CONTACT: OSHA, Jeff Franklin on 11Aug75, Re: No slot

(OSHA) Contact report 26268 1

(DATE) 11 Aug 75 1a

(BY) Lieberman 1b

(ATTENDEES) 1c

Jeff Franklin - OSHA 1c1

Robert Lieberman - SRI-ARC 1c2

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

(MEDIUM) PHONE 1e

(WHERE) Menlo Park, CA and Silver Spring, MD 1f

(ACTION-ITEMS) 1g

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

(DISTRIBUTION) ARC-LOG DCE JCN RLL 1h

(REFERENCES) 25261 25954 25741 25740 1i

(DOCUMENTS) Hard copy given and received 1j

(GIVEN) Date and documents given 1j1

(RECEIVED) Date and documents received 1j2

(REMARKS) 1k

Jeff called me to say that OSHA would NOT give us a contract for a utility slot, 1k1

It seems that the money has to be spent on the three software contractors currently under contract, 1k2

I mentioned the possibility of having one of them subcontract to use, but Jeff did not think that possibility was high, 1k3

He did not seem especially interested in the AKW seminar to be given this August, but I will send him a copy of announcement, 1k4

RLL 13-AUG-75 18:36 26268

CONTACT: OSHA, Jeff Franklin on 11Aug75, Re: No slot

(J26268) 13-AUG-75 18:36;;;; Title: Author(s): Robert N. Lieberman/RLL; Distribution: /ARC-LOG([INFO-ONLY]) DCE([INFO-ONLY]) JCN([INFO-ONLY]) ; Sub-Collections: SRI-ARC ARC-LOG; Clerk: RLL;

RLL 13-AUG-75 17:43 26269

BUG: dspyes does not refresh.

The type "dspyes" for the "dpset" procedure dos not work. No refreshing is done.I had to use "dspallf". "dspstrc" And "dsprfmt" works OK. 1

RLL 13-AUG-75 17:43 26269

BUG: dspyes does not refresh.

(J26269) 13-AUG-75 17:43;;;; Title: Author(s): Robert N. Lieberman/RLL; Distribution: /FEED([ACTION]) NDM([INFO-ONLY]) JHB([INFO-ONLY]) JDH([INFO-ONLY]) JCN([INFO-ONLY]) ; Sub-Collections: SRI-ARC; Clerk: RLL;

JDH 13-AUG-75 17:59 26270

Proposed changes for handling Right Margins on different devices

This proposed change might affect you drastically. Any comments will be appreciated.

JDH 13-AUG-75 17:59 26270

Proposed changes for handling Right Margins on different devices

See (26217,) for considerations leading to this proposal. 1

Assuming we are willing to make the changes required to make the right margin settings for quickprint, TNLS print and display mean roughly the same thing, we propose the following additional changes: 2

1) Make quickprint and TNLS print behave like the output processor right margin setting, i.e, make that setting a "printing character" or "visible" right margin. 2a

In the default case where the margin is set at 72 and a line is formatted with 72 printing characters followed by a number of non-printing characters, none of these non-printing characters will be put at the beginning of the next line. (unless, of course, they contain a CR followed by more non-printing characters) 2a1

2) For the display, use the preceding algorithm to the extent possible, formatting non-printing characters beyond the right margin setting up to the limit of the physical bounds of a particular display before putting them at the beginning of the next line. 2b

These changes give excellent likeness between the four cases for ordinary written material. However, they may actually agravate difficulties with highly formatted material, especially when designed for printers wider than a display or TTY. 3

It seems reasonable from the implementation viewpoint to add a new "wraparound" or "absolute" display boundary setting for use with material formatted wider than the particular terminal available for composition. 3a

With this boundary set less than the printing "right margin", line breaks would occur at exactly the number of characters specified, breaking in the middle of words, visibles, or invisibles if need be. 3a1

There are two choices considered unlikely to cause a nervous breakdown and/or revolution among the applications programmers: 3b

1) Use the physical maximum display width as the "wraparound" point when the "right margin" is set beyond this point. 3b1

This has the advantage that it doesn't require a new "SET" or "USEROPTION" command which would require some memory shuffling. 3b1a

Also, we also don't have to figure out what to call it. 3b1b

2) Make it user settable at a single number. It would take effect any time it was set less than the display area width and the right margin setting. 3b2

This has the advantage that the user could adjust it to miss a print field which might fall around the physical display maximum. 3b2a

Your comments are requested on the desirability of this feature. People who strongly favor implementation 2) should also include suggestions about what to call it. 3c

I repeat my earlier warning that any change in these areas stands some chance of hurting a very few people very badly. The changes outlined seem desireable and reasonable. Your comments are requested within a week so we can reach a decision. 4

JDH 13-AUG-75 17:59 26270

Proposed changes for handling Right Margins on different devices

(J26270) 13-AUG-75 17:59;;;; Title: Author(s): J. D. Hopper/JDH;
Distribution: /SRI-ARC([ACTION]) JDH([INFO-ONLY]) ;
Sub-Collections: SRI-ARC; Clerk: JDH;

26270 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, J. D. Hopper,
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

JAKE 15-AUG-75 00:42 26272

DPCS Queen for a Day

Here are some of my 'make-it-nice' 'what-the-world-needs-now' 'lucy-in-the-sky-with-diamonds' requests for DPCS and related pursuits: 1

1. Statement names only (preferably as a viewspec) 1a

(I would like a decent XGP-type printer) 1a1

2. Underlining both in hardcopy and online. I would like to be able to put some invisible character before and after text I want underlined or bug text in underline mode and have it happen online and in quickprint as well as through output processor. 1b

(I would like a decent XGP-type printer) 1b1

3. Statement numbers that are not buried in the text of the statement. What is the point of having a nice outline format if all the outline designators are buried so as to be difficult to read and find. 1c

(I would like a decent XGP-type printer) 1c1

4. An 'insert before' command that is not part of a special subsystem so that I can insert a character at the beginning of a statement easily. 1d

(I would like a decent XGP-type printer) 1d1

5. An indexing subsystem with a scenario something like this...one could specify whole strings of text (such as titles) for KWIC index handling (would extract significant words). In addition, one could bug any word in text (while in index mode) and this would become an index term. I would like the KWIC and KEYWORD options to work either together or separately depending upon the user's need. For any item that has been indexed previously I would like to have some sort of invisible marker for each index entry so that I could at some later date go back and pull out index entries. I would like to be able to extract index entries from several cited documents and have a bibliography print out which contains the subject index entry followed by the bibliographic citation in a standard format, e.g.: 1e

