

U.S money and effort went into accomplishing this impossible feat,

1

(J26232) 4-AUG-75 16:08;;; Title: Author(s): Ann Weinberg/POOH;
Distribution: /PAW2([ACTION]) PKA([ACTION]) FFL([ACTION])
SGR([INFO-ONLY]) ; Sub-Collections: SRI=ARC; Clerk: POOH;

26232 Distribution

Priscilla A. Wold, Pamela K. Allen, Flanoid F. Lejeune, Susan Gail
Roetter,

direps

now is the time for all good men to come to the aid of their country,
with except of cmsgt harvey d guest.

1

direps

(J26233) 4=AUG=75 16:08;;; Title: Author(s): Susan Gail
Roetter/SGR; Distribution: /PAW2([ACTION]) JOC([ACTION]) PKA([ACTION]) JLC([ACTION]) FFL([ACTION]) SGR([INFO=ONLY]) ;
Sub=Collections: SRI=ARC; Clerk: SGR;

26233 Distribution

Priscilla A. Wold, Jim O. Calvin, Pamela K. Allen, Johnny L.
Crabtree, Flanoid F. Lejeune, Susan Gail Roetter,

Knock on wood

When ISIC came back up I had a totally bad file!!! Me and my big
mouth.

KIRK 31-JUL-75 17:32 26234

Knock on wood

(J26234) 31-JUL-75 17:32;;; Title: Author(s): Kirk E. Kelley/KIRK;
Distribution: /JML([INFO-ONLY]) ; Sub=Collections: SRI=ARC; Clerk:
KIRK;

Directives for beginning and ending tables

I think I told you IfirstShow. It should instead be PxIfirstShow.

Directives for beginning and ending tables

In first level title of table place ",PxIFirstShow=0;".
After the last item in the table, place ",PxIFirstShow=<7;".

Directives for beginning and ending tables

(J26235) 31-JUL-75 20:04;;; Title: Author(s): Kirk E. Kelley/KIRK;
Distribution: /SGR([INFO-ONLY]) ; Sub-Collections: SRI-ARC; Clerk:
KIRK;

26235 Distribution
Susan Gail Roetter,

YBS, PxNShow needed for AFM tables

Instead of IFirstShow and SN, the following are the proper directives,

Before the table, insert

" ,PxIFirstShow=0; ",

3 After the table, insert

" ,PxIFirstShow=<7; ",

If you don't want spaces between the statements in the table, leave the YBS directive off,

Statements comprising the table must be below level 1,

The most up-to-date information concerning this kind of stuff should be in the AFMFORMAT help file <xhelp,afmformat,> at ISIC and BBNB. I am responsible for this file. Let me know if you would like to add to or modify any of the information in it,

1

2

2a

3

3a

4

5

6

KIRK 31-JUL-75 23:20 26236

YBS, PxNShow needed for ApM tables

(J26236) 31-JUL-75 23:20;;; Title: Author(s): Kirk E. Kelley/KIRK;
Distribution: /SRL([INFO-ONLY] this obsoletes Previous message about
IFirstShow) POOH([INFO-ONLY]) EKM([INFO-ONLY]) EAR([INFO-ONLY
]) EFF([INFO-ONLY]) ; Sub=Collections: SRI=ARC; Clerk: KIRK;

26236 Distribution

Susan R, Lee, Ann Weinberg, Elizabeth K, Michael, Elizabeth A,
Riddle, Elizabeth F, Finney,

LETTER: sent to Widasky of Hawaiian law firm

This letter was sent along with announcement of AKW seminar.

LETTER: sent to Widasky of Hawaiian law firm

Dr. Robert N. Lieberman
Stanford Research Institute
Augmentation Research Center
333 Ravenswood Avenue
Menlo Park, California 94025

Ms. Ethel K. Widasky
Carlsmith, Carlsmith, Wichman and Case
Attorneys at Law
P.O. Box 656
Honolulu, Hawaii 96809

dear Ms. Widasky:

Thank you for your letter of 25 July 75 inquiring about the Augmented Knowledge Workshop Seminar. I have enclosed the announcement of this week long seminar,

1

We feel that attendance could make a substantial contribution to your future plans for state-of-the-art word processing and subsequent implementation of a system,

2

The seminar will be oriented toward managers with experience in assessing current and future needs of an office. Conversations with Mr. Mark Michael indicated that you would be the ideal person to attend,

3

A substantial part of the week's time will be spent in training and actual hands-on use of our interactive system,

4

The rest of the time will be spent in discussing the potential of the system in different situations. This would require a good knowledge of how computers are/could be used for information handling in an organization,

5

Mr. Michael mentioned to us that one of the general partners will be on the mainland later this month. We would be pleased to show him the system and talk with him. Please let me know a tentative date if this is possible,

6

LETTER: sent to Widasky of Hawaiian law firm

Thank you very much.

7

Sincerely,

Robert N. Lieberman

LETTER: sent to Widasky of Hawaiian law firm

(J26237) 12=AUG-75 15:03;;; Title: Author(s): Robert N.
Lieberman/RLL; Distribution: /JCN([INFO-ONLY]) EKM([INFO-ONLY])
; Sub-Collections: SRI=ARC; Clerk: RLL; Origin: < ARC=LOG,
WIDASKY,NLS;8, >, 6=AUG-75 14:22 RLL ;;;; #####

Weekly Report

4-August-75	1
Last Week	1a
nsw protocols	1a1
-worked not at all on updating the file package document as agreed to at the june protocol meeting	1a1a
-worked on the "pseudo user telnet" program for old tool interaction	1a1b
project management	1a2
-Meeting Notes	1a2a
+Review of near term milestones	1a2a1
+Our milestone for FE-10 (old tool) is too late for COMPASS	1a2a1a
+need to debug the pseudo telnet old tool path	1a2a1b
+involves CHI JBP and Schantz	1a2a1b1
+Need to develop per person level milestones	1a2a2
+Need to develop a note on relationship between the proposal tasks and thecharge number subs	1a2a3
+Pull together the design documents and deliver to compass by 15-AUG	1a2a4
-Milestones sent to Compass	1a2b
arpa protocols	1a3
-Confered with Jim on the response to the Message Protocol	1a3a
Next Week	1b
nsw protocols	1b1
-get the "pseudo user telnet" program for old tool interaction to an operational state	1b1a
-complete updating the file package specification documents	1b1b
project management	1b2

Weekly Report

-revise and refine milestones	1b2a
-send updated milestones to Compass, re their questions (26194,)	1b2b
-collect the set of design documents, transmit the table of contents to COMPASS	1b2c
-prepare a note relating project account numbers to contract work statement tasks	1b2d
-have NLS 8,5 as standard NLS at ISIC	1b2e
arpa protocols	1b3
-read INWG notes	1b3a
-read Network Measurement notes	1b3b
vacation	1b4
9-17 August gone to yosemite	1b4a

Weekly Report

(J26238) 5=AUG=75 11:46;;; Title: Author(s): Jonathan B.
Postel/JBP; Distribution: /ARC=DEV([INFO=ONLY]) ; Sub=Collections:
SRI=ARC ARC=DEV; Clerk: JBP;

26238 distribution

Mary Ann Kellan, Andy Poggio, David L. Retz, Jan A. Cornish, Larry L. Garlick, Delorse M. Brooks, Beverly Boli, James E. (Jim) White, Ann Weinberg, Kenneth E. (Ken) Victor, Dirk H. Van Nouhuys, Jonathan B. Postel, Elizabeth K. Michael, David S. Maynard, Karolyn J. Martin, Harvey G. Lehtman, Kirk E. Kelley, Charles H. Irby, Joseph L. Ehardt, Robert Louis Belleville, Don I. Andrews, Richard W. Watson, Douglas C. Engelbart,

CONTACT: Tom Humphrey of SRI - re: EPC on 5AUG75

(EPC) Contact report 26239 1

(DATE) 5 Aug 75 1a

(BY) Lieberman 1b

(ATTENDEES) 1c

 Thomas Humphrey - SRI-ISG 1c1

 Robert Lieberman - SRI-ARC 1c2

(ADDRESSES) Full name of organization, address, and phone number 1d

(MEDIUM) FACE-TO-FACE 1e

(WHERE) SRI, Menlo Park, CA 1f

(ACTION=ITEMS) 1g

 Actions taken, to be taken, etc., dated 1g1

(DISTRIBUTION) ARC=LOG DCE JCN RLL JHB BJP 1h

(REFERENCES) 1i

(DOCUMENTS) Hard copy given and received 1j

 (GIVEN) Date and documents given 1j1

 (RECEIVED) Date and documents received 1j2

(REMARKS) 1k

 I visited Tom's office for a few minutes to ask him about the latest on the EPC proposal he is rewriting, 1k1

 Tom is writing it now and will hopefully resubmit it by next week to NSF. This time SRI will have some \$35,000 of our own money it as cost sharing. 1k2

 He indicated that TNLS would be used for the control of publication within this proposed experiment. 1k3

 However, DNLS was definitely NOT suitable for editing due to the VERY slow response a user gets. This made it totally objectionable for an operational service. 1k4

 He has several possible avenues in which to he may go. First,

CONTACT: Tom Humphrey of SRI - re: EPC on 5AUG75

he might adapt MAE to this service (unlikely), second he might develop an intelligent front end to the NLS front end so that editing can be done locally.

1k5

What is important here is the adamant attitude that DNLS is not suitable for editing. Once again we might take heed of this prevalent opinion by our users and do something about it.

1k6

RLL 5-AUG-75 16:46 26239

CONTACT: Tom Humphrey of SRI - re: EPC on 5AUG75

(J26239) 5-AUG-75 16:46;;; Title: Author(s): Robert N.
Lieberman/RLL; Distribution: /ARC=LOG([INFO-ONLY]) DCE([INFO-ONLY
]) JCN([INFO-ONLY]) RLL([INFO-ONLY]) JHB([INFO-ONLY]) BJP(
[INFO-ONLY]) ; Sub=Collections: SRI=ARC ARC=LOG; Clerk: RLL;

26239 Distribution

James C. Norton, Log Augmentation, Douglas C. Engelbart, James C.
Norton, Robert N. Lieberman, James H. Bair, Buddie J. Pine,

Listings with SIDs and statement numbers

List print and how it works 1

The user subsystem listprint is now in directory weinberg at office-1 and directory michael at isic, 2

It prints statement numbers for all top level statements and for each statement at the top of a page, 3

It prints sids for every statement, 4

It puts blank lines between statements, 5

It creates a k'printer' file in the CONNECTED directory. You may then copy this file to the line printer or to a terminal, 6

HOW IT WORKS 7

Create a file that has a list of your file names preceded by "Print" 8

 e.g, statement 1: Print <weinberg>vviich1 8a

 statement 2: Print <weinberg>vviich2 8b

 etc 8c

Execute Programs Load Program weinberg,listprint 9

Load the file that contains the list of Print commands 10

Goto Listprint 11

When the spooler is working you can get multiple copies by doing (in listprint) a Set Copies n - where n is some number > 1 12

To create the printer file: Listprint Group/Plex 13

 Then give it the sids or statement numbers of the group or plex you want printed, 13a

Do your copy net trick to the lineprinter or copy to a terminal 14

If you are doing this from a TI listprint will print the name of each file it has completed, 15

Listprint also changes your print list file. It replaces the words

Listings with SIDs and statement numbers

Print with Finished, THEREFORE BEFORE YOU RUN IT AGAIN YOU HAVE TO
SUBSTITUTE Print for Finished. 16

This is all much easier than it may sound. 17

Listings with SIDs and statement numbers

(J26240) 5-AUG-75 17:08;;; Title: Author(s): Elizabeth K.
Michael/EKM; Distribution: /POOH([ACTION]) SGR([ACTION]) JDH([
INFO-ONLY]) LAC([INFO-ONLY]) ; Sub-Collections: SRI=ARC; Clerk:
EKM; Origin: < MICHAEL, EKM,NLS;1, >, 21-JUL-75 12:24 EKM
;;;####;

26240 Distribution

Ann Weinberg, Susan Gail Roetter, J. D. Hopper, Lawrence A. Crain,

Code Shuffling in GNLS: NDDT as User Program

Don't forget: the running NLS at ISIC has the new file system supporting graphics in it. If you notice any bad files due to unexplainable circumstances, please let me know immediately!

Code Shuffling in GNLS: NDDT as User Program

I have fixed GNLS to load NDDT as a user program when control=H is armed. When the user program buffer is reset, control=H is disarmed.

1

This gives us about 10 pages of needed space. I have also moved BINTNLS from the high segment to the low segment because we were overflowing into the user profile area. We currently have about 2 pages in high segment and 128 pages in low segment.

2

These changes are not necessary for NSW. The next things to be moved if we need the space are SENDMAIL, IDENT, and maybe we'll have to make the EXEC stuff a separate subsystem.

3

We should make the code which loads these things as user programs general. (Currently the code is duplicated with minor variations in at least three places.)

4

The system should be brought up as the running system at ISIC as soon as the undefineds in help are resolved by Kirk.

5

Code Shuffling in GNLS: NDDT as User Program

(J26241) 6=AUG=75 11:49;;; Title: Author(s): Harvey G.
Lehtman/HGL; Distribution: /EKM([ACTION]) RLB2([ACTION]) DSM([ACTION]) JDH([ACTION]) KJM([ACTION]) KIRK([ACTION])
SRI=ARC([INFO-ONLY]) ; Sub=Collections: SRI=ARC; Clerk: HGL;

26241 Distribution

Elizabeth K. Michael, Jonathan B. Postel, Elizabeth J. Feinler, Kirk E. Kelley, N. Dean Meyer, James E. (Jim) White, Douglas C. Engelbart, Martin E. Hardy, J. D. Hopper, Charles H. Irby, Harvey G. Lehtman, James C. Norton, Jeffrey C. Peters, Dirk H. Van Nouhuys, Kenneth E. (Ken) Victor, Richard W. Watson, Don I. Andrews, Elizabeth K. Michael, Robert Louis Belleville, David S. Maynard, J. D. Hopper, Karolyn J. Martin, Kirk E. Kelley, Mary Ann Kellan, Buddie J. Pine, Andy Poggio, David L. Retz, Laura J. Metzger, Karolyn J. Martin, Jan A. Cornish, Larry L. Garlick, Priscilla A. Wold, Pamela K. Allen, Delorse M. Brooks, Beverly Boli, Rita Hysmith, Log Augmentation, Joseph L. Ehardt, Raymond R. Panko, Susan Gail Roetter, Robert Louis Belleville, Rene C. Ochoa, Ann Weinberg, Joan Hamilton, Adrian C. McGinnis, Robert S. Ratner, David S. Maynard, Robert N. Lieberman, Sandy L. Johnson, James H. Bair, Jeanne M. Leavitt, Rodney A. Bondurant, Jeanne M. Beck, Marcia L. Keeney

hard to fine archived items

The message generated when a journal item you are jumping to is not on line (ie "catalog item" "file not online use interrogate" should also tell what directory the file is in (eg HJOURNAL), --jon,

1

JBP 6=AUG=75 14:59 26242

hard to fine archived items

(J26242) 6=AUG=75 14:59;;; Title: Author(s): Jonathan B.
Postel/JBP; Distribution: /FEEDBACK([ACTION]); Sub=Collections:
SRI=ARC FEEDBACK; Clerk: JBP;

26242 distribution
Special Jhb Feedback,

Delete Modifications too easy

Delete modifications should requires some complicated confirmation like your password rather than a simple command accept, --jon,

JBP 6=AUG=75 15:03 26243

Delete Modifications too easy

(J26243) 6=AUG=75 15:03;;; Title: Author(s): Jonathan B.
Postel/JBP; Distribution: /FEEDBACK([ACTION]); Sub-Collections:
SRI=ARC FEEDBACK; Clerk: JBP;

26243 Distribution
Special Jhb Feedback,

CONTACT: BPA, Marge Lambie on 5 AUG 75

(BPA) Contact report 26244 1

(DATE) 5 Aug 75 1a

(BY) Lieberman 1b

(ATTENDEES) 1c

Marge Lambie - BPA 1c1

Robert Lieberman - SRI-ARC 1c2

(ADDRESSES) Full name of organization, address, and phone number 1d

Phone 503-234-3361 1d1

(MEDIUM) PHONE 1e

(WHERE) Menlo Park, CA and Portland, Oregon 1f

(ACTION=ITEMS) 1g

Actions taken, to be taken, etc., dated 1g1

(DISTRIBUTION) ARC-LOG DCE JCN RLL 1h

(REFERENCES) 1i

(DOCUMENTS) Hard copy given and received 1j

(GIVEN) Date and documents given 1j1

(RECEIVED) Date and documents received 1j2

(REMARKS) 1k

I called Marge to find out what the latest status of BPA was with regard to buying a utility slot. 1k1

Things are moving slowly up there. Additionally, it seems that they might not have as great a document load as originally thought. This would mean it would be harder to justify spending \$40,000 for our service. However, this is not final and the load might be large enough. 1k2

As far as interest in the short seminar that we are giving this August, she felt that it sounded like a good idea but the cost of travel and the cost of the seminar would probably preclude anyone from attending. 1k3

CONTACT: BPA, Marge Lambie on 5 AUG 75

She also mentioned that it would be nice to have many people on (logged on) the system for the one slot. I explained the pie slice scheduler and that this would permit more than one person logged on per slot. However, cautioned her that it would not be advisable to have more than 2 or 3 people on at one time.

