, CT

UNIX Contact
Wish List Contact
  Bradford Lubell
  L.A. Heart Lab
  Los Angeles, CA

RT-11 Hardware Contact
C Contact
  Carl Lowenstein
  University of Calif., San Diego
  San Diego, CA

TSX Contact
  Jack Peterson
  Horizon Data Systems
  Richmond, VA

Library Contact
Tape Copy Distribution
  Tom shinal
  General Scientific Corp.
  Rockville, MD

BASIC Contact
  Ed Stevens
  EMDA Inc.
  Pasadena, CA

CAMAC Contact
  J.W. Tippee
  Kinetic Systems Inc.
  Lockport, IL

Personal Computer Contact
LUG Contact
  William K. Walker
  Monsanto Research Corp.
  Miamisburg, OH

FORTRAN Contact
RUNOFF Contact
  Robert Walraven
  Multiware, Inc.
  Davis, CA

## FROM THE EDITOR

As you read this the Spring Symposium in Dallas will be over. I hope many of you went, and learned a lot from the formal sessions, as well as the informal get-togethers in the RT-11 campground. To those who missed it, begin to plan now to be in San Francisco October 6-10, 1986.

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

## 100,000th RT-11 License

Congratulations to John Crowell of Sandia Labs. He was recently presented the 100,000th RT-11 license by Ken Olsen, DEC President, and Diana Miller, RT-11 Product Manager. We will have more coverage on this in the June Mini-Tasker.

Jack serves the RT-11 SIG as Product Planning Contact and as TECO Contact. If you have trouble understanding Jack, it is because he talks in TECO. (Is there any other language?)

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

I have received the following note from Diana L. Miller, RT-11 Product Manager.

### Communications area and RT-11.

The RT-11 Product Group is looking at the communications area and RT-11. We would like to hear what you feel the problem(s) is/are in this area. A few ideas to get you going (as if anyone needed any):

    Problem: Ability to transfer large amounts of data quickly.

    Is there a need for Task-to-Task (two way) communications?
    Is VTCOM/TRASFER fast enough via a serial line?
        Would this capacity (one way) over Ethernet be enough?
    What is needed in DECnet/RT?
        Enhanced serial line product?
        Ethernet or nothing at all?

Please be realistic with your requests. Please prioritize them. Please add comments/reasons to help me understand.

Regards, Diana

[Requests and comments may be sent to the Editor at the address below.]

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

## TSXLIB UPDATED

Nick Bourgeois has announced that TSXLIB, DECUS #11-490, has been updated for TSX-Plus V6.0. The new EMTs have been implemented, along with an update of one old EMT. In addition, a KWIK index of all of the TSXLIB routines has been added to the manual. Nick gave a paper on the subject in Dallas on Friday, session RT001, and audio tapes are available.

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

## FALL 1985 RT-11 SIG TAPE

Rally Barnard has put together a fine FALL 1985 RT-11 SIG TAPE, and I have included a directory in this months issue. Tapes can be ordered (for free) from your Local User Group (LUG), as well as from the DECUS Library (at a nominal charge). In addition, Tom Shinall has put together four sites for on-line distribution (see article this issue).

Free tapes are an advantage of belonging to a LUG, as well as the fellowship and general information gained from attending LUG meetings.

You can also contribute to the tape yourself. Please look around your site for some interesting program, and consider submitting it in San Francisco.

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

## Tidbits to remember

In public we say the race is to the strongest; in private we know that a lop-sided person runs the fastest along the little sidehills of success. - - Frank Moore Colby
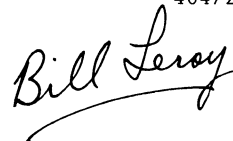
Most people, when they think of success, think about it as the end result of long hours, endless pressures, and bumper-to-bumper driving down smog-bound super-highways. May I present for your consideration, however, the theory that success is the means to an end. Getting there is half the fun. - a Sears-Roebuck executive

And finally, I am always looking for something to print.

Please send your submissions to The Mini-Tasker to me at:

Bill Leroy
The Software House, Inc.
2964 Peachtree Road, NW #320
P. O. Box 52661
Atlanta, GA 30355-0661

404/231-1484 or DCS (LEROY)

```
*********************************************************************
                  DECUS Symposium RT-11 SIG Tape
                             Fall 1985

                        Anaheim, California

                        Annotated Directory

*********************************************************************

                            IMPORTANT

                Read the file, README.1ST, first.

   README.1ST    10   21-Feb-86   SIG tape copy instructions
                            and new information for everyone.




   NOTE!  We are interested in maintaining the quality of the submis-
   sions to the RT SIG tape.  Therefore, we welcome feedback
   regarding your use of these files, any bugs you find, and any bug
   fixes or improvements you devise.  Please send any correspondence
   regarding the tape to:

        John Crowell
        Crow4ell Ltd.*                   * (But not very)
        145 Andanada
        Los Alamos, NM 87544
        (505) 662-3893

        DCS - CROWELL


   **********************************************************




VIRTUL - Subdevice retriever for RSTS/E.

E. F. Beadle, Jr., Manager            (315) 341-3055
CAUSE Instructional Computer Center
SUNY at Oswego
Oswego, NY 13126
This program allows RSTS/E users to break down the subdevice files
from  this  tape after they have been copied to disk.  It has been
modified by David Smith, Galileo Computer Center, to remove a  few
bugs,  and  to  be  able  to  read multi-segment directories.  See
README.1ST for details.

VIRTUL.BAS   1 file,   43 blocks
```

```
DIRTWO - Annotated tape directories, part 2.

R. W. Barnard
Sandia National Laboratories
Minicomputer Software Division 7523
P. O. Box 5800
Albuquerque, NM 87185
(505) 844-5115

        DIRTWO contains annotated directories of the DECUS Symposia
RT-11 tapes from the Fall of 1981 through the Fall of 1985 (this
symposium).

DIRTWO.DSK    9 Files, 428 Blocks
*********************************************************************
```

```
KER* - File transfer protocol for PDP-11's.
K11*.HEX

Brian Nelson
Computer Services, University of Toledo
2801 West Bancroft
Toledo, OH 43606
(419) 537-2841

        This is release 2.39 of Kermit-11.  It requires RT11 Version
4.0 or later, TSX+, RSTS Version 7.2 or later, RSX11M v4.0 or
later, RSX11M Plus Version 2.0 or later, P/OS Version 2.0 or
PRO/RT11 Version 5.1.  See the files K11AAA.AAA, and K11INS.DOC
for more information.  Edit history is given in the file
K11CMD.MAC.

        The distribution has been subdivided roughly by operating
system.  The subdevice files KERCM*.DSK contain documentation and
files common to all operating systems.  The other subdevices are
operating-system specific.  The distribution also contains both
save (binary executable) images and .HEX (ASCII) versions of the
save images.  See the installation document for information on how
to create a binary from the hex file.  The HEX files are not con-
tained in subdevices.  Please note that the allocation of specific
files to the operating system-specific subdevices was done without
a great deal of research - If you can't find a file, try another
subdevice!

KERCM1.DSK   10 Files, 472 Blocks   (Common Files)
KERCM2.DSK   20 Files, 449 Blocks   (Common Files)
KERCM3.DSK   10 Files, 369 Blocks   (Common Files)
KERCM4.DSK    5 Files, 362 Blocks   (Common Files)
KERRT1.DSK   13 Files, 310 Blocks   (RT Files)
KERRT2.DSK    2 Files, 318 Blocks   (RT Files)
K11RT4.HEX    1 File,  363 Blocks   (RT File)
K11RXM.HEX    1 File,  363 Blocks   (RT File)
```

```
KERST1.DSK  14 Files, 465 Blocks  (RSTS Files)
KERST2.DSK   7 Files, 435 Blocks  (RSTS Files)
K11NRS.HEX   1 File,  792 Blocks  (RSTS File)
KERSX1.DSK  16 Files, 416 Blocks  (RSX Files)
KERSX2.DSK   3 Files, 461 Blocks  (RSX Files)
K11RSX.HEX   1 File,  607 Blocks  (RSX File)
K11POS.HEX   1 File,  432 Blocks  (POS File)
KERI31.DSK   4 Files, 186 Blocks  (IAS File)
K11I31.HEX   1 File,  395 Blocks  (IAS File)
K11XXX.HEX   1 File,  489 Blocks
***********************************************************
         FALL, 1985 RT SIG TAPE DIRECTORY
```

RUNOF* - Bonner Lab Runoff for RT-11.

Submitted by:
Robert Walraven
Multiware, Inc.
139 G Street, Suite 161
Davis, CA 95616
(916) 756-3291

This version of Bonner Lab RUNOFF has been modified significantly
since the last release (Fall 1984). Many bugs have been elimin-
ated, some new features have been added, and the documentation has
been extensively rewritten to improve its readability. The RT-11
version of Bonner Lab RUNOFF is now being supported by the RT-11
SIG in cooperation with John Clement, the author. The RUNOF*.DOC
files are a printer-ready copy of the full documentation.

```
RUNOF1.DSK   8 Files, 383 Blocks
RUNOF2.DSK   7 Files, 306 Blocks
RUNOF3.DSK  14 Files, 446 Blocks
RUNOF4.DSK  11 Files, 368 Blocks
RUNOF5.DSK  27 Files, 463 Blocks
RUNOF6.DSK  24 Files, 265 Blocks
RUNOF7.DSK  26 Files, 146 Blocks
RUNOF8.DSK  19 Files,  37 Blocks
RUNOF9.DSK 109 Files, 274 Blocks
RUNO10.DSK  33 Files, 126 Blocks
RUNOF1.DOC   1 File,  400 Blocks
RUNOF2.DOC   1 File,  304 Blocks
***********************************************************
         FALL, 1985 RT SIG TAPE DIRECTORY
```

F77KIT - FORTRAN-77 OTS Update.
VIRDSK - Virtual-to-disk Mechanism.
DOHAND - Diagnostic Overlay Handler.
MTOPEN - FORTRAN-77 Support for Mag tape.
AIRPLN - Airplane Lander Game.
DATUTL - FORTRAN Utilities.

Robert Walraven
Multiware, Inc.
139 G Street, Suite 161
Davis, CA 95616
(916) 756-3291

F77KIT - Upgrade kit for FORTRAN-77/RT, Release 1. Makes anything
associated with virtual work better, and fixes a few bugs, includ-
ing two for unformatted reads. NOTE: this does not contain many
of the modifications for release 2 of FORTRAN-77/RT.

VIRDSK - Module to force VIRTUAL arrays to a disk file rather than
extended memory.

DOHAND - A diagnostic overlay handler that gives an error report
if you try to destroy the return path in an overlayed program.

MTOPEN - Replacement module for F77OTS Release 1 to provide sup-
port for MAGTAPE sequence numbers.

AIRPLN - Aircraft flight simulator game for VT100.

DATUTL - A collection of FORTRAN-callable subroutines to provide a
variety of disk services.

```
F77KIT.DSK   8 Files,  28 Blocks
VIRDSK.DSK   4 Files,  32 Blocks
DOHAND.DSK  16 Files,  80 Blocks
MTOPEN.DSK   2 Files,   5 Blocks
AIRPLN.DSK   5 Files, 128 Blocks
DATUTL.DSK  13 Files,  42 Blocks
****************************************************************
         FALL, 1985 RT SIG TAPE DIRECTORY
```

UCLPL* - User Command Language (UCL) Program.

William K. Walker
Monsanto Research Corp.
P. O. Box 32    OS-123
Miamisburg, OH  45342
(513) 865-3557

    UCL+ is upward-compatible with the UCL distributed with RT-
11, Version 5.1B and later. The Version submitted to this tape is
V07.49, an update from all previous versions. UCL+ contains a
number of extensions, including chaining to additional UCL's,
"run-by-name", path definition, display of command expansions,
etc. Symbols are defined by entering a "symbol definition string"
in the format: symbol==definition. The DISPLAY command can be
used to output ASCII strings to the console or printer (handy for
sneaky escape sequences). A UC "pseudo-device" handler is provid-
ed as an option which allows UCL+ to "remember" the "input-spec"
part of the last UCL+ command. This text can be retrieved, at
the command level, by using the "^" character in place of the
argument in a subsequent command.

    This Version of UCL+ supports many new features of RT-11 and
TSX+. It can be used with TSX+ as a "User Command Interpreter"
(UCI). It minimizes disk access to improve efficiency; included
on this distribution is a "memory-resident" UCL.

```
UCLPL1.DSK  26 Files, 394 Blocks
UCLPL2.DSK   2 Files, 252 Blocks
****************************************************************
```

DAYLOG - "Foolproof" Way for Setting Date and Time.

Gary F. Sallee
Sallee Software
19912 Fernglen Drive
Yorba Linda, CA 92686
(714-970-2864)

Set the RT-11 time and date the easy way with DAYLOG.SAV.
The DAYLOG.SAV program for RT-11 is yet another variation of the
DATIM functionality, but with a twist. DAYLOG is easy to use by
computer-ignorant people. But DAYLOG also has built-in shortcuts
the knowledgeable person. DAYLOG maintains a .LOG file to form
defaults for all of the questions. DAYLOG is intended to run from
the STARTx.COM file, but can be run at any time. DAYLOG will run
under RT-11, V4 or V5.

DAYLOG.DSK   6 Files, 335 Blocks
*************************************************

DIRUT* - Directory, and Other Utilities.

Glenn A. Bever
NASA Ames/Dryden Flight Research Facility
Code OFA
P.O. Box 273
Edwards, CA    93523
805-258-3747

These are a group of handy utilites for general useage.
PRH is a print utility (date/time stamped headers).
SDIR searches an RT-11 volume and its subdevices for specified
filenames. It has been updated to include some date qualifiers
(/D, /B, /S).
ELXSI and ELXSIW are mag tape read/write routines.
Some useful control files are included that compare directories,
print directories, backup and restore directories in a format com-
patible with SDIR. The latter files have been modified since last
year to include some MSCP support and fix a few bugs.

DIRUT1.DSK  43 Files, 230 Blocks
DIRUT2.DSK  15 Files, 308 Blocks
*************************************************

FLECS(1,2) - DECUS RT Flecs Translator.

Dennis Jensen
Ames Laboratory, USDOE
Iowa State University
310 Metallurgy
Ames, IA 50011
(515) 294-4823

This is an update to a previous DECUS release of the Flecs
translator. Changes have been made to fix the expansion of tabs
used in source code. The FLECS.OLD is also a working  version
but uses the older technique of INCLUDE. It requires less memory
than Flecs.AL, so may be required by some users.

FLECS1.DSK  36 Files, 448 Blocks
FLECS2.DSK   4 Files, 424 Blocks
*****************************************************************

FLECS(A,B,C,D,E) - FLECS translator.

Karl L. Danneil
General Electric Corporation
6767 Pittsfield Road
Nassau, NY 12123
(413) 494-2907

The FLECS (Fortran Language with Extended Control Structures)
program is a pre-processor for FORTRAN programs. It will process
standard FORTRAN source programs (causing no changes or errors).
Other pre-processors such as RATFOR have so corrupted the FORTRAN
language that they can't be used on standard FORTRAN source files.
There are several versions of the precompiler included on the
disk. They represent various corruptions of implementation mostly
varying in the details of STRING handling.

FLECSA.DSK  27 Files, 446 Blocks
FLECSB.DSK  29 Files, 485 Blocks
FLECSC.DSK  14 Files, 474 Blocks
FLECSD.DSK   5 Files, 402 Blocks
FLECSE.DSK   2 Files, 255 Blocks
*****************************************************************

FORTHP - FORTH Programming Environment.

M. P. Hanson
Department of Chemistry
Humboldt State University
Arcata, CA 95521
(707) 826-4286

This is a public domain version of the FORTH programming
environment which has been revised for use on RT-11 (and other
operating systems). This FORTH system has full-length names,
extensive compile-time checks, 32-bit integer support, string-
handling routines, a string-search editor, linked vocabularies,
and a FORTH assembler which permits structured, interactive devel-
opment of device handlers. Full documentation is in the file
FORTH.MAC.

FORTHP.DSK   5 Files, 374 Blocks
*****************************************************************

HGRAF* - RT Graphics Package, Release 4.

Dennis Jensen
Ames Laboratory, USDOE
Iowa State University
310 Metallurgy
Ames, IA 50011
(515) 294-4823

This release of HGRAPH is an update to provide for virtual arrays. The use of virtual arrays allows the PDP-11 HGRAPH user to access more program memory without the need to overlay. This often makes the use of virtual arrays preferable over real arrays for those who can use them.

HGRAF1.DSK  20 Files, 418 Blocks
HGRAF2.DSK   2 Files, 455 Blocks
***************************************************************

SYDATE - "System" Handler for Saving Date.

Taken from the Fall, 1979,
RT SIG tape by
popular demand.

This handler stores the current system date, so that it will automatically be provided to RT-11 on boot-up. A very handy feature for frequent reboots. To use, type
SET SY [NO]DATE.

SYDATE.DSK   1 File,    4 Blocks
***************************************************************

The Fall, 1985, RT SIG tape contains 56 Files, 18097 Blocks.