NETWORK INFORMATION CENTER 1e1

Engelbart, D. C. and English, W. How we did the NIC 10 years
ago, pp 243-52 IN: COMPCON Proceedings, AFIPS Press, Wherever, 1977. 1e1a

Feinler, E. J. My Life and Hard Times at NIC, ARC Journal, Vol. 25, (No. 12345, 1:w) 23-30 (Jan. 1984) 1e1b

Jernigan, M. Confessions of a NIC User, ARC Journal , Vol. 5, (No. 76543,) 8-10 (June 1976) 1e1c

Watson, R. W. You can't do a NIC for 100K! ACM J., Vol. 63, No. 4, 7-9 (May 1980) 1e1d

For instance, if I had 20 indexed journal entries, perhaps I would make a list of these entries as links and then say something like 'compile index' (or whatever) at which time a program would jump to the link of the text, extract the index entries, put them in a sublist or file as links, page numbers, or full citations (hazy about this at moment) and then sort the whole thing, first by index entry, then by entity under each index entry. 1f

I would also want to be able to create a Thesaurus of all the terms I had used in any given set of documents along with exclusion lists (words to be ignored by the KWIC index program), equivalencies (NIC = Network Information Center, which see) and cross references (Arpanet Information Center SEE Network Information Center). 1g

Well, enough...this gives a flavor of possibilities. 1h

6. TABS that work. 1i

(I would like a decent XGP-type printer) 1i1

7. A complete revamp of output processor directives...why GCR for 'grab carriage return' when the whole world recognizes CR for carriage return. 1j

(I would like a decent XGP-type printer) 1j1

8. Better still a graphics document formatter that allows me to do the following kind of neat things: Load a document onto a screen, go into formatting mode, call up a given font or format, then bug the text I would like to be in that font or format, and 'voila!' it is displayed thusly, next I would shove text around until it looks just the way I think is nice, then maybe I would specify other features like page headers that look a certain way, numbering, whatever, then say ZAP! and the system interprets all this into its own jargon and spits me out a page proof. If I liked the page proof I could ZAP ZAP! and specify a few more things like color of cover, number of copies, paper stock, etc.,

and it would go to COM and eventually appear as a Cinderella document on my desk. 1k

In the middle of any text I could, of course, edit, draw pictures, put things in boxes, insert previously numbered tables, graphs, etc,, or leave space for a given sized art insert. I would not even have to know that an output processor existed, just keep trying and looking until I got the layout I liked. There would also be online several samples of layouts and fonts done by experts, so that I would not have to do everything from scratch if one of them was suited to a particular job. (I would also like a bottle that says 'drink me' so I can explore this little hole in the ground I noticed the other day....) 1l

(I would like a decent XGP-type printer) 1l1

9. A column formatter that would generate a table consisting of columns of a specified width and containing information which I would supply to come under appropriate headers. For instance I would go into column-producing mode and specify page width (and maybe length), number of columns, Header corresponding to each column, and data to fit into each column per statement. The program would take the specified data and fit it into the specified column size under the right header automatically. 1m

(I would like a decent XGP-type printer) 1m1

10. Some kind of column recognition scheme so that I could do the usual Move, Transpose, Delete, Replace, Sort Column,...this would save hours of time spent hand-doing this job for different views of the same data. 1n

(I would like a decent XGP-type printer) 1n1

11. A link verifier that goes around and says 'Hey, you dummy, there isn't any document or file at the end of that link. Better check it'. This of course presupposes that there is a journal system where 9/10ths of it is not archived at all times. I would like to be able to trace links backwards and forwards to make sure they are valid. 1o

(I would like a decent XGP-type printer) 1o1

12. A journal system that can handle Xdoc entries, their indexing, and abstracts as an integral part of the system. However, I would want to be able to separate those entries whose full text had never been online from those whose full text was online. 1p

(I would like a decent XGP-type printer) 1p1

13. I would like to see us use some editing standards for our documents, particularly syntax standards. When I came here I was presented with a detailed syntax and told to use it as it was ARC's standard. It wasn't perfect but it was reasonable and consistent. To the best of my knowledge I am the only one who ever tried to follow it faithfully (including its author). At present I do not believe there are two ARC Manuals in the whole place that follow the same style or syntax. This is very trying for users. 1q

(I would like a decent XGP-type printer) 1q1

14. I would like to have (and so would many government agencies) the Cosati Corporate Authority Headings file available online in a searchable manner as well as various standard Thesauri of terms. It would be nice to be able to use a subset of these index terms for a small data base and know that the indexes could be merged later because they all used primarily the same indexing scheme. 1r

(I would like a decent XGP-type printer) 1r1

15. I would like to have a standard network reference-citation protocol based upon how references are cited not upon how they are cataloged. I also would like to have user options that would permit me to arrange the pieces of a reference for bibliographic citation according to given publication standards, e.g., IEEE, Amer. Chem. Soc., ACM, etc.. 1s

(I would like a decent XGP-type printer) 1s1

16. I would like to have a 'shelf list' of what journal number has been assigned to what document and what file it lives in online as well as what shelf it exists on offline. I would like to know what journal numbers were never used so I do not spend a lot of time trying to account for the document that corresponds to one of these unused numbers. I would like to also have a running account somewhere of what journal documents are online at any given moment (that is active and not archived). 1t

(I would like a decent XGP-type printer) 1t1

17. I would like to have a standard journal citation header for our journal items and have the journal be published offline in volumes like more traditional journals. This presupposes that trivial messages and private items would be either separate or easily removed. Depending on the size of the whole thing perhaps several offshoots would develop...such as publications of special

interest groups would make useful hardcopy journals. How or whether they should be refereed or copyrighted I won't even begin to approach. 1u

(I would like a decent XGP-type printer) 1u1

18. I would like to have a regional network copy center for hardcopy distribution that could publish and distribute any documents a user would like to have produced and distributed. Any user who wanted to receive either all or selected pieces of his incoming mail, documentation, journal items or whatever, in hardcopy could so specify and he would be billed the prevailing rates for the amount of work done and the goods would be delivered within a reasonable length of time (say a week formost, much longer for some and special delivery one day turnaround for a few). 1v