1k4

CONTACT: BPA, Marge Lambie on 5 AUG 75

(J26244) 6=AUG=75 17:39;;; Title: Author(s): Robert N.
Lieberman/RLL; Distribution: /ARC-LOG([INFO-ONLY]) DCE([INFO-ONLY
']) JCN([INFO-ONLY]) RLL([INFO-ONLY]) ; Sub-Collections:
SRI=ARC ARC-LOG; Clerk: RLL;

26244 Distribution

James C. Norton, Log Augmentation, Douglas C. Engelbart, James C.
Norton, Robert N. Lieberman,

FUTURE NEEDS OF HELP

I just added this section to <xhelp, helpd,>. I think we need to get funding for these things, What do you think?

FUTURE NEEDS OF HELP

Do feature:

The "do" feature is the ability of having Help execute a command for you or a task consisting of a scenario of commands. Should user specification be necessary in the process, the do feature will tell in English sentences what is expected at every step of the way. This is an active tutorial/example/service which should make Help much more valuable as a teaching aid and provide a new service as a task doer. It would be implemented by writing command branches for each command. The branches are processed when the user selects them because of a special symbol placed by the Help description writer after the right anglebracket of a link. Uparrow or ^, indicates that Help is to process the commands in the branch addressed by the link. Backarrow or ←, indicates that Help is to process the command(s) between the link delimiters,

1

It would take approximately two calendar weeks to implement the software for this feature. The do descriptions would probably take a couple of person months to complete for all commands,

1a

Comment feature:

The comment feature allows any arbitrary amount of designated text to disappear when viewspec capital T is in effect. The text reappears when viewspec capital S is turned on. This feature is desperately needed for four separate functions,

2

Making user-invisible comments to other Help writers,

This would do away with the current percent sign convention which requires a special sequence generator or content analyzer pattern as it is not a part of the standard nls capabilities. Unfortunately, content analyzers are not additive,

2a

Making output processor directives invisible in help,

This would do away with having to have two separate directories of files. One with the OP directives and one without. And it would do away with having to delete directives and update to the second directory whenever a modification is made,

2b

Placing links invisibly next to referenced text,

This is needed to place "ugly" link syntax in a node to define how that node will be viewed and to link to references,

2c

A back-link facility where the links to a node are invisibly backlinked from a node,

The back-link feature is crucial to a workable multi-file index, generation and maintenance procedure,

2d

Backlinks: automatic link maintenance and "forward" references

It is essential that automatic link maintenance via back-links be implemented in order to reduce the tremendous overhead and inherent

FUTURE NEEDS OF HELP

mistakes in the current procedure of discovering bad links and updating them manually. This has the added benefit of allowing "forward references" which allow users to see what has been written about a topic subsequent to its publication,

3

Menu viewspec:

viewspec capital M turns on "menu" numbers (a la Help). Capital I or capital J turns Capital M off. If menu numbers are turned on via viewspecs instead of via a special sequence generator, we should be able to implement bugging and the use of other standard viewspecs in a more straight forward way. In addition, this feature would then be generally available in all tools in the core AKW,

4

Warp viewspecs:

A warp is a link that has been turned into a window to the text it addresses. This is currently implemented in help via "included" text and is always on. It cannot be turned off by the user or the help writer. The capital W viewspec would turn on a warp or "include" the text addressed by the first link of each statement if it is a valid link. Additional W viewspecs "include" text in "included" text. The capital X viewspec turns off all warps. If warps are implemented as viewspecs instead of via a special sequence generator, we should be able to implement bugging and the use of other standard viewspecs in a more straight forward way. In addition, this feature would then be generally available in all tools in the core AKW,

5

Index generation and maintenance:

automatic creation and maintenance of an alphabetic index to all named statements and perhaps all meaningful words in a multifile database. This needs the automatic link maintenance facility described in the preceding paragraph,

6

Boolean searches and other AI type input

once automatic index maintenance is implemented, we can generate efficient files for searching the index. This should allow standard Boolean arguments such as "X AND Y OR Z AND NOT a" and eventually, natural language typed in (and spoken?) english queries,

7

FUTURE NEEDS OF HELP

(J26245) 7-AUG-75 06:13;;; Title: Author(s): Kirk E. Kelley/KIRK;
Distribution: /JAC3([INFO-ONLY]) DVN([INFO-ONLY]) BEV([INFO-ONLY]) KS([INFO-ONLY]) ; Sub-Collections: SRI-ARC; Clerk:
KIRK;

DEX Modifications as per Training Staff Request

Today (7 August 1975) I brought up new versions of the DEX program <REL=NLS>DXCTL,REL; at ISIC, BBNB and OFFICE-1. The up to date source code may be found in <NLS>DXCTL,NLS; at ISIC. The following modifications were made, but should be verified by applications staff under production situations. The changes should be officially documented and released to users of DEX. Additionally, applications programmers should consider modifying the NLS code so the program loaded is obtained from <NETSYS> or <SUBSYS> rather than <REL=NLS> in all but experimental systems. The projected changes to CASSETTE have not been made, but will be completed upon Jan Kremers return next week.

1

Changes to DEX

2

Delete word: The control character "-" indicating delete word means the following: All trailing non-printing characters preceded by all printing characters are deleted. Leading non-printing characters remain.

2a

Delete line: all text from the delete cluster up to but not including the first preceding EOL (or CR LF) is deleted.

2b

Null characters which get into the sequential file when TENEX translates CR LF are deleted.

2c

These modifications were made upon consultation with and under the insistence of the Applications training group staff. Given the nature of DEX, I felt they should be brought up immediately. If any changes are necessary, I should be notified as soon as possible. Please incorporate these changes into official documentation which I may verify before the documentation's release.

3

DEX Modifications as per Training Staff Request

(J26246) 7-AUG-75 09:56;;; Title: Author(s): Harvey G. Lehtman/HGL;
Distribution: /POOH([ACTION]) JDH([ACTION]) JCN([ACTION])
JMB([ACTION]) JHB([ACTION]) RWW([ACTION]) SRI=ARC([
INFO=ONLY]) ; Sub=Collections: SRI=ARC; Clerk: HGL;

26246 Distribution

Jonathan B. Postel, Elizabeth J. Feinler, Kirk E. Kelley, N. Dean Meyer, James E. (Jim) White, Douglas C. Engelbart, Martin E. Hardy, J. D. Hopper, Charles H. Irby, Harvey G. Lehtman, James C. Norton, Jeffrey C. Peters, Dirk H. Van Nohuys, Kenneth E. (Ken) Victor, Richard W. Watson, Don I. Andrews, Ann Weinberg, J. D. Hopper, James C. Norton, Jeanne M. Beck, James H. Bair, Richard W. Watson, Mary Ann Kellan, Buddie J. Pine, Andy Poggio, David L. Retz, Laura J. Metzger, Karolyn J. Martin, Jan A. Cornish, Larry L. Garlick, Priscilla A. Wold, Pamela K. Allen, Delorse M. Brooks, Beverly Boli, Rita Hysmith, Log Augmentation, Joseph L. Ehardt, Raymond R. Panko, Susan Gail Roetter, Robert Louis Belleville, Rene C. Ochoa, Ann Weinberg, Joan Hamilton, Adrian C. McGinnis, Robert S. Ratner, David S. Maynard, Robert N. Lieberman, Sandy L. Johnson, James H. Bair, Jeanne M. Leavitt, Rodney A. Bondurant, Jeanne M. Beck, Marcia L. Keeney, Elizabeth K. Michael

BEV 7-AUG-75 12:32 26247

NATIONAL SOFTWARE WORKS DEVELOPMENT

This is the last Quarterly Management Report for the last NSW Contract. Dick Watson will be sending you the 'official' hardcopy in the mail.--Bev Boli

BEV 7-AUG-75 12:32 26247
17 July 1975

Quarterly Management Report 11
Covering the Period 1 July 1974 to 17 July 1975
Stanford Research Institute Project 4015

NATIONAL SOFTWARE WORKS DEVELOPMENT

by

Richard W. Watson

Contract F30602-75-C-0156

Prepared for

Rome Air Development Center (ISIM)
Griffiss Air Force Base
Rome, New York 13440

Attn: Duane L. Stone

NATIONAL SOFTWARE WORKS DEVELOPMENT

ARPA Order Number and Program:	1
Title: NATIONAL SOFTWARE WORKS DEVELOPMENTS	1a
Contractor: Augmentation Research Center, Stanford Research Institute	1b
Date of Contract: 1 July 1974	1c
Amount of Contract: \$701,635	1d
Contract Number: F30602-75-C-0156	1e
Principal Investigator: Dr. Richard W. Watson, phone (415) 326-6200, ext. 2013	1f
Contract Expiration Date: 17 July 1975	1g

NATIONAL SOFTWARE WORKS DEVELOPMENT

I RESEARCH PROGRAM AND PLAN

2

As per our proposal and contract, work has progressed in the following areas:

2a

1) The NSW frontend System, which provides terminal access to the ARPANET, a set of services creating a coherent NSW user environment, and an environment to decrease the cost of new tool creation,

2a1

2) protocols and conventions between the Frontend and Works Manager, Frontend and tools, and Works Manager and tools,

2a2

3) NLS Tool restructuring and enhancement to integrate NLS into the NSW,

2a3

II MAJOR ACCOMPLISHMENTS

3

NSW Frontend

3a

Command Language Interpreter

3a1

An initial version of the NSW Command Language Interpreter (CLI) was designed and written for the PDP-10 and PDP-11 for typewriter-like terminals. The first release of the CLI for the PDP-10 was made on 15-May-75. The CLI is now able to parse all commands for the Works Manager, the NLS Editor tool, and some commands for other tools including the NSW debugger. The CLI communicates with tool processes and the Works Manager via the Distributed Programming System (DPS) or via a shared page,

3a1a

Associated with the CLI is a formal language, called the Control Meta Language (CML), and its compiler. The output of this compiler serves as the program that is interpreted by the CLI in the course of interacting with the user. Most of the features that were originally specified for this language are now implemented,

3a1b

L10 Compiler for the PDP-11

3a2

Since it was required that the CLI be able to execute on a PDP-11 or a PDP-10, a cross-compiler L1011 was written to enable us to write the CLI in the single language L10,

3a2a

In the course of writing the L1011 compiler, it was necessary and desirable to incorporate into the formal L10 language new constructs that provided for machine-independent data structure declaration and

manipulation as well as better exception handling mechanisms (e.g., list features, catch phrases, and co-routines),	3a2b
The enhanced version of the L10 compiler has been quite thoroughly debugged and the L1011 compiler is currently being optimized to produce fewer instructions of code for the same high level constructs. In addition, the runtime package has been designed and implemented.	3a2c
Operating System Interface	3a3
Since the environment is quite different for the two machines, we also developed a level of software called the Operating System Interface (OSI). An OSI has been written and debugged for the PDP-10 and has been partially written for the PDP-11. The CLI has been compiled for the PDP-11 and debugging is now in progress.	3a3a
ELF Operating System	3a4
It was necessary for ARC staff to invest some time in shaking down the ELF operating system for use in the NSW. ELF now serves as a reliable terminal support system for ARC and as a development vehicle for the CLI and DPS development.	3a4a
NSW Debugger	3a5
The NSW debugger has been designed and partially coded and debugged. To facilitate NLS and CLI development activities, a special debugging package was provided that performs all of its functions within a structure very similar to that of the eventual debugger and thus served as a testbed for some of the basic low level primitives and approaches.	3a5a
Stand-alone PDP-11 Debugging Environment	3a6
To facilitate initial L1011 testing and CLI debugging, a stand-alone debugging and cross-net loading facility was developed. This is still being used for some debugging work today.	3a6a
Protocols	3b
Distributed Programming System (DPS)	3b1
A protocol for communicating between procedures in the various NSW modules (i.e. Front End, Works Manager, and tools) called the Distributed Programming System (DPS) was	

designed and documented (24852), implemented for Tenex, and partially debugged. This protocol is in use in the debugging of the Works Manager, the Front end CLI, and the NLS tool back end. 3b1a

A set of procedure packages was designed and documented (24830). Among these packages are: 3b1b

file Package--The set of procedures to be implemented in each host to facilitate the movement of files between workspaces in either the same computer or different computers is called the "File Package". First-pass design and documentation (23926) has been completed. The File Package specification is greatly rewritten and enlarged to include specifics on file types and scenarios for file movement (25850). 3b1b1

debugger--First-pass design and documentation (24579) of a low-level debugger has been completed. 3b1b2

Remote Job Entry Service--First-pass design and documentation (23927) has been completed. 3b1b3

Documentation of FTpFRK 3b2

Documentation was completed (23649) for the inter-host file transfer module (FTpFRK) implemented in the previous year for use by the Journal. 3b2a

Tools 3c

NLS 7 to NLS 8 Conversion 3c1

The conversion of NLS 7 to NLS 8 was completed. NLS 8 was brought up as the running system first at ARC, then, in November, at Office-1. This included conversion of the Ident System, the Journal, and Class 1 user programs. 3c1a

System Conversion documentation 3c2

Various special purpose documents for converting NLS 7 to NLS 8 were produced. 3c2a

NLS Frontend/Backend Code 3c3

The design for splitting NLS code into separate Frontend and Backend components has been completed. The grammar for the NLS Editor has been written and almost debugged. A Backend

Editor Interface has been written for the NSW DPS environment and is almost debugged.	3c3a
NLS File System	3c4
NLS's structured file system has been generalized to include two new concepts, the property and the inferior tree. Properties are typed data blocks that are chained to the nodes within an NLS file. The familiar NLS statement is a property of typed text. In addition to textual properties, several properties have been added to support the storage of graphics within NLS files. Both properties and inferior trees will find use in applications ranging from comments and heading entries to the storage of digitized speech strings.	3c4a
NLS Graphics	3c5
The capability for the manipulation of line drawings has been added to NLS. The lineprocessor has been modified to service a standard storage tube graphic display. By utilizing mouse and keyset, the graphics user can create and edit line drawings such as flowcharts and block diagrams. The diagrams are stored within the NLS file system.	3c5a
New NLS Features	3c6
New document production tools and COBOL programming aids for NLS-9 in the NSW environment were completed. These include an automatic Editing system, and a COBOL system for augmenting creation of source code and reformatting for remote batch job entry on the B47.	3c6a
File conversion Software	3c7
Software was produced to convert IBM EDCDIC files to ASCII Tenex files on a PDP10. This program has been used to create NLS files from text files originally generated on an MTST.	3c7a
Software for Singer 6000	3c8
Software was designed to make the Output Processor compatible with the Singer 6000. With George Lithograph, software has been developed for the Singer 6000 COM machine to process virtual COM files.	3c8a

Standard Air Force Manual Format

3c9

The text entry process for standard Air Force manuals was specified. A system has been implemented for generating Output Processor directives to create standard Air Force formats both for line printers and COM output. We worked with Air Force personnel on the production of a 4000 page manual to test this system, and are now ready for production.

3c9a

NLS 8 Help System

3c10

From preliminary software and a sketchy data base, Help was made operational for NLS 8, debugged, and its data base completed. Further use of the Help system led to its revision into the Multi-file Help.