It was prepared by:

R. W. Barnard
Sandia National Laboratories
Division 7523
P. O. Box 5800
Albuquerque, NM 87185

DCS - BARNARD

It is available from the following sources:

DECUS NLO TAPE TREE                    DECUS LIBRARY
c/o Robert N. Perry
Tektronix, Inc.                        DECUS Program Library
PO Box 500                             BP02
MS: 19-333                             249 Northboro Road
Beaverton, OR  97077                   Marlboro, MA 01752
(503) 527-5410

DCS - PERRY

File date: 28-Feb-1986
Printing date: March 3, 1986

Cross-Reference Index

```
                    FALL 1985                        Host:   RDB/ALPHA SYSTEMS & SOFTWARE
               ANAHEIM SIMPOSIA TAPE                 Service:        Limited Binaries and Index
               ON-LINE DISTRIBUTION                  Protocol:       KERMIT and VTCOM/TRANSF
                                                     Phone:          513-426-0344
The distribution of the Fall 1985 RT-11 Simposia tape is now available    Time Zone:      Eastern
from the NLO (National LUG Organization). All LUGs who desire the tape     Hours:          8:00 PM to 6:00 AM
are to contact the NLO to make arrangements to receive it.                 Data Rate:      1200/2400 bps
                                                     Logon:          DECUS
There are many individuals who do not have access to a LUG  or  simply     Password:       GUEST
do  not  have  a  tape  drive and can not take advantage of all of the
useful programs available. To accommodate  these  individuals,  a  new
voluntary  program has been initiated by the RT-11 SIG which will make
the tape index and the save images available on line. There  are  pre-   HOST:   COMPUSERVE
sently  four  sites  which  will  initiate service as indicated below.   Consult your Compuserve  manual  for  information.  Programs  will  be
Depending upon the test results, more will follow.                        available within the PDP-11 SIG area.


-------------------------------------------------------------------------

Host:   GENERAL SCIENTIFIC CORPORATION                        CAVEAT - CAVEAT
Service:        Binaries and Index
Protocol:       KERMIT and VTCOM/TRANSF                These services are on a volunteer basis and the system  resources  are
Phone:          301-340-2776                           also  on  a  donated  basis. Please  don't  abuse  them else we loose
Time Zone:      Eastern                                them....
Hours:          6:00 PM to 8:00 AM
Data Rate:      1200 bps                               Also please note carefully the indicated times of availability.  Calls
Logon:          DECUS                                  outside  of  these times will be rudely disconnected and you will have
Password:       GUEST                                  willingly contributed to the pension and profit-sharing plan  of  your
                                                       local telephone company.
Service:        Sources and Index
Protocol:       Kermit and VTCOM/TRANSF
Phone:          301-340-2776
Time Zone:      Eastern
Hours:          2:00 AM to 8:00 AM
Data Rate:      1200 bps
Logon:          SOURCE
Password:       GUEST


Host:   SIDLINGER COMPUTER CORPORATION
Service:        Binaries and Index
Protocol:       KERMIT
Phone:          512-344-4845
Time Zone:      Central
Hours:          11:00 PM to 8:00 AM
Data Rate:      1200 bps
Logon:          DECUS
Password:       GUEST

Service:        Sources and Index
Protocol:       KERMIT
Phone:          512-344-4845
Time Zone:      Central
Hours:          2:00 AM to 8:00 AM
Data Rate:      1200 bps
Logon:          SOURCE
Password:       GUEST
```

02-Jan-1986

### What is ModulaWare's Modula-2 Programming System ?

© (1985)
Günter Dotzel
ModulaWare GmbH
Wilhelmstr. 17A
D-8520 Erlangen/West Germany.

Introduction: This paper is only relevant to you if you are using or if you are interested in PDP-11/LSI-11 with either RT-11 or SHAREplus operating system.

History: ModulaWare's Modula-2 Programming System for PDP-11/RT-11 is based upon the M2RT11 release of the Institut für Informatik, ETH-Zürich and is in production use since early 1981. For more information on the programming language Modula-2 see [Wirth 85]. My first industrial project using Modula-2 on PDP-11/23 under RT-11 was with Siemens AG, Erlangen, in summer 1981. In an experiment four ultra-sonics transducers in combination with four transient recorders (wave form digitizers) were sensed via CAMAC with a sampling rate of 20MHz. The application for stress detection and localization in components of a nuclear power station was tested and simulated using a pulsed laser. Modula-2 programs were limited to 56K bytes sharing this addressing space with RT-11's RMON, handlers and USR. At this time, it was the only compiler running under RT-11 which could generate native code for the floating point unit (FPU) of the new LSI-11/23. Due to this fact and because of Modula-2's real-time and debugger features, the compiler quickly gained acceptance at several installations.

I'm involved with computers since 1976, when I started to study computer sciences. In 1981, I purchased a PDP-11/23 system and founded my own company. Besides working as a consultant, I developed several modules for the Modula-2 programming system, including MathLib and real number conversion, since they were not part of the M2RT11 release. The screen-editor was released in 1982. The development of the raster-graphics software was started in 1983. In 1984, my company got the new name "ModulaWare" and extended the product range, selling GDC-11 (graphic display controller) Q-Bus boards and Modula-2 software. In summer 1985, the company got four additional letters and is called "ModulaWare GmbH" now. The abbreviation GmbH means Limited (Ltd).

Memory prices dropped dramatically since 1981. Most PDP-11 systems now have at least 1M byte of main memory. But the extended memory was not directly available to the Modula-2 programmer. More sophisticated controllers, such as raster-scan graphic boards with on-board graphic processor, became available. More sophisticated software for interactive bit-mapped graphics, computer aided design and database systems were developed. Application programs importing such modules did not fit into the limited addressing space.

In 1983, Modula-2/VRS became available. VRS means virtual run-time system. With Modula-2/VRS programs are no longer restricted to fit into PDP-11's 16 bit directly addressable memory space. The Modula-2/VRS System is based upon the XM2-System released by the Technical Unversity of Munich (TUM). VRS is in production use since early 1984.

ModulaWare is allowed to maintain, to support and to international distribute these ETH and TUM based software.

The Modula-2 Programming System: The Modula-2 distribution kit consists of compiler, linker, screen-editor, debugger, resident monitor with run-time system and command interpreter, library, decoder for sym, ref, lnk, lod and xml files and documentation.

Here are some details concerning the ModulaWare's Modula-2 distribution kit.

(A) Compiler: switch selectable /FIS or /FPP (FPU) code generation, logical units for compilers overlays (VC), work files (VS) (can be placed in VM for example) and for the listing output device (VL; .assign TT VL shows the listing on console). Code size up-to 4K byte per procedure. There is only one version of compiler which runs on any hardware configuration (with our without FIS/EIS/FPU/MMU) running under RT11SJ, RT11XM (VBGEXE) or SHAREplus.

(B) Linker: The linker was corrected, since the original linker LINK.LOD sometimes destroyed the default device's directory when there was no room for the output file. The linker now handles procedures up to 4K byte in size and looks for library modules to be linked to the master file, not only on DK and SY, but also on logical devices DKA, DKB, ... DKH with the new option /L (lookup path). This allows the configuration of implementation dependent libraries under RT-11. Example: the implementation modules for monochrome graphics may be located on DKA and the color graphics modules on DKB. Your master program may be on DK and can be linked to DKA or DKB depending on what you have assigned. Under SHAREplus, the .assign/path command serves the same purpose.

(C) Screen-Editor: An integrated screen-editor called MEd was developed by ModulaWare in 1982. MEd works à la UCSD p-system editor, supporting semi-automatic interactive source line indentation, string/literal searching/replacement, listing of other text files, merging text files, listing of device directories without leaving the editor and last but not least optional encryption/decryption of text files. MEd may be called directly from command interpreter.

(D) Debugger: DBUG is a post-mortem dump analyser at source-code-level. DBUG shows the program's termination point in the text window, displays the procedure chain, allows the inspection of local and global data, including processes. DBUG fully decodes REALs, RECORDs, PROCESSes, ARRAYs and any user defined data structure. DBUG was developed by Georg Maier at ETH Zürich; Gerhard Koller at TUM designed it's VRS version. Note: Run-time debugging is possible with TRACE-11 (TR.SYS or TRACE.FEA under SHAREplus; for availability see below). TR is a pseudo device handler which intercepts all operating system calls (EMTs, e.g. the I/O requests) to display the type and it's parameters. This cuts debugging from hours to seconds.

(E) RTS/VRS resident monitor, run-time system (RTS) and virtual run-time system VRS: RTS and VRS were improved and extended; they are now so-called well-behaved RT-11 programs, i.e. RTS doesn't steal operating system's trap vectors and doesn't even perform any direct I/O-page access. VRS saves all vectors (including 4 and 10) and restores all vectors on termination. There are two run-time systems each for VRS and RTS. One version is with floating point unit support and the other with FIS and EIS emulation. VRS always needs the memory management unit (MMU). RTS runs under
(E1) RT11SJ monitor in the back-ground area.
(E2) RT11XM monitor in extended memory (using the VBGEXE.SAV utility).
(E3) SHAREplus multi-user RT-11 operating system (some customers use RTS under TSXplus).
VRS runs under
(E4) RT11SJ monitor.
The memory size available with different operating and Modula-2 run-time systems is
(E1): up to 51K byte depending on which handlers are resident or loaded. Direct I/O-page and RMON data base access possible.
(E2): 64K byte, no direct I/O-page access under RT11 version 5.1.
(E3a): 62K byte, no direct I/O-page access, but RMON fixed offset data available.
(E3b): 55K byte, direct I/O-page and RMON fixed offset data access possible.

(E4): 40K byte for directly addressable Modula-2 program data, heap and stack; up-to 2M byte of Modula-2 program code and up-to 2M byte of virtual addressable data in extended memory, depending on how much memory is installed.
Some modifications and extensions of the RTS/VRS concern the multi-user environments such as SHAREplus and the floating point error recovery of the KEF11AA FPU-chip for LSI-11/23 or FPU of the LSI-11/73. The VRS Upgrade was tuned to be fully upward compatible. Since the standard VM.SYS handler doesn't work with VRS, the virtual memory handler VN.SYS was developed by ModulaWare in 1985. VN.SYS is VRS compatible and even bootable. The typical memory layout with 1M byte total storage is (B means octal):
0 to 20000B run-time system VRS
20000B to 133000B (to max. 140000B) data, heap and stack
133000B to 160000B RT-11's USR, handlers and RMON
160000B to 400000B (128K byte) virtual program code segments
400000B to 1000000B (256K byte) virtual user data
1000000B to 4000000B (1M byte) VN handler with 1536 virtual disk blocks in total.

Modula-2/VRS Upgrade: The upgrade kit consists of an alternate run-time system with integrated extended memory loader (VRS), the VRS linker which allocates the virtual code segments separating code and data and the VRS debugger. Modula-2 programs linked and executed under VRS run under RT-11SJ (i.e. not RT-11XM). You can develop your Modula-2 programs under the normal unmapped Modula-2 system (RTS) and link, execute, and debug them using the VRS linker and debugger when the program becomes too large to fit into PDP-11's directly addressable low-memory (16-bit) space of the so-called "RT-11/16". VRS has build in MMU support running unmapped operating system RT-11SJ. Since RTS and VRS programs use different file extensions (.LOD and .XML), you may switch between the run-time systems without changing your system device. Even the commands and operations are the same in RTS and VRS:

.RUN MODULA (may be renamed to RTS and invoked by .RUN RTS or simply .RTS)
RTS>comp
...
RTS>link
...

.RUN VRS
VRS>comp
...
VRS>link
...

When you are running RTS you must share 56K byte with the RMON, handlers and USR. When using VRS you must share your stack, heap and normal data as with RTS. But the program code is virtual and can be as large as your actual main memory size. This makes the PDP-11/23 or -11/73 look like a little VAX, efficiently running Modula-2 programs under the so-called "RT-11/(18,22)" (18 or 22 bit addressing capability and Q- or Q22-bus). If you would like to see how Modula-2/VRS works, see [Dotz 86a].

(E continued) Loader: The loader which is part of the run-time system (RTS), was tuned. Instead of reading single blocks the loader performs very few read requests using a larger buffer on the stack. This speeds up program and overlay loading especially when using floppy disk.

(E continued) Command Interpreter: The verbous command interpreter eases error diagnosis, since not only the error message (e.g. Input/Output error) is displayed, but also -where possible- the reasons which caused the error (e.g. device or directory full, output file protected, input file exhausted, handler not loaded).

(F) Library: Beside the standard modules providing keyboard and screen (standard console) access, file system interface for manipulation of random and sequential files, storage handling for the heap and overlay loading, a flexible Modula-2 I/O-library called Univ* (universal I/O) is supplied. Univ* allows you to read and write any data type (including REAL, LONGCARDINAL, system date and time and file names) from and to any device. The Modula-2 library includes two MathLibs (one is part of the run-time system - written in MACRO and the other written in Modula-2). The Modula-2 library which is currently released in source form also, supports all data types for any I/O devices without overhead. There is video terminal independent retyping feature when reading strings, numbers and file names, called TIRF. TIRF is unique across all system and application programs.

Not all extensions and improvements to the Modula-2 Programming System can be listed here. For more details see [Dotz 86] and [Dotz 84].

(G) Documentation consists of 110 pages in A4 format and covers user guide with appendix for RTS and VRS, RT-11 and SHAREplus installation hints, MEd screen-editor manual with quick-reference guide, description of utilities and library and comprehensive Modula-2 bibliography.

(H) Maintenance and updates: ModulaWare produces currently about three updates per year. Since the main system components are very stable, there are only minor changes. The customers are informed about the new releases. The updates are send for free in the first year.

(I) Policy: The Modula-2 Programming System RTS/VRS is available as a ready-ready-to-use binary kit. The source code of the original Modula-2/VRS upgrade is also available and includes compiler, VRS-linker, VRS-debugger, utilities, library modules, examples - all written in Modula-2, run-time system written in Macro-11, and documentation in ASCII. Note, that although the system compiles itself under RT-11, modifications on the compiler are not recommended and you have to sign the source license agreement, not to alter the language.

References
[Dotz 84] Dotzel, Günter and Moritzen Klaus: Modula-2 and its Environment. In: Journal of Pascal, Ada & Modula-2. Vol. 3, Nr. 4, West Publishing Co., Orem UT, 1984, pp. 42-46.
[Dotz 84a] Dotzel, Günter and Moritzen Klaus: LIDAS: A Relational Database System. In: Journal of Pascal, Ada & Modula-2. Vol. 3, Nr. 5, West Publishing Co., Orem UT, 1984, pp. 32-37, 52.
[Dotz 85] Dotzel, Günter: Benchmark Test Results for Modula-2 Compiler. In: Journal of Pascal, Ada & Modula-2. Vol. 4, Nr. 2, John Wiley & Sons, Inc., New York, 1985, pp. 28-32. (Wiley is the new publisher of this journal since Jan-85)
[Dotz 85a] Dotzel, Günter: Modula-2 References. In: Journal of Pascal, Ada & Modula-2. Vol. 4, Nr. 4, John Wiley & Sons, Inc., New York, 1985, pp. 33-37.
[Dotz 86] Dotzel, Günter: Modula-2 I/O Library Proposal. Accepted for publication by Wiley & Sons, Inc., New York, in the Jan/Feb '86 issue of "Journal of Pascal, Ada & Modula-2".
[Dotz 86a] Dotzel, Günter: On LSI-11, Megabytes of Memory and Modula-2/VRS. Will be published in the Jan '86 issue of the DEC® PROFESSIONAL, The Magazine for DEC Users, Professional Press, Spring House, PA.
[Wirth 85] Wirth, Niklaus: Programming in Modula-2. Springer Verlag, New York, 1983 (3nd ed.).
[Zehn 83] Zehnder, C.A.: Database Techniques for Professional Workstations, ETH-Report No. 55. ETH Zürich, Institut für Informatik, CH-8092 Zürich/Switzerland.

DECUS
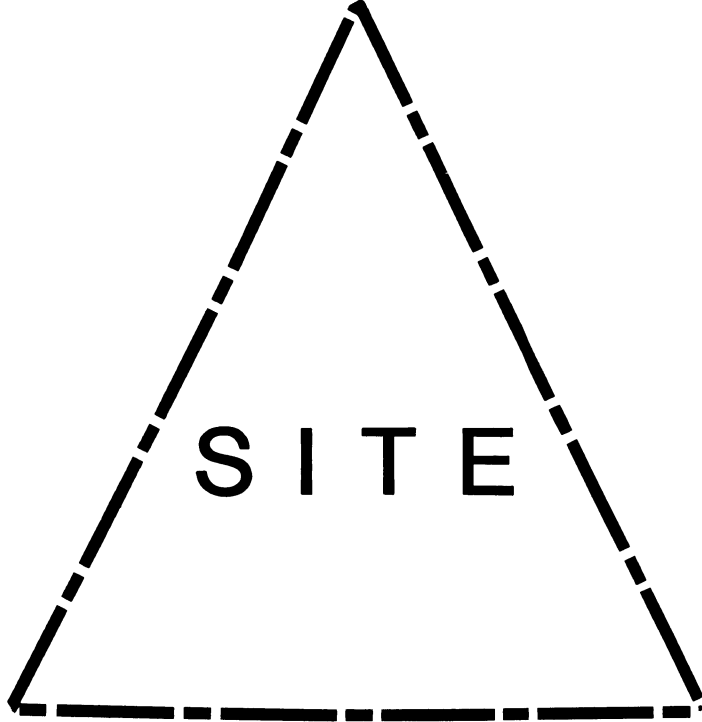
SITE

# SITE

Chairman
HMS SIG Liason
RT SIG Liason
    David Hunt
    Lawrence Livermore National Lab
    MS L-230
    P.O. Box 808
    Livermore CA. 94550
    (802) 656-3190

Symposia Coordinator
VAX SIG Liason
    Michael Weaver
    OJCS/JAD/TSD
    Room 1D940
    Washington D.C. 20301-5000
    (202) 694-6772

Communications Committee Representative
AI SIG Liason
    Terry C. Shannon
    Digital Review
    160 State St.
    6th Floor
    Boston, MA. 02109
    (617) 367-7190

Newsletter Editor
Networks SIG Liason
OA SIG Liason
    Gregory N. Brooks
    Washington University
    Behavior Research Labs
    1420 Grattan St.
    St. Louis, MO. 63104
    (314) 241-7600 ext. 257

Session Note Editor
Large SIG liason
    Gary Bremer
    Emerson Electric Co.
    8100 W. Florisant
    St. Louis, MO. 63136
    (314) 553-4448

Product Coordinator
HMS SIG liason
    Jim Corrigan
    KI Research
    2116 E. Arapaho
    Suite 521
    Richardson, TX.
    (214) 960-5674 or (714) 633-9308
Assistant Library Coordinator
RSTS SIG Liason
    Timothy Frazer
    Specialized Bicycle Components
    15130 Concord Circle #77
    Morgan Hill, CA. 95037
    (408) 779-6229

Library Coordinator
DMS SIG Liason
Large SIG Liason
    Larry W. Hicks
    Relational Database services
    P.O. Box 644
    121 S. Main St.
    Kernersville, NC. 27285-0644
    (919) 996-4882

Hardware Coordinator
HMS SIG Liason
    Emily Kitchen
    A.H. Robbins Co.
    1211 Sherwood Ave. RT-2
    Richmond, VA. 23220
    (804) 257-2925

Staff Management
    Adam Zavitski
    720 N. Zenith
    Tulsa, OK. 74127

Pre-Symposia Seminar Coordinator
    Phillip Ventura

Members-At_Large

    George Hilton
    Compucard International
    777 Summer St.
    Stamford, CT. 06091
    (203) 324-9261

    Debbie Boole
    Texas Instruments
    13510 N. Central
    M/S 437
    Dallas, TX. 75266
    (214) 995-4629

DEC Counterparts

    Sonja Sokal
    Digital Equipment Corp.
    Stow MA.