(I would like a decent XGP-type printer) 1v1

19. I would like to see an online index to government grants (ala the Grants Index) maintained online and centrally located that pertained only to research projects being carried out on the Arpanet. It would include the usual information such as title, funding agency, PI, amount, purpose, dates, etc. 1w

Well that is enough to keep most all of us busy until the year 2000. One of the joys of what we are doing is that there are infinite interesting possibilities! 1x

Please Note Subliminal Advertising 1x1

DPCS Queen for a Day

(J26272) 15-AUG-75 00:42;;;; Title: Author(s): Elizabeth J. Feinler/JAKE; Distribution: /DPCS([ACTION]) RLB2([ACTION]) SRI-ARC([INFO-ONLY]) ; Sub-Collections: SRI-ARC DPCS; Clerk: JAKE; Origin: < FEINLER, DCPS-WISHLIST,NLS;3, >, 15-AUG-75 00:38 JAKE
;;;;####;

26272 Distribution

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, 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,
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, Robert Louis Belleville, 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

JEW 15-AUG-75 12:18 26273

The Printer

One has only to read between the lines of Jake's (26272,) to sense her need for a jazzier printer. May I suggest that a knowledge workshop needs, at very least, a FASTER printer (and/or printer configuration; perhaps ELF and/or the Net are partially to blame). I resorted to AI's printer for my weekly listings of DPS during the last few months but, in retrospect, consider that taking unfair advantage of a friend. 1

The Printer

(J26273) 15-AUG-75 12:18;;;; Title: Author(s): James E. (Jim) White/JEW; Distribution: /SRI-ARC([INFO-ONLY]) DPCS([INFO-ONLY]) RLB2([INFO-ONLY]) ; Sub-Collections: SRI-ARC DPCS; Clerk: JEW;

26273 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, 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, Robert Louis Belleville,
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

RLL 15-AUG-75 12:46 26274

Announcement of AKW workshop - 25 to 29 Aug 1975

This is the announcement of the seminar on AKW to be given the week of 25 Aug, If interested, it is not too late to notify me. Please let me know if you have anyone in mind. This item was sent to potential participants two weeks ago. We plan to given this seminar again in a few months (subject to demand), Your suggestions and leads are welcomed, thank you, Rob

Announcement of AKW workshop - 25 to 29 Aug 1975

A SEMINAR ON THE AUGMENTED KNOWLEDGE WORKSHOP

The Augmentation Research Center (ARC) of Stanford Research Institute (SRI) has been developing for the past 13 years a computer-augmented workshop to help people with their information handling. Our goal is the evolution of a coherent set of facilities to aid people with their reading, writing, communicating, and management of information.

We feel that a large and diverse community of users participating in an augmented environment is necessary for the continuation of workshop developments.

As part of the effort to transfer our developments to a growing community of users, we are offering an intensive, one week seminar on the capabilities and use of ARC's workshop facilities. The participants will gain insight into the tools, techniques, and methodology of the augmented workshop.

This seminar is intended for those people who wish to begin assessing the workshop capabilities and its potential value to their organization.

Those individuals whose job is to give experience-based appraisals of new working methods and new technology will find this week particularly beneficial.

Training in many of the workshop features will be given with several additional hours of individual, on-line, interactive computer experience.

Documentation, training aids, and demonstrations will illustrate, in more breadth and depth, the workshop's facilities.

Discussion sessions on the application of this technology to each organization will highlight the week's work. Emphasis will be placed on each participant's introduction to the potential of these capabilities.

The topics to be covered will include the following:

- Studying online documents
- Document production
- Organizational and community collaborative dialogue
- Organizational intelligence
- Personal data management
- Software engineering augmentation

This five day seminar will be held at SRI from 25 August to 29 August

1975. The seminar will be limited to ten active participants so that individualized and intensive training, consulting, and discussion can take place. 11

The cost of this seminar will be $1000 per person and will include all materials and computer time. Travel, local lodging, and meals will be paid by each client directly. 12

If you cannot attend this seminar but are interested in future seminars, please contact us so that a more convenient date can be arranged. 13

For additional information and registration contact: 14

Dr. Robert Lieberman 14a
Augmentation Research Center 14b
Stanford Research Institute 14c
333 Ravenswood Road 14d
Menlo Park, CA 94025 14e
415-326-6200 x4119 14f

Announcement of AKW workshop - 25 to 29 Aug 1975

Registration Form 15

A SEMINAR on the 15a

AUGMENTED KNOWLEDGE WORKSHOP 15b

at 15c

STANFORD RESEARCH INSTITUTE 15d

25-29 August 1975

15e

Name..... 15f

Organization..... 15g

Address..... 15h

City, State, Zip..... 15i

Phone..... 15j

Authorizing signature..... 15k

Would you like motel information? 15l

Payment is 15m

enclosed..... 15m1

to be billed to: 15m2

Name..... 15m2a

Address..... 15m2b

City, State, Zip..... 15m2c

RLL 15-AUG-75 12:46 26274

Announcement of AKW workshop - 25 to 29 Aug 1975

(J26274) 15-AUG-75 12:46;;;; Title: Author(s): Robert N. Lieberman/RLL; Distribution: /KWAC([INFO-ONLY]) SRI-ARC([INFO-ONLY]) ; Sub-Collections: SRI-ARC KWAC; Clerk: RLL;

26274 Distribution

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, 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 F. Finney, Lawrence A. Crain, E. S. VonGehren, Glenn A. Sherwood, Kathey L. Mabrey, Jeanne M. Beck, David A. Potter, Robert N. Lieberman, Terry H. Proch, Ronald P. Uhlig, Susan Gail Roetter, Michael A. Placko, Stanley M. (Stan) Taylor, Elizabeth J. Feinler, Rudy L. Ruggles, Frank G. Brignoli, Robert M. Sheppard, Richard W. Watson, Douglas C. Engelbart, James C. Norton, James H. Bair, Duane L. Stone, Inez M. Mattiuz, Connie K. McLindon, 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

RLL 15-AUG-75 17:27 26275

CONTACT: Commission on Civil RIghts, Jackson on 4 Aug 75 Re: RFP

The upshot is that we will not bid on this RFP.