3c10a

Multi-file Help System

3c11

The NLS Help system has been redesigned and coded to work across multiple file data bases, and to serve as the NSW Help system for any tool that provides the necessary data base. New tool data bases have been added (e.g. Calculator tool), while revisions on the existing "Core" data base and individual tool data bases have begun.

3c11a

Secretarial Functions Guide

3c12

A self-instructional Guide to performing secretarial functions using NLS tools was completed. This document contains introductory information about the system and individual instructional modules covering tasks such as writing and sending memos, creating and revising drafts of reports, and writing and formatting letters.

3c12a

Introductory Documentation

3c13

Completed documents include: the "Preface to NLS Tools", which introduces the new user to basic NLS concepts and commands; the "NLS 8 Command Summary"; the hard copy "NLS 8 Glossary"; the "Introduction to NLS Documentation", a discursive introduction outlining NLS capabilities for documentation production; and the "Format Library", a guide to the various formats available through the Format subsystem.

3c13a

NATIONAL SOFTWARE WORKS DEVELOPMENT

III PROBLEMS ENCOUNTERED 4

No problems were found that require government action. 4a

IV FISCAL STATUS 5

Estimated expenditures and commitments to date are \$689,173
excluding computer and other lease commitments. Estimated
additional funds required to complete the work are \$0. 5a

V ACTION REQUIRED BY THE GOVERNMENT 6

None. 6a

VI NEXT QUARTER PLANS 7

This is the final quarter of this contract. 7a

Approved by:

8

Richard W. Watson, Principal Investigator 8a

NATIONAL SOFTWARE WORKS DEVELOPMENT

(J26247) 7-AUG-75 12:32;;; Title: Author(s): Beverly Boli/BEV;
Distribution: /DLS([ACTION]); Sub-Collections: SRI=ARC; Clerk:
BEV; Origin: < BOLI, QMR,NLS;14, >, 4-AUG-75 13:31 BEV ;;;
####;

seminar in August

Bob, we have had some problems getting good copy from reproduction
(also I was sick for a week), Will send it as soon as it gets BACK
FROM REPRO. If you want an advanced copy , print out a file
<office=1,lieberman,seminar,> Regards Rob

1

seminar in August

(J26248) 7-AUG-75 15:16;;; Title: Author(s): Robert N.
Lieberman/RLL; Distribution: /RDA([ACTION]); Sub-Collections:
SRI=ARC; Clerk: RLL;

What We Can Learn from the DPS Decision

What We Can Learn from the DPS Decision

Making mistakes is one of the more common ways to learn. The trick is to learn from them. I have been thinking alot about the events of the past week and trying to see both what can be learned and what opportunities it presents us. Hidden within everything is its opposite. That is no exception here, but that is the subject of another note.

First, let me say clearly that careful analysis of the situation has me completely convinced that if there is any finger pointing to be done the person pointed at is me. I just did not see clearly that we have a technology transfer problem with DPS, CLI, Debugger etc. that takes the same care and attention following the same principles that we use to transfer technology to our end NLS user groups.

We need to provide the same level of assistance in training, usable documentation, being sure they understand what we have provided them and how it works, personal on ste handholding, feedback responsiveness etc. We can have as I believe we did in DPS a very fine product, but if we do not pay full attention to the above types of details the transfer will not take properly if at all.

WE need to be sure that in our CLI and Debugger work we reexamine this whole set of transfer needs and evaluate what we should be doing to profit from the DPS experience. In the PDS case we probably should have had almost daily phone calls, better documentation before turning it over to them, and I should have asked Jim to fly back to MCA two to three weeks ago and to have stayed there long enough so that all was going smoothly. The result was they preferred to use a kludge they could understand and to rebuild the whole thing themselves to guarantee that they understood it and were in control. It just like an NLS user left on his own who could get frustrated and go back to a typewriter or pencil and paper. Jim Dave Charles and the others who have been working so hard on DPS please accept my strong regrets that I had just not recognized the technology transfer problem clearly earlier.

What We Can Learn from the DPS Decision

(J26249) 7-AUG-75 19:22;;; Title: Author(s): Richard W. Watson/RWW;
Distribution: /ARC-DEV([ACTION]) ARC-APP([INFO-ONLY]) DCE([
INFO-ONLY]) ; Sub-Collections: SRI-ARC ARC-DEV ARC-APP; Clerk: RWW;

26249 Distribution

Susan Gail Roetter, Raymond R. Panko, Adrian C. McGinnis, James C. Norton, J. D. Hopper, Elizabeth J. Feinler, James H. Bair, Robert N. Lieberman, N. Dean Meyer, Sandy L. Johnson, Martin E. Hardy, Douglas C. Engelbart, Mary Ann Kellan, Andy Poggio, David L. Retz, Jan A. Cornish, Larry L. Garlick, Delorse M. Brooks, Beverly Boli, James E. (Jim) White, Ann Weinberg, Kenneth E. (Ken) Victor, Dirk H. Van Nouhuys, Jonathan B. Postel, Elizabeth K. Michael, David S. Maynard, Karolyn J. Martin, Harvey G. Lehtman, Kirk E. Kelley, Charles H. Irby, Joseph L. Ehardt, Robert Louis Belleville, Don I. Andrews, Richard W. Watson, Douglas C. Engelbart, Buddie J. Pine, Laura J. Metzger, Priscilla A. Wold, Pamela K. Allen, Joan Hamilton, Rene C. Ochoa, Jeffrey C. Peters, Marcia L. Keeney, Jeanne M. Beck, Geoffrey S. Goodfellow, Rodney A. Bondurant, Douglas C. Engelbart, Jeanne M. Leavitt

test

this is a test of sendmail from isic using the supposed nls 8.5 (nee gnls) on the evening of 7 aug 75, --jon.

1

test

(J26250) 7-AUG-75 23:14;;; Title: Author(s): Jonathan B.
Postel/JBP; Distribution: /JBP([INFO-ONLY]); Sub-Collections:
SRI-ARC; Clerk: JBP;

JAC3 8-AUG-75 10:30 26251

unrecorded

This is an unrecorded journal message sent by ekm logged in as jac.

1

unrecorded

(J26251) 8-AUG-75 10:30;;; Title: (Unrecorded) Title: Author(s):
Jan A. Cornish/JAC3; Distribution: /JCP([ACTION]) JBP([ACTION])
CHI([INFO-ONLY]) EKM([INFO-ONLY]) ; Sub-Collections: SRI-ARC;
Clerk: JAC3;

Proposed CLI MSG conventions

The following is a proposed convention to be used by the CLI when interacting with any other process via MSG. I have attempted to minimize the amount of existing "shared page" code that has to be changed, while trying to simplify things where possible. In this vain, then, I propose the following:

Parameters passed in these types of messages should be encoded in PCPB36 as is now done, but without the extra layers of ABC (Since such arguments will upon receipt generally be immediately converted to some other form, it is not obvious to me that there is much value in it). There should be no absolute pointers within the page being sent with a message,

The words of the message page will be denoted (starting with word zero) as DBOX, SBOX, LENGTH, P1 through P509. LENGTH is the total length in words of the message. LENGTH may be set to 512 initially for the intra-host case. P1 would be used to hold a type-of-message code, P2 through P509 would be used differently based on the value in P1. This would be of the form:

invoke-function: invoke the specified function/action

P1 = 1

P2 = displacement to beginning of argument list, a PCPB36 LIST, where each element of the list is an argument, or -1 meaning none,

P3 = boolean (TRUE => acknowledgement required, FALSE => acknowledgement not expected)

a word of zero will be FALSE, anything else will be TRUE,

p4 = beginning of an ASCII string specifying the action to be performed (the procedure to call),

Note 1: This incorporates the DPS notion of NOTE, in that no acknowledgement need be sent if the parameter p3 is FALSE. Since there is no notion of a call stack or thread of control, this should suffice. Thus, for example, the encapsulator would invoke a check-sockets function in the FE, with no acknowledgement required, to give the FE a chance to verify that the right processes are hooked together via TELNET,

Note 2: This encompasses the DPS notions of help and co-routine returns in that a process achieves these capabilities by invoking functions (such as "help" or "show" in the FE) in its "caller".

Proposed CLI MSG conventions

In the case of the function "help" in the FE, the arguments would be help(helpcode, helpmsg, abortmsg, param1, ..., param8), where param1 thru param8 are optional.

1b1f1

Note 3: This may all fall apart if a process invokes several functions in another process, specifying that acknowledgement is required, without waiting for each to be acknowledged (This is as close as MSG can come to DPS "out of line" calls).

1b1g

acknowledge: the last requested function/action that required acknowledgement has been processed; p2 indicates whether the action was successfully performed or not,

1b2

P1 = 2

1b2a

P2 = boolean (TRUE => succeeded, FALSE => failed)

1b2b

P3: = begining of results PCPB36 LIST, or -1 meaning none,

1b2c

In the case of a failure to process the request, the first result is an error code and the second a diagnostic message.

1b2c1

Note: The acknowledgement will be matched to the last invoke-function issued to the sending process which had the p4 parameter = TRUE. If there is no outstanding request for which an acknowledgement is expected, then the acknowledgement and its results are discarded.

1b2d

Proposed CLI MSG conventions

(J26252) 8-AUG-75 17:21;;; Title: Author(s): Charles H. Irby/CHI;
Distribution: /CHI([INFO-ONLY]); Sub-Collections: SRI-ARC; Clerk:
CHI; Origin: < IRBY, PROC-CALL=MSG,NLS;3, >, 8-AUG-75 17:10 CHI
;;;####;

EKM 9-AUG-75 13:59 26253

demo

this is a demo

1

demo

(J26253) 9-AUG-75 13:59;;; Title: Author(s): Elizabeth K.
Michael/EKM; Distribution: /EKM([ACTION]) ; Sub-Collections:
SRI-ARC; Clerk: EKM;

Gunter Report for week ending 8/10/75

In addition to all this I want to inform you of my new address while I am here: 1007B SKI Lodge, Montgomery, Alabama, phone number to come next week,, You all come on down and visit me real soon ya hear,.....

Gunter Report for week ending 8/10/75

This file will contain weekly progress reports on NSW at Gunter Air Force base. The first branch each week will summarize the various groups that are working and include any global comments. The second branch will be daily notes.

The Week Ending 8/10/75

Week Summary

66-1

The volumes 2 and 7 that were put online at the pentagon are being revised at Gunter. The work is being done by the group in LG as well as about twenty people from other bases. After a slow start, procedures were set up that allowed us to keep up to date on most edits, here is the procedure we are currently using:

Procedures for editing 66-1

The various groups make their edits and/or additions to the current drafts of the various chapters, each chapter is a separate file,

they submit their edits to the "Murder Board" for approval,

The "Murder Board" gives the ready to be edited copies to Sgt. Albano,

Sgt. Albano, gives the copies to Ann who then gives them to the appropriate editor, (Some edits are more difficult and some editors are more experienced than others,)

editors:

The editors make the changes on the files located in directory AFM,

About once every fifteen pages, the editors do an update,

When finished, the editos do an update compact,

When finished, the editors mark on the draft: "edited, the date and their initials,"

The editor gives the draft to Ann,

1

2

2a

2a1

2a1a

2a1a1

2a1a1a

2a1a1b

2a1a1c

2a1a1d

2a1a1e

2a1a1e1

2a1a1e2

2a1a1e3

2a1a1e4

2a1a1e5

printing 2a1a1f

The revised version is printed on the Gunter printer in the block house 2a1a1f1

Ann informs Sgt. Albano that a new draft is ready and he picks it up at the block house. 2a1a1f2

Sgt. Albano gives the new draft to Major Garrett who then recycles it through the groups. 2a1a1g

Copies of the updated files are transferred to ISIC for backup. 2a1a1h

Old drafts are stored on shelves that have large signs, "DO NOT USE. 2a1a1i

I eventually hope to phase myself out of this procedure when there are appropriate people to do the printing and the transferring of files. 2a1a2

Training: 2a2

Susan and Priscilla ran two training classes: one in the morning for people who had already had training and one in the afternoon for new people. About eight people went through training. They will probably submit a detailed report. 2a2a

PR Test Document: 2a3

By the end of the week, the two people in PR (Cindy and Jo) were beginning to put in the test document. This is currently being done online. 2a3a

Base Tops 2a4

This group decided not to do the entire manual using NLS, but made the online files available to anyone who wanted them. One group in PR had a small 3 section (3 files) part that they decided to do with NLS. By the end of the week Mr Spires who is in charge of these sections had some editing to be done. It was done by Cindy. He projects 20-25 new pages to be added next week, and needs a rough draft (quickprint) by next Friday. 2a4a

Quality Control 2a5

Through several meetings and a demo, the groundwork was laid

Gunter Report for week ending 8/10/75

to work with this group who gives final approval to the PR test document. They agreed to allow us to use the format that has been developed for 66-1. Mr Fisher who is head of this group is rather skeptical about NLS. Future plans are to have one his small montly reports put on line, updated with revisions and printed for him as he needs it.

2a5a

DIREPS

2a6

Sgt. Crabtree, an avid NLS fan, is working with problems that come in that need immediate answers. He has outlined his needs and Susan will discuss these with appropriate ARC people to devise a plan for him to use.

2a6a

ELF

2a7

BY the end of the week, the ELF printer was working nicely. There are still some problems that make it impossible to have any terminal hooked up, but Larry Crain is working on those.

2a7a

Daily Reports

2b

Monday

2b1

AM:spent in LG helping them organize for 66-1.,they had not chosen editors and felt non-trained people could learn in one or two days to do the work.,problems with using people from other groups.,arranged for training of new people and set up directory for AFM where work will be done.

2b1a

PM. worked with Larry on getting printer running.,worked from ELF console but crashed after printing of each single file. Paul Williamson from ADR was there but knew nothing about ELF.

2b1b

Tuesday

2b2

AM: met with Mr. Fisher of Quality Control and several others to discuss format of test PR document. agreed on 66-1 format.,set up demo for Thursday pm.,worked on printer with Larry able to get Susan from another site at Gunter to print on the printer.

2b2a

PM: LG in panic.,had begun changes using files in wrong directory.,fixed it.,by the end of day working groups had made changes that needed to be edited on three files., Susan and Priscilla and I spent 5-6 hours putting in the changes.