    Lil Holloway
    Digital Equipment Corp.
    Bedford MA.

    Susan Porada
    Digital Equipment Corp.
    Marlboro, MA.

    Gary Siftar
    Digital Equipment Corp.
    Tulsa, OK.

**LETTER FROM THE EDITOR**               Gregory N. Brooks

Welcome to the Site SIG Newsletter for May.

Since many of us will have just spent a brain racking week in
Dallas by the time this reaches the membership, I thought we might
turn to the lighter side of our profession with the article " The
Computer Experts Guide to Life" by Charles Bassine. ( Is that his
real name ?)

Even though we are all in the computer business, we all have a
favorite tale of the, so-called, computer expert that we
periodically drege up from the past.

On the serious side, the Site Newsletter is desperately in need of
contributions for the up coming issues. If you have something to
offer to your fellow Site SIG members, please send it in.

I have met so many bright, articulate SIG members at Symposia so I
know there has to be excellent quality material to be had, if only
it was down on paper. The biggest obstacle is the nature of the
business. We are in high demand at our work places. It's hard to
find time to do the work, much less write about it.

If you have anything to submit, you can send it to me at:

                    Gregory N. Brooks
                    Washington University
                    Department of Psychology
                    Behavior Research Labs
                    1420 Grattan St.
                    St. Louis, MO. 63104
                    (314) 241-7600 ext. 257

I can accept just about anything you submit.(that includes hard
copy and hand written) I would prefer Runoff files on VAX backup on
mag tape or RT-11 on RX02 or mag tape, but if you can send it, I
will worry about getting in the right format.

Look forward to seeing many of you in Dallas. Please come by our
table on Sunday or by the Suite on the designated night. Be prepared
for some wild (job related) story telling. Be sure to express
yourself on what we are doing (or not doing) with the Site SIG.

Keep your eyes peeled for SITEMAN. He's coming!


                            Greg Brooks

## THE COMPUTER EXPERT'S GUIDE TO LIFE

### (How to profit from the information revolution

### while avoiding hard work.)

### By Charles Bassine

Computers are not necessarily a good thing. Although computer
games offer diversion, there is little practical opportunity in
computers unless you are serious about computer crime.

Computer crime, however, requires insight, dedication, and
careful planning, and so is not for everyone. An easier way to
make money with computers is to become a Computer Expert. This is
not as rewarding as computer crime, but it is less risky and
requires much less work.

What is it really like to be a Computer Expert? It is like
this:

1. Your boss will not understand what you do. His eyes will
glaze if you try to explain it to him.

2. No one will ever know where you are, and if you aren't in
the office, every one will suppose you are in Poughkeepsie.

3. You will not be required to produce anything, ever.

Best of all, computers will be largely excluded from your
life. This is important in a world where everyone from typists to
lawyers is expected to be using, or thinking about using,
computers in his job.

This article provides essential guidance to Right Thinking
in planning a computer career. The guidance is sound and useful
because it is based on a simple principle; your success as a
Computer Expert varies inversely with your interest and
competence in computing.

Where to look for employment. A good rule is to avoid any
company that uses a great many computers to run its business. The
problems companies try to solve with computers are even less
interesting than computers themselves. Also, there is the
possibility, admittedly slim, that after enormous expense in
acquiring computers, people will notice that nothing is happening
and ask why.

If you have to work for a computer user, here are some
guidelines for selecting an appropriate employer:

--The company should have the most obsolete computing equipment
in the industry. This shows a proper view of the value of

computing.

--The funds spent on computing should be controlled by a Financial Officer at least 73 years old. He will not fret about why you are not suggesting computer solutions.

--The computer activity should be part of an enormous bureaucracy that has a reputation for being completely unresponsive to the needs of the operational departments of the company.

Never accept a job in a department that has its own computer and is trying to solve its own problems. The department you wish to work for is called, variously, MIS (Management Information Systems) or DPD (Data Processing Department). Its identifying characteristic is that it runs all of the computers for its own convenience, and tries to discourage others from using them.

The best employment for a Computer Expert is working for a manufacturer of computers. There are several reasons why this is so. For one thing, computer manufacturers make minimal use of computers in their own businesses. For another, knowing nothing about computers will help you get promoted to management in this environment. And finally, large computer manufacturers have a list of odd people who know and love computers and keep the company in business. If you can keep off this list, no one will notice you are around.

There are a few disadvantages, however. If you like city life you will have trouble. Perhaps for reasons of security--computers make people angry--computer manufacturers prefer more rural environments. Another factor may be a secondary interest in agricultural products. In the environs of one of the largest computer manufacturers, there are apple orchards, orange groves, and cattle. This manufacturer is not yet convinced that the computer business is here to stay, and is hedging its bet. It may also mean that the manufacturer feels that Computer Experts would pick fruit at low rates.

The working environment. A Computer Expert, like anyone else, must pay attention to external appearances. There are fundamental rules for going about this:

1. Keep your desk constantly cluttered with papers and articles. This suggests you are into things, learning, and doing.

2. Write memos. About anything. Choose subject headings judiciously, and don't worry about the body being relevant to the subject.

3. Always express concern and anxiety to your colleagues. Never be pleased with any technical decision your company has made.

4. Promise anything. Management forgets.

## BECOME A KNOW-NOTHING

Not choosing a specialty. The Expert must decide whether to be a generalist or a specialist. In general, one cannot start off by being a generalist. It takes time (but happily, very little effort). One must spend time in buildings with a variety of different kinds of projects. One need not be otherwise associated with them. Being a generalist is a long-term goal. What is a generalist like? First, he knows almost nothing about almost every aspect of computing, and can use jargon words in apparently correct sentences. To wit: "I think we generated the macros in the same way in 1959 on COMPSAC/465L SACCS on the old FSQ/38. Trouble was, we found the subroutine linkages very expensive."

In generalism, one claims to have done everything and to now know better. A generalist can also imply that any aspect of computing not known to him is not worth knowing, or dangerous. He does this with remarks like, "Well, of course, you can write articles about stuff like that, but I don't see it ever being practical." Equally useful is the opposite tack: "These people don't have the theoretical background they need. It's all seat-of-the pants and ad hoc. Goin' to get in real trouble."

Generalism is a good way to avoid the accusations of technological obsolescence Computer Experts are always using to discredit each other. A good generalist can take the offensive with cracks like, "He thinks a high-level language is one where you write programs on a tall stool." And, in the U.K., "The last manager to take him seriously about database design was William I."

Generalists are like designated hitters because they command high salaries and go on forever. As they go on, they discourage young computer science graduates with the feeling that it has all been done or that it is too hard to do. After speaking with a Senior Consultant, one young PhD from MIT (they deserve all they get, too) wrote to his mother that things were "so complex it sometimes seems nothing is worth doing."

This young man has had a serious insight: nothing is worth doing. It is the ability to move young people in this direction that makes an older generalist so valuable to a company.

Choosing a specialty. Since the status of generalist comes only with time, the new Computer Expert must start by picking a computing specialty. Do not be discouraged by the dreary list of possible specialties. A specialist need know only enough about his area to make nine simple declarative sentences in a meeting. It is important to know which verbs take direct objects and which nouns are inherently plural. For example:

Wrong: lock management are a very important part of distributed databases.

Right: lock management is a very important part of distributed databases.

Some specialties are better than others. Physics and electrical engineering are bad because they require formal education and have a high possibility of getting you work with a measurable result. These bad specialties have to do with hardware, the essential electronic componentry of computers. Because it is associated with bad specialties, there is getting to be less and less hardware in computer and more and more firmware and software.

CHOOSING THE BEST SPECIALTY

Software is a good specialty, and the best software specialties have to do with programs that no one really believes are necessary. Top candidates include:

--Operating system design. There is no known way to be successful at this and consequently no known way to fail at it. It is the creation of programs that make computers easy to use and efficient. It's best to design very large operating systems because such systems are never actually completed, no one knows what they do, and no one likes them anyway.

--Product requirements. This means predicting what people will want computers to be like in the future. No one will believe your predictions. No one will know what to do about them if they believe them because there is no known way of translating product requirements into actual computer products. This is a very good thing since no one really wants the computers they say they want anyway.

Most product requirements indicate that people want computers that are easy to use, reliable, easy to install, and easy to extend. This is just not true. People's jobs depend upon computers being hard to use, unreliable, difficult to install, and almost impossible to enhance. This is why people who actually make new computer products do what they damn please, and then announce that the machine has "good human factors."

Probably the best current specialty is computer education. This is an area with a number of subspecialties:

1. Educating people who know nothing but would like to use computers. Avoid this.

2. Educating people who know nothing but someone is insisting they find out. This is not bad. You will be talking to people much like yourself.

3. Educating computer experts who are trying to keep up with changing technology and advanced topics. Make it up as you go along; you are with friends.

Basic career tactics. We are in a men's room of the headquarters of a top computer company. An Industrial Democratization Program mistranslated from the Japanese forces Top Executives to share this facility with diverse Computer Experts. (The democratization does not, oddly, include the elevator. No corporate executive is willing to use elevators that

are used by Computer Experts.) As a Top Executive turns from the urinal he sees a Computer Expert. The Expert is at a neighboring urinal and remains there during much of the conversation. This is thought by many to be a masterful touch.

Exec: "Well, Siggy, are you excited about our announcement of Model 549 III?"

Expert: "Actually, I'm concerned."

Exec (still smiling): "Oh? concerned? Why?"

Expert: "I wonder if we have the security design right yet. You know Gilstrap and Fungio published a very good paper. It seems to me we designed..."

Exec (smile slipping): "Of course, I left the design to the team, I mean I have every confidence in Tom...."

Expert: "Well, I think there are some security problems in the addressing scheme. I think it is still possible to generate an address to protected data. Gilstrip shows that..."

Exec: "The machine is all right. Isn't it? We can sell it without going to the slammer? What will it cost to fix this? Can we meet delivery schedules? Good Lord. Half a billion bucks. And the damn thing doesn't work. I knew we should have stayed in the toaster business. Look, if it doesn't work, can you head up a task force?"

Expert: "Oh, it basically works. But it is not elegant. What Gilstrap and Fungio point out is..."

Exec: "Well, good night, Siggy. Don't stay too long."

Now, do we have here a scene in which an Expert sincerely interested in good computing is talking to a barbarian businessman who cares only about the bottom line? This is not likely because everyone is in it for the money. What we have seen, in fact, is an uncommonly elegant use of the Flake Play.

TRULY ADVANCED WORK

The Flake Play suggests that the Expert is an important thinker who really cares about good computing even at very private moments, but is too unstable to be relied upon for actual work. Staying at the urinal is very effective. It is truly advanced work.

An important aspect of Expert-Executive relations is never to be positive. An Expert must never say to an Exec anything like "Looks great, chief," or "Looks great, Fred."

Top Managements do not respect Experts who are polite and agreeable. They begin to think they are staff, or that they have something to hide. Be careful, though; it is equally true that Top Execs do not tolerate even the most trivial kind of dissent. The essence of Top Exec interactions is informed but ineffective

opposition. Make waves if you must, but don't make trouble.

Notice the Expert made no comment about market acceptance of the new computer. He spoke only of an obscure problem in machine organization. This is good work. A frank and critical comment about market impact would reveal that the Expert has no clue about what computers are used for, who would want them, and why they would want them.

Another important point is that a competent Expert would not do this scene unless Gilstrap and Fungio are real people who have written an article on computer design. It is not necessary, however, that:

1. Expert has read or understood the article.

2. Expert has any idea why the article is relevant to his company's new computer.

3. Expert really needs to stay at the urinal for so long.

Properly executed, the Flake Play is one of the truly elegant career plays, computing's equivalent to scoring from first on a long single. It is not to be used without great finesse, however. Its major risk is that it seems to demonstrate knowledge about computers to an executive of a company that designs computers.

The desired result of the Flake Play is for the Top Executive to draw certain conclusions about the Computer Expert: that he is not steady, that his judgments are not balanced, that he may have a bladder problem. It is the lack of good character that makes the computer knowledge acceptable and makes the Flake a valuable resource person--never fired and never thought of as a source of productive labor. This is the status that one searches for while in the full flower of a career in computing.

## THE COMPUTER EXPERT'S GLOSSARY

--ADA: something you need only know the name of to be an Expert in Computing. Useful in sentences like, "We had better develop an ADA awareness."

--Bug: an elusive creature living in a program that makes it incorrect. The activity of "debugging," or removing bugs from a program, ends when people get tired of doing it, not when the bugs are removed.

--Cache: a very expensive part of the memory system of a computer that no one is supposed to know is there.

--Design: what you regret not doing later on.

--Documentation: instructions translated from Swedish by Japanese for English-speaking persons.

--Economies of scale: the notion that bigger is better. In particular, that if you want a certain amount of computer power,

it is much cheaper to buy one biggie than a bunch of smallies. Accepted as an article of faith by people who love big machines and all that complexity. Rejected as an article of faith by those who love small machines and all those limitations.

--Hardware: the parts of a computer system that can be kicked.

--Information center: a room staffed by professional computer people whose job it is to tell you why you cannot have the information you require.

--Information processing: what you call data processing when people are so disgusted with it they won't let it be discussed in their presence.

--Machine-independent program: a program that will not run on any machine.

--Meeting: an assembly of computer experts coming together to decide what person or department not represented in the room must solve a problem.

--Minicomputer: a computer that can be afforded on the budget of a middle-level manager.

--Office automation: the use of computers to improve efficiency in the office by removing anyone you would want to talk with over coffee.

--On-line: the idea that a human being should always be accessible to a computer.

--Pascal: a programming language named after a man who would turn over in his grave if he knew about it.

--Performance: a statement of the speed at which a computer system works. Or rather, might work under certain circumstances. Or was rumored to be working over in Jersey about a month ago.

--Priority: a statement of the importance of a user or a program. Often expressed as a relative priority, indicating that the user doesn't care when the work is completed so long as he is treated less badly than someone else.

--Quality control: assuring that the quality of a product does not get out of hand and add to the cost of its manufacture or design.

--Regression analysis: Mathematical techniques for trying to understand why things are getting worse.

--Strategy: a long-range plan whose merit cannot be evaluated until sometime after those creating it have left the organization.

--Systems programmer: a person in sandals who has been in the elevator with a senior vice president and is ultimately responsible for a phone call you are to receive from your boss.

Charles Bassine is a top Computer Expert and an internationally
recognized authority. He recently received the Baud Institute's
Rita Hayworth Award for Uncompleted Projects in Information
Processing.