RLL 15-AUG-75 17:27 26275

CONTACT: Commission on Civil RIghts, Jackson on 4 Aug 75 Re: RFP

(RIGHTS) Contact report 26275 1

(DATE) 4 Aug 75 1a

(BY) Lieberman 1b

(ATTENDEES) 1c

Rita Jackson - CCR 1c1

Robert Lieberman - SRI-ARC 1c2

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

U. S. Commission on Civil Rights 1d1

(MEDIUM) PHONE 1e

(WHERE) Menlo Park, CA and Washington, DC 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 Ms. Jackson to find out more about an RFP that reached ARC (request for text editing services). 1l

What they want is ATS. This was concluded after a two month study!!! 1m

In any case, she would be pleased to read about other services such as ours. 1n

RLL 15-AUG-75 17:27 26275
CONTACT: Commission on Civil Rights, Jackson on 4 Aug 75 Re: RFP

(J26275) 15-AUG-75 17:27;;;; 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;

26275 Distribution
James C. Norton, Log Augmentation, Douglas C. Engelbart, James C. Norton, Robert N. Lieberman,

CONTACT: Bechtel, Ahrendes on 30 Jul 75 Re: Seminar

(BECHTEL) Contact report 26276 1

(DATE) 30 Jul 75 1a

(BY) Lieberman 1b

(ATTENDEES) 1c

John Ahrendes - BECHTEL 1c1

Robert Lieberman - SRI-ARC 1c2

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

(MEDIUM) PHONE 1e

(WHERE) Menlo Park, CA and San Francisco, CA 1f

(ACTION-ITEMS) 1g

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

(DISTRIBUTION) ARC-LOG DCE JCN RLL 1h

(REFERENCES) 25939 26079 1i

(DOCUMENTS) Hard copy given and received 1j

(GIVEN) Date and documents given 1j1

(RECEIVED) Date and documents received 1j2

(REMARKS) 1k

I called John to find out the status of his seminar, 1k1

He plans to have us lecture at his seminar in September, 1k2

August 11 he is having a orientation meeting for his seminar, 1k3

He is interested in our AKW seminar but not until after we give our lecture at his, 1k4

RLL 15-AUG-75 17:29 26276

CONTACT: Bechtel, Ahrendes on 30 Jul 75 Re: Seminar

(J26276) 15-AUG-75 17:29;;;; 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;

26276 Distribution
James C. Norton, Log Augmentation, Douglas C. Engelbart, James C. Norton, Robert N. Lieberman,

RLL 15-AUG-75 17:32 26277

CONTACT: City of Tracy, Garvin on 22 July 75 Re: seminar

(TRACY) Contact report 26277 1

(DATE) 22 July 75 1a

(BY) Lieberman 1b

(ATTENDEES) 1c

Dan Garvin - City of Tracy 1c1

Robert Lieberman - SRI-ARC 1c2

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

(MEDIUM) PHONE 1e

(WHERE) Menlo Park, CA and Tracy, CA 1f

(ACTION-ITEMS) 1g

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

(DISTRIBUTION) ARC-LOG DCE JCN RLL 1h

(REFERENCES) 26085 25988 1i

(DOCUMENTS) Hard copy given and received 1j

(GIVEN) Date and documents given 1j1

(RECEIVED) Date and documents received 1j2

(REMARKS) 1k

Dan is now preparing a document which includes information on our workshop for presentation at the Council of Governments. 1k1

He talked to Postman of San Jose. They will get together next month. 1k2

Dan and Postman think it best to pursue the possibliity of using NLS with a multi-city project. 1k3

Our documentation production facilities have much appeal to him. 1k4

The teleconferencing among several cities has great potential in this multi-city arena. 1k5

RLL 15-AUG-75 17:32 26277

CONTACT: City of Tracy, Garvin on 22 July 75 Re: seminar

He will be talking to state officials and the council of governments on possibility getting together several cities for a project. 1k6

RLL 15-AUG-75 17:32 26277

CONTACT: City of Tracy, Garvin on 22 July 75 Re: seminar

(J26277) 15-AUG-75 17:32;;;; 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;

26277 Distribution
James C. Norton, Log Augmentation, Douglas C. Engelbart, James C. Norton, Robert N. Lieberman,

RLL 15-AUG-75 17:36 26278

CONTACT: OOT, B, Wieder on 22 Jul 75 Re: seminar and his interest in AKW

(OOT) Contact report 26278 1

(DATE) 22 Jul 75 1a

(BY) Lieberman 1b

(ATTENDEES) 1c

Bernard Wieder - OOT 1c1

Robert Lieberman - SRI-ARC 1c2

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

Program Development Office 1d1

Office of Telecommunication 1d2

U.S. Dept. Of Commerce 1d3

1325 G Street NW 1d4

Washington, DC 20005 1d5

202-967-4888 1d6

(MEDIUM) PHONE 1e

(WHERE) Menlo Park, CA and Washington, DC 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

He has read the literature I sent him, It looks good, 1k1

RLL 15-AUG-75 17:36 26278

CONTACT: OOT, B. Wieder on 22 Jul 75 Re: seminar and his interest in AKW

He will be visiting us in August or September (more likely Sept.). 1k2

The AKW seminar sounds interesting and he might consider it. 1k3

RLL 15-AUG-75 17:36 26278

CONTACT: OOT, B. Wieder on 22 Jul 75 Re: seminar and his interest in AKW

(J26278) 15-AUG-75 17:36;;;; 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;

26278 Distribution
James C. Norton, Log Augmentation, Douglas C. Engelbart, James C. Norton, Robert N. Lieberman,

RLL 15-AUG-75 17:38 26279

CONTACT: Mitre Corp., L. Thomas on 22 Jul 75 Re: seminar

(MITRE) Contact report 26279 1

(DATE) 22 Jul 75 1a

(BY) Lieberman 1b

(ATTENDEES) 1c

Louis Thomas - Mitre 1c1

Robert Lieberman - SRI-ARC 1c2

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

Mitre general number - 617-271-2000 1d1

Thomas - 617-271-4872 1d2

(MEDIUM) PHONE 1e

(WHERE) Menlo Park, CA and Bedford, MA 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 Lou to find out what was happening since he and others from Mitre visited us. 1k1

Lou is no longer in charge of the projects related to COTCO, so he is not the person to contact anymore (according to him). 1k2

Nancy Goodwin and John Mitchell are still involved. Lou will tell Mitchell to call me. 1k3