2b2b

Gunter Report for week ending 8/10/75

Wednesday

2b3

to LG to set up final routine,,they were duly impressed that they had the printouts with all the changes they had submitted the night before, i was assured of all the people support i needed,,the bulk of the material is expected to arrive beginning Thursday,

2b3a

PM: Larry and I with Joe and Dave worked on getting ELF running--it is up!!! spoke to Capt. Davis in Base Tops, decided not to do whole document with NLS, but made files available to groups that want to use it,,one group in PR (Mr. Spires) would like to make his changes using NLS,,said he had trained people but it turned out to be some of the people that were beign trained right at this time,,made an appt for him tomorrow morning,

2b3b

Thursday

2b4

AM: met with Mr. Spires,,his part of Base Tops is small and he anticipates only 25 additional pages,,he will have some first edits available this afternoon and we will use them in class,,spoke with Crabtree on his DIREPS problem will have him explain it to susan before she goes,

2b4a

PM: demo for Mr. Fisher and several others,,the idiots who saw it loved it and could see all sorts of possibilities,,Fisher was critical of print quality from the ELF printouts,,discovered that Crabtree with his DIREPS deals with JO who is one of the people being trained in PR,,LG had some heavy editing that was beginning to come in,

2b4b

Friday

2b5

AM: in LG holding hands,,by noon we had two people making edits on different files,,everyone seemed satisfied,

2b5a

PM: three people making edits and routine is going well,,got printouts of one chapter that had been submitted in the AM,

2b5b

Saturday

2b6

more edits coming in and two people for morning and one in afternoon were around to make these,,had to do a few global changes and that went well,,Col W....? happened to come by just to visit and we sat and talked a bit,,he is in charge

Gunter Report for week ending 8/10/75

of the whole group and was vaguely aware of things that were going on,

2b6a

Gunter Report for week ending 8/10/75

(J26254) 9=AUG-75 14:40;;; Title: Author(s): Ann Weinberg/POOH;
Distribution: /SRI-ARC([INFO-ONLY]) LAC([INFO-ONLY]) ;
Sub-Collections: SRI-ARC; Clerk: POOH; Origin: < WEINBERG,
GUNTERREPORT,NLS;2, >, 9=AUG-75 14:36 POOH ;;;;####;

26254 distribution

Douglas C. Engelbart, Martin E. Hardy, J. D. Hopper, Charles H. Irby, Harvey G. Lehtman, James C. Norton, Jeffrey C. Peters, Dirk H. Van Nouhuys, Kenneth E. (Ken) Victor, Richard W. Watson, Don I. Andrews, Lawrence A. Crain, Mary Ann Kellañ, Buddie J. Pine, Andy Poggio, David L. Retz, Laura J. Metzger, Karolyn J. Martin, Jan A. Cornish, Larry L. Garlick, Priscilla A. Wold, Pamela K. Allen, Delorse M. Brooks, Beverly Boli, Rita Hysmith, Log Augmentation, Joseph L. Ehardt, Raymond R. Panko, Susan Gail Roetter, Robert Louis Belleville, Rene C. Ochoa, Ann Weinberg, Joan Hamilton, Adrian C. McGinnis, Robert S. Ratner, David S. Maynard, Robert N. Lieberman, Sandy L. Johnson, James H. Bair, Jeanne M. Leavitt, Rodney A. Bondurant, Jeanne M. Beck, Marcia L. Keeney, Elizabeth K. Michael, Jonathan B. Postel, Elizabeth J. Feinler, Kirk E. Kelley, N. Dean Meyer, James E. (Jim) White

ARPANET book

Friday I talked to Becker and Hayes (subsidiary of McGraw Hill who is editing the Arpanet Book) and to Dr. Licklider. The contract concerning writing and editing the book has been approved and B&H will be contacting authors for their chapters soon. (They hope to finish the whole thing by early next year). B&H will also publish a bibliography of pertinent articles along with hardcopy and microfilm of all the articles in the bibliography. They contacted me to supply the original articles. I said I could not do this unless money was provided for manpower and also would want a set of the documents for NIC use as well. Lick thought this was reasonable and will check with Craig (who was out that day) and let me know. I told them I could handle nothing along these lines until the Resource Handbook is published.

1

ARPANET book

(J26256) 9-AUG-75 18:33;;; Title: Author(s): Elizabeth J.
Feinler/JAKE; Distribution: /DCE([INFO-ONLY]) JCN([INFO-ONLY])
RWW([INFO-ONLY]) JBP([INFO-ONLY]) ; Sub-Collections: SRI=ARC;
Clerk: JAKE;

CLI MSG protocol conventions

The following are the conventions used by the CLI when interacting with any other process via MSG, 1

The words of the message page will be denoted (starting with word zero) as LENGTH, DBOX, SBOX, RESERVED1, RESERVED2, RESERVED3, RESERVED4, RESERVED5, P1 through P504. LENGTH is the total length in words of the message. LENGTH may be set to 512 initially for the intra-host case. P1 is used to hold a type-of-message code, P2 through P504 are used differently based on the value in P1. 1a

invoke-function: invoke the specified function/action 1a1

P1 = 1 1a1a

P2 = displacement to beginning of argument list, a PCPB36 LIST, where each element of the list is an argument, or -1 meaning none. 1a1b

P3 = integer (> 0 => acknowledgement required, interger is to be used as a transaction identifier (TID) in the acknowledgement; 0 => acknowledgement not expected) 1a1c

p4 = beginning of an ASCIZ string specifying the action to be performed (the procedure to call). 1a1d

acknowledge: a previously requested function/action that required acknowledgement has been processed; p2 indicates whether the action was successfully performed or not. 1a2

P1 = 2 1a2a

P2 = boolean (TRUE => succeeded, FALSE => failed) 1a2b

a word of zero will be FALSE, anything else will be TRUE. 1a2b1

P3 = Transaction Identifier (TID) specified in the invoke-function request. 1a2c

P4: = beginning of results PCPB36 LIST, or -1 meaning none, 1a2d

In the case of a failure to process the request, the first result is an error code and the second a diagnostic message. 1a2d1

Announcement and registration form in the mail

copy of online file is in the mail to you. sorry for the delay, Rob

1

Announcement and registration form in the mail

(J26258) 11-AUG-75 18:41;;; Title: Author(s): Robert N.
Lieberman/RLL; Distribution: /RDA([INFO=ONLY]) ; Sub=Collections:
SRI=ARC; Clerk: RLL;

TNLS Q bug printg fix

Here is a copy of the fix I made in printg at ISIC in NLS directory
to fix the relative indenting bug,

TNLS Q bug printg fix

```

(printg) %***% PROCEDURE (da, stid1, stid2, type, oldsw);          1
%print the structure (type) addressd by stid1,stid2; oldsw is the
%address of an already open sequence work area or zero. If zero, a
%sequence work area is opened by printg and closed; if non-zero,
%the sequence is not closed by printg. %                          1a
%-----%                                                         1b
LOCAL                                                             1c
    i, %control variable for loop%                                1c1
    char, %current character%                                     1c2
    gapcol, %column of character preceding last invisible char%  1c3
    gapptr, %pointer to character preceding last invisible char% 1c4
    startp, %pointer to start of line character%                 1c5
    printd, %number of columns to indent%                        1c6
    lincnt, %count of lines printed for current statement%      1c7
    stid, %current stid being printed%                           1c8
    maxcol, %maximum number of columns for current line%        1c9
    stnrt, %whether statement numbers are to be printed on right% 1c10
    extraline, %true if statement number put on separate line%  1c11
    stnlength, %save length of stmt no,%                         1c12
    sw, %address of sequence generator work area%                1c13
    top, %counter for spacing at top of page%                    1c14
    vspc1, %viewspecs from sequence work area %                  1c15
    vspc2,                                                         1c16
    head, %address of file header%                                1c17
    cntrlstr, %address of non-printing char string%              1c18
    mkrptr; %marker table pointer%                                1c19

```

TNLS Q bug printg fix

```

LOCAL STRING                                1c20
    str[100],    %scratch string%          1c20a
    stnsig[50]; %holds statement number or signature% 1c20b
LOCAL TEXT POINTER z1;                      1c21
REF oldsw, sw, da, mkrptr;                 1c22

IF stid1 = endfil THEN err($"End of file."); 1d
da,dacrow = 0; %start a page for possible pagination later% 1e
IF da,davspec,vspagf THEN                 1f
    BEGIN                                  1f1
        top = 3;                          1f2
        UNTIL (top := top - da,davinc) <= 0 DO da,dacrow = da,dacrow +
        da,davinc;                        1f3
    END;                                   1f4
pageno = 0;                               1g
&sw = IF &oldsw THEN &oldsw              1h
    ELSE <SEQGEN, openseq> (stid1, IF da,davspec,vspagf THEN stid1
    ELSE stid2 % plex only should be handled by the calling routine
    since stid1 and stid2 supposedly form a legitimate group! %,
    da,davspec, da,davspc2, da,dausqcod, da,dacacode); 1h1
ON SIGNAL ELSE                             1i
    IF NOT &oldsw THEN <SEQGEN, closeseq>(&sw); 1i1
z1[0] = IF stid1,stastr THEN da,dacsp ELSE stid1; 1j
UNTIL ((stid = seqgen (&sw)) = endfil) OR
(type = stmtv AND sw,swcstid # stid1) DO 1k
    BEGIN                                  1k1
        % get viewspecs from sequence work area. % 1k2
        vspec1 = da,davspec = sw,swvspec; 1k2a

```

TNLS Q bug printg fix

```

vspc2 = da,davspc2 = sw,swvsp2;                                1k2b
%set up work area s2work for READC%                             1k3
s2work = stid;                                                  1k3a
s2work[1] =                                                     1k3b
    IF vspc1,vsnamf OR stid,stastr THEN 1 %print name%         1k3b1
    ELSE fchtxt (stid); %skip past name%                       1k3b2
fechc1 (forward, $s2work);                                     1k3c
%markers (not allowed for now)%                                 1k4
prmkrf = 0;                                                    1k4a
IF FALSE AND vspc1,vsmkrf AND NOT stid,stastr THEN           1k4b
    BEGIN %see if marker in statemnt%                          1k4b1
        &mkrptra = &mkrtb = $filhed +                          1k4b2
            (head = <FILMNP, filhdr>(stid,stfile));           1k4b2a
        mkrend = &mkrptra +                                    1k4b3
            [$mkrtb1 = $filhed + head] * mkrl;                1k4b3a
        mkrf1g = FALSE;                                        1k4b4
    FOR mkrptra UP mkrl UNTIL = mkrend DO                       1k4b5
        IF mkrptra,mkpsid = stid,stepsid THEN                 1k4b5a
            BEGIN                                              1k4b5a1
                mkrf1g = TRUE;                                  1k4b5a2
                mkrcnt = mkrptra,mkccnt + 1;                  1k4b5a3
            EXIT;                                              1k4b5a4
            END;                                               1k4b5a5
        END;
    END;
    END;
printd = %indentation%                                         1k5

```

```

CASE TRUE OF
    = vspci,vsrind;
        IF (vspci,vsbrof OR vspci,vsplx) THEN
            MAX (tppoffset, MIN (da,daind *
                (sw,swclvl-sw,swslvl) + tppoffset, da,damind,
                spacestr,M))
        ELSE tppoffset;
    = vspci,vsindf;
        MAX (tppoffset, MIN (da,daind * (sw,swclvl-1) +
            tppoffset, da,damind, spacestr,M));
ENDCASE tppoffset;

% TPOFFSET is a global which the user can set via Execute
Viewchange. This allows the user to control the left margin
of his print out, %
*prbuf* = *spacestr* [empty + 1 TO printd];
maxcol = da,damcol; %max no. cols%
stnrt = FALSE;
IF vspci,vsstnf AND stid,stpsid # origin AND
NOT stid,stastr THEN
    IF NOT vspci,vsstnr THEN
        BEGIN %print statement number%
            IF vspci,vssidf
                THEN % display sid's %
                    *prbuf* = *prbuf*, *0, STRING( getsid(stid) )
                ELSE % display line numbers %
                    fechm (sw,swsvw, sprbuf);
            *prbuf* = *prbuf*, SP;
        END

```

TNLS G bug printg fix

```

ELSE % statement numbers go on the right %           1k9b
  BEGIN                                               1k9b1
    *stnsig* = NULL;                                  1k9b2
    IF vspcl,vssidf                                   1k9b3
      THEN % display sid's %                          1k9b3a
        *stnsig* = '0, STRING( getsid(stid) )        1k9b3a1
      ELSE % display line numbers %                   1k9b3b
        fechnm (sw,swsvw, sstnsig);                  1k9b3b1
    stnlength = stnsig,L;                             1k9b4
    strt = TRUE;                                       1k9b5
  END;                                                 1k9b6

  gapcol = da,daccol = prbuf,L * da,dahinc;          1k9c
%line printing loop%                                  1k10
FOR lincnt = 0 UP UNTIL = vspcl,vstrnc DO            1k11
  BEGIN                                               1k11a
    IF inptrf THEN EXIT 2;                             1k11b
    gapptr = startp = prbuf,L;                         1k11c
    UNTIL da,daccol >= maxcol + 1 DO                  1k11d
      CASE char = READC ($s2work) OF                  1k11d1
        =SP:                                           1k11d1a
          BEGIN                                         1k11d1a1
            gapptr = prbuf,L;                          1k11d1a2
            gapcol = da,daccol;                        1k11d1a3
            da,daccol =
            da,daccol+p,putchr(&da,char,da,daccol,$prbuf); 1k11d1a4
          END
        OTHERWISE
          gapcol = gapcol + 1;
      END
    END
  END

```

TNLS Q bug printg fix

```

END; 1k11d1a5
:=ENDCHR, =EOL, =CR; 1k11d1b
BEGIN 1k11d1b1
gapptr _ prbuf,L; 1k11d1b2
gapcol _ da,daccol; 1k11d1b3
EXIT LOOP; 1k11d1b4
END; 1k11d1b5
:=TAB; 1k11d1c
BEGIN 1k11d1c1
gapptr _ prbuf,L; 1k11d1c2
gapcol _ da,daccol; 1k11d1c3
da,daccol _
da,daccol+putchr(&da,char,da,daccol,sprbuf); 1k11d1c4
END; 1k11d1c5
:=CA, =C,, =LF, =CD, =BC, =BW, =$ascalt, IN [1B,32B],
IN [34B,36B]: %an acceptable non-printing character% 1k11d1d
BEGIN 1k11d1d1
c_ntrlstr _ <DSPGEN, npstrad>(char);%get np
representation (address)% 1k11d1d2
%see if non-printing character string will exceed
current line% 1k11d1d3
IF [c_ntrlstr].L + da,daccol >= maxcol THEN 1k11d1d4
BEGIN 1k11d1d4a
s2work[1] _ s2work[1] - 1; 1k11d1d4b
fecnci (forward, ss2work); %repeat last char
next time through% 1k11d1d4c
REPEAT CASE (char _ EOL); 1k11d1d4d

```

TNLS Q bug printg fix

```

        END                                1k11d1d4e
    ELSE %fits on current line%           1k11d1d5
        FOR i = 1 UP UNTIL > [cntrlstr],L DO 1k11d1d5a
            BEGIN                            1k11d1d5a1
                char = *[cntrlstr]*[i];      1k11d1d5a2
                da,daccol =
                da,daccol+putchr(&da,char,da,daccol,sprbuf);
                                                    1k11d1d5a3
            END;                               1k11d1d5a4
        END;                                1k11d1d6
    ENDCASE                                1k11d1e
        BEGIN                                1k11d1e1
            da,daccol =
            da,daccol+putchr(&da,char,da,daccol,sprbuf); 1k11d1e2
        END;                                1k11d1e3
%by here, a line has been constructed, fix gapptr and gapcol
if there were no invisibles or if the last invisible was the
last character%                            1k11e
    IF startp = gapptr THEN %no invisibles in line% 1k11e1
        BEGIN                                1k11e1a
            %an extra character was collected in case it was a
            gap; it wasn't, so don't print it on this line% 1k11e1b
            gapptr = prbuf,L = 1;            1k11e1c
            gapcol = da,daccol = 1;         1k11e1d
        END;                                1k11e1e
    IF gapptr = prbuf,L = 1 THEN %last char a gap
    (SP)%                                    1k11e2
        BEGIN %two extra spaces at end of statement
        allowed%                             1k11e2a

```

TNLS Q bug printg fix

```

char = READC(ss2work);                                1k11e2b
IF char # SP THEN                                      1k11e2c
    %just store it for the next line of this statement%
    da,daccol = da,daccol +                            1k11e2c1
    putchr(&da,char,da,daccol,sprbuf);                 1k11e2c2
END;                                                  1k11e2d

IF (char = ENDCHR OR lincnt = vspc1,vstrnc) AND strt AND
da,daccol + stnsig,L + 2 <= maxcol THEN extraline = FALSE 1k11f

    % can cram statement number on right of this line (which
    is the last line of the statement), so don't print it out
    yet%                                               1k11f1

ELSE                                                  1k11g

BEGIN                                                1k11g1

extraline = TRUE;                                    1k11g2

IF NOT prtype (&da, gapptr) THEN EXIT 2; %type the line% 1k11g3

IF NOT (char = ENDCHR OR lincnt + 1 = vspc1,vstrnc) THEN 1k11g4

    *prbuf* [prindt + 1 TO prbuf,L] = *prbuf* [gapptr + 2
    TO prbuf,L]                                       1k11g4a

ELSE                                                1k11g5

    *prbuf* = *spacestr*[empty + 1 TO prindt];       1k11g5a

gapcol = da,daccol = prbuf,L*da,dahinc;            1k11g6

END;                                                1k11g7

z1[0] = stid;                                       1k11h

IF char = ENDCHR THEN EXIT;                          1k11i

END;                                                1k11j

% The statement is completed,%                        1k12