UNISIG

DECUS

UNI
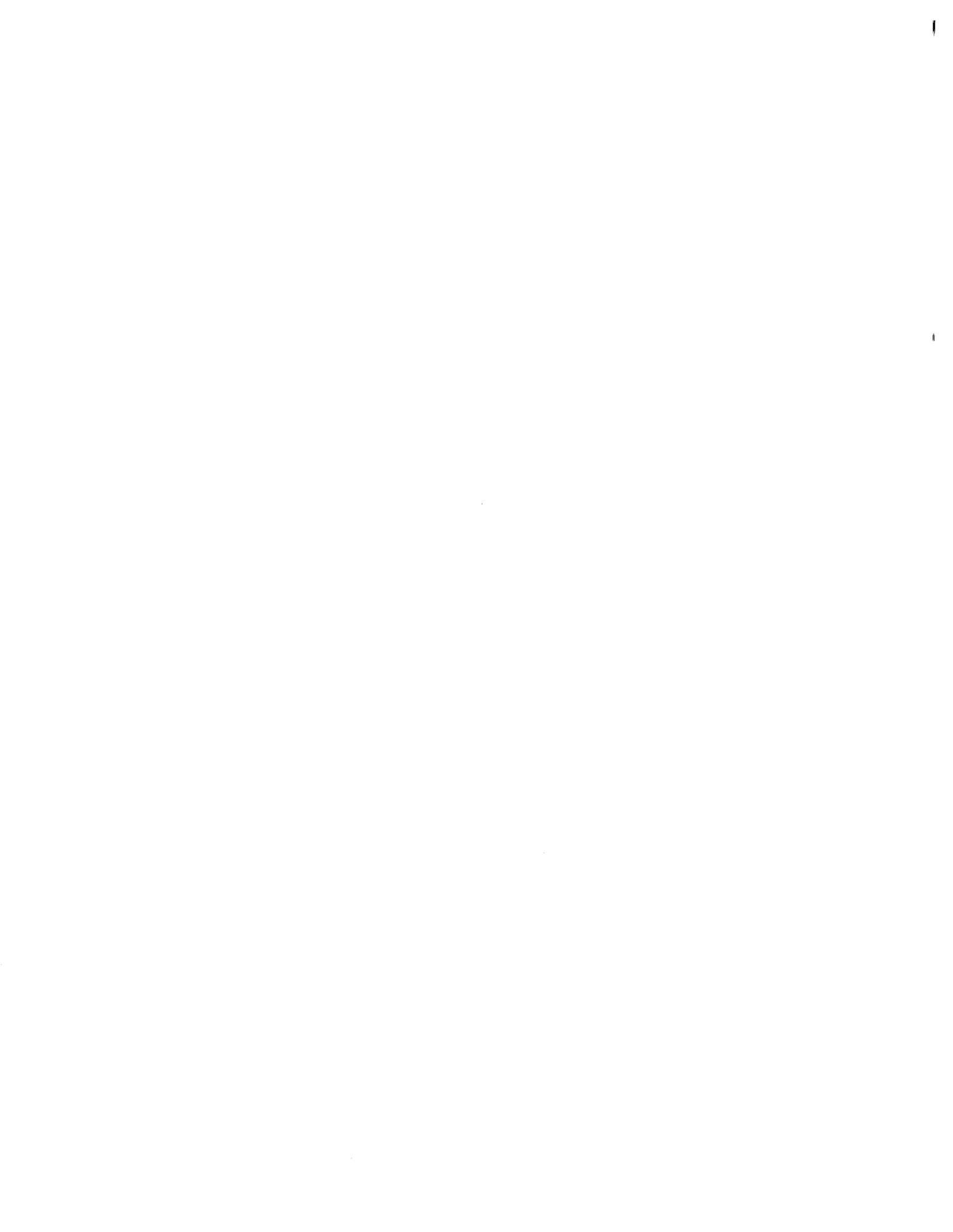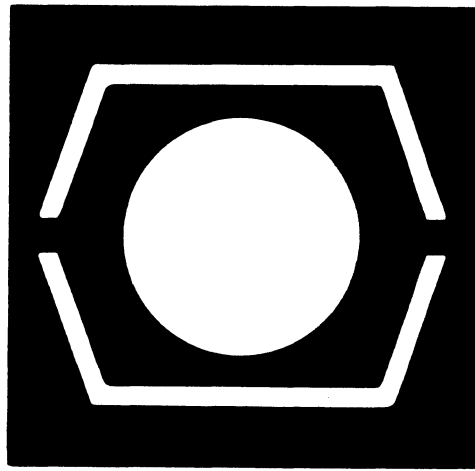
## UNISIG Chair

James W. Livingston, Jr.
Measurex Corporation
1 Results Way
Cupertino, CA  95014
408-255-1500 x5556

ihnp4!decwrl!jwl

## UNISIG Symposia Coordinator

Stephen M. Lazarus
Ford Aerospace, MS X-20
3939 Fabian Way
Palo Alto, CA  94303
415-852-4203

ihnp4!fortune!wdl1!sml

## UNISIG Session Notes Editor

Kurt L. Reisler
Hadron Incorporated
9990 Lee Highway
Fairfax, VA  22030
703-359-6100

decvax!seismo!hadron!klr

## UNISIG Newsletter Co-editor

William Toth
Harvard-Smithsonian
  Center for Astrophysics
60 Garden Street, P-353
Cambridge, MA  02138
617-495-7181

harvard!hrvsmth!toth

## UNISIG Newsletter Co-editor

James W. Livingston, Jr.

## UNISIG Administrative Daemon

Dorothy Geiger
The Wollongong Group
49 Showers Drive, 451
Mountain View, CA  94040
415-948-1003

ihnp4!decwrl!dgeiger

## UNISIG Tape Librarian

Carl Lowenstein
Marine Physical Laboratory
Scripps Institute of Oceanography, P-004
La Jolla, CA  92093
619-294-3678

(ihnp4|decvax|akgua|dcdwest|ucbvax)
  !sdcsvax!mplvax!cdl

## UNISIG Usenet Liason

Joe Kelsey
FlexComm Corporation
711 Powell Ave. SW
Renton, WA  98055

allegra!fluke!joe

## UNISIG Standards Coordinator

Jeff Gilliam
National Semiconductor
2900 Semiconductor Drive, MS C2303
Santa Clara, CA  95051
408-721-3801

ihnp4!nsc!voder!jeff

## UNISIG Minister Without Portfolio

Norman Wilson
Bell Laboratories, 2C-529
600 Mountain Avenue
Murray Hill, NJ  07974
201-582-2842

(decvax|ihnp4)!research!norman

## UNISIG DEC Counterpart

Roseann Maclean
DEC, MKO2-1/H10
Continental Boulevard
Merrimack, NH  03054
603-884-5702

decvax!maclean

# NEWSLETTER OF THE VAX SYSTEMS SIG

# Pageswapper

*Our Mascot*

VAX

General material for publication in the Pageswapper should be sent (US mail only -- no "express" services please) to:

Larry Kilgallen, PAGESWAPPER Editor
Box 81, MIT Station
Cambridge, MA  02139-0901
USA

Preference is given to material submitted as machine-readable text (best is Runoff source).  Line length should not exceed 64 characters.  Please do not submit program source, as that is better distributed on the VAX SIG tape.

Change of address, reports of non-receipt, and other circulation correspondence should be sent to:

DECUS U.S. Chapter
Attention: Publications Department
249 Northboro Road (BPO2)
Marlborough, MA  01752
USA

Only if discrepancies of the mailing system are reported can they be analyzed and corrected.

## Software Reviewers Wanted

One of the principal benefits of being associated with DECUS is the public domain software available on the Symposia tapes and from the DECUS Software Library.  Much of this software is outstanding and you can't beat the price - Symposia tapes free from your local LUG Librarian, a whole catalog of items at a nominal charge from the Library.

We pay a price for this software in other ways though:

1.  It is not offically supported,

2.  Documentation is frequently less than adequate,

3.  It may be dangerous.

Here is what I am getting around to.  You, as a user of a DECUS Library or Symposia tape program, can help the rest of us avoid the above expenses.  If you use something and like it, let us know.  We want to try it, too.  If you use something and don't like it, let us know.  We may still want to try it, but your comments can be valuable.

I am not soliciting reviews about copying problems.  If you have a copying problem you should get back to the person supplying the tape immediately and let him know that he has a problem.

I am soliciting reviews about DECUS Library offerings and SIG tape programs.  For widest dissemination an article for the SIG newsletter is appropriate.  If you have something to say about a VAX Library or tapecopy program but not the time to write it down come to the VAX Tapecopy User's Forum at the next DECUS Symposium.  If you just have a short comment you can give it to me over the telephone and I will pass it on.

Joe Bingham
VAX Systems SIG Librarian
ManTech Services Company
232Ø Mill Road
Alexandria, VA 22314
(7Ø3) 838-56ØØ

# Editor's Workfile

by Larry Kilgallen, Pageswapper Editor

DBMS Documentation Update –

Submitting this to the Data Management SIG Newsletter does not seem worth the effort, considering its (lack of) length, so here goes. If you are updating a DBMS documentation set to V3.0, beware of the Introduction to Data Manipulation. Although it comes in an orange cover (or did in my set) it is actually an UPDATE section, where you have to interleave the pages. If you follow the obvious visual clues and discard your old AA-Y306A-TE without taking the plastic off the new one, you will be in for a surprise. Of course the TRULY paranoid among us have a cardboard box in which we save all the pages ever removed in updating over the past 3 years for fear we made a mistake.

Digital's Sales System –

A coworker was moved by my plight and when attending DECworld ordered a brochure by the aforementioned title. It turns out to be about some software DEC is offering for sale, but the note my coworker sent to me with the brochure said: "I thought that I was ordering a brochure on DEC's labyrinth of sales organizations."

# Letters to the Pageswapper

Ken Levitt
Informed Computer Solutions, Incorporated
9 Linn Lane
Wayland, MA 01778
617-237-7427

March 15, 1986

A few years back, if you were talking about a VAX, you were talking about a 780 or a 750. These were large expensive machines which almost always had a 9 track tape drive or a network connection. If you didn't have either of these things, you were in a very small minority. Then came the 730 and 725. These were strange little machines which might have had a 9 track tape drive but so few people bought them that no one gave them much thought. Now we have reached modern times and DEC has shipped thousands of MicroVAXes with TK-50's rather than 9 track tapes. If the MicroVAXes don't already outnumber all the other VAXes combined, they soon will. Most people who would be reading this letter know all this. However, the people at the DECUS Library seem to be still living in the past. A call to the folks at the library sounds like they never heard of a TK-50 tape cartridge and that they don't know that thousands of users have them as their only source of magnetic input.

I ask that the VAX Systems SIG put pressure on the DECUS Library to offer TK-50 distributions. I also think that the symposium Tape Copy Facility should accept and distribute on TK-50. I have been trying to get a copy of the last symposium tape on TK-50, but have thus far been unsuccessful.

I understand that there are plans for DECUS to distribute software over the phone lines. I strongly support such a move, but I doubt that it would work out for very large distributions.

On another topic, a MicroVAX that I installed is used 99% of the time for one custom application. A few hours a month one user would like to use an accounting package and a work processor. The company is very small and has very simple needs. For the cost of the software on the VAX, I could buy a complete Rainbow and software to go with it. The problem is that I don't want to buy a Rainbow:
I want access to the system printer.

I want to pass some data from the custom application to the accounting system.

I don't want to buy DECnet for my VAX.

I want to be able to dial in at night and use the word
processor.

I like knowing that all of my data files are getting backed
up every night.

The MicroVAX has excess capacity.

I don't want to move to another office to do word
processing when I already have a terminal on my desk.

In short, I hate distributed processing.

In summary, why can't I buy a single user VAX license for a word
processor and an accounting package at an affordable price? If
anyone out there knows of software that fits these requirements
and doesn't require a compiler on the VAX, please let me know.

Ken Levitt

## DR780 Issues

by Mark Willie Wilson
Texas Instruments, Incorporated
Image Processing Lab
Post Office Box 405 M/S 3407
Lewisville, Texas 75067

February 27, 1986

Now that the Fall Symposium is over, I would like to fulfill my
promise to be a point-of-contact for DR780 issues. Other people
have expressed interest in the DR780 besides those in the DAARC
SIG, VAX Systems SIG and HMS SIG. A few working groups have
been organized to address problems of "High throughput IO" or
"Realtime IO on VMS".

We all have had the same question at these meetings, i.e., "How
do I get more IO bandwidth on my VAX". The DR780 is a general
purpose data port, capable of moving up to 6 Megabyte/sec on a
VAX 11/780. This product has limited documentation. The
impression I have gathered from the few people that own a DR780
is that there is hidden work involved in implementing this
interface. Some applications have required writing an ACP.

I would welcome any comments, war stories, applications, and
recommendations. Any caveats, warnings, workarounds, code, or
spare parts would be welcomed and published with my own
experiences as long as the need and I exist. Any other product
that attaches to a DR780 could also be discussed in working
group or newsletter form.

# Maturity of Products vs

## Maturity of Engineers and Product Managers

by Marian J. Kowalski
Commercial Union Insurance Companies
One Beacon Street, Boston, MA  02108

Having used Digital products for the last six years, I have seen quite a few products come and go.  I have also seen some products that didn't quite get here and some that Digital shooed out the door as soon as they arrived.  These have caused me some doubts about Digital's seriousness about its commitments to its products and customers...and about its understanding of what it means to be in business.  The decision to call a halt to EDT enhancements in favor of TPU is the latest example of this behavior, and it leads me to recall a few similar experiences.

In early 1982, Digital announced that it was selling its own C compiler-- VAX-11 C.  At that time, my company was developing software to run on a variety of micro- and minicomputers for insurance agents.  We went shopping for a language that would allow us to do development on the VAX, taking advantage of its many development tools, and then port the code to the various machines for agents.  At the Spring 82 Symposium, ı attended several session about the new C compiler and heard the promises of related tools that were soon to come--especially a C debugger.  C appeared to be the answer to our prayers, and we jumped into it with both feet.  In spite of a downturn in the insurance industry, we spent hundreds of thousands of dollars on training programmers in C, buying C compilers for the various machines that would use it, and beginning the project with another vendor's VAX C compiler, with the expectation that Digital's superior compiler and debugger were just around the corner.  By the time that I attended the Fall 83 Symposium, there was still no C debugger, and we had abandoned the use of C as impractical in its absence.  At the symposium, I had the pleasure of hearing a speaker from the original C compiler development group.  He talked about the fun they had in dreaming up a hare- brained project like a C compiler and the many laughs they had in implementing it.  And, what the heck, so what if there's no debugger and everyone on the development team has moved on to more interesting products?  Isn't that what we're all in business for--a laugh?

A year ago, our Digital salesman was working to sell us a new terminal server system: the LAT-11.  At the time, we had other priorities, but another company in this area bought one.  A few months ago, Digital told them that it had become a mature product--at Version 1.0--and that there would be no enhancements to it.  Is that the way that your company expects you to operate your department--just abandon programs when you've had enough?

Although its lifetime was a few years longer, EDT has been declared a mature product, too.  Its replacement is inadequate, although I've heard of plans to improve that.  But why should we have to go through the same agonies with TPU to get the features that we worked to get into EDT?  I've heard that EDT's performance is Digital's reason for moving on to TPU.  What are the odds that TPU's performance will be any better when it has the same features as EDT?  When I hear reasons like that from a department in my own company, it usually means that the group's manager has no business sense or is being blackmailed by a bored hot-shot (what my father called a Bright Young Engineer) who wants a new toy.

There may very well be legitimate reasons for some of the cases that I have described.  As a businesswoman, however, I suspect that some of them were also cases of a bored product manager or a group of engineers who declined to perform maintenance/enhancement work.  It's tough work--much more demanding than development, since it requires an understanding of a system rather than just a program, as well as the ability to read another programmer's code and mind; also, maintenance work is expected to be perfect the first time, while developers are given more leeway.  Perversely, it has never had the prestige of development work.  In fact, many foolish managers consider maintenance fit only for trainees' time and effort.

So I wouldn't blame any of Digital's engineers for wanting to move on to a nice, fresh development project.  But I would blame their managers for allowing them to do so at the expense of customers who are using existing products.  Maturity for engineers and product managers means recognizing that maintenance work is implied in Digital's commitment to a product.

## Making Timesharing Volume Backups Less Intrusive

by Larry Kilgallen, Software Consultant
Box 81, MIT Station, Cambridge, MA  02139-0901

Although others might find something in this article of passing interest, it is written with a view toward VAX shops which are used mainly for timesharing purposes (be it in-house or service bureau) rather than shops where one central piece of software consumes the whole machine or cluster. Likewise the presumption below is that the shop has operators on duty, or someone who performs that function, rather than a situation of "everyone for himself" when it comes to backups.

### Do You Really Care?

If your volume backup habits are to use the /IMAGE qualifier once a year, whether you need it or not, then this article is not for you. Likewise, if you feel no desire to get a "static" snapshot of your disk which can be restored in the event of catastrophe, then a noontime Backup with all the files still open may meet your goals.

If you are looking for guidance on basic use of the Backup command, then this is also not the article for you.

### The Historical Method - SHUTDOWN

With a single-disk system, Stand-alone Backup is all that one can hope for. (If you EVER have bad feelings about Stand-alone Backup, have someone who has been around for a while recount to you the stories of its predecessor, Stand-alone DSC).

A single-disk system leads to a great deal of simplification of the volume backup planning process. Once you have committed to doing your volume backups "cleanly", with the volume in question dismounted, it is obvious that the whole system is going to be shut down, because the disk in question IS the whole system.

With a multi-disk system, you may yearn for the standardization provided by DCL command procedures, and elect to boot a minimal VMS system from the "other" disk in order to back up a disk. (Use VMSKITBLD.COM to put a minimal copy of VMS on this "other"

disk for use backing up your normal system disk.)

In case there are any beginners still reading, make sure you use the ANALYZE/DISK/REPAIR command before doing a volume backup, as it reduces the chance of getting error messages from Backup which are unconnected with the Backup process itself. In the case of single-disk systems, this means using the ANALYZE/DISK/REPAIR command before shutting the system down.

### The Large System Problem

If your VAX has 8 disks, however, bringing the whole system down for backups can lead to much grumbling about availability from the user community. The prime reason is that you probably don't have 8 tape drives. So you end up backing up the disks one-by-one. (If you DO happen to have 8 tape drives you better have an 8600 to be able to run 8 simultaneous copies of Backup at reasonable speed.)

So if backing up your RP07 takes x microfortnights, backing up all 8 of them will usually take 8x microfortnights of wall-clock time.

### Dismounting Just One User Disk

Well, users realize that their disk has to be down SOME part of the time in order for you to get a good backup copy (I know the users weren't born with that knowledge, I presume you have beat it into them). The trick is to let users on disk 3 continue to run while disk 2 is down for volume backups.

This certainly involves some custom software on your part, because there is no DEC code to shoo just certain users off the machine. In some environments knowing the usernames gives you information about what disks they use, but more often you will have to automate the process. Parsing the output of the SHOW DISK/FILES command will let you know which users are currently USING disk 2, but not a word about which users might reasonably EXPECT TO USE disk 2.