Thomas thought Mitchell was going to contact us anyway. 1k4

RLL 15-AUG-75 17:38 26279

CONTACT: Mitre Corp,, L. Thomas on 22 Jul 75 Re: seminar

(J26279) 15-AUG-75 17:38;;;; 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;

26279 Distribution
James C. Norton, Log Augmentation, Douglas C. Engelbart, James C. Norton, Robert N. Lieberman,

RLL 15-AUG-75 17:41 26280

CONTACT: RADC, Stone on 22 jul 75 Re: AKW seminar

(RADC) Contact report 26280 1

(DATE) 22 July 75 1a

(BY) Lieberman 1b

(ATTENDEES) 1c

Duane Stone - RADC 1c1

Robert Lieberman - SRI-ARC 1c2

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

(MEDIUM) PHONE 1e

(WHERE) Menlo Park, CA and Rome, NY 1f

(ACTION-ITEMS) 1g

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

(DISTRIBUTION) ARC-LOG DCE JCN RLL JHB 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 Duane about our seminar to be given in August, 1k1

He said not to make it a proposal, 1k2

Probably IBM should not be asked since they will only have a short term contract, 1k3

He will think and ask DMA (Defense Mapping Agency) if they are interested, 1k4

Maybe someone at RADC at the executive level should attend, He will think more on this, 1k5

He asked about the terminal letter proposal and the possibility of getting terminals on order before money arrives, 1k6

RLL 15-AUG-75 17:41 26280

CONTACT: RADC, Stone on 22 jul 75 Re: AKW seminar

(J26280) 15-AUG-75 17:41;;;; Title: Author(s): Robert N. Lieberman/RLL; Distribution: /ARC-LOG([INFO-ONLY]) DCE([INFO-ONLY]) JCN([INFO-ONLY]) RLL([INFO-ONLY]) JHB([INFO-ONLY]) ; Sub-Collections: SRI-ARC ARC-LOG; Clerk: RLL;

26280 Distribution
James C. Norton, Log Augmentation, Douglas C. Engelbart, James C. Norton, Robert N. Lieberman, James H. Bair,

RLL 15-AUG-75 17:45 26281

CONTACT: NSWC, Bob Archer on 22 Jul 75 Re: AKW seminar

(NSWC) Contact report 26281 1

(DATE) 22 July 75 1a

(BY) Lieberman 1b

(ATTENDEES) 1c

Bob Archer - NSWC 1c1

Robert Lieberman -SRI-ARC 1c2

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

(MEDIUM) PHONE 1e

(WHERE) Menlo Park, CA and Silver Spring, MD 1f

(ACTION-ITEMS) 1g

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

(DISTRIBUTION) ARC-LOG DCE JCN RLL JHB 1h

(REFERENCES) 1i

(DOCUMENTS) Hard copy given and received 1j

(GIVEN) Date and documents given 1j1

(RECEIVED) Date and documents received 1j2

(REMARKS) 1k

He does wish to appraise NLS, 1k1

He has had only a little training via FGB, I suggested he talk to Frank about possibly more training, 1k2

He only needs a formal announcement in order to attend the course, 1k3

Bob will call me back when he knows, 1k4

RLL 15-AUG-75 17:45 26281

CONTACT: NSWC, Bob Archer on 22 Jul 75 Re: AKW seminar

(J26281) 15-AUG-75 17:45;;;; Title: Author(s): Robert N. Lieberman/RLL; Distribution: /ARC-LOG([INFO-ONLY]) DCE([INFO-ONLY]) JCN([INFO-ONLY]) RLL([INFO-ONLY]) JHB([INFO-ONLY]) ; Sub-Collections: SRI-ARC ARC-LOG; Clerk: RLL;

26281 Distribution
James C. Norton, Log Augmentation, Douglas C. Engelbart, James C. Norton, Robert N. Lieberman, James H. Bair,

JEW 15-AUG-75 18:14 26282

DPS Implementer's Guide
DPS Version 2.5

15-AUG-75

James E. White
Augmentation Research Center

Stanford Research Institute
Menlo Park, California 94025

The Distributed Programming System (DPS) provides a framework in which processes on arbitrary ARPANET hosts can communicate at the procedure call level. This Implementer's Guide describes the inter-process protocol underlying DPS Version 2.5.

This Implementer's Guide describes DPS as it existed when cut from NSW.

Preface 1

The Distributed Programming System (DPS) provides a framework in which independent processes on arbitrary ARPANET hosts can communicate at the procedure call level. In effect, DPS provides a common virtual run-time programming environment in which all remote processes can be assumed to run. DPS also provides the nucleus for what may one day become a virtual network operating system. 1a

The present document describes the protocol exchanges that take place between the system components of DPS processes. Its primary purpose is to present to DPS implementers the details of their interface with other processes. The reader should refer to other documents for a higher-level description of the DPS model. 1b

Process Structure 2

A DPS process consists, at least conceptually, of a "system component" (SC) implementing DPS and a "user component" (UC) containing the process proper. Ideally, the SC is implemented once for each class of host machines, or at worst once for each programming language or environment on that machine. 2a

SC and UC communicate by means of an internal protocol specified by the SC's implementer. The DPS-10 Programmer's Guide (26271,) provides an example of such an interface. 2b

The SCs within different processes communicate over a communication channel according to the protocol described in this document. The communication channel is either a pair of Network connections (established by ICP to a DPS contact socket or negotiated by a third process) if the processses reside on different hosts, or a host-dependent inter-process communication channel if they reside within a single host. 2c

The DPS protocol specifies the several messages which implement the fundamental DPS call and return mechanism (CRM), transmission formats for those messages, and a family of system procecures invokable by means of the CRM. 2d

Handles and Their Scope 3

Those DPS messages or procedures that create or access DPS entities (e,g, packages, logical channel segments) return INDEX "handles" to which the processes can refer in subsequent dialog. Some also accept as one of their arguments the "scope" of the handle to be assigned. 3a

The scope of a handle is simply the domain, with respect to the process P which requests its assignment, within which the handle will be defined (i,e, recognized by DPS). Scope is an INDEX with one of the following values: 3b

PS=3: a handle with scope PS is known only to P. 3b1
AL=4: a handle with scope AL is known throughout the process tree. 3b2

The classes of entities assigned handles are listed below, along with the handle's mneumonic and legal values for its scope: 3c

Call	CAH	PS --	3c1
Data lock	DLH	PS AL	3c2
Package	PKH	PS AL	3c3
Port	POH	-- AL	3c4
Process	PSH	PS --	3c5

Abbreviations 4

The following abbreviations are used throughout this document: 4a

[x] X or EMPTY; i,e, x is optional. 4b

%x% X is a comment. 4c

System Data Types 5

Introduction 5a

The following sections describe the "system data types" whose definition permits convenient shorthand description of various message parameters and procedure arguments and results throughout this document. The names of system data types appear in the descriptions suffixed with an asterisk ('*'). 5a1

Processes 5b

USERINFO
User Information 5b1

LIST (%user% CHARSTR, %password% CHARSTR,
%account% CHARSTR)

USERINFO specifies the upper-case USER, PASSWORD, and ACCOUNT to which use of a process is to be billed.