```

TNLS G bug printg fix

```

%statement numbers on right%                                1k13
  IF strt THEN                                              1k13a
    BEGIN                                                    1k13a1
      (stnort);                                             1k13a2
    IF extraline THEN                                       1k13a3
      BEGIN          %putting number on separate line%    1k13a3a
        *prbuf* _ *spacestr*[1 TO maxcol = stnsig,L=2];    1k13a3b
      END                                                    1k13a3c
    ELSE                                                     1k13a4
      BEGIN                                                    1k13a4a
        *prbuf* _ *prbuf*, *spacestr*[1 TO maxcol = da,daccol
        = stnsig:L =2];                                     1k13a4b
      END;                                                    1k13a4c
      *prbuf* _ *prbuf*, Sp,Sp, *stnsig*;                  1k13a5
      IF NOT (extraline AND vspci,vsidtf AND NOT stid,stastr)
      THEN prtype(&da, prbuf,L);                             1k13a6
    END                                                       1k13a7
  ELSE extraline _ FALSE;                                    1k13b
%blank line and signature%                                  1k14
  IF vspci,vsidtf AND NOT stid,stastr THEN                  1k14a
    BEGIN                                                    1k14a1
      *stnsig* _ NULL;                                       1k14a2
      fechsig (stid, $stnsig);                                1k14a3
      IF extraline AND stnlength + stnsig,L +3 > maxcol -
      print THEN                                             1k14a4
        BEGIN                                                1k14a4a

```

TNLS Q bug printg fix

```

prtype(&da, prbuf,L);                                1k14a4b
extraline = FALSE;                                    1k14a4c
END;                                                    1k14a4d
IF extraline THEN *prbuf*[maxcol - stnlength - stnsig,L
-2 TO maxcol - stnlength -3] = *stnsig*                1k14a5
ELSE *prbuf* =                                          1k14a6
    *spacestr* [1 TO maxcol - stnsig,L], *stnsig*;     1k14a6a
prtype (&da, prbuf,L);                                1k14a7
END                                                    1k14a8
ELSE IF vspci,vsblkf AND NOT extraline THEN prtype(&da, 0); 1k14b
END;                                                    1k15
IF NOT &oldsw THEN <SEQGEN, closeseq> (&sw);          11
%wait for output buffer empty%                        1m
!dobe(777777B);                                       1m1
RETURN(z1, 1);                                        1n
END,                                                    1o

```

TNLS Q bug printg fix

(J26259) 11-AUG-75 22:18;;; Title: Author(s): Kirk E. Kelley/KIRK;
Distribution: /JDH([ACTION]) ; Sub=Collections: SRI-ARC; Clerk:
KIRK;

SCHEDULE AND RESPONSIBILITY FOR FINAL REPORT

SCHEDULE AND RESPONSIBILITY FOR FINAL REPORT

Week end of Aug 9-10

Dee checks references and headers to make sure correct and consistent throughout; makes printout for RWW to look at First thing week of Aug 11,

Jeanne Leavitt has sent form 1472 to Duane Stone. This is all that was necessary besides the online file,

Week starting Monday Aug, 11

RWW reads report, Makes (HOPEFULLY MINOR) changes for Dee to put in at end of week, Report should be next to perfect by the end of week,

Dee fills out Report Approval Form so it is ready for the signing off procedure. If dee has questions in editing she should be able to consult DVN. Especially if she notices any hanging output processor directives or messed up formatting, Dean can also help here with expert advise on OP things,

MONDAY AUG 18 and days following:

Doug returns, There should be a copy ready for him to see, with the Report Approval Form as a cover sheet. Doug should make less than .0008% changes and be able to sign it in good faith,

When Doug has signed it, then Bart Cox has to sign it. He will read it very carefully and probably find the remaining typos. Before Dee puts in those few changes, Spencer Floyd must sign it. His main interest, is there anything contractual we have overlooked, Spencer will be very glad to see the report going out the door, I guarantee that,

Dee puts in final changes, THEN:

The file must be tranfered from ISI to Office-1. Dee should have help from Jeff and it would be nice if DVN could oversee as a guiding angel,

Then Doug or DVN can notify Duane Stone of the location of the file at Office-1, and our mission will be complete. Dee should send a Xerox copy of the Report Approval Form to Duane so he knows all the sign-offs really happened,

MISSION ACCOMPLISHED. Good luck, and I hope when I return it has all been brought to successful completion. I've worked very hard

SCHEDULE AND RESPONSIBILITY FOR FINAL REPORT

on this report and gotten very involved in seeing it get done and
come out right,

4f

If you lose your copy of this file, which I don't even have time
to journalize or edit because am racing off to the bank and then
into the mountains, it is at ISI, <srife>final, Jon Postel knows
the password,

4g

SCHEDULE AND RESPONSIBILITY FOR FINAL REPORT

(J26260) 12-AUG-75 08:36;;; Title: Author(s): Jeanne M,
Leavitt/JML; Distribution: /DVN([ACTION]) DMB([ACTION]) RWW([
ACTION]) DCE([ACTION]) DPCs([INFO-ONLY]) DIRT([INFO-ONLY])
JML([INFO-ONLY] Dirk journalized this) ; Sub-Collections: SRI-ARC
DPCS DIRT; Clerk: JML; Origin: < SRIFE, FINAL,NLS;1, >, 8-AUG-75
17:00 JML ;;;;###;

26260 Distribution

Ann Weinberg, Kenneth E. (Ken) Victor, Douglas C. Engelbart, James H. Bair, Elizabeth K. Michael, Richard W. Watson, Elizabeth J. Feinler, Harvey G. Lehtman, Kirk E. Kelley, Laura E. Gould, Jeanne M. Beck, Dirk H. Van Nouhuys, James C. Norton, Jeanne M. Leavitt, Dirk H. Van Nouhuys, Delorse M. Brooks, Richard W. Watson, Douglas C. Engelbart, Delorse M. Brooks, Elizabeth F. Finney, Beverly Boli, Joseph L. Ehardt, James H. Bair, Robert N. Lieberman, Pat Whiting O'Keefe, James H. Bair, Robert Louis Belleville, Ann Weinberg, Thomas L. Humphrey, Jeanne M. Leavitt, Kirk E. Kelley, Duane L. Stone, Elizabeth J. Feinler, N. Dean Meyer, Dirk H. Van Nouhuys, Douglas C. Engelbart, James C. Norton, Richard W. Watson, Charles H. Irby, Jonathan B. Postel, Priscilla A. Wold, Rita Hysmith, Pamela K. Allen, Delorse M. Brooks, Elizabeth F. Finney, Beverly Boli, Lawrence A. Crain, Kirk Sattley, Susan Gail Roetter, Robert N. Lieberman

People in Santa Cruze, Help Needs

Don and Dianna Cooper, 512 Du Jour st, (403)423-7846. He is a gardner and thinker; she is a paramedical office manager and poet. Jim Gaw, 1222 Laurel st., 426 1443, he is a research biochemist and mountain climber. He owns, by the way, that house and shares it ou, normally with outhere people at the university. I will drop him a card who you are. I just got to reading your journal item on Help needs and will comment further soon.

1

People in Santa Cruze, Help Needs

(J26263) 12-AUG-75 23:05;;; Title: Author(s): Dirk H. Van
Nouhuys/DVN; Distribution: /KIRK([INFO-ONLY]) ; Sub-Collections:
SRI-ARC; Clerk: DVN;

Smiles of Menlo Park

I just got to reading 26254, sounds like great stuff, I think it should be recorded for DPCS posterity, Why don't you send future ones to DPCS or &DPCS if you don't want all those people to get it?..Your pictures turned out well and will be in the division slide show, The pictures of Chauncy, by the way, turned out beautiful! Is there snow in the ski lodge?

1

Smiles of Menlo Park

(J26264) 12-AUG-75 23:35;;; Title: Author(s): Dirk H. van
Nouhuys/DVN; Distribution: /POOH([ACTION]); Sub-Collections:
SRI-ARC; Clerk: DVN;

recorded or unrecorded, that is the question

Elizabeth, the "unrecorded" message you sent us was RECORDED (26251)!
This feature ought to be fixed or not advertised. -- Charles,

1

recorded or unrecorded, that is the question

(J26265) 13-AUG-75 12:13;;; Title: Author(s): Charles H. Irby/CHI;
Distribution: /EKM([ACTION]) JBP([INFO-ONLY]) JCP([INFO-ONLY]
); Sub-Collections: SRI-ARC; Clerk: CHI;

Final Version of DPSJSYS

Includes the new features that were in the works when DPS was cut from NSW, Published for the record only; replaced by a more complete "programmer's guide".

Final Version of DPSJSYS

Introduction

1

This document describes the internal structure of a Distributed Programming System (DPS) process on Tenex. My apologies for the terseness of the document, but time and computer resources are scarce. The reader is assumed to have as background for this present offering, a thorough understanding of the several more verbose PCP documents which have preceded it. The primary purpose of this document is to present to process implementers the details of their interface with DPS. The services provided by DPS are, roughly speaking, a superset of those described in previous documents. Comments, bug/deficiency reports, and questions are welcome.

1a

Fork structure

2

A DPS process consists of a "controlling fork" (CF) containing the DPS implementation, and one or more "processing forks" (PFs) running beneath the CF, which contain user code. A PF requests the DPS services it needs by executing JSYS 400, which the CF intercepts via Tenex's JSYS trap facility and processes. A process' PFs are partitioned into one or more "subprocesses", the first called the "process leader" and created as part of the process' creation. The first PF of each subprocess is called the "subprocess leader" and is created as part of the subprocess' creation. All PFs within a subprocess execute copies of the same SAV file, and the CF routes incoming procedure calls to the appropriate subprocess and schedules its execution in an available PF within that subprocess.

2a

Operations

3

Three of the six low-level "operations" implemented by the CF and accessible via JSYS 400 -- IVdPS, RRdPS, and dRpPS -- provide the PFs with access to a whole set of DPS "virtual JSYSs" (VjSYSSs) implemented by the CF. Another two operations -- GTDPS and PTDPS -- provide the CF with access to a set of "virtual JUSRs" (VJUSRs; virtual jump-to-user's) implemented by the PFs. The fifth operation -- PGDPS -- provides a simple mechanism for VJUSR dispatch for "sequential processors" (SPs) which possess the following two characteristics: they are active only while executing VJUSRs, and they execute one VJUSR at a time.

3a

A PF declares itself to be a sequential processor by means of a flag bit in the S1PR VJSYS, and then invokes PGDPS. PGDPS blocks the SP until the first VJUSR is required. The SP executes the VJUSR whose number it finds in AC 0 (arguments in ACs 1-4), and then returns its results via PGDPS, which again blocks the processor. An SP need never issue either RDYPR or TSTEV/WAIEV.

3b

Procedures executed by SPs can be aborted, but cannot be interrupted. To abort a procedure invoked via the PECAL VJUSR, the CF interrupts the SP on a PSI channel specified via SIPR; the SP may either ignore the interrupt, or promptly make an abort return from PECAL. SPs need not supply the PEABR, PEINT, or PERSM VJUSRs,

3c

Some of the arguments and results of certain operations (and VJUSRs and VJSYSs) are stored in "blocks". A block is M+1 contiguous words of memory, of which the first contains a header (XWD M,L) and the next L, data. "ABC (x)s" stands for the Address of a Block CONTAINING zero or more x's (or exactly one, if "s" is absent). "ABF (x)s" stands for the Address of a Block FOR zero or more x's (or exactly one, if "s" is absent).

3d

IVDps (Op 0)
Invokes VJSys,

3e

ACCEPTS IN

3e1

0; XWD

3e1a

op [0],

3e1a1

ABC (

3e1a2

XWD

3e1a2a

event handle / 0 (meaning block)

3e1a2a1

(signalled upon completion of VJSYS with
completion code = XWD systemcall handle, VJSYS
number),

3e1a2a1a

VJSYS number

3e1a2a2

XWD

3e1a2b

call handle on whose behalf the VJSYS is being
executed) / 0

3e1a2b1

(meaning none),

3e1a2b1a

ABF (use by DPS in returning VJSYS results) / 0

3e1a2b2

(if not blocking operation))

3e1a2b2a

1-4: VJSYS arguments

3e1b

Final Version of DPSJSYS

RETURNS +	3e2
systemcall handle in 0	3e2a
1: unsuccessful,	3e2b
error number in 1,	3e2b1
byte pointer to ASCIZ diagnostic in 2	3e2b2
2: successful, VJSYS results in 1-4	3e2c
RRDPS (OP 1)	
Retrieves results of VJSYS,	3f
ACCEPTS IN	3f1
0: XWD op [1], systemcall handle	3f1a
1: ABF (use by DPS in returning VJSYS results)	3f1b
RETURNS +	3f2
1: unsuccessful,	3f2a
error number in 1,	3f2a1
byte pointer to ASCIZ diagnostic in 2	3f2a2
2: successful, VJSYS results in 1-4	3f2b
DRDPS (OP 2)	
Discards results of VJSYS,	3g
ACCEPTS IN	3g1
0: XWD op [2], systemcall handle	3g1a
RETURNS +	3g2
1: unsuccessful, error number in 1, 0 in 2	3g2a
2: successful	3g2b
GTDPDS (OP 3)	
Gets VJUSR arguments from DPS,	3h
ACCEPTS IN	3h1

Final Version of DPSJSYS

0: XWD op [3], usercall handle	3h1a
1: ABF (use by DPS in returning VJUSR arguments)	3h1b
RETURNS +	3h2
1: unsuccessful,	3h2a
error number in 1,	3h2a1
byte pointer to ASCIZ diagnostic in 2	3h2a2
2: successful,	3h2b
XWD	3h2b1
VJUSR number,	3h2b1a
requesting process handle / 0	3h2b1b
(meaning local DPS environment) in 0,	3h2b1b1
VJUSR arguments in 1-4	3h2b2
PTDPS (OP 4)	
Returns VJUSR results to DPS,	3i
ACCEPTS IN	3i1
0: XWD	3i1a
op [4],	3i1a1
ABC (3i1a2
XWD	3i1a2a
error code / 0 (meaning successful),	3i1a2a1
usercall handle)	3i1a2a2
1-4: VJUSR results	3i1b
(or, if error code specified,	
byte pointer to ASCIZ diagnostic in 1)	3i1b1
RETURNS +	3i2
1: unsuccessful, error number in 1, 0 in 2	3i2a

Final Version of DPSJSYS

2: successful	312b
PGDPS (OP 5)	
[Returns previous VJUSR's results to DPS and]	
gets next VJUSR's arguments from DPS,	3j
ACCEPTS IN	3j1
0: XWD	3j1a
op [5],	3j1a1
ABC (3j1a2
XWD	3j1a2a
error code / 0	3j1a2a1
(meaning previous VJUSR successful/nonexistent),	3j1a2a1a
ABF (use by DPS in returning next VJUSR's	
arguments))	3j1a2a2
1-4: previous VJUSR's results	3j1b
(or, if error code specified, byte pointer to ASCIZ	
diagnostic in 1)	3j1b1
RETURNS +	3j2
1: unsuccessful,	3j2a
error number in 1,	3j2a1
byte pointer to ASCIZ diagnostic in 2	3j2a2
2: successful	3j2b
XWD next VJUSR number, requesting process handle / 0	3j2b1
(meaning local DPS environment) in 0	3j2b1a
next VJUSR's arguments in 1-4	3j2b2
ERDPS (OP 6)	
Retrieves diagnostic message for DPS error,	3k
ACCEPTS IN	3k1

0: XWD op [6], error number	3k1a
1: ABF (use by DPS in returning diagnostic)	3k1b
RETURNS +	3k2
1: unsuccessful, error number in 1, 0 in 2	3k2a
2: successful, byte pointer to ASCIZ diagnostic in 1	3k2b
VJSYSs for manipulating remote processes	4
processes	4a
CRTPS (VJSYS 1) Creates remote process,	4a1
ACCEPTS IN	4a1a
1: byte pointer to ASCIZ process address	4a1a1
2: XWD	4a1a2
ABC (PCPB36 startup info) / 0	4a1a2a
(meaning EMPTY),	4a1a2a1
ABC (byte pointers to	4a1a2b
ASCIZ user name, password, and account)	4a1a2b1
3: XWD	4a1a3
package scope / 0 (meaning open no packages),	4a1a3a
process scope	4a1a3b
4: XWD	4a1a4
ABC (4a1a4a
ABC (PCPB36 package startup info) / 0	4a1a4a1
(meaning EMPTY))s	4a1a4a1a
/ 0 (meaning all EMPTY),	4a1a4a2
ABC (byte pointer to ASCIZ package name)s / 0	4a1a4b