One technique is to have the system login command procedure mount each user's SYS$LOGIN_DEVICE for them /SHARE (in addition to the /SYSTEM mount it already has) and then use the mount count to see if any users based on that disk are still logged on when the backup is about to start. This requires that users who

are using disks other than their SYS$LOGIN_DEVICE mount those
disks /SHARE as well. There is no VMS software short of SDA to
tell you WHO has those outstanding mounts, so a little "local
knowledge" comes in handy.

Another method of learning which users have disk 2 as their
SYS$LOGIN_DEVICE is to scan SYSUAF, but again users who expect
to access disks other than their SYS$LOGIN_DEVICE must mount the
disks /SHARE (on top of the /SYSTEM mount) to stake their claim.

As for shooing the errant users away, if the system manager does
not write a command procedure, then the operator will, so you
might as well accept the need from the beginning and take a look
at SHUTDOWN.COM as an example. If you have a protective
attitude toward users, your site-specific shutdown command
procedure probably already takes precautions to not blow batch
and print jobs out of the water on shutdown without operator
involvement; remember to carry those techniques over to the
command procedure you create to clear a disk for backup.

Beyond interactive and batch users, there is another "use" of
the disk which must be run down -- installed images (known
files). Here is where you get a chance to parse the output of
SHOW DEVICE/FILE, unless you can be SURE that no image is
installed without being entered in some table you can read to
find out what to de-install.

Once you have dismounted the disk and mounted it privately,
users cannot submit any batch or print jobs which reference it,
but what about jobs they might have submitted earlier which had
not yet started or had a /AFTER qualifier?


### A Role for Queue Characteristics


I have always wondered what these were for, and why there were
128 of them. I know that sometimes people have special print
jobs and special printers, but I couldn't face up to the fact
that there was anything in VMS of which there were 128 and I
hadn't figured out how to use up more than a handful. Well,
here it is:

Assign a characteristic to each disk volume used on your
machine/cluster (including dismountables). There is no
requirement that you assign names to characteristics, but I use
DISK_GREECE as the name for the characteristic associated with
DISK$GREECE (using DISK$GREECE does not produce the desired
result because VMS attempts logical name translation on
characteristics). This characteristic on a queue means that the
queue currently has access to that volume. This characteristic

on a job means that to run successfully the job must have access
to that volume.

At system boot, you give all the execution queues for your node
the characteristics for all the disk volumes presently mounted.
(DCL syntax of the SET QUEUE command requires that you give the
whole list in one command rather than "adding" them on.
Characteristics don't "add".)

At backup time, you TAKE AWAY from all execution queues the
characteristic for the disk volume to be backed up. This will
prevent new jobs which have the characteristic from starting.
Note that to clear a single characteristic from the set
associated with a queue requires learning the full list and
setting the full list again, minus the one you want to take
away. I certainly do not recommend attempting this in DCL (even
if you get a copy of DCL with an F$GETQUI lexical function). So
pick a compiled language, bite the bullet, and call SYS$SNDJBC
(note that the binary format it uses for expressing
characteristics is entirely different from that returned by
SYS$GETQUI).

The politically astute reader has realized that users will not
stand for requirements that they specify that extra
/CHARACTERISTICS qualifier on every little SUBMIT or PRINT
command. For MOST jobs you can relieve users of the burden of
adding /CHARACTERISTICS by adding certain characteristics to the
jobs algorithmically. (Your program was going to scan the
queues with SYS$GETQUI anyway). For EVERY job in EVERY queue
(execution or generic) ensure that at LEAST the following
characteristics are set:

   o  Device for each file in the job (the job controller in
      its befuddlement makes you reason backward from
      physical device names)

   o  For batch jobs - SYS$LOGIN_DEVICE for the username

   o  For batch jobs - disk specified for any explicit /LOG
      qualifier


Having taken care of those algorithmic chores, you leave the
users to specify SUBMIT/CHARACTERISTICS for any OTHER disks
referenced by their batch jobs (remember that you must perform a
logical OR of what you find already specified for a job with
what you decide based on your queue scan).

It turns out that you cannot modify the characteristics of a
running job, but that is all right, because if you really need
to change a running job (because it uses the disk you are about
to back up) you cannot proceed with the backup until that job
finishes anyway. So make a distinction between those

alterations absolutely required for this backup and those performed gratuitously so the job will be fully qualified with respect to other disks when the time comes for THEIR backup. If the change your program would make to an executing job does not set the same characteristic you are removing from queues, they you can just bypass the executing job. If the change your program would make to an executing job DOES add the characteristic you are removing from queues, you cannot complete your action until that job finishes executing. You may, however, choose to continue scanning the rest of the queues and before exiting your program catch up with all the jobs which were executing. You DO have to attempt to change their characteristics then, since the job might still exist in a queue (not necessarily the same queue) through operator action or the /RETAIN qualifier on a queue, and thus be subject to re-release.


                        After the Backup


When the backup is complete, mount the disk and set its characteristics back into the execution queues.


                        About Clusters


On clusters shooing the users off is more complicated since it must be accomplished on a distributed basis. There are two possible avenues (for those of us without the SCS programmers handbook): DECnet and Batch.

Everything you did on a single node, do it on all cluster members and synchronize. One consideration I failed to recognize from the beginning is that the DCL command DISMOUNT completes asynchronously. Once the disk has been dismounted on all nodes, it may still show up on various nodes as still being mounted on other nodes. So far, a WAIT 0:5 command seems to solve the problem for me.

## Fall 1985 VAX Systems SIG Symposium Tape (Disneyland)


The Fall VAX tape was sent to the DECUS Program Library and to the National LUG Organization (NLO) in February. The tape is available form the Library (order number V-SP-49), will be on sale at the DECUS Store at the Dallas Symposium (April 28-May02, I know, history by the time you get this.) and from your local LUG Librarian through the NLO distribution.

This tape consists of the VAX submissions to the Tapecopy project at the Fall 1985 DECUS Symposium in Anaheim, CA. As usual there is a large quantity of valuable material. A very brief description follows. For a more complete description of the contents of each submission see the AAAREADME.TXT files in each submission or the concatenated version of the AAAREADME's in [VAX000]; for documentation check for pointers in the AAAREADME.TXT'S, for files containing the string READ and for files with .DOC, .TXT, .MEM, .RNO and .1ST extensions in the submissions.

NOTES:

1.  Many of the submissions were submitted with VMS version 4 filenames. These files were renamed to names compatible with VMS version 3 and command procedures were constructed to restore the version 4 filenames. Therefore, version 3 sites will have no trouble loading the tape but some filenames will be inconsistent with documentation and command procedures. Version 4 sites will want to follow the instructions in [VAX000]LOADING.TXT to rename the affected files back to their original names.

2.  This tape does not contain the [VAX000.INDEX] directory. We expect to provide index files for this tape and a composite index file with the Spring 1986 tape.

3.  If you get your tape through the NLO I presume the format of the tape will be as I submitted it to them, i. e. three save sets, VAX000, VAX85C and VAX85D which will fit on two 2400 foot tapes at 1600 bpi (VAX000 and VAX85C on one and VAX85D on the other) or on one tape at 6250 bpi. If you get your tape from the Library everything will be in one save set with the three directory trees as subdirectories to a top level directroy added by the Library. This should present no problems - directory tree depth is not a problem with this tape and the command procedures to rename files use relative path names.

This tape was put together by:

Joe Bingham
ManTech Services Company   and
2320 Mill Road
Alexandria, VA 22314
(703) 838-5600

Glenn Everhart
RCA A&D Engineering
Rt 38, Bldg 206-1
Cherry Hill
New Jersey 08358

Fall 1985 Tapes VAX85C and VAX85D Submissions Overview

VAX85C Tape ------ ----

[.AMS]
DEPROC - Header for formatting DECUS proceedings article usinbg TEX.  Barbara Beeton, Am.  Math. Soc.

[.BATTELLE]
FILES - find files by size or owner.  ZDEC  Zero Device  Error  Count for VMS V4.  BYDISK proc to do a DCL proc for all disks on a system.  KRON - proc  to do things at scheduled times.  PERMIT - easy ACL editing.  PTY - Pseudo terminal  driver and  session logger (PHOTO) for VMS V4.  By Gary Grebus and Mark Oakley, Battelle.

[.BELONIS]
HOST and MODEM7 - micro compatible file transfer utils.   XMODEM   -   ditto.   ACCOUNT  -  V4.x accounting  system.   QPOST  -  Talaris  printer support  for  TEX.   MOVE - another SET DEFAULT program.   SIX  -  simple  extractor  showing selected  lines  from files.  TERMS - LOGIN.COM compiler.  VVCODE - translates VMS files/headers to/from  printable text for comm.  transmission. Jim Belonis, Univ.  of Washington.

[.BONNER]
Bonner  Lab  RUNOFF  (in  NATIVE MODE!)  large superset  of  DSR  with  MANY  extensions.  John Clement, Rice Univ.

[.BRIDGE]
DOCUMENT - extract material from src files  with delimiters.   WHO - who's on the system.  LIMS - programs  and  report  writers.  Adam  Bridge, Multiware.

[.BULLETIN]
VMS Bulletin Board from ARPAnet.   Mark  London, MIT.

[.COSTELLO]
TPC  format  independent  tape-disk-tape  copy routine  in  native  mode  for VMS.  FAT - Pixel douibler  for  dumping ReGIS images on LN03. Dennis Costello and Larry Muray, Cornell Univ.

[.CWAX]
PVT -  print  VT1xx  or  VT2xx  terminal  screen images  on  printer.  Andrew Wax, Chemical Bank, NYC.

[.DFWLUG]
ALLIEDELEC - SMG utility library.  WHERE - shows where  one  is.   Utility  to get current system uptime into a symbol.  Simple  snapshot  to  see how  much  free space ther is on all disks, plus SNAPSHOT for V4.  Dallas/Fort Worth LUG.

[.DMREV]
DM - Directory  Management  for  VMS  V4.   Rich Gregory, Pharmaceutical Research Assoc.

[.ERI]
HALFTONE - convert  gray  scales  for  print  on LA50.   MACSNVAX  -  Macintosh  file  transfer utility and several Macintosh applications.  Bob Goldstein, Daniel Smith, Eye Research Institute.

[.EVEPLUS]
Extensions to EVE interface of TPU.

[.EYE]
DISKMON - watches disk usage, warns of impending running out of room.  Eric Richards, Gould OSD.

[.FERMILAB]
ALLOCWATCH - watches allocation  of  devices  so other  cluster  members can wait for them.  EDTX - expanded EDT  with  file  mem,  wildcard  files, spawn,  etc.   EXPAND  -  allow conditionalizable FORTRAN  programs.   LA100SMB  -  flag  pages  at dense  modes for LA100.  NODEIDS - get node names as  identifiers  for  ACLs  to  use.   SETUSER - eliminate   need   to   explicitly  provide  UIC associated  with  a  username.   STARTUP  -  system startup  command file examples.  TECO macros for mass  changes in UAF records.  TELLSELF - allow detached  or batch jobs to BROadcast messages to parent process.  More.  Frank Nagy, Fermilab.

[.GARMAN]
DFRAG - Disk  Fragmentation  reporter.   Charles Garman, RCA MSRD.

[.INQUIRE]
INQUIRE and  PROTO  -  DTR  update  and  program generators.  Florence Bowden, UCC.

[.LEVINE]
INDEX - super FORTRAN indexer, cross referencer, and  static  analysis  tool.  Also  some  VT200 utilities  including FONT  to  create/edit  VT200 fonts.    Directory   tree   painter,   disk fragmentation   utilities,   inactive   terminal process  killer,  more.  Michael Levine, Naval Weapons Ctr.

[.LJK]  Pageswapper articles since Spring 1985 tape. Larry Kilgallen, Cambridge, Mass.

[.MANTECH]  OBSERVE - allows you to watch another (TT or TX) terminal on your machine. No extra hardware required. DEFAULT - Set defualt program. Mark Crego and Joe Bingham, Mantech.

[.NSWC]  SCHEDULER - run processes on future dates based on logic (for periodic, longterm repeated needs; able to run procedure on day before holidays that are on weekdays, other unusual needs). REMINDER - improved calendar and tickler. Fixed for better cluster work. SUBMITIF conditional submission. Notify message sender. Alan Zirkle, NSWC.

[.PFILE]  Kernel mode no delete protection for files. Good for synonym files and directories. Ken Coar, Gen. Dynamics.

[.RAINIER]  Ada tools and support environment. Stephen Rainier, MITRE.

[.RAWIO]  Unix style raw and cbreak i/o. Forrest Kenney, DEC.

[.SAOSTOIC]  STOIC - stack oriented language like FORTH. RED - screen editor/WP. CALC - RPN calculator. Updates for VMS V4. Jonathan Mark and Roger Hauck, Smithsonian Instutute.

[.SKUNK]  SETDEF program. EDT with spawn. SEND messages to users.' Skunk LUG. (Ames, Iowa)

[.SMGLIB]  SMG screen library. Ken Messier, Allied Electronics. (may be same as DFWLUG one).

[.TIMELINE]  A number of VMS utilities, notabiby VERB to extract verb definitions from DCLTABLES. Joe Meadows, Timeline/Why Systems.

[.TPUEDT]  Documentation on TPU EDT simulator and customizer. Richard Piccard, Kalamazoo College.

[.TSUME]  C program to solve mating problems in Japanese chess. Nigel Haslock, AGS Computers.

[.UAB]  LIST - TPU template for file lister. GRADE - class grader system. SMAUG - CPU hog priority adjuster. Mark Vevle, Univ. of Alabama at Birmingham.

[.UWRF]  ACCESS - system for easy add/delete user account information. CALENDAR - print calendar of any month with text in blocks. EBS - Emergency broadcast utility (also with /SELF qualifier). FRAGMENTATION - show disk fragmentation. LISTER - source lister with titles on each page. PEN - Pascal Environment. PRIV - get privs in a subprocess if parent is authorized but not enabled. PROSE - Runoff-like processor. QUOTA - gets acct quotas in V3 format (for VMS V4). RESERVE - terminal reservation system. SCRUNCH - remove comments from DCL. SETFEEDER - set up Diablo 1630 sheet feeder. SGO - GO game. VAX users' guide. More. Marlys Nelson, Univ. of Wisconsin, River Falls.

[.VFE]  Block oriented, type insensitive file editor. Ward Condit, Maricopa Community Colleges

[.WATCHDOG]  V4 update of WATCHDOG, idle process killer. George Walrod III, American Satellite Company.

VAX85D Tape

------ ----

[.BNELSON]  Kermit-11 (and .EXE for VMS Kermit) communication program. VT100 emulating IBM PC Kermit. TED - good full screen editor/WP for VAX, PDP11, micros...

[.CI]  DROIDS game and SYSTATUS4 system status, from Ken Richardson.

[.DENNISON]  Grammar correction system and spelling checker for VMS V4. Dennison University.

[.FORTH]  FIG Forth for VMS (native mode). Roderick Eldridge, Central Iowa Forth Interest Group.

[.HALL]  Game manager (and large collection of games). GETWS - show work set stats. NEWS program. REMINDER update (patched to eliminate time bomb) [n.b. - this version NOT tested by librarian to verify time bomb is reset.] SB - limited login time enforcer. Rand Hall, Northeastern University.

[.KMSKIT]  Vax Professional Workstation (i.e., most all of All-in-1 free!) with various additions since Spring '85. Many MANY other utilities included (including NOTEBOOK, which is handy for doing short procedures that DTR startup overhead makes

impractical with DTR.) Complete window and graphics subsystem for VMS. Jim Downward, KMS Fusion.

[.RCAF85]   ANALYTICALC - spreadsheet/database system for VAX (native mode) and PDP11, 32000 by 32000 cells, DTR-32 interface, graphics, data file read, WP integration features, spawn DCL, MUCH more. (Also faster than most commercial spreadsheets...), with new micro-like command mode option, batch support, etc. DTCVAX - Desk Top Calendar update from Charles Garman that works from 1 AD to 9999 AD, many enhancements and speedups. VPW mods (as illustrations) showing how VPW from this tape can be customized for YOUR site's software. Database mgr. More. Glenn Everhart, Charles Garman, RCA.

[.UALR]   BBS - VAX Bulletin Board system. POSTERS - biggest collection of ([fortunately!] squeezed) posters ever. Menu driven operator system. University of Arkansas at Little Rock.

[.VASSAR]   Full screen spelling corrector update (uses SMG now). BB - VMS Bulletin Board system. RESERVE - terminal reservation system. Mark Resmer, Vassar College.