PSDESC
Process Description 5b2

LIST (%name% CHARSTR, %address% CHARSTR, LOCATION*,
%user% CHARSTR)

PSDESC identifies a process to a human user, e.g. in a debugging situation: its NAME (an arbitrary character string), ADDRESS, and LOCATION, along with the name of the USER to whom its use is being billed.

The upper-case process address must have the following format:

<action> [<SP> <host>] <SP> <intrahost address>

where ACTION is either "CRT" or "SPL", meaning a new process was created or an existing one spliced to, respectively; HOST is a decimal host address or standard host name; and INTRAHOST ADDRESS is a SAV filename (on Tenex) or a decimal ICP contact socket number for CRT and SPL, respectively.

LOCATION
Location 5b3

LIST (%host% INDEX, %job% INDEX, %process% INDEX)

LOCATION specifies a process' physical location within the Network, i.e. its HOST address, its JOB number, and its (local) PROCESS number.

NOTE: as currently defined, LOCATION is insufficiently general and should be changed to:

LIST (%host% INDEX, %intra-host address% any)

Packages 5c

PKINFO
Package Information 5c1

LIST (LIST (%name% CHARSTR, ...),
[LIST (%startup info% any, ...)], %scope% INDEX)

PKINFO specifies the parameters necessary to open one or more processes in the destination process: their upper-case NAMEs, STARTUP INFO for each, and the SCOPE of the package handles to be returned.

Procedures 5d

ARGLMASK
Argument List Mask 5d1

LIST (INDEX [CALLER=1] / DSELECTOR*, ...)

ARGLMASK (default is LIST (INDEX [CALLER=1])) specifies the list of respective transformations to be applied to a list of arguments by the process that consumes them.

Each argument may be either transmitted as specified to the destination procedure, or first replaced by the contents of the data store whose DSELECTOR is specified.

If the mask has fewer elements than the argument list, the mask is extended by replicating the last element the requisite number of times. If the mask has more elements than the argument list, the excess elements are ignored.

RESLMASK
Result List Mask 5d2

LIST (INDEX [CALLER=1/DISCARD=2] / DSELECTOR*, ...)

RESLMASK (default is LIST (INDEX [CALLER=1])) specifies the list of respective transformations to be applied to a list of results by the process that generates them.

Each result may be transmitted as specified to the destination procedure, deposited in the data store whose DSELECTOR is specified and replaced by EMPTY, or simply discarded and replaced by EMPTY.

If the mask has fewer elements than the result list, the mask is extended by replicating the last element the requisite number of times. If the mask has more elements than the result list, the excess elements are ignored.

Data Stores 5e

DSELECTOR
Data Store Selector 5e1

LIST (%psh% INDEX, %pkh% INDEX, %name% CHARSTR,
[ESELECTOR*])

DSELECTOR denotes the data store whose process handle PSH, package handle PKH, and upper-case NAME are specified, or the substructure within it whose element selector (if any) is specified.

ESELECTOR
Element Selector 5e2

LIST (<BOOLEAN [KEY=TRUE/INDEX=FALSE]> any/INDEX, ...)

ESELECTOR specifies a substructure within a data store. Each element of the eselector identifies, by INDEX or KEY, an element of the LIST selected by the previous eselector element. If two or more elements within a list have the same key, the first one is selected.

Channels 5f

CLASS
Channel Class 5f1

<%format% INDEX (PCPB36=1/PCPB8=2)>
%type% INDEX (INTERPS=1/INTERHOST=2/INTERJOB=3)

CLASS specifies the TYPE and FORMAT of a particular class of physical inter-process communication channels.

TYPE characterizes the software techniques employed to interconnect and exchange messages between two processes (e.g. ARPANET connections (INTERHOST), address space windows on Tenex (INTERPS)).

FORMAT specifies the encoding of the DPS data structures which constitute the inter-process messages (see Appendices A and B).

PORT
Port 5f2

%socket% INTEGER / %page% INDEX / any

PORT specifies a port to which a physical inter-process communication channel can be established.

An inter-host port is a receive SOCKET number R, and the channel a pair of Network connections terminating at sockets R and R+1.

An inter-process port on Tenex is a PAGE number, and the channel a window comprising pages P through P+L of an address space, where L is a compile-time constant of the DPS-10 implementation.

Arbitrary host-dependent ports are permitted.

Messages 6

Introduction 6a

The following sections describe the four messages which DPS implementations exchange over the physical channel between them. 6a1

Messages are LIST data structures whose first and second elements are an INDEX routing code (see CRTCHH system procedure) and an INDEX op code, respectively, and whose remaining elements are message-specific parameters. 6a2

CALPE (OP 1)
Calls procedure in destination process. 6b

LIST (route, 1, cah, pkh, name, arguments, arglmask, reslmask, priority) 6b1

CALPE invokes at the indicated PRIORITY (whose semantics are unspecified by the protocol) and with the provided ARGUMENTs (default is none), the procedure whose package handle PKH and NAME are specified. 6b2

If PKH is EMPTY, NAME is an INDEX and designates one of the system procedures described later in this document; if PKH is an INDEX, NAME is an upper-case CHARSTR and designates a procedure within the user program. 6b3

Before delivery to the callee, the arguments are to be filtered through the argument list mask ARGLMASK. Similarly, the first set of results returned by the callee are to be filtered through the result list mask RESLMASK before transmission to the caller's process. 6b4

The call handle CAH serves to identify the call in subsequent inter-process dialog, e.g. in the RTNPE message. 6b5