Final Version of DPSJSYS

(if no packages to be opened)	4a1a4b1
RETURNS IN	4a1b
1: XWD ABC (Package handle)s, process handle	4a1b1
DELPS (VJSYS 2)	
Deletes previously created remote process,	4a2
ACCEPTS IN	4a2a
1: process handle / 0 (meaning all)	4a2a1
RETURNS IN	4a2b
1: cost in cents	4a2b1
ITDPS (VJSYS 3)	
Introduces two remote processes to one another,	4a3
ACCEPTS IN	4a3a
1: XWD	4a3a1
ABC (PCPB36 startup info 1) / 0	4a3a1a
(meaning EMPTY),	4a3a1a1
process handle 1	4a3a1b
2: XWD	4a3a2
ABC (PCPB36 startup info 2) / 0	4a3a2a
(meaning EMPTY),	4a3a2a1
process handle 2	4a3a2b
3: XWD	4a3a3
flags,	4a3a3a
B0 on: logical channel only	4a3a3a1
scope	4a3a3b
RETURNS IN	4a3b
1: XWD	4a3b1

<u>flags</u>	4a3b1a
B0 on: physical channel established,	4a3b1a1
introduction handle	4a3b1b
2: XWD ph12, ph21	4a3b2
SEPPS (VJSYS 4)	
Separates two previously introduced remote processes,	4a4
ACCEPTS IN	4a4a
1: introduction handle / 0 (meaning all)	4a4a1
RETURNS IN	4a4b
1: cost 1 in cents	4a4b1
2: cost 2 in cents	4a4b2
INFPs (VJSYS 52)	
Retrieves information about a remote process,	4a5
ACCEPTS IN	4a5a
1: XWD information type, process handle	4a5a1
RETURNS IN	4a5b
1: information	4a5b1
Packages	4b
OPNPK (VJSYS 5)	
Opens remote packages,	4b1
ACCEPTS IN	4b1a
1: XWD scope, process handle	4b1a1
2: XWD	4b1a2
ABC (4b1a2a
ABC (PCPB36 startup info) / 0	4b1a2a1
(meaning EMPTY)	4b1a2a1a

Final Version of DPSJSYS

)s / 0 (meaning all EMPTY),	4b1a2a2
ABC (byte pointer to ASCIZ package name)s	4b1a2b
RETURNS IN	4b1b
1: ABC (package handle)s	4b1b1
CLSPK (VJSYS 6)	
Closes previously opened remote packages,	4b2
ACCEPTS IN	4b2a
1: XWD process handle, ABC (package handle)s	4b2a1
RETURNS IN	4b2b
1: ABC (cost in cents)s	4b2b1
Procedures	4c
CALPE (VJSYS 7)	
Calls remote procedure,	4c1
ACCEPTS IN	4c1a
1: XWD	4c1a1
ABC (PCPB36 result list mask) / 0	4c1a1a
(meaning LIST (INDEX [CALLER])),	4c1a1a1
addr of Tenex-format procedure selector	4c1a1b
2: XWD	4c1a2
ABC (PCPB36 argument list mask) / 0	4c1a2a
(meaning LIST (INDEX [CALLER])),	4c1a2a1
ABC (ABC (PCPB36 argument) / 0 (meaning EMPTY))s / 0	4c1a2b
(meaning none)	4c1a2b1
3: priority	4c1a3
RETURNS IN	4c1b
1: XWD	4c1b1

Final Version of DPSJSYS

outcome,	4c1b1a
ABC (ABC (PCPB36 result) / 0 (meaning EMPTY))s / 0	4c1b1b
(meaning none)	4c1b1b1
2: cost in cents	4c1b2
VISPE (VJSYS 10)	
Visits remote callee/caller,	4c2
ACCEPTS IN	4c2a
1: XWD	4c2a1
ABC (PCPB36 result list mask) / 0	4c2a1a
(meaning LIST (INDEX [CALLER])),	4c2a1a1
call handle	4c2a1b
2: XWD	4c2a2
ABC (PCPB36 argument list mask) / 0	4c2a2a
(meaning LIST (INDEX [CALLER])),	4c2a2a1
ABC (ABC (PCPB36 argument) / 0 (meaning EMPTY))s / 0	4c2a2b
(meaning none)	4c2a2b1
RETURNS IN	4c2b
1: XWD	4c2b1
outcome,	4c2b1a
ABC (ABC (PCPB36 result) / 0 (meaning EMPTY))s / 0	4c2b1b
(meaning none)	4c2b1b1
ALOCH (VJSYS 11)	
Allocates call handle for remote procedure call,	4c3
ACCEPTS IN	4c3a
1: XWD	4c3a1
priority,	4c3a1a

Final Version of DPSJSYS

addr of Tenex-format procedure selector	4c3a1b
RETURNS IN	4c3b
1: call handle	4c3b1
RELCH (VJSYS 12) [Aborts remote callee and] releases call handle.	4c4
ACCEPTS IN	4c4a
1: call handle / 0 (meaning all)	4c4a1
RETURNS IN	4c4b
1: cost in cents	4c4b1
ACQPE (VJSYS 13) Acquires control from remote callee/caller.	4c5
ACCEPTS IN	4c5a
1: call handle	4c5a1
RETURNS IN	4c5b
1: XWD	4c5b1
outcome,	4c5b1a
ABC (ABC (PCPB36 result) / 0 (meaning EMPTY))s / 0	4c5b1b
(meaning none)	4c5b1b1
RELPE (VJSYS 14) Releases control to remote callee/caller.	4c6
ACCEPTS IN	4c6a
1: XWD	4c6a1
ABC (PCPB36 result list mask) / 0	4c6a1a
(meaning LIST (INDEX [CALLER])),	4c6a1a1
call handle	4c6a1b
2: XWD	4c6a2

Final Version of DPSJSYS

ABC (PCPB36 argument list mask) / 0	4c6a2a
(meaning LIST (INDEX [CALLER])),	4c6a2a1
ABC (ABC (PCPB36 argument) / 0 (meaning EMPTY))s / 0	4c6a2b
(meaning none)	4c6a2b1
3: event handle	4c6a3
(signalled upon return of remote procedure with completion code = XWD call handle, outcome)	4c6a3a
SIGPE (VJSYS 56)	
Signals remote callee/caller.	4c7
ACCEPTS IN	4c7a
1: call handle	4c7a1
2: XWD	4c7a2
ABC (PCPB36 argument list mask) / 0	4c7a2a
(meaning LIST (INDEX [CALLER])),	4c7a2a1
ABC (ABC (PCPB36 argument) / 0 (meaning EMPTY))s / 0	4c7a2b
(meaning none)	4c7a2b1
INTPE (VJSYS 15)	
Interrupts remote callee.	4c8
ACCEPTS IN	4c8a
1: call handle / 0 (meaning all)	4c8a1
RSMPE (VJSYS 16)	
Resumes previously interrupted remote callee.	4c9
ACCEPTS IN	4c9a
1: call handle / 0 (meaning all)	4c9a1
NTEPE (VJSYS 17)	
Makes event known to remote caller.	4c10
ACCEPTS IN	4c10a

Final Version of DPSJSYS

```

1: XWD 4c10a1
      ABC (PCPB36 event description) / 0 4c10a1a
      (meaning EMPTY), 4c10a1a1
      event code 4c10a1b
HLPPE (VJSYS 20)
Solicits help from remote caller, 4c11
ACCEPTS IN 4c11a
1: XWD 4c11a1
      ABC (PCPB36 problem description) / 0 4c11a1a
      (meaning EMPTY), 4c11a1a1
      problem code 4c11a1b
RETURNS 4c11b
1: ABC (PCPB36 solution) / 0 (meaning EMPTY) 4c11b1
Data Stores 4d
CRTDT (VJSYS 21)
Creates remote data store, 4d1
ACCEPTS IN 4d1a
1: XWD 4d1a1
      scope, 4d1a1a
      addr of Tenex-format data store selector 4d1a1b
2: ABC (PCPB36 initial value) / 0 4d1a2
      (meaning EMPTY) 4d1a2a
DELDT (VJSYS 22)
Deletes previously created remote data store, 4d2
ACCEPTS IN 4d2a
1: addr of Tenex-format data store selector 4d2a1

```

Final Version of DPSJSYS

RDDT (VJSYS 23)	
Reads remote data store,	4d3
ACCEPTS IN	4d3a
1: addr of Tenex-format data store selector	4d3a1
RETURNS IN	4d3b
1: ABC (PCPB36 value) / 0 (meaning EMPTY)	4d3b1
WRDT (VJSYS 24)	
Writes remote data store,	4d4
ACCEPTS IN	4d4a
1: XWD	4d4a1
ABC (PCPB36 value) / 0 (meaning EMPTY),	4d4a1a
addr of Tenex-format data store selector	4d4a1b
LCKDT (VJSYS 25)	
Locks remote data store,	4d5
ACCEPTS IN	4d5a
1: XWD	4d5a1
scope,	4d5a1a
addr of Tenex-format data store selector	4d5a1b
2: XWD	4d5a2
flags,	4d5a2a
B0 on: abort if lock not settable immediately	4d5a2a1
lock type	4d5a2b
RETURNS IN	4d5b
1: datalock handle	4d5b1
ULKDT (VJSYS 26)	
Unlocks previously locked remote data store,	4d6
ACCEPTS IN	4d6a

Final Version of DPSJSYS

1: XWD	4d6a1
dataLock handle,	4d6a1a
addr of Tenex-format data store selector	4d6a1b
Channels	4e
CRTCH (VJSYS 27)	
Creates channel between two remote processes,	4e1
ACCEPTS IN	4e1a
1: XWD process handle 1, process handle 2	4e1a1
2: scope	4e1a2
RETURNS IN	4e1b
1: channel handle	4e1b1
2: XWD port handle 1, port handle 2	4e1b2
DELCH (VJSYS 30)	
Deletes previously created channel between two remote processes,	4e2
ACCEPTS IN	4e2a
1: channel handle / 0 (meaning all)	4e2a1
Debugging	4f
SETRC (VJSYS 61)	
Sets remote DPS trace word,	4f1
ACCEPTS IN	4f1a
1: process handle	4f1a1
2: trace word value	4f1a2
Bit 0 on: trace incoming inter-process messages	4f1a2a
Bit 1 on: trace outgoing inter-process messages	4f1a2b
Bit 2 on: trace inter-process messages to self	4f1a2c
Bit 3 on: trace VJSYS arguments	4f1a2d

Final Version of DPSJSYS

Bit 4 on:	trace VJSYS results	4f1a2e
Bit 5 on:	trace vJSyS aborts	4f1a2f
Bit 6 on:	trace VJUSR arguments	4f1a2g
Bit 7 on:	trace VJUSR results	4f1a2h
Bit 8 on:	trace VJUSR aborts	4f1a2i
Bit 9 on:	trace internal interlock activity	4f1a2j
Bit 10 on:	trace DPS-detected errors	4f1a2k

VJSYSS for manipulating local process	5
---------------------------------------	---

Subprocesses	5a
--------------	----

CRTSP (VJSYS 31)	
Creates local subprocess,	5a1

ACCEPTS IN	5a1a
------------	------

1: byte pointer to ASCIZ subprocess address	5a1a1
---	-------

2: XWD	5a1a2
--------	-------

scope,	5a1a2a
--------	--------

ABC (pCpB36 startup info) / 0	5a1a2b
-------------------------------	--------

(meaning EMPTY)	5a1a2b1
-----------------	---------

3: priority	5a1a3
-------------	-------

RETURNS IN	5a1b
------------	------

1: subprocess handle	5a1b1
----------------------	-------

DELSP (VJSYS 32)	
Deletes previously created local subprocess,	5a2

ACCEPTS IN	5a2a
------------	------

1: subprocess handle / 0 (meaning all)	5a2a1
--	-------

RETURNS IN	5a2b
------------	------

1: cost in cents	5a2b1
------------------	-------

Final Version of DPSJSYS

Processors	5b
CRTPR (VJSYS 33) Creates local processor.	5b1
ACCEPTS IN	5b1a
1: XWD scope, subprocess handle	5b1a1
2: XWD	5b1a2
ABC (PCPB36 startup info) / 0 (meaning EMPTY),	5b1a2a
priority	5b1a2b
RETURNS IN	5b1b
1: processor handle	5b1b1
DELPR (VJSYS 34) Deletes local processor.	5b2
ACCEPTS IN	5b2a
1: processor handle / 0	5b2a1
(meaning all within subprocess but leader)	5b2a1a
RETURNS IN	5b2b
1: cost in cents	5b2b1
SIPR (VJSYS 35) Signs in local processor.	5b3
ACCEPTS IN	5b3a
1: byte pointer to ASCIZ process name	5b3a1
(ignored except from first process-leader processor)	5b3a1a
2: XWD	5b3a2
flags	5b3a2a
B0 on: auto processor creation	5b3a2a1
(CF to create/delete processors as required;	

	ignored except from first process=leader processor)	5b3a2a1a
B1 on:	sequential processor	5b3a2a2
	(processor will use the PGDPS operation as its VJUSR dispatch mechanism)	5b3a2a2a
B2 on:	auto ready	5b3a2a3
	(CF will simulate a call to RDYPR after SIPR and after each PTDPS; processor need never invoke RDYPR explicitly)	5b3a2a3a
B3 on:	splicable process	5b3a2a4
	(CF will accept attempts by remote processes to splice to local process; ignored except from process leader)	5b3a2a4a
B4 on:	encapsulated subprocess	5b3a2a5
	(CF with encapsulate subprocess; ignored except from subprocess leader)	5b3a2a5a
ABC	(byte pointer to ASCIZ package name)s / 0	5b3a2b
	(meaning none; list index serves as an "internal package handle")	5b3a2b1
3:	QWD	5b3a3
	first page of subprocess=global storage	5b3a3a
	(ignored except from subprocess leader),	5b3a3a1
	last page of subprocess=global storage	5b3a3b
	(ignored except from subprocess leader; first greater than last implies none),	5b3a3b1
	0,	5b3a3c
	PSI channel / -1 (meaning none)	5b3a3d
	(either for VJUSR request event, or, for sequential processors, to abort a procedure)	5b3a3d1
RETURNS IN		5b3b

Final Version of DPSJSYS

1: XWD	5b3b1
ABC (pCpB36 [sub]process[or] startup info) / 0	5b3b1a
(meaning EMPTY),	5b3b1a1
event handle / 0 (if sequential processor)	5b3b1b
(signalled by CF to request a VJUSR with completion code = XWD usercall handle, VJUSR number),	5b3b1b1
2: flags	5b3b2
B0 on: local process is at root of tree	5b3b2a
B1 on: local subprocess is process leader	5b3b2b
B2 on: local processor is subprocess leader	5b3b2c
SOPR (VJSYS 53)	
Signs out local processor/subprocess/process,	5b4
RDYPR (VJSYS 36)	
Readys local processor for next service request (INIPK / TRMPK / PECAL / LRDDT / LWRDT),	5b5
ITDFK (VJSYS 54)	
Introduces fork to DPS,	5b6
ACCEPTS IN	5b6a
1: XWD	5b6a1
ABC (PCPB36 startup info) / 0 (meaning EMPTY),	5b6a1a
fork handle	5b6a1b
RETURNS IN	5b6b
1: processor handle	5b6b1
SEPFK (VJSYS 55)	
Separates fork from DPS,	5b7
ACCEPTS IN	5b7a
1: processor handle / 0	5b7a1
(meaning all introduced by invoking processor)	5b7a1a

Final Version of DPSJSYS

Channels	5c
SNDCH (VJSYS 37)	
Outputs portion of PCPB36 data structure on local channel.	5c1
ACCEPTS IN	5c1a
1: XWD ABC (portion), port handle	5c1a1
RCVCH (VJSYS 40)	
Inputs next portion of PCPB36 data structure from local channel.	5c2
ACCEPTS IN	5c2a
1: port handle	5c2a1
RETURNS IN	5c2b
1: ABC (portion)	5c2b1
Locks	5d
CRTLK (VJSYS 41)	
Creates local lock.	5d1
ACCEPTS IN	5d1a
1: scope	5d1a1
RETURNS IN	5d1b
1: lock handle	5d1b1
DELLK (VJSYS 42)	
Deletes local lock.	5d2
ACCEPTS IN	5d2a
1: lock handle / 0 (meaning all)	5d2a1
SETLK (VJSYS 43)	
Sets local lock.	5d3
ACCEPTS IN	5d3a
1: XWD	5d3a1
scope (value ALL illegal),	5d3a1a