## VAX System SIG Committee List

As of January 8, 1985

Osman K. Ahmad - Large Systems Integration Working Group
        Association of American Railroads
        Technical Center, Research and Test Department
        3140 South Federal Street
        Chicago, IL 60616

Joe Angelico - Assistant Symposium Coordinator
        US Coast Guard CCGD8(DT)
        Hale Boggs Federal Building
        500 Camp Street, New Orleans, LA 70130

Elizabeth Bailey - Volunteer Coordinator
        222 CEB
        Tennessee Valley Authority
        Muscle Shoals, AL 35660

June Baker - Advisor
        Computer Sciences Corporation
        6565 Arlington Boulevard
        Falls Church, VA 22046

Joe L. Bingham - Librarian
        Mantech International
        2320 Mill Road
        Alexandria, VA 22314

Bob Boyd - Commercial Working Group
        GE Microelectronics Center
        MS 2P-04
        Post Office Box 13409
        Research Triangle Park, NC 27709

C. Douglas Brown - Security
        Sandia Labs
        Division 2644
        P.O. Box 5800
        Albuquerque, NM 87185

Jim Caddick - VAXcluster
        General Datacom
        Strait Turnpike
        Middlebury, CT 06762-1299

Jack Cundiff - Symposium Coordinator
        Horry-Georgetown
        Post Office Box 1966
        Conway, SC 29526

Tom Danforth - Handout Editor
    Woods Hole Oceanographic Institute
    Woods Hole, MA 02543

Jim Downward - Migration and Host Development, VAXintosh Working
    Group
    KMS Fusion Incorporated
    3941 Research Park Drive
    Ann Arbor MI 48106

Jane Furze - Campground
    3830 West Cochise
    Phoenix, AZ 85064

Dennis Frayne - Real Time/Process Control Working Group
    McDonnell Douglas
    5301 Bolsa Avenue
    Huntington Beach, CA 92646

Carl E. Friedberg - Internals Working Group
    In House Systems
    165 William Street
    New York, NY 10038

Don Golden - Communications Committee Representative
    c/o Shell Oil Company
    Westhollow Research Center
    Post Office Box 1380, Room D2132
    Houston, TX 77001

Gary Grebus - System Improvement Request
    Battelle Columbis Labs
    Room 11-6011
    505 King Avenue
    Columbus, OH 43201-2693

B. Hancock - Network Working Group
    Dimension Data Systems, Incorporated
    2510 Limestone Lane
    Garland, TX 75040
    (214) 495-7353

Jeffrey S. Jalbert - Historian
    J C C
    Post Office Box 381
    Granville, OH 43023
    614-587-0157

Ray Kaplan - MicroVAX Working Group
    Pivotal Incorporated
    6892 East Dorado Court
    Tucson, AZ 85715

Lawrence J. Kilgallen - Newsletter Editor
    Box 81, MIT Station
    Cambridge, MA 02139-0901

Margaret Knox - Chair
    Computation Center
    University of Texas
    Austin, Texas 78712

Art McClinton - Advisor
    MITRE
    1820 Dolley Madison Boulevard
    McLean, VA 22102

Ross W. Miller - Vice Chair and Working Group Coordinator
    Online Data Processing, Inc.
    N 637 Hamilton
    Spokane, WA 99202

Eugene Pal - Multiprocessor Working Group
    US Army
    CAORA (ATOR-CAT-C)
    Fort Leavenworth, KA

Susan Rehse - System Management Working Group
    Lockheed Missiles
    3251 Hanover Street
    Palo Alto, CA 94301-1187

Bob Robbins - Advisor
    Array Computer Consultants
    5364 Woodvale Drive
    Sarasota, FL 33582

Larry Robertson - Real Time/Process Control Working Group
    Bear Computer Systems Inc.
    5651 Case Avenue
    North Hollywood, CA

David Schmidt - LUG Coordinator, Hardware Working Group
    Management Sciences Associates
    5100 Centre Avenue
    Pittsburgh, PA 15232

Al Siegel - Advisor
    Battelle Memorial Institute
    505 King Avenue
    Columbus, OH 43201-2693

D. Slater - Artificial Intelligence Working Group
    Institute for Defense Analysis
    1801 North Beavregard Street
    Alexandria, VA 22314

# INPUT/OUTPUT

A SIG Information Interchange

A form for INPUT/OUTPUT submissions is available at the back of the issue.

INPUT/OUTPUT 487

Caption:  RX02 Drive Errors

Message:  We run a VAX-11/780 shop that includes the RX02 dual diskette drive.  The drive should read eight-inch single and double density diskettes, and we use it for reading information that has been written onto diskettes by microcomputers or other main frame computers.  We have a program which uses the SYS$QIOW system service that can even read IBM diskettes in EBCDIC.  Obviously this is a great convenience.

The problem is that occasionally for no reason that I can determine the diskette controller, when attempting to initialize or mount a diskette that is identical in physical characteristics to diskettes that we have successfully read, will fail with a "FATAL CONTROLLER ERROR" message.  And that is that; we can do nothing with that particular diskette.  The problem appears to be related to software rather than hardware for the following reason:

I own a Radio-Shack Model-16 micro and purchased some diskettes for it.  If I take a new Radio-Shack diskette, fresh from the wrapping paper, it can be initialized with no problem on the VAX.  But if I first initialize it on the micro and they try to initialize or mount it on the RX02, then the fatal controller error message appears.  So the problem can't be hardware.  The problem is very irritating because sometimes a user comes with a diskette with information that he wants to read on the VAX, and when we try to read it the error message occurs.  It is hard to explain to the disgruntled user why we usually can read diskette information, but for him we can't.

Has anyone an idea what might be the origin of this strange problem?  Has anyone encountered a similar problem?

Contact:  Dr. Richard Branham
          Servicio Centralizado de Computacion
          Centro Regional de Investigaciones Cientificas y Tecnologicas
          Mendoza, Argentina

Date:     February 4, 1986

INPUT/OUTPUT 488

Caption:  Using incoming modems as outgoing modems - Reply to I/O # 477

Message:  Use SET HOST/DTE ddcu:.  This V4.x command does not require that DECnet be running on the processor.  The optional /LOG=filename qualifier logs the session to a file.  If a DF03 autodial modem is used, the SET HOST command allows the phone number to be specified via /DIAL=.

The user MUST have RW access to the modem port or the necessary privileges.  The command SET PROTECTION=(W:RW)/DEVICE ddcu: will allow users to allocate the modem for dial-out purposes.  Note that this opens security problems via password grabbers.  The use of device ACLs may be more appropriate, i.e., SET ACL/ACL=(IDENT=identifier,ACCESS=READ+WRITE)/OBJECT=DEVICE ddcu:.  If the dial-out modem port is not set at the dial-out modem's speed (i.e., if it's AUTOBAUD), then you MUST change the speed of the modem's port.  The following will suffice:

    $ ALLOCATE ddcu:
    $ SET TERMINAL /SPEED=1200 ddcu:

The port will automatically reset to the original /PERMANENT speed when it is DEALLOCATED.

Programs on the VAX SIG DECUS tapes, such as KERMIT and HOST, can be used instead of the SET HOST command.

Contact:  Mark A. Holomany
          JCPDS-International Centre for Diffraction Data
          1601 Park Lane
          Swarthmore, PA 19081
          Telephone (215) 328-9403

Date:     March 9, 1986

## INPUT/OUTPUT 489

Caption:  Real Estate Software?

Message:  We are a large real estate company in Hawaii, running
          two 750 systems.  We are looking for real estate
          oriented accounting/sales-tracking software.  If
          anyone knows of programs of this type for the VAX, I
          would appreciate any information you can provide.

Contact:  Jack Willey
          1339 Hunakai Street
          Honolulu, Hawaii 96816
          Telephone (808) 735-4200

Date:     March 13, 1986

## INPUT/OUTPUT 490

Caption:  BUSUSE modified for Unibus

Message:  Barbara Dow-Pleines offered source for the above
          (Pageswapper February 1985).  We've been unable to get
          an answer at the given phone.  We're interested in
          getting the Unibus mods.

Contact:  June Templin
          Goshen College
          1700 South Main
          Goshen, IN 46526
          Telephone (219) 533-3161 Ext. 551

Date:     March 13, 1986

## INPUT/OUTPUT 491

Caption:  I/O redirection and STR$MATCH_WILD

Message:  I too thought * wasn't working when I first used
          STR$MATCH_WILD.  My problem, maybe yours, was due to
          gratuitous trailing spaces in FORTRAN's fixed-length
          CHARACTER variables, e.g. if PATT is CHARACTER*4, then
          PATT='*' actually sets PATT to '*bbb' (where b
          signifies a space), which only matches 'ABCbbb', NOT
          'ABC'.  Use STR$TRIM on the pattern string to get its
          real length, as:
                  PATT='*'
                  CALL STR$TRIM(PATT,PATT,LP) ! LP = trimmed
                  length

                  IF (STR$MATCH(CAND,PATT(1:LP)) ...

Contact:  C J Doran
          Sira Limited
          Souht Hill, Chislehurst, Kent, BR7 5EH, England
          Telephone +44 1 467 2636 x 325

Date:     March 11, 1986
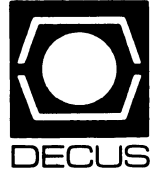
# LIBRARY
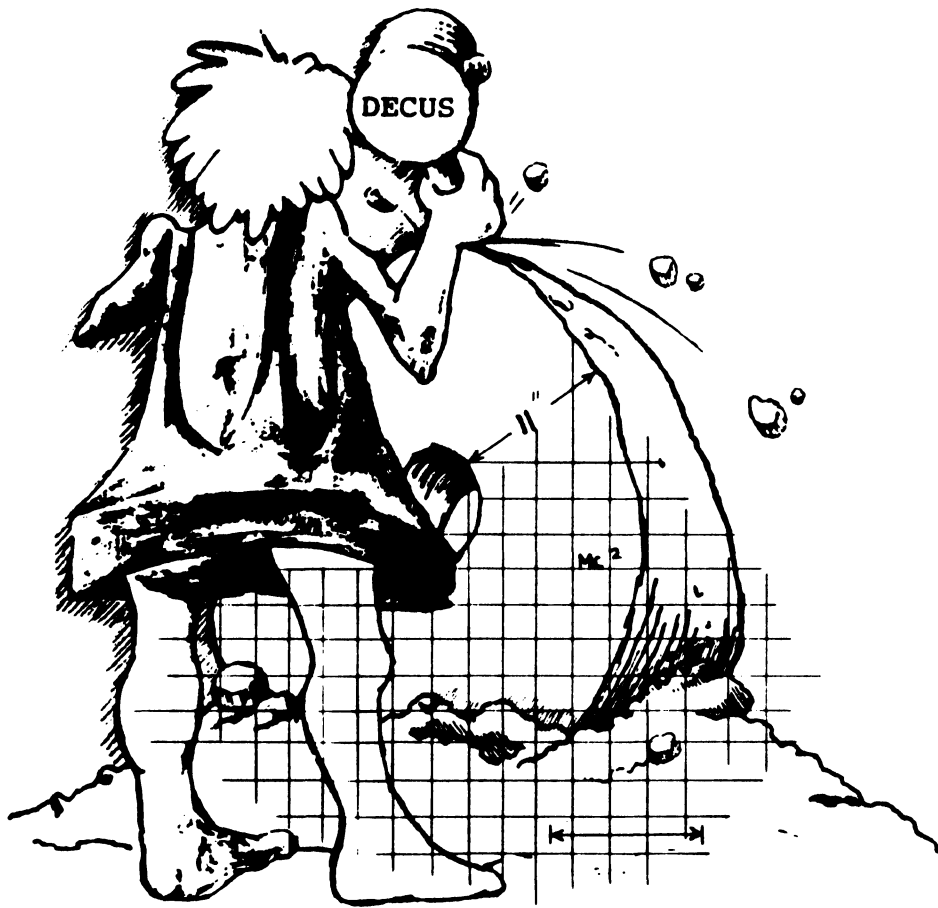
# DECUS PROGRAM LIBRARY

## NEW LIBRARY PROGRAMS AVAILABLE FOR THE PDP-11 COMPUTER FAMILY

DECUS NO: 11-822, Title: VT-200 SET UP, Version: V1, September 1985

Submitted by: Mark Northrup, Operating System: RSX-11M-PLUS V2.1E Source Language: MACRO-11, Memory Required: 11264 Bytes Software Required: TT Driver, FILES-11, Special Hardware Required: VT-200 Terminal, Disk or File System

Abstract: This module was designed to down load pre-set commands to the VT-200 terminal.

The first position in the file (VT200SETU.PRM) must be either a semi-colon or the first digit of the function key number as defined in the "VT-200 Programmers Pocket Guide", page 39, and page 83 of the "VT-200 Programmer Reference Manual". If the first character is not a semi-colon, the first three characters must have the form "nn/" where nn= the function key value, i.e. Function key 6 has a value of 17. After the "/", the command line that is to be displayed when that key is pressed, this program translates the string into the hex pairs required by the VT-200. A tilde will be translated into a carriage return, all other symbols will be translated directly into their hex pair equivalent (including semi-colons that are not in position 1). To use the defined keys, press the shift key and the function key simultaneously and the value of that key will be displayed. If a carriage return (indicated by a "tilde" in the file VT200 SETU.PRM) was placed in the string, it will execute.

Restrictions: Only works on VT-200

Documentation on magnetic media.

Media (Service Charge Code): One RX01 Diskette (KA), Format: FILES-11, 600' Magnetic Tape (MA), Format: FILES-11

DECUS NO: 11-823, Title: Task to Task Communications, Version: V1.01, November 1985

Submitted by: Eddy Fey, Utah Power & Light, Salt Lake City UT, Operating System: RSX-11M-PLUS V2.1, Source Language: MACRO-11, FORTRAN 77

Abstract: This package contains a task-to-task communications and flying install subsystem. Some of the individual programs in the package are:

RUNNER and INSREM - Set of programs used to send

parameters to a task, install the task, activate it and remove it upon exit. This is a complete 'flying install' subsystem and will work for permanently installed tasks as well (see RUNNER. DOC).

. VXDRV and VXUTIL — This is yet another re-write of the VSDRV by Osudar. It provides multiple units each capable of having 16kb of pool space. This will only work on RSX11M+ with I/D.

. CMB — Compare binary. Compares two fixed length binary files for equality. Can be used for task images, data files, etc.

. WATCHER — Nifty patch for finding executive space memory clobbers.

. IODUMP — Dumps buffers in hex-ascii byte or word format.

. CDUMP — Searches a binary CDA file for a given pattern.

Documentation on magnetic media.

Media (Service Charge Code): 600' Magnetic Tape (MA), Format: FILES-11

## NEW LIBRARY PROGRAMS AVAILABLE FOR THE VAX/VMS FAMILY OF COMPUTERS

DECUS NO: V-SP-49, Title: Symposium Tape from VAX SIG, Fall 1985, Anaheim Version: Fall 1985

Author: Various, Submitted by: J. L. Bingham, Mantech Services Corporation, Alexandria VA, Operating System: VAX/VMS V3.X or V4.X, Source Language: FORTH, STOIC, PASCAL, MACRO-32, VAX-11, FORTRAN, DCL, VAX-11 COBOL, C, VAX-11 BASIC

Abstract: These tapes consist of the VAX submissions to the Tapecopy project at the Fall 1985 DECUS Symposium in Anaheim, CA. As usual, there is a large quantity of valuable material. A very brief description follows. For a more complete description of the contents of each submission, see the AAAREADME.TXT files in each submission or the concatenated version of the AAAREADME's in [VAX000]; for documentation check for pointers in the AAAREADME.TXT's, for files containing the string READ and for files with .DOC, .TXT, .MEM,.RNO and .1ST extensions in the submissions.

SUBMISSION OVERVIEWS FOR [VAX85C...]

| | |
|---|---|
| [.AMS] | DEPROC - Header for formatting DECUS proceedings with TeX. |
| [.BATTELLE] | FILES - find files by size or owner. ZDEC -Zero Device Error Count for VMS V4. BYDISK - do a DCL pro for all disks on a system. KRON - do things at scheduled times. PERMIT - easy ACL editing. PTY - Pseudo terminal driver and session logger. |
| [.BELONIS] | HOST, MODEM7 and XMODEM - micro compatible file transfer utilities. ACCOUNT - V4.x accounting system. QPOST - Talaris printer support for TEX. MOVE - another SET DEFAULT program. SIX - simple extractor showing selected lines from files. TERMS - LOGIN.COM compiler. VVCODE - translates VMS files/headers to/from printable text for comm. transmission. |
| [.BONNER] | Bonner Lab RUNOFF (in NATIVE MODE!) large superset of DSR sith MANY extensions. |
| [.BRIDGE] | DOCUMENT - extract material from src files with delimiters. WHO - who's on the system. LIMS - programs and report writers. |
| [.BULLETIN] | VMS Bulletin Board. |
| [.COSTELLO] | TPC format independent tape-disk-tape copy routine in native mode. FAT - Pixel doubler for dumping ReGIS images on LN03. |
| [.CWAX] | PVT - print VT1xx or VT2xx terminal screen images on printer. |
| [.DFWLUG] | ALLIEDELEC - SMG utility library. WHERE - show where one is. Utility to get current system uptime into a symbol. Simple snapshot to see how much free space there is on all disks, plus SNAPSHOT for V4. |
| [.DMREV] | DM - Directory Management for VMS V4. |
| [.ERI] | HALFTONE - convert gray scales for print on LA50. MACSNVAX - Macintosh file transfer utility and several Macintosh applications. |
| [.EVEPLUS] | Extensions to EVE interface of TPU. |
| [.EYE] | DISKMON - watches disk usage, warns of impending running out of room. |
| [.FERMILAB] | ALLOCWATCH - watches allocation of devices so other cluster members can wait for them. EDTX - expanded EDT with file mem, wildcard files, spawn, etc. EXPAND - allow conditionalizable FORTRAN programs. LA100SMB - flag pages at dense modes for LA100. NODEIDS - get node names as identifiers for ACLs to use. SETUSER - eliminate need to explicitly provide UIC associated with a username. STARTUP - system startup command file examples. TECO macros for mass changes in UAF records. TELLSELF - allow detached or batch jobs to broadcast messages to parent process. |
| [.GARMAN] | DFRAG - Disk fragmentation reporter. |
| [.INQUIRE] | INQUIRE and PROTO - DTR update and program generators. |
| [.LEVINE] | INDEX - super FORTRAN indexer, cross referencer, and static analysis tool. Also some VT200 utilities including FONT to create/edit VT200 fonts. Directory tree painter, disk fragmentation utilities, inactive terminal process killer. |
| [.LJK] | Pageswapper articles since Spring 1985 tape. |
| [.MANTECH] | OBSERVE - allows you to watch another (TT or TX) terminal on your machine. No extra hardware required. DEFAULT - set default program. |
| [.NSWC] | SCHEDULER - run processes on future dates based on logic. REMINDER - improved calendar and tickler. Fixed for better cluster work. SUBMIT_IF - conditional submission. Notify message sender. |
| [.PFILE] | Kernel mode no delete protection for files. Good for synonym files and directories. [.RAINIER] Ada tools and support environment. |
| [.RAWIO] | Unix style raw and cbreak i/o. |
| [.SAOSTOIC] | STOIC - stack oriented language like FORTH. RED - screen editor/WP. CALC - RPN calculator. Updates for VMS V4. |
| [.SKUNK] | SETDEF program. EDT with spawn. SEND messages to users. |
| [.SMGLIB] | SMG screen library, (may be same as DFWLUG one). |
| [.TIMELINE] | A number of VMS utilities, notably VERB to extract verb definitions from DCLTABLES. |
| [.TPUEDT] | Documentation on TPU EDT simulator and customizer. |

| | |
|---|---|
| [.TSUME] | C program to solve mating problems in Japanese chess. |
| [.UAB] | List- TPU template for file lister. GRADE- class grader system. SMAUG- CPU hog priority adjuster. |
| [.UWRF] | ACCESS- system for easy add/delete user account information. CALEN-DAR- print calendar of any month with text in blocks. EBS- Emergency broadcast utility (also with /SELF qualifier). FRAGMENTATION - show disk fragmentation. LISTER - source lister with titles on each page. PEN- Pascal Environment. PRIV- get privs in a subprocess if parent is authorized but not enabled. PROSE- Runoff-like processor. QUOTA - gets acct quotas in V3 format (for VMS V4). RESERVE- terminal reservation system. SCRUNCH- remove comments from DCL SETFEEDER - set up Diablo 1630 sheet feeder. SGO - GO game. VAX users's guide. |
| [.VFE] | Block oriented, type insensitive file editor. |
| [.WATCHDOG] | V4 update of WATCHDOG, idle process killer. |