Parameter types: 6b6

```
cah       - INDEX
pkh       - [INDEX]
name      - CHARSTR / INDEX
arguments - [LIST]
arglmask  - [ARGLMASK*]
reslmask  - [RESLMASK*]
priority  - INDEX
```

```
RECPE (OP 2)
Recalls procedure in destination process,                          6c

   LIST (route, 2, cah, arguments, arglmask, reslmask)             6c1

      RECPE transmits ARGUMENTs (default is none) and
   relinquishes control to the callee with handle CAH,             6c2

      Before delivery to the callee, the arguments are to be
   filtered through the argument list mask ARGLMASK.  Similarly,
   the next set of results returned by the callee are to be
   filtered through the result list mask RESLMASK before
   transmission to the caller's process,                           6c3

   Parameter types:                                                6c4

      cah       = INDEX
      arguments = [LIST]
      arglmask  = [ARGLMASK*]
      reslmask  = [RESLMASK*]
```

RTNPE (OP 3)
Returns results of procedure to destination process. 6d

LIST (route, 3, cah, results, reslmask, arglmask,
outcome, cost) 6d1

RECPE transmits RESULTs (default is none) and relinquishes control of the call with handle CAH in any of the following ways, distinguished by OUTCOME: 6d2

VISIT=1

The caller or callee has relinquished control temporarily.

SUCCESS=2 OR FAILURE=3

The procedure has completed, and the callee has returned the COST in cents of its execution and permanently relinquished control; the call handle is invalidated.

ABORTED=4

The procedure has aborted, and the callee has returned the COST in cents of its execution and permanently relinquished control; the call handle is invalidated. RESULTS is:

LIST (%error number% INDEX, %diagnostic% CHARSTR)

Before delivery to the destination procedure, the results are to be filtered through the result list mask RESLMASK, unless the procedure aborted. Similarly, the next set of arguments (if any) returned to the source procedure are to be filtered through the argument list mask ARGLMASK before transmission to the source process. 6d3

Parameter types: 6d4

cah - INDEX
results - [LIST]
reslmask - [RESLMASK*]
arglmask - [ARGLMASK*]
outcome - INDEX
cost - INTEGER

ERROR (OP 4)
Reports error to destination process. 6e

LIST (route, 4, number, diagnostic) 6e1

ERROR reports to the destination process the protocol violation whose NUMBER and DIAGNOSTIC are specified. 6e2

The destination process should log ERROR messages for later examination by a systems programmer. 6e3

Parameter types: 6e4

number - INDEX
diagnostic - CHARSTR

System Procedures 7

Introduction 7a

The inter-process messages described in the preceeding section implement the fundamental DPS call and return mechanism. Exploiting this mechanism, all other protocol exchanges are carried out as calls upon one of the "system procedures", described below, offered by every DPS implementation. 7a1

System procedures are distinguished from user procedures by an EMPTY package handle in the CALPE message. 7a2

Throughout the following descriptions, "local" refers to the process containing the procedure being described. 7a3

Processes 7b

INIPS (PROC 1)
Initializes local process. 7b1

```
INIPS (program, userinfo, startupinfo, psdesc1, pkinfo
       => psdesc2, pkhs)
```

INIPS initializes the local process.

INIPS verifies the user, password, and account specified by USER INFO, loads into the process the user PROGRAM specified in upper-case, delivers the STARTUP INFO to it, and opens and returns handles PKHs to the packages (if any) specified by PKINFO.

The local and invoking processes also exchange process descriptions PSDESC1 and PSDESC2.

INIPS must be the first procedure called after the local process' creation.

Argument/result types:

```
program     - CHARSTR
userinfo    - USERINFO*
startupinfo - any
psdesc1     - PSDESC*
pkinfo      - [PKINFO*]
psdesc2     - PSDESC*
pkhs        - [LIST (INDEX, ...)]
```

TRMPS (PROC 2)
Terminates local process. 7b2

TRMPS (=> cost)

TRMPS terminates the local process and returns the COST of its use in cents.

TRMPS must be the last procedure called before the local process' deletion.

Argument/result types:

cost = INTEGER

CRTCHH (PROC 3)
Creates half of logical channel. 7b3

CRTCHH (psdesc, startupinfo, chainsgh, modelsgh
=> sgh, psh)

CRTCHH enables the invoking process to introduce two processes to one another, by establishing half of the logical channel required to interconnect them.

A "logical channel" is a software-maintained communication path exploiting several shorter physical channels, which interconnects two processses P(1) and P(n) by means of one or more intermediate processes P(2) through P(n-1).

Every such process P(i) maintains one "segment" of the linked list which constitutes the logical channel. A segment is simply a table entry containing handles for the adjacent processes P(i-1) and P(i+1) and their respective segments. The channel's two terminal segments, of course, contain only a single process and segment handle.

Control messages (i.e. CALPE, RECPE, RTNPE, and ERROR) can be relayed between terminal processes by means of the logical channel. The source process (say P(1)) transmits a message via its physical channel to P(2), with the message's ROUTE field addressing the appropriate segment within P(2). Upon receipt of the message, P(2) locates its table entry with handle ROUTE, replaces the message's ROUTE field with the handle to P(3)'s segment, and forwards the message to P(3). Eventually the message reaches P(n), where its ROUTE field contains the special relative segment handle SELF=1 and where it is therefore consumed.

The simplest logical channel (n=3) is established by P(2), who calls CRTCHH in both terminal processes P(1) and P(2). When invoked in P(1), for example, CRTCHH specifies P(3)'s process description PSDESC, STARTUP INFO for P(1), and a handle CHAINSGH for the segment allocated by P(2); MODELSGH is EMPTY. CRTCHH returns P(1)'s segment handle SGH and the process handle PSH by which P(3) will be known within P(1).

More complicated logical channels (n>3) result when one or both of the terminal processes is itself linked by a logical channel to the process performing the introduction. In this case, CRTCHH is employed in much the same manner, except that MODELSGH is the handle of an existing segment within the local process which is to be replicated by a recursive call on CRTCHH.

Argument/result types:

```
psdesc      - PSDESC*
startupinfo - any
chainsgh    - INDEX
modelsgh    - [INDEX]
sgh         - INDEX
psh         - INDEX
```

DELCHH (PROC 4)
Deletes half of logical channel. 7b4

DELCHH (sgh -> cost)

DELCHH enables the invoking process to separate two introduced processes, by dismantling half of the logical channel which interconnects them. DELCHH invalidates both segment and process handles.