Final Version of DPSJSYS

lock handle	5d3a1b
2: XWD	5d3a2
flags,	5d3a2a
B0 on: abort if lock not settable immediately	5d3a2a1
lock type	5d3a2b
RETURNS IN	5d3b
1: lockset handle	5d3b1
REMLK (VJSYS 44) Unsets local lock,	5d4
ACCEPTS IN	5d4a
1: XWD	5d4a1
lock handle,	5d4a1a
lockset handle	5d4a1b
Events	5e
CRTEV (VJSYS 45) Creates local event,	5e1
ACCEPTS IN	5e1a
1: XWD	5e1a1
scope (value ALL illegal),	5e1a1a
PSI channel to be interrupted when event signalled / =1	5e1a1b
(meaning none)	5e1a1b1
2: max length	5e1a2
RETURNS IN	5e1b
1: event handle	5e1b1
DELEV (VJSYS 46) Deletes local event,	5e2

Final Version of DPSJSYS

ACCEPTS IN	5e2a
1: event handle / 0 (meaning all)	5e2a1
SIGEV (VJSYS 47) Signals a local event.	5e3
ACCEPTS IN	5e3a
1: event handle	5e3a1
2: completion code (non-zero)	5e3a2
TSIEV (VJSYS 50) Tests for and clears signalled local event.	5e4
ACCEPTS IN	5e4a
1: event handle	5e4a1
RETURNS IN	5e4b
1: completion code / 0 (meaning unsignalled)	5e4b1
2: new length	5e4b2
WAI EV (VJSYS 51) Waits for to be signalled and clears [any] one of a list of local events.	5e5
ACCEPTS IN	5e5a
1: ABC (event handle)s	5e5a1
RETURNS IN	5e5b
1: XWD	5e5b1
block offset to left-most signalled event handle,	5e5b1a
new total length	5e5b1b
2: completion code for left-most signalled event	5e5b2
Timers	5f
SETMR (VJSYS 57) Sets interval timer.	5f1

ACCEPTS IN	5f1a
1: interval in ms	5f1a1
2: XWD scope, event handle	5f1a2
(signalled upon expiration of timer with completion code = XWD timer handle, interval)	5f1a2a
RETURNS IN	5f1b
1: timer handle	5f1b1
TSTMR (VJSYS 60) Tests [and cancels] previously set interval timer.	5f2
ACCEPTS IN	5f2a
1: XWD flags, timer handle	5f2a1
B0 on: cancel timer	5f2a1a
RETURNS IN	5f2b
1: ms gone	5f2b1
2: ms left	5f2b2
VJUSRs implemented by (every processor in) every subprocess	6
Processors	6a
PRSO (VJUSR 1) Solicits signout of local processor/subprocess/process.	6a1
Packages	6b
INIPK (VJUSR 2) Initializes local package for subprocess.	6b1
ACCEPTS IN	6b1a
1: internal package handle	6b1a1
RETURNS IN	6b1b
1: package version number	6b1b1

TRMPK (VJUSR 3)	
Terminates local package for subprocess,	6b2
ACCEPTS IN	6b2a
1: internal package handle	6b2a1
Procedures	6c
PECAL (VJUSR 4)	
Calls local procedure on behalf of remote caller,	6c1
ACCEPTS IN	6c1a
1: internal package handle	6c1a1
2: byte pointer to ASCIZ procedure name	6c1a2
3: XWD	6c1a3
call handle,	6c1a3a
ABC (6c1a3b
ABC (PCPB36 argument) / 0 (meaning EMPTY)	6c1a3b1
)s / 0 (meaning none)	6c1a3b2
RETURNS IN	6c1b
1: XWD	6c1b1
outcome,	6c1b1a
ABC (6c1b1b
ABC (PCPB36 result) / 0 (meaning EMPTY)	6c1b1c
)s / 0 (meaning none)	6c1b1d
PEINT (VJUSR 5)	
Interrupts previously called local procedure on behalf of remote caller,	6c2
ACCEPTS IN	6c2a
1: call handle	6c2a1
PERSM (VJUSR 6)	

Final Version of DPSJSYS

Resumes previously interrupted local procedure on behalf of remote caller,	6c3
ACCEPTS IN	6c3a
1: call handle	6c3a1
PEABR (VJUSR 7) Aborts previously called local procedure on behalf of remote caller,	6c4
ACCEPTS IN	6c4a
1: call handle	6c4a1
PENTE (VJUSR 10) Makes event detected by remote callee known to local caller,	6c5
ACCEPTS IN	6c5a
1: XWD	6c5a1
call handle for local caller,	6c5a1a
call handle for remote callee	6c5a1b
2: XWD	6c5a2
ABC (PCPB36 event description) / 0 (meaning EMPTY),	6c5a2a
event code	6c5a2b
pEHLp (VJUSR 11) Solicits help from local caller on behalf of remote callee,	6c6
ACCEPTS IN	6c6a
1: XWD	6c6a1
call handle for local caller,	6c6a1a
call handle for remote callee	6c6a1b
2: XWD	6c6a2
ABC (PCPB36 problem description) / 0 (meaning EMPTY),	6c6a2a
problem code	6c6a2b

RETURNS IN	6c6b
1: ABC (PCPB36 solution) / 0 (meaning EMPTY)	6c6b1
Data Stores	6d
LVRDT (VJUSR 12) Verifies existence of local data store,	6d1
ACCEPTS IN	6d1a
1: internal package handle	6d1a1
2: byte pointer to ASCIZ data store name	6d1a2
LRDDT (VJUSR 13) Reads local data store on behalf of remote process,	6d2
ACCEPTS IN	6d2a
1: XWD	6d2a1
ABC (PCPB36 element selector) / 0	6d2a1a
(meaning whole data store),	6d2a1a1
internal package handle	6d2a1b
2: byte pointer to ASCIZ data store name	6d2a2
RETURNS IN	6d2b
1: ABC (PCPB36 value) / 0 (meaning EMPTY)	6d2b1
LWRDT (VJUSR 14) Writes local data store on behalf of remote process,	6d3
ACCEPTS IN	6d3a
1: XWD	6d3a1
ABC (PCPB36 element selector) / 0	6d3a1a
(meaning whole data store),	6d3a1a1
internal package handle	6d3a1b
2: byte pointer to ASCIZ data store name	6d3a2

Final Version of DPSJSYS

3: ABC (PCPB36 value) / 0 (meaning EMPTY) 6d3a3

VJUSRs implemented by process leader 7

Processes 7a

OKIPS (VJUSR 15)
OKs introduction of remote process to local process. 7a1

ACCEPTS IN 7a1a

1: XWD 7a1a1

ABC (PCPB36 startup info) / 0 7a1a1a

(meaning EMPTY), 7a1a1a1

new process handle 7a1a1b

OKSPS (VJUSR 16)
OKs separation from local process of previously introduced
remote process. 7a2

ACCEPTS IN 7a2a

1: old process handle 7a2a1

OKLPS (VJUSR 24)
OKs splicing of remote process to local process. 7a3

ACCEPTS IN 7a3a

1: XWD 7a3a1

ABC (PCPB36 startup info) / 0 7a3a1a

(meaning EMPTY), 7a3a1a1

new process handle 7a3a1b

2: byte pointer to ASCIZ user name 7a3a2

OKUPS (VJUSR 25)
OKs unsplicing from local process of previously spliced remote
process. 7a4

ACCEPTS IN 7a4a

1: old process handle 7a4a1

Final Version of DPSJSYS

Packages	7b
OKOPK (VJUSR 17) OKs opening of [and initializes] local package by remote process.	7b1
ACCEPTS IN	7b1a
1: XWD scope, new package handle	7b1a1
2: byte pointer to ASCIZ package name	7b1a2
3: XWD	7b1a3
internal package handle (meaning INIPK too) / 0,	7b1a3a
ABC (PCPB36 startup info) / 0 (meaning EMPTY)	7b1a3b
RETURNS IN	7b1b
1: package version number / 0	7b1b1
(if no internal package handle specified)	7b1b1a
OKCPK (VJUSR 20) OKs closing of [and terminates] local package by remote process.	7b2
ACCEPTS IN	7b2a
1: XWD	7b2a1
internal package handle (meaning TRMPK too) / 0,	7b2a1a
old package handle	7b2a1b
2: byte pointer to ASCIZ package name	7b2a2
Channels	7c
OKCCH (VJUSR 21) OKs creation of channel to local process.	7c1
ACCEPTS IN	7c1a
1: new port handle	7c1a1
OKDCH (VJUSR 22) OKs deletion of previously created channel to local process.	7c2

Final Version of DPSJSYS

ACCEPTS IN	7c2a
1: old port handle	7c2a1
NTLCH (VJUSR 23)	
Notes loss of channel to remote process.	7c3
ACCEPTS IN	7c3a
1: XWD	7c3a1
flags,	7c3a1a
BO on: process, rather than port handle	7c3a1a1
handle	7c3a1b
Data Type Assignments	8
Argument list mask LIST (INDEX [CALLER=1] / DSELECTOR*, ...)	8a
Code INDEX	8b
(event, problem, error)	8b1
Completion code INTEGER (non-zero)	8c
Cost INTEGER	8d
Data store selector	8e
LIST (%ph% INDEX, %pkh% INDEX, %data store% CHARSTR, %element% ESELECTOR*)	8e1
Depth INTEGER	8f
diagnostic CHARSTR	8g
Element selector	8h
LIST (<BOOLEAN [KEY=TRUE / INDEX=FALSE]> %element% any/INDEX, ...)	8h1
Handle INDEX	8i
(systemcall, usercall, process [SELF=1/SUPER=2], subprocess [SELF=1/LEADER=2], processor [SELF=1/LEADER=2], package, internal package, call, introduction, channel, port, lock, lockset, datalock, event, timer)	8i1

Final Version of DPSJSYS

Lock type	INDEX [SHARE=1/EXCLUSIVE=2]	8j
Login parameter	CHARSTR	8k
	(user, password, account)	8k1
Name	CHARSTR	8l
	(process, package, data store)	8l1
Number	INDEX	8m
	(VJSYS, VJUSR)	8m1
Outcome	INDEX [VISIT=1 / SUCCESS=2 / FAILURE=3 / SIGNAL=4]	8n
Priority	INDEX	8o
Procedure selector		8p
	LIST (%ph% INDEX, %pkh% INDEX, %pname% CHARSTR)	8p1
Process address	CHARSTR	8q
	<action> [<SP> <host address>] <SP> <intrahost address>	8q1
	Action is either "CRT", meaning create a new process, or "SPL", meaning splice to an existing process.	8q1a
	Host address is a decimal host addr or standard host name (defaulting to that of the local host).	8q1b
	Intrahost address is a SAV filename on Tenex (for CRT), or a decimal ICP contact socket number (for SPL).	8q1c
Process information type	INDEX [HOSTADDR=1]	8r
Result list mask	LIST (INDEX [CALLER=1/DISCARD=2] / DSELECTOR*, ...)	8s
Scope	INDEX [PROCESSOR=1/SUBPROCESS=2/PROCESS=3/ALL=4]	8t
Startup info	any	8u
Subprocess address	CHARSTR	8v
	<intrahost address>	8v1
PCPB36 Data Structure Format		9

Final Version of DPSJSYS

Bit	0 If set, key data structure follows	9a
Bits	1-13 Unused (zero)	9b
Bits	14-17 Data type	9c
	EMPTY =1 INTEGER=4 LIST=7	9c1
	BOOLEAN=2 BITSTR =5	9c2
	INDEX =3 CHARSTR=6	9c3
Bits	18-20 Unused (zero)	9d
Bits	21-35 Value or its length	9e
	EMPTY unused (zero)	9e1
	BOOLEAN 14 zero-bits + 1-bit value (TRUE=1 / FALSE=0)	9e2
	INDEX unsigned value	9e3
	INTEGER unused (zero)	9e4
	BITSTR unsigned bit count	9e5
	CHARSTR unsigned character count	9e6
	LIST unsigned element count	9e7
Bits	36-?? Value	9f
	EMPTY unused (nonexistent)	9f1
	BOOLEAN unused (nonexistent)	9f2
	INDEX unused (nonexistent)	9f3
	INTEGER two's complement full-word	9f4
	BITSTR bit string + zero padding to word boundary	9f5
	CHARSTR ASCII string + zero padding to word boundary	9f6
	LIST element data structures	9f7
	Tenex Data Structure Formats	10
	Procedure selector	10a

Final Version of DPSJSYS

Block containing process handle, package handle, and byte pointer to ASCIZ procedure name	10a1
Data store selector	10b
Block containing process handle, package handle, byte pointer to ASCIZ procedure name, and ABC (PCPB36 element selector) or zero (meaning whole data structure)	10b1
Error Messages	11
Below are listed the errors currently detected and reported by DPS-10: their mnemonics within the DPS source code, their numbers in decimal, and their associated diagnostic messages. Error numbers greater than 1000 may occasionally be reported, and are generated by DPS' L10 runtime environment.	11a
%data structures%	11b
eddst = 1 "?."	11b1
edkey = 2 "Duplicate key."	11b2
eddrky = 3 "Key has key."	11b3
eddsto = 4 "Duplicate data store name."	11b4
edlidx = 5 "Illegal INDEX."	11b5
edipdl = 6 "Illegal PSEL/DSEL."	11b6
edluif = 7 "Illegal USERINFO."	11b7
edmkey = 8 "Missing key."	11b8
edodcl = 9 "LIST too long to decode."	11b9
edolst = 10 "Maximum LIST size exceeded."	11b10
edostr = 11 "CHRSTR too long to decode."	11b11
edufty = 12 "Undefined data type."	11b12
eduity = 13 "Undefined informal data type."	11b13
edusto = 14 "Undefined data store name."	11b14
edwesl = 15 "Non-LIST addressed by ESEL."	11b15

Final Version of DPSJSYS

edwidx = 16	"No such index,"	11b16
edwkey = 17	"No such key,"	11b17
edwtyp = 18	"Wrong data type,"	11b18
edwpmc = 19	"Incorrect number of parameters,"	11b19
edoabc = 20	"Data structure overflows source block,"	11b20
%errors%		11c
eefimp = 51	"Not implemented,"	11c1
eefops = 52	"Operating system error,"	11c2
eemerr = 53	"Unidentifiable operating system error,"	11c3
eeuern = 54	"Undefined error number,"	11c4
eefl10 = 55	"L10 run-time error,"	11c5
%events%		11d
evfacq = 101	"Won't delete ALOCH event,"	11d1
evfjuv = 102	"Won't delete SIPR event,"	11d2
evilen = 103	"Illegal event length,"	11d3
evoecb = 104	"Event overflow,"	11d4
evftmr = 105	"Won't delete SETMR event,"	11d5
%folders%		11e
efodr_n = 151	"Won't create record while folder drained,"	11e1
eforun = 152	"RUNFLD overrun,"	11e2
efuift = 153	"Undefined record information type,"	11e3
efurop = 154	"Undefined RUNFLD operation,"	11e4
%inter-process communication%		11f
ecdcc = 201	"Channel already created,"	11f1
ecutyp = 202	"Undefined channel type,"	11f2