SUBMISSION OVERVIEWS FOR [.VAX85 D...]

| | |
|---|---|
| [.BNELSON] | Kermit-11 (and .EXE for VMS Kermit) communication program. VT100 emulating IBM PC Kermit. TED - good full screen editor/WP for VAX, PDP11, Micros. |
| [.CI] | DROIDS game and SYSTATUS4 system status. |
| [.DENNISON] | Grammar correction system and spelling checker for VMS V4. |
| [.FORTH] | FIG Forth for VMS (native mode). |
| [.HALL] | Game manager (and large collection of games). GETWS- show work set stats. NEWS program. REMINDER update SB - limited login time enforcer. |
| [.KMSKIT] | VAX Professional Workstation (i.e., most of all of ALL-in-1 free!) with various additions since Spring '85. Many other utilities included (including NOTEBOOK, which is handy for doing short procedures that DTR startup overhead makes impractical with DTR). Complete window graphics subsystem for VMS. |
| [.RCAF85] | ANALYTICALC - spreadsheet/database system for VAX and PDP11. DTCVAX - Desk Top Calendar update Garman that works from 1 AD to 9999 AD. VPW mods (as illustrations) showing how VPW from this tape can be customized for YOUR site's software. |
| [.UALR] | BBS - VAX Bulletin Board system. POSTERS - biggest collections of posters ever. Menu driven operator system. |
| [.VASSAR] | Full screen spelling corrector update (uses SMG now). BB- VMS Bulletin Board system. RESERVE -terminal reservation system. |

No guarantees are made as to the completeness, usability or quality of the programs on the tape and the material has not been checked or reviewed.

Note: Many of the submissions were submitted with VMS version 4 filenames. These files were renamed to names compatible with VMS version 3 and command procedures were constructed to restore the version 4 filenames. Therefore, version 3 sites will have no trouble loading the tape but some filenames will be inconsistent with documentation and command procedures. Version 4 sites will want to follow the instructions in [VAX000] LOADING.TXT to rename the affected files back to their original names.

This tape does not contain the [VAX000.INDEX] directory. We expect to provide index files for this tape and a composite index file with the Spring 1986 tape.

**Complete sources not included. Documentation on magnetic media.**

**Media (Service Charge Code):** 2400' Magnetic Tape (PS), Format: VMS/BACKUP

DECUS NO: VAX-148, Title: DELTREE. Version: V1.2, September 1985

**Submitted by:** Eyal Bartfeld, Hebrew University Medical School, Jerusalem 91010 Israel, **Operating System:** VAX/VMS V4.1, **Source Language:** C

Abstract: DELTREE is a program that makes life easier when one wants to delete an entire directory tree. Today, VMS responds to a command like: $DELETE USER$DISK:[JONES..] *.*;* by many warning messages when attempting to delete not-empty directory files. In order to accomplish the deletion of a directory tree, one has to repeat the above command line a few times. Each iteration deletes a higher level of the directory tree, until no more files are left. The number of iterations depends on the directory tree depth and the number of warning messages depends on the amount of 'branching' in the directory tree.
    DELTREE overcomes this problem by marking files for deletion in a recursive manner. This method ensures that all files in a directory are deleted before the directory file itself is deleted, so one can delete directory trees in a clean and elegant manner.

Options are:
. Delete an entire directory tree, starting from the root.

. Delete only files, and leave directory files, (makes a 'skeleton' of this directory tree).

. List files to be deleted, (presumably by another Deltree operation).

More features of DELTREE:
. In case of incorrect command line, a help text is typed.

. Deletion can be controlled and tuned by protecting files against deletion. The program notifies the fact that certain files were not deleted, and continues.

. Final report is typed on end.

. Easy installation.

**Documentation on magnetic media.**

**Media (Service Charge Code):** 600' Magnetic Tape (MA), Format: VMS/BACKUP

DECUS NO: VAX-161, Title: IOU-HELP, Version: V1.0, November 1985

**Submitted by:** Mark Moore, University of Texas, Sanng System: VAX/VMS V4.1, **Source Language:** DCL, VAX-11 BASIC **Special Hardware Required:** VT-100

Abstract: Information for Online Users, commonly referred to as IOU-HELP, is a set of DCL command procedures that allow easy retrieval of online documentation. The system is menu driven and users have the option of viewing or printing the document. IOU-HELP allows access by multiple users, maintains statistics of usage, and allows a user to enter his comments at the end of each session. This system was designed to be used primarily by novice computer users, but can be a useful tool for anyone wishing to make online documentation available to a large group of users. The system is in the form of a tree structure of directories. Documents are grouped together by some common denominator (subject, machine, etc.) and are stored in a common directory. If a new document is to be added, it is simply placed in the appropriate directory and will automatically appear on the menu. The main categories are hard coded in the program but can be easily changed to meet the needs of the individual site.
    This system was designed to work under a captive account. All of the installation procedures included on this tape assume the tape will be loaded into the users root directory.

**Documentation on magnetic media.**

**Media (Service Charge Code):** 600' Magnetic Tape (MA), Format: VMS/BACKUP

## NEW LIBRARY PROGRAMS AVAILABLE FOR THE DECSYSTEM-20 FAMILY OF COMPUTERS

DECUS NO: 20-184, Title: 2022, Version: V117.A(20), January 1986

**Submitted by:** David L. Wodelet, Strathcona County, Sherwood Park Alberta, Canada T8A 3W7, **Operating System:** TOPS-20 release 5.1, **Source Language:** MACRO-10, Software Required: System 1022 from Software House

Abstract: 2022 is a TOPS-20 front-end command parser for the System 1022 data base management system from Software House. Through use of TOPS-20 COMND% jsys, 2022 provides escape recognition and the "?" help feature for nearly all 1022 commands. The only commands NOT implemented in 2022 are those commands specific to TOPS-10 or those used exclusively within PL1022 or report programs.

Note: Sources for Software House System 1022 are unavailable.

**Complete sources not included. Documentation on magnetic media.**

**Media (Service Charge Code):** 600' Magnetic Tape (MA)

### DECUS Program Library CHANGES:

. DECUS Order Number: VAX-70, SLY: A Program for Changing Process' Name and UIC, is not compatible with Version 4.0 or later.
. DECUS Order Number: RB-101, DTC/PC: Desk to Calendar for MSDOS on the Rainbow, needs MS FORTRAN 77 compiler. Program needs to be recompiled because of a bug.
. NOTE: Micro-11 software will no longer be identified by an "M" number. Please make the following changes:
   - DECUS Order Number: M11-101, CGL to ReGIS VT240 Converter, should now read, DECUS Order Number: 11-828.
   - DECUS Order Number: M11-103, Kermit-11 for P/OS and Micro-RSX, should now read, DECUS Order Number: 11-829.
   - DECUS Order Number: M11-104, Kermit-11 for Micro-RSTS/E and RT-11, should now read, DECUS Order Number: 11-830.
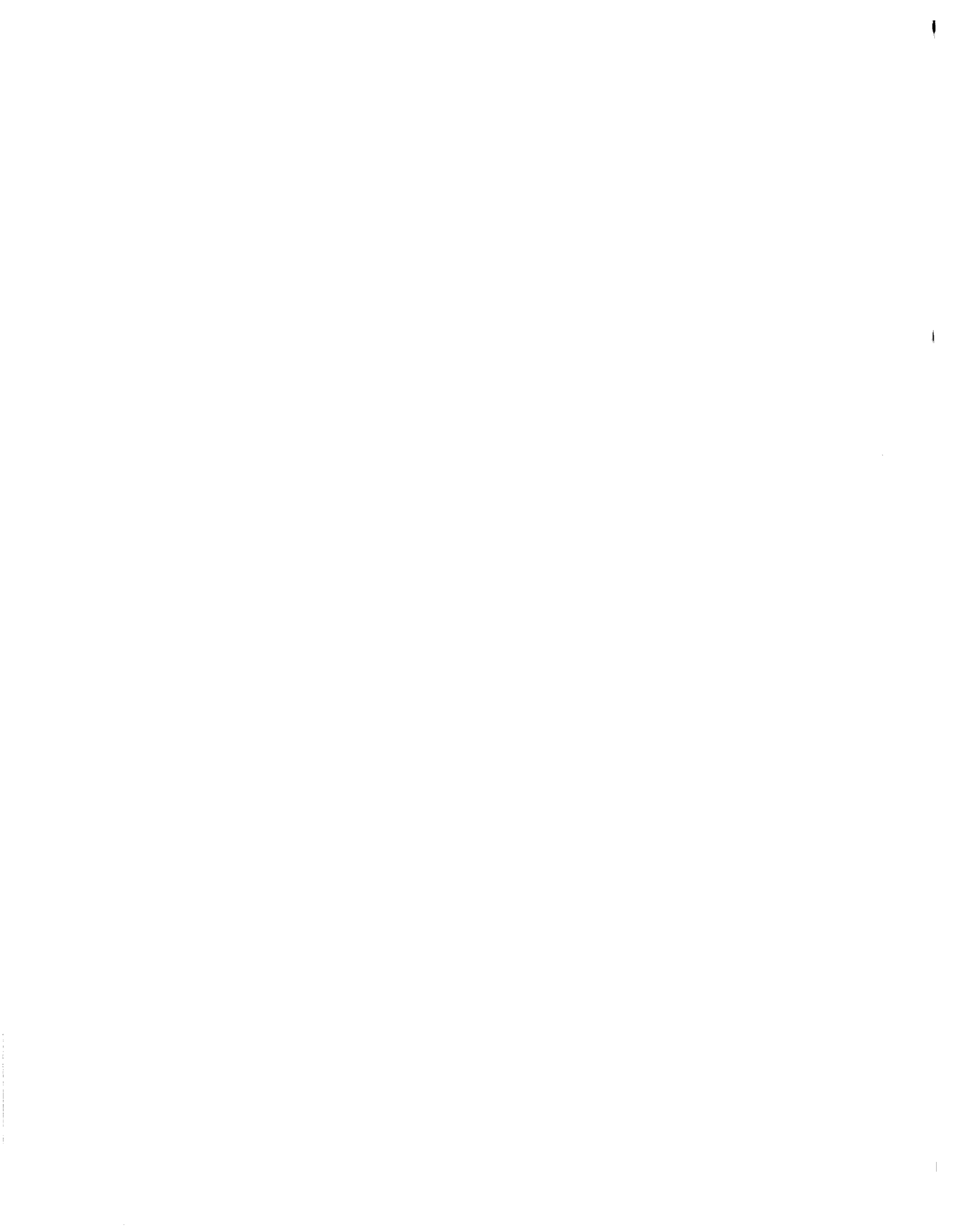
## HOW TO SUBMIT TO A SPECIFIC SECTION OF THE NEWSLETTER

The following is a listing of the Newsletter Editors with their addresses and phone numbers. All submissions to the newsletter should be submitted directly to the appropriate Editor.

ARTIFICIAL INTELLIGENCE
Terry Shannon
160 State Street
Boston, MA 02109
(617) 367-7190

BUSINESS APPLICATIONS
Thomas Byrne
L. Karp & Sons
1301 Estes
Elk Grove, IL 60007
(312) 593-5705

DATA MANAGEMENT SYSTEMS
Russ Poisson
Seed Software Corp.
2121 Eisenhower Avenue
Alexandria, VA 22314
(703) 783-4944

DAARC
Ellen Reilly
William H. Rorer
500 Virginia Drive
Ft. Washington, PA 19034
(215) 628-6547

GRAPHICS APPLICATION
Michael Anton
P.O. Box 591293
Houston, TX 77259-1293
(713) 928-4838

IAS
John Ross Roman
McDonnell Douglas
600 McDonnell Blvd.
Hazelwood, MO 63042
(314) 234-0984

LARGE SYSTEMS
Michael Joy
1st Church of Christ
Scientist
Boston, MA 02115
(617) 262-2300 x 3903

NETWORKS
Vicki Hancock
2510 Limestone Lane
Garland, TX 75040
(214) 495-7353

PERSONAL COMPUTER
Caroline Mack
9007 Mears Street
Fairfax, VA 22031
(703) 280-4404
[Upload submissions to
Wash-A-Rug Fido
(703) 359-6549]

RSX
Dominic DiNollo
Loral Electronics
Engineering Computer Center
Ridge Hill
Yonkers, NY 10710
(914) 968-2500 x2210

SITE MANAGEMENT & TRAINING
Gregory Brooks
Washington University
Behavior Research Lab.
1420 Gratton St.
St. Louis, MO 63104
(314) 241-7600 x 257

VAX SYSTEMS
Larry Kilgallen
c/o DECUS Office
219 Boston Post Road. (BP02)
Marlboro, MA 01752

APL
Doug Bohrer
Bohrer & Company
903 Ridge Road, Suite 3
Wilmette, IL 60091
(312) 251-9449

COMMERCIAL LANGUAGES
Ted Bear
RAMTEK
2211 Lawson Lane
Santa Clara, CA 95950
(408) 988-2211

DATATRIEVE
Donald E. Stern, Jr. c/o
Warner Lambert Company
10 Webster Road
Milford, Ct. 06460
(203) 783-0238

EDUSIG
Fred Bell
Taft College
29 Emmons Park Drive
P.O. Box 1437
Taft, CA 93268
(805) 763-4282

HMS
William Walker
Monsanto Research Corp.
P.O. Box 32 A-152
Miamisburg, OH 45342
(513) 865-3557

LANGUAGES & TOOLS
Alan Folsom Jr.
Fischer & Porter Company
E. County Line Road
Warminster, PA 18974
(215) 674-7154

MUMPS
Janet Berryman
2405 N. Bush
Santa Ana, CA 92706
(714) 953-1025

OFFICE AUTOMATION
Margaret Drake
Univ. of TX Health Science Ctr.
7703 Floyd Curl Drive
San Antonio, TX 78284
(512) 691-6105

RSTS
Bill Hobbs
ComManD. Inc.
6535 E. 82nd St., Suite 102
Indianapolis, IN 46250
(317) 842-5320

RT
Bill Leroy
The Software House, Inc.
2964 Peachtree RDNW #320
P.O. Box 52661
Atlanta, GA 30355
(404) 231-1484

UNISIG
William Toth
Harvard-Smithsonian Ctr. for
Astrophysics
60 Garden Street P353
Cambridge, MA 02138
(617) 495-7181

Bruce Bergman
UserWare International
2235 Meyer Avenue
Escondido, CA 92025-1070
(619) 741-8825

HOW

---
## SUBMITTING ARTICLES TO THE HMS SIG NEWSLETTER
---

The purpose of the HMS SIG Newsletter is to serve as a forum
to share information related to DEC hardware with the
members of the SIG. As such, the existence of the
newsletter is entirely dependent on your contributions. If
you have an HHK item, a better or safer way to do something,
product news, a tutorial article of general interest, etc.,
we are interested in publishing it in the newsletter. It is
intended that the HMS SIG Newsletter be published at least
four times a year.

There are several ways to submit material for the
newsletter:

> o The Hardware Submission Form in the back of the
>   newsletter can be used for brief items (there is
>   not enough room if you have a lot to say).
>
> o You can send me camera-ready hard-copy (this saves
>   me a lot of typing).
>
> o I will accept submissions on floppys. I can handle
>   RX50's or 8" diskettes (either density, single or
>   double sided). I prefer RT-11 format, if possible,
>   but I can probably handle RSX or VMS stuff somehow.
>   I will return your diskette(s), of course.
>
> o Those of you that have access to DCS can send
>   things to username WALKER. I check DCS daily.
>
> o I am also on CompuServe as "Bill Walker 71066,24".

In any event, if you have anything to submit, send it! If
it is a mess, but I can read it, I will get it in the
newsletter somehow. Finally, if you have any question about
submitting material, call me. My telephone number is listed
below.