The simplest logical channel (n=3) is dismantled by P(2), who calls DELCHH in both terminal processes P(1) and P(2). When invoked in P(1), for example, DELCHH specifies S(1)'s segment handle SGH, and returns the COST of P(3)'s use of P(1).

More complicated logical channels (n>3) are dismantled in the same manner. In this case, however, the call to DELCHH is propagated down the logical channel until the terminal process (who supplies COST) is reached.

A logical channel may only be dismantled by the process which created it.

Argument/result types:

sgh - INDEX
cost - INTEGER

Packages 7c

OPNPK (PROC 5)
Opens local packages. 7c1

OPNPK (pknames, startupinfos, scope -> pkhs)

OPNPK opens and returns handles PKHs for the packages, in the local process, whose upper-case PACKAGE NAMEs and STARTUP INFOs (defaults are EMPTY) are specified.

The handles returned are to have the specified SCOPE.

Argument/result types:

```
pknames      - LIST (CHARSTR, ...)
startupinfos - [LIST]
scope        - INDEX
pkhs         - [LIST (INDEX, ...)]
```

CLSPK (PROC 6)
Closes local packages. 7c2

CLSPK (pkhs -> costs)

CLSPK closes the packages with handles PKHs in the local process, and then invalidates the handles.

CLSPK returns the COSTs of the packages' use in cents.

Argument/result types:

```
pkhs  - LIST (INDEX, ...)
costs - LIST (INTEGER, ...)
```

Procedures 7d

INTPE (PROC 7)
Interrupts local procedure, 7d1

INTPE (cah)

INTPE interrupts (i.e. freezes) the local callee with handle CAH, currently being executed within the local process. INTPE delays indefinitely, transmission of the RTNPE message which would terminate the procedure.

If the procedure cannot be interrupted, or if the interruption of procedures is unsupported (in general) by the local process, the system procedure may be aborted, in which case execution of the procedure is presumed to continue.

Argument/result types:

cah - INDEX

RSMPE (PROC 8)
Resumes local procedure, 7d2

RSMPE (cah)

RSMPE resumes (i.e. thaws) the local callee with handle CAH, currently interrupted within the local process.

Argument/result types:

cah - INDEX

ABRPE (PROC 9)
Aborts local procedure. 7d3

ABRPE (cah)

ABRPE solicits the abortion (i.e. transmission of an appropriate RTNPE message) of the local callee with handle CAH, currently being executed within the local process.

If the procedure cannot be aborted, or if the abortion of procedures is unsupported (in general) by the local process, the system procedure may be aborted, in which case execution of the procedure is presumed to continue.

Argument/result types:

cah - INDEX

SIGPE (PROC 22)
Signals local procedure. 7d4

SIGPE (cah, arguments, arglmask)

SIGPE signals the local procedure with handle CAH (without releasing control of the call) and delivers to it the next set of ARGUMENTs (default is none).

Before delivery to the procedure, the arguments are to be filtered through the argument list mask ARGLMASK.

SIGPE may be invoked on behalf of either caller or callee.

NOTE: implementation of the signalling function as a system procedure, rather than as a message, provides a crude form of flow control which, however, should not be allowed to delay the user program unless two signals are generated in rapid succession.

Argument/result types:

```
cah       = INDEX
arguments = [LIST]
arglmask  = [ARGLMASK*]
```

NTEPE (PROC 21)
Reports event to local caller. 7d5

NTEPE (cah, number, description)

NTEPE reports to the local caller an event detected by the callee with handle CAH in the invoking process.

NTEPE specifies the event's NUMBER and DESCRIPTION, which should be propagated up the thread of control by the local process via a recursive call to NTEPE.

The remote callee retains control of the call.

NOTE: implementation of the note function as a system procedure, rather than as a message, provides a crude form of flow control which, however, should not be allowed to delay the user program unless two notes are generated in rapid succession.

Argument/result types:

cah	- INDEX
number	- INDEX
description	- any

HLPPE (PROC 10)
Solicits help from local caller. 7d6

HLPPE (cah, number, description -> solution)

HLPPE solicits help from the local caller with a problem encountered by the caller with handle CAH in the invoking process.

HLPPE specifies the problem's NUMBER and DESCRIPTION, which should first be presented to the local caller for consideration and then, if no solution to the problem is provided, propagate the request up the thread of control via a recursive call to HLPPE. Only if the control thread ends with the local caller should the HLPPE procedure be aborted.

The callee relinquishes control of the call.

Argument/result types:

cah	- INDEX
number	- INDEX
description	- any
solution	- any

Data Stores 7e

CRTDT (PROC 11)
Creates local data store. 7e1

CRTDT (dselector, value, scope)

CRTDT creates the local data store whose DSELECTOR and SCOPE are specified, and assigns it an initial VALUE.

Argument/result types:

dselector - DSELECTOR*
value - any
scope - INDEX

DELDT (PROC 12)
Deletes local data store. 7e2

DELDT (dselector)

DELDT deletes the local data store whose DSELECTOR is specified, and discards its VALUE.

Argument/result types:

dselector - DSELECTOR*

RDDT (PROC 13)
Reads local data store. 7e3

RDDT (dselector -> value)

RDDT returns the current VALUE of the local data store [substructure] whose DSELECTOR is specified.

Argument/result types:

dselector - DSELECTOR*
value - any

WRDT (PROC 14)
Writes local data store. 7e4

WRDT (dselector, value)

WRDT replaces with a new VALUE the current contents of the local data store [substructure] whose DSELECTOR is specified.

Argument/result types:

dselector - DSELECTOR*
value - any

LCKDT (PROC 15)
Locks local data store. 7e5

LCKDT (dselector, setting, scope, mode -> dlh)

LCKDT locks the local data store whose DSELECTOR is specified, thereby granting processes within the SCOPE of the invoking one either shared or exclusive control of the data store, according to SETTING. LCKDT returns a data lock handle DLH for subsequent use by the invoking process in unlocking the data store via ULKDT.

Shared control prevents outside processes from gaining exclusive control or modifying the data store (via WRDT); exclusive control prevents them from setting the lock in any way or reading the data store (via RDDT