Final Version of DPSJSYS

ecwmnu = 203	"Channel type menu mismatch,"	11f3
ecwpkl = 204	"Inconsistent packet length,"	11f4
%locks%		11g
elfded = 251	"Deadlock,"	11g1
elfdel = 252	"Sought lock deleted,"	11g2
elflck = 253	"Lock attempt failed,"	11g3
elmswp = 254	"Non-existent LCB to be swapped,"	11g4
elmstk = 255	"Lock stack underflow,"	11g5
elostk = 256	"Lock stack overflow,"	11g6
elistk = 257	"Lock stack surplus,"	11g7
%packages%		11h
ekfded = 301	"Package dead,"	11h1
ekupkn = 302	"Undefined package,"	11h2
%procedures%		11i
epfnoh = 351	"No help available,"	11i1
epfpio = 352	"No processor with sufficient priority,"	11i2
epfsab = 353	"Won't abort system procedure,"	11i3
epfsin = 354	"Won't interrupt system procedure,"	11i4
epiacq = 355	"Context prohibits ACQPE,"	11i5
epiaid = 356	"Context prohibits any action by local procedure,"	11i6
epihlp = 357	"Context prohibits HLPPE,"	11i7
epint = 358	"Context prohibits INTPE,"	11i8
epimsk = 359	"Illegal argument/result list mask,"	11i9
epinte = 360	"Context prohibits NTEPE,"	11i10

Final Version of DPSJSYS

epiote = 361	"Illegal system procedure outcome,"	11111
epirel = 362	"Context prohibits RELPE,"	11112
epirms = 363	"Context prohibits RSMPE,"	11113
epixhp = 364	"Context prohibits call to XHLPPE,"	11114
epixin = 365	"Context prohibits call to XINTPE,"	11115
epixnt = 366	"Context prohibits sending XNTEPE,"	11116
epixrc = 367	"Context prohibits sending XRECPE,"	11117
epixrm = 368	"Context prohibits call to XRSMPPE,"	11118
epixrn = 369	"Context prohibits sending XRTNPE,"	11119
epuote = 370	"Undefined procedure outcome,"	11120
epurtn = 371	"Undefined return type,"	11121
epusyn = 372	"Undefined system procedure number,"	11122
epwvis = 373	"Unplanned for visit,"	11123
epfqin = 374	"Procedure can't be interrupted,"	11124
episig = 375	"Context prohibits SIGPE,"	11125
epixsg = 376	"Context prohibits call to XSIGPE,"	11126
%processes%		11j
esdpoh = 401	"POH already associated with process,"	11j1
esfded = 402	"Process dead,"	11j2
esipsa = 403	"Syntax error in process addr,"	11j3
esircy = 404	"No POH via which to receive message,"	11j4
esisup = 405	"Not direct superior,"	11j5
esmpoh = 406	"No POH via which to send message,"	11j6
esumsg = 407	"Undefined message number,"	11j7
esuser = 408	"Undefined user name,"	11j8

esuwt	= 409	"Undefined watchdog code,"	11j9
eswqak	= 410	"?,"	11j10
%processors%			11k
erigtd	= 451	"Context prohibits GTDPS,"	11k1
eriptd	= 452	"Context prohibits PTDPS,"	11k2
erisin	= 453	"Not signed in,"	11k3
erorsb	= 454	"ABF overflow,"	11k4
erosml	= 455	"Small block overflow,"	11k5
erowin	= 456	"Processor window overflow,"	11k6
erualo	= 457	"Undefined entity type allocated,"	11k7
eruinf	= 458	"Undefined process information type,"	11k8
eruopn	= 459	"Undefined operation number,"	11k9
erupml	= 460	"Undefined parameter location,"	11k10
erupsi	= 461	"Undefined PSI channel number,"	11k11
erurde	= 462	"Undefined entity type read,"	11k12
erusc	= 463	"Undefined scope,"	11k13
erusc	= 464	"Undefined system call number,"	11k14
eruusc	= 465	"Undefined user call number,"	11k15
eruwre	= 466	"Undefined entity type written,"	11k16
erfded	= 467	"Processor dead,"	11k17
erdsin	= 468	"Processor already signed in,"	11k18
%storage%			11l
emfexh	= 501	"CF storage exhausted,"	11l1
emient	= 502	"Negative entity size,"	11l2
emuent	= 503	"Undefined entity type,"	11l3

Final Version of DPSJSYS

```

emibyp = 504 "Illegal user-supplied byte pointer," 11l4
emiadr = 505 "Illegal user-supplied address," 11l5
%undefined handles% 11m
egmhca = 551 "Undefined call handle," 11m1
egmhcn = 552 "Undefined channel handle," 11m2
egmhdt = 553 "Undefined data store handle," 11m3
egmhev = 554 "Undefined event handle," 11m4
egmhlk = 555 "Undefined lock handle," 11m5
egmhls = 556 "Undefined lockset handle," 11m6
egmhmg = 557 "Undefined manager handle," 11m7
egmhpk = 558 "Undefined package handle," 11m8
egmhpo = 559 "Undefined port handle," 11m9
egmhpr = 560 "Undefined processor handle," 11m10
egmhps = 561 "Undefined process handle," 11m11
egmhsq = 562 "Undefined segment handle," 11m12
egmhsu = 563 "Undefined subprocess handle," 11m13
egmhsy = 564 "Undefined system call handle," 11m14
egmhus = 565 "Undefined user call handle," 11m15
egmhtm = 566 "Undefined timer handle," 11m16
%no more handles% 11n
ehohca = 601 "No call handle available," 11n1
ehohcn = 602 "No channel handle available," 11n2
ehohdt = 603 "No data store handle available," 11n3
ehohcv = 604 "No event handle available," 11n4
ehohlk = 605 "No lock handle available," 11n5

```

ehohls = 606	"No lockset handle available,"	11n6
ehohmg = 607	"No manager handle available,"	11n7
ehohpk = 608	"No package handle available,"	11n8
ehohpo = 609	"No port handle available,"	11n9
ehohpr = 610	"No processor handle available,"	11n10
ehohps = 611	"No process handle available,"	11n11
ehohsg = 612	"No segment handle available,"	11n12
ehohsu = 613	"No subprocess handle available,"	11n13
ehohsy = 614	"No system call handle available,"	11n14
ehohus = 615	"No user call handle available,"	11n15
ehontm = 616	"No timer handle available,"	11n16

Appendix -- Change Summaries

12

Summary of 6-AUG-75 Changes

12a

1) Operation gR_{DPS} has been added to retrieve the diagnostic message associated with a specified pPS error number, furthermore, IV_{DPS} , RR_{DPS} , GT_{DPS} , and PG_{DPS} (i.e. all operations which supply DPS with an ABP) now return a byte pointer to the diagnostic in $AC 2$ (DR_{DPS} and PT_{DPS} return zero in $AC 2$),

12a1

2) $VJSYS$ $SETRC$ has been added to set the DPS trace word $DTRACE$ in a specified remote process (or, of course, self),

12a2

3) $VJSYS$ $SOPR$ has been added to sign out the invoking processor, subprocess, or process. The $VJUSR$ $SOPR$ has been renamed $PRSO$ (and replaces $NTDPS$), and now solicits signout, rather than accomplishing it. These changes allow the signout of a created processor/subprocess/process to be initiated by its creator (as before), as well as permitting the suicide of the root processor of the entire process tree (which was previously impossible),

12a3

4) $VJSYSs$ $ITDFK$ and $SEPFK$ have been added to permit a processor to make forks it creates known to DPS so they can invoke $VJSYSs$,

12a4

Final Version of DPSJSYS

- 5) VJSYS SIGPE has been added to permit a procedure to transmit to its caller/callee without surrendering control. This action is reported to the remote procedure by means of outcome SIGNAL=4 in ACQPE, and should be followed by another invocation of ACQPE. SIGPE differs from NTEPE in the following ways: it's reported to the remote procedure via ACQPE, rather than by PENTE; its use is not restricted to the callee; no constraints are imposed upon its argument list; and it's not propagated up the thread of control. 12a5
- 6) VJSYSS SETMR and TSTMR have been added to permit the setting of virtual interval timers (this facility is required for the CF's own internal use, and is simply made available to the PF). 12a6
- 7) VJUSRs OKLPS and OKUPS have been added to OK the splicing of a remote process to the local process. 12a7
- 8) If B4 of AC 2 is raised in SIPR, the subprocess will be encapsulated with ENCAP,SAV in the connected directory. 12a8
- 9) If B3 of AC 2 is raised in SIPR, remote processes will be permitted to splice to the local process. 12a9
- 10) VJSYS ITDPS has been changed to ignore the request for a physical channel if only a logical one can be established, and to report the kind created via B0 of AC 1. 12a10
- Summary of 16-JUL-75 Changes (26100,) 12b
- 1) Operation PGDPS has been added to simplify VJUSR dispatch for "sequential" processors (see discussion) who so identify themselves via a new SIPR flag bit. 12b1
- 2) If B2 of AC 2 is raised in SIPR, PTDPS (and the SIPR itself) will thereafter be understood to imply R_DYPR. 12b2
- 3) The calling sequences for OKOPK/OKCPK have been modified to allow calls to INIPK/TRMPK to be piggybacked on them for packages in the subprocess leader. 12b3
- 4) The calling sequence for CRTPS has been modified to allow a call to OPNPK to be piggybacked on it, with a savings of two inter-process messages. 12b4
- 5) The event provided in IVDPS is signalled with the following completion code (formerly 1): 12b5
- XWD systemcall handle, VJSYS number 12b5a

Final Version of DPSJSYS

- 6) The event provided in RELPE is signalled with the following completion code (formerly 1): 12b6
- XWD call handle, outcome 12b6a
- 7) The following entities are converted to upper-case by DPS whenever accepted from the user (in one process), and therefore will appear in upper-case if ever presented to the user (in another process): 12b7
- process addresses and names; package, procedure, and data store names; and elements of user information (i.e. user, password, and account). 12b7a
- 8) Subprocesses with no packages may present a zero, rather than the address of a zero-length list of package names, in the RH of AC 2 in SIPR. 12b8
- 9) The "all" option has been deleted from CLSPK. 12b9
- 10) GTDPS returns the VJUSR number in the LH of AC 0. 12b10
- 11) Clarification: To indicate to SIPR that no (zero) address space pages are to be shared by processors within the subprocess, make the "first" page greater than the "last". Setting both to zero implies that one page (namely, page 0) is to be shared. 12b11
- 12) Clarification: Whenever a byte pointer contains -1 as its left half, the usual Tenex default (namely 440700) is assumed. 12b12
- 13) Clarification: Whenever an address is presented to DPS in a full word, the LH is ignored and may (as far as DPS is concerned) contain trash. 12b13
- 14) Clarification: If in doubt, set scope to ALL=4, priorities to 1, and version numbers (INDEXs) to 1. 12b14
- Summary of 10-JUN-75 Changes 12c
- 1) Claimed change to PECAL (see 30-MAY # 11), omitted by mistake, actually made. 12c1
- Summary of 30-MAY-75 Changes 12d
- 1) PF requests CF services via JSYS 400, rather than JSYS. Sorry for the flipflop, but note that it's a one-instruction change. 12d1

Final Version of DPSJSYS

- 2) The length (L) field of all ABF's should be zero when supplied by the programmer, 12d2
- 3) In GTDPS, the usercall handle and ABF are placed in the ACs, rather than in an ABC, 12d3
- 4) INFPS (VJSVS 52) has been added for use by the NVT package in locating a process within the network, 12d4
- 5) The priority argument has been moved from VISPE and RELPE to ALOCH, 12d5
- 6) The event handle argument has been moved from ALOCH to RELPE, 12d6
- 7) INTPE and RSMPE accept zero as a call handle, meaning all, 12d7
- 8) The call handle argument has been deleted from NTEPE and HLPPE, being redundant (i.e. supplied in IVDPS), 12d8
- 9) Abort provisions have been eliminated from SNDCH and RCVCH, 12d9
- 10) CRTEV requires an additional "length" argument specifying the number of completions simultaneously storable within the event, TSTEVE and WAIEV return the number of completions which remain stored in the specified event(s) after the operation, 12d10
- 11) Wherever DPS communicates an ASCII string to the PF, whether as a VjSYS result or as a VjUSR argument, it supplies a byte pointer to the ASCII string, rather than an ABC. Specifically, this change is made to the calling sequences of PECAL (procedure name); LVRDT, LRDDT, LWRDT (data store name); and OKOPK, OKCPK (package name), 12d11
- 12) The subprocess handle result has been deleted from OKOpK, 12d12
- 13) NTDPS (VJUSR 24) has been added to inform the process leader of the process' termination, prior to initiation of signout, 12d13
- 14) Process addresses must contain one of the following verbs: "CRT", meaning create a new process; or "SPL", meaning splice to an existing process, 12d14
- 15) The Tenex format for a data store selector contains ABC (PCPB36 element selector), rather than "zero or more ABC (element of PCPB36 element selector)", 12d15
- Summary of 27=APR=75 Changes 12e

- 1) PF requests CF services via HALTF, rather than JSYS, 12e1
- 2) PF provides a single block of storage for use by DPS in returning parameters to the PF, rather than a separate block for each VJSYS result / VJUSR argument. If insufficient storage is provided, the operation is aborted and the supplied block's L set to the size required; a second call to RRDFS (providing increased storage) may be employed to recover from the error. 12e2
- 3) CALPE, INTPE replaced by CALPE, VISPE, ALOCH, RELCH, ACQPE, RELPE, INTPE, RSMPE, NTEPE, HLPPE, 12e3
- CALPE is basically shorthand for the sequence ALOCH - VISPE - RELCH, 12e3a
 - VISPE is basically shorthand for the sequence RELPE - ACQPE, 12e3b
 - Help (HLPPE) and note (NTEPE) returns have been factored out into separate VJSYSs, 12e3c
 - Return type and subtype have been combined into a single "outcome", except that abort returns are represented as VJSYS / VJUSR failures (i.e. return +1). 12e3d
 - The EVH supplied to ALOCH designates an event to be signalled whenever the remote caller/callee returns to the local procedure and therefore acts as a cue to issue ACQPE, 12e3e
 - INTPE suspends the remote caller and must be followed by RSMPE/RELCH, 12e3f
 - LCAPE, LINPE replaced by PECAL, PEINT, PERSM, PEABR, PENTE, PEHLP, 12e3g
- 4) LVRDT VJUSR added to verify existence of local data store, 12e4
- 5) SOPR changed from a VJSYS to a VJUSR; signout event eliminated from SIPR, 12e5
- 6) CALPE, VISPE, RELPE, CRTSP, CRTPR allow priority, 12e6
- 7) LCKDI, SETLK allow abort if lock can't be set immediately, 12e7
- 8) CRTLK, CRTPS, ITDPS, CRTCH require scope, 12e8
- 9) REMLK requires lock handle, 12e9

Final Version of DPSJSYS

- | | |
|---|-------|
| 10) ULKDT requires data store selector, rather than process handle. | 12e10 |
| 11) SIPR allows automatic processor creation. | 12e11 |
| 12) WAIEV returns number of signalled events. | 12e12 |
| 13) OPNPK returns package handles (omitted by accident). | 12e13 |
| 14) VJSYSs, VJUSRs renumbered. | 12e14 |

Final Version of DPSJSYS

(J26266) 13-AUG-75 13:55;;; Title: Author(s): James E. (Jim)
White/JEW; Distribution: /SRI-ARC([INFO=ONLY]) ; Sub-Collections:
SRI-ARC; Clerk: JEW; Origin: < JWHITE, DPSJSYS,NLS;4, >
13-AUG-75 13:54 JEW ;;;;####;

26266 Distribution

Douglas C. Engelbart, Martin E. Hardy, J. D. Hopper, Charles H. Irby, Harvey G. Lehtman, James C. Norton, Jeffrey C. Peters, Dirk H. Van Nouhuys, Kenneth E. (Ken) Victor, Richard W. Watson, Don I. Andrews, Mary Ann Kellan, Buddie J. Pine, Andy Poggio, David L. Retz, Laura J. Metzger, Karolyn J. Martin, Jan A. Cornish, Larry L. Garlick, Priscilla A. Wold, Pamela K. Allen, Delorse M. Brooks, Beverly Boli, Rita Hysmith, Log Augmentation, Joseph L. Ehardt, Raymond R. Panko, Susan Gail Roetter, Robert Louis Belleville, Rene C. Ochoa, Ann Weinberg, Joan Hamilton, Adrian C. McGinnis, Robert S. Ratner, David S. Maynard, Robert N. Lieberman, Sandy L. Johnson, James H. Bair, Jeanne M. Leavitt, Rodney A. Bondurant, Jeanne M. Beck, Marcia L. Keeney, Elizabeth K. Michael, Jonathan B. Postel, Elizabeth J. Feinler, Kirk E. Kelley, N. Dean Meyer, James E. (Jim) White