Contributions can be sent to:

```
    HMS Editor               William K. Walker
    DECUS            OR      Monsanto Research Corp.
    BP02             ==      P.O. Box 32   A-152
    249 Northboro Road       Miamisburg, OH   45342
    Marlboro, MA 01752       (513) 865-3557
```

If you need to get something to me quickly, send it to my
work address.

## DECUS SUBSCRIPTION SERVICE
### SIGs NEWSLETTERS
### U.S. CHAPTER MEMBERS ONLY

As a member of DECUS U.S. Chapter, you are entitled to contribute and subscribe to the DECUS monthly publication, **SIGs Newsletters.** You also have the opportunity to subscribe to the Symposia Proceedings which are a compilation of the reports from various speakers at the U.S. National DECUS Symposia.

- **No Purchase Orders will be accepted.**
- The order form below must be used as an invoice.
- All checks must be made payable to DECUS.
- All orders MUST be paid in full.
- No refunds will be made.
- The address provided below will be used for all DECUS mailings; i.e. Membership, Subscription Service and Symposia.
- SIGs Newsletters Price is for a one-year subscription beginning the month following receipt of payment.

Name _____ DECUS Member No. _____
Company _____
Address _____

City _____ State _____ Zip _____
Phone _____

| Subscription Service Offering | Qty. | Unit Price | Total |
|---|---|---|---|
| SIGs Newsletters | _____ | $35.00 | _____ |
| Fall '85 Proceedings (FA5) | _____ | 15.00 | _____ |
| Spring '86 Proceedings (SP6) | _____ | 15.00 | _____ |
| Fall '86 Proceedings (FA6) | _____ | 15.00 | _____ |
| Spring '87 Proceedings (SP7) | _____ | 15.00 | _____ |

**TOTAL COST OF SUBSCRIPTION**                            $ _____

☐ MASTERCARD ☐ VISA ☐ DINERS CLUB/CARTE BLANCHE®

_____ Exp. Date _____

I understand that there will be no refunds even if I decide to cancel my subscription.

Signature: _____

| FOR DIGITAL EMPLOYEES ONLY | FOR DECUS OFFICE ONLY |
|---|---|
| Badge No. _____ CC: _____ | Check No. _____ |
| CC Mgr. Name _____ | Bank No. _____ |
| CC Mgr. Signature _____ | Amount $ _____ |

Subscription Service, DECUS (BP02), 219 Boston Post Road, Marlboro, MA 01752-1850, (617) 480-3418.

# DECUS U.S.CHAPTER
# APPLICATION FOR MEMBERSHIP

☐ New Membership  ☐ Update to current membership profile  Current DECUS Member. # ___ ___ ___ ___ ___ ___

*NOTE: PLEASE PRINT CLEARLY OR TYPE!*
**PLEASE PROVIDE A COMPLETE MAILING ADDRESS, INCLUDE ZIP CODE IN ACCORDANCE WITH POSTAL REGULATIONS FOR YOUR LOCALITY.**

**ARE YOU AN EMPLOYEE OF DIGITAL EQUIPMENT CORPORATION?** ☐ YES  ☐ NO

Name: _____
     (first)          (Middle Intial)       (Last/Family Name)

Company: _____

Address: _____

_____

_____

City/Town: _____ State: _____ Zip: _____

_____

Telephone:  Home ( ___ ) _____  Work ( ___ ) _____

**HOW DID YOU LEARN ABOUT DECUS?  Please check applicable item.**

| | | |
|---|---|---|
| 1 ☐ ANOTHER DECUS MEMBER | 4 ☐ DIGITAL SALES | 13 ☐ LOCAL USER GROUP |
| 2 ☐ SYMPOSIA | 5 ☐ HARDWARE PACKAGE | 14 ☐ SPECIAL INTEREST GROUP |
| 8 ☐ DECUS CHAPTER OFFICE | 6 ☐ SOFTWARE PACKAGE | 7 ☐ SOFTWARE DESPATCH |
| 10 ☐ DIGITAL STORE | 12 ☐ ADVERTISING | (DIGITAL Newsletter) |

**DO YOU WISH TO BE INCLUDED IN MAILINGS CONDUCTED BY DIGITAL** (for Marketing purposes etc.?)  ☐ Permission
    ☐ Refusal

**TYPE OF DIGITAL HARDWARE USED:** Please check those applicable to you.

| | | | |
|---|---|---|---|
| 20 ☐ DECMATE | 52 ☐ LSI-11 | 21 ☐ PROFESSIONAL | 5 ☐ WPS-8 |
| 82 ☐ DECsystem-10 | 3 ☐ PDP-8 FAMILY | 22 ☐ RAINBOW | 51 ☐ WPS-11 |
| 83 ☐ DECSYSTEM-20 | 50 ☐ PDP-11 FAMILY | 54 ☐ VAX FAMILY | |

**MAJOR OPERATING SYSTEMS? LANGUAGES USED:** Please check those applicable to you

| | | | | |
|---|---|---|---|---|
| 1 ☐ ADA | 26 ☐ CORAL-66 | 47 ☐ FOCAL | 67 ☐ OS/8 | 109 ☐ RT-11 |
| 2 ☐ ALGOL | 28 ☐ COS | 48 ☐ FORTRAN | 68 ☐ PASCAL | 97 ☐ TECO |
| 5 ☐ APL | 34 ☐ DATATRIEVE | 51 ☐ GAMMA | 72 ☐ PL-11 | 70 ☐ TOPS-10 |
| 7 ☐ BASIC | 35 ☐ DBMS | 110 ☐ IAS | 92 ☐ RPG | 71 ☐ TOPS-20 |
| 17 ☐ BLISS | 38 ☐ DECnet | 53 ☐ IQL | 81 ☐ RSTS/E | 104 ☐ VMS |
| 19 ☐ C | 43 ☐ DIBOL | 58 ☐ MACRO | 83 ☐ RSX | 107 ☐ WPS-8 |
| 22 ☐ COBOL | 45 ☐ DOS-11 | 65 ☐ MUMPS | 91 ☐ RMS | |

## TYPE OF BUSINESS (ENVIRONMENT)/COMPUTER APPLICATIONS
Please check that which best describes your business/application

| | | |
|---|---|---|
| 21 ☐ ACCOUNTANCY | 1 ☐ EDUCATION/PRIMARY | 73 ☐ NUMERICAL CONTROL |
| 7 ☐ BANK | 2 ☐ EDUCATION/SECONDARY | 68 ☐ OEM-COMMERCIAL |
| 64 ☐ BUSINESS/COMMERCIAL | 61 ☐ EDUCATION-TECHNOLOGY | 78 ☐ OEM-TECHNICAL |
| 74 ☐ BUSINESS/INFORMATION SYSTEMS | 3 ☐ EDUCATION/UNIVERSITY | 56 ☐ PHYSICAL SCIENCES |
| 57 ☐ CHEMISTRY | 67 ☐ ENGINEERING | 20 ☐ RESEARCH/DEVELOPMENT |
| 54 ☐ CLINICAL LABORATORY | 65 ☐ FINANCE/ACCOUNTING | 10 ☐ RETAIL |
| 63 ☐ COMPUTATION | 77 ☐ GOVERNMENT | 76 ☐ SOFTWARE DEVELOPMENT |
| 11 ☐ CONSUMER ELECTRONICS | 75 ☐ GRAPHICS | 53 ☐ TELECOMMUNICATIONS |
| 18 ☐ CONSULTANT | 4 ☐ HOSPITAL | 19 ☐ TELEPHONE/UTILITIES |
| 72 ☐ DATA ACQUISITION | 62 ☐ INDUSTRIAL | 51 ☐ TIMESHARING |
| 52 ☐ DATA COMMUNICATIONS | 55 ☐ LABORATORY/SCIENTIFIC | 80 ☐ TRAINING/INSTRUCTION |
| 13 ☐ DATA PROCESSING SERVICES | 14 ☐ LIBRARY | 66 ☐ TYPESETTING/PUBLICATION |
| 71 ☐ DATA REDUCTION | 58 ☐ LIFE SCIENCES | |
| 17 ☐ DIGITAL EMPLOYEE-ENGINEERING | 70 ☐ MANUFACTURING | |
| 15 ☐ DIGITAL EMPLOYEE-MARKETING | 79 ☐ MARKETING | |
| 16 ☐ DIGITAL EMPLOYEE-SERVICE GROUP | 59 ☐ MEDICAL RESEARCH | |
| 60 ☐ EDUCATIONAL ADMINISTRATION | 6 ☐ MILITARY INSTALLATION | |

## SPECIAL INTEREST GROUP (SIGs) ENROLLMENT
I wish to participate in the following DECUS U.S. Chapter Special Interest Groups.

| | | |
|---|---|---|
| 33 ☐ APL SIG | 11 ☐ HARDWARE AND MICRO | 36 ☐ PERSONAL COMPUTER |
| 2 ☐ COMMERCIAL | 35 ☐ IAS | 18 ☐ RSTS/E |
| LANGUAGES | 31 ☐ DAARC(LABS) | 17 ☐ RSX |
| 6 ☐ DATA MGMT.SYS. | 27 ☐ LARGE SYSTEMS | 19 ☐ RT-11 |
| 5 ☐ DATATRIEVE | 16 ☐ LANG. AND TOOLS | 32 ☐ SITE MGMT.& TRNG |
| 7 ☐ BUSINESS APPL. | 14 ☐ MUMPS | 21 ☐ UNISIG |
| 8 ☐ EDUSIG | 15 ☐ NETWORKS | 26 ☐ VAX SYSTEMS |
| 10 ☐ GRAPHICS APPL | 34 ☐ OFFICE AUTOMATION | |

## JOB TITLE/POSITION - Please check:

| | |
|---|---|
| 1 ☐ CORPORATE STAFF | 101 ☐ CORPORATE DIRECTOR OF DP/MIS |
| 2 ☐ DIVISION OR DEPARTMENT STAFF | 102 ☐ ADMINISTRATIVE ASSISTANT |
| 3 ☐ SYSTEMS ANALYSIS | 103 ☐ TECHNICAL ASSISTANT |
| 4 ☐ APPLICATIONS PROGRAMMING | 104 ☐ SERVICES COORDINATOR |
| 5 ☐ SYSTEMS ANALYSIS/PROGRAMMING | 105 ☐ MANAGER |
| 6 ☐ OPERATING SYSTEM PROGRAMMING | 106 ☐ ANALYST |
| 7 ☐ DATABASE ADMINISTRATION | 107 ☐ PROGRAMMER |
| 8 ☐ DATA COMMUNICATIONS/TELECOMMUNICATIONS | 108 ☐ DATABASE MANAGER |
| 9 ☐ COMPUTER OPERATIONS | 109 ☐ DATABASE ADMINISTRATOR |
| 10 ☐ PRODUCTION CONTROL | 110 ☐ MANAGER OF DP OPERATIONS |

CITIZEN OF UNITED STATES OF AMERICA?  ☐ Yes   ☐ No   Country:_____

Signature: _____ Date: _____

**Forward To:**

DECUS U.S. CHAPTER, MEMBERSHIP PROCESSING GROUP
219 BOSTON POST ROAD
MARLBORO, MA 01752, USA
PHONE: (617) 480-3418

# INPUT/OUTPUT Submission Form

A SIG Information Interchange

Please reprint in the next issue of the Pageswapper

If this is a reply to a previous I/O, which number?  _____

Caption:  _____

Message:  _____

_____

_____

_____

_____

_____

_____

_____

Contact:

Name  _____

Address  _____

_____

_____

_____

Telephone  _____

Signature  _____  Date  _____

Mail this form to:  Larry Kilgallen, PAGESWAPPER Editor
Box 81, MIT Station, Cambridge, MA  02139-0901, USA

QU

Tear out or photocopy reverse to submit an I/O item

Larry Kilgallen, PAGESWAPPER Editor
Box 81, MIT Station
Cambridge, MA  02139-0901
USA

# System Improvement Request Submission Form

Page 1 of _____

Submittor:                    Firm:

Address:                      Phone:

How to write an SIR:
Describe the capability you would like to see available on VAX
systems.  Be as specific as possible.  Please don't assume we
know how it's done on the XYZ system.  Justify why the capability
would be useful and give an example of its use.  If you wish,
suggest a possible implementation  of your request.

Abstract (Please limit to four lines):

Description and examples (use additional pages if required)

Tear out or photocopy reverse to submit an SIR

Gary L.  Grebus
Battelle Columbus Division
Room 11-6011
505 King Avenue
Columbus, Ohio 43201-2693
USA

# Ask the WOMBAT WIZARD Submission Form

To submit a problem to the WIZARD, please fill out the form below and send it to:

> Donald E. Stern, Jr., DTR/4GL SIG Newsletter Editor
> Warner Lambert Company
> 10 Webster Road
> Milford, CT  06460

Name:_____

DECUS Membership No. _____

Affiliation:_____

Address:_____

_____

Telephone Number:  _____

Statement of Problem:

_____

_____

_____

_____

Guidelines and rules for submitting questions to the Wombat Wizard:

1. If you are trying to demonstrate a method or a concept, please simplify the procedures, records, and other information to the shortest form possible. Avoid long procedures where only a small portion of the procedure is required to demonstrate the concept.

2. Annotate your attachments. Simple comments or handwritten notes ("Everything worked until I added this statement.") go a long way toward identifying the problem.

3. Keep an exact copy of what you send. And number the pages on both copies. But send everything that is related to your question, even remotely.

4. Wombat Wizard is not the Telephone Support Center, nor is it part of DEC's Software Performance Reporting (SPR) system. Our goal is to answer "how to" or "how come" questions in an informative and instructive fashion - not to be a clearinghouse for software performance problems.

5. If you would like a direct response or would like your materials returned, please don't forget to include a stamped, self-addressed envelope large enough to hold the materials you send.

(fold here)
------------------------------------------------------------------------

                    Donald E. Stern, Jr., DTR/4GL SIG Newsletter Editor
                    Warner Lambert Company
                    10 Webster Road
                    Milford, CT   06460

------------------------------------------------------------------------
(fold here)

# OFFICE AUTOMATION SIG
# SYSTEM IMPROVEMENT REQUEST SUBMISSION FORM

**Name** _____     **Address** _____

**Firm** _____     _____

**Telephone** _____     _____

**INSTRUCTIONS:** System Improvement Requests (SIR) can be either hardware of software; please check the category addressed by this SIR. Under ABSTRACT, give a brief definition of the capability you would like. In the DESCRIPTION section, give a detailed description and examples of what you want. Be specific; don't assume that we know how other products function. Justify the usefulness of the capability and give an example of its use.

**HARDWARE IMPROVEMENT**           **SOFTWARE IMPROVEMENT**

DECmate _____     ALL-IN-1 _____     WPS _____

PRO-Series _____     CP/M (DECmate) _____     P/OS _____

Rainbow _____     CP/M (Rainbow) _____     MS-DOS _____

Other _____     Other _____

**ABSTRACT** _____

_____

_____

**DESCRIPTION** _____

_____

_____

_____

_____

_____

_____

_____

E. Catherine Ditamore
ARA Services
Corp MIS
Independence Square West
Philadelphia, Pa. 19106

# DATAGRAM

DATAGRAMs are short messages, comments, requests, or answers that are published in NETwords. Please fill in the sections below and send the DATAGRAM to:

> **Vickie Hancock**
> **NETWords Editor**
> **2510 Limestone Ln.**
> **Garland, Tx. 75040**

**Title:** _____

**Message:** _____
_____
_____
_____
_____
_____
_____
_____
_____
_____
_____
_____
_____
_____
_____
_____

**Your Name:** _____
**Address:** _____
_____
**Telephone:** _____

**If this is a reply to a previous DATAGRAM, what #?** __

**Signature:** _____ **Date:** ____

Vickie Hancock
NETWords Editor
2510 Limestone Ln.
Garland, Tx.  75040

Fold Here

# VAX Systems SIG Spring 1986 SIR Ballot

DECUS membership number _____ (six digits)

Our site uses the following VAX models (check all that apply)

8600 ____        11/782 ____        11/780,11/785 ____        11/750 ____
11/730,11/725 ____        MicroVAX ____

We use VAX's in the following applications (Check all that apply)

Business EDP ____                    Software Development ____
Education ____                       Computer Science Research ____
Data Acquisition/Control____         CAD/CAM ____
Service Bureau ____                  Hardware Development ____
Scientific/Engineering ____          Office Automation ____
Telecommunications ____
Other _____

I support the following as the most important System Improvement
Requests. (List from zero to fifteen SIR's):

SIR Number:


_____        _____        _____        _____        _____

_____        _____        _____        _____        _____

_____        _____        _____        _____        _____


I oppose the following SIR's as detrimental. (List from zero to
five SIR's):

SIR Number:


_____        _____        _____        _____        _____


Mail to:

Gary L. Grebus
Battelle Columbus Division
Room 11-6011
505 King Avenue
Columbus, OH  43201

To be counted, you ballot must be received by April 1.

Tear out or photocopy reverse to vote on SIRs

Gary L.  Grebus
Battelle Columbus Division
Room 11-6011
505 King Avenue
Columbus, Ohio 43201-2693
USA

DECUS

## STATUS CHANGE

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

(    ) Change of Address
(    ) Please Delete My Membership Re
    (I Do Not Wish To Remain A Mem

DECUS Membership No: _____

Name: _____

Company: _____

Address: _____

_____

State/Country: _____

Zip/Postal Code: _____

Affix mailing label
el is not
print old
e. Include
stallation,
university.

Mail to: DECUS - Attn: Subscription Sen
      219 Boston Post Road, BP02
      Marlboro, Massachusetts 01752-

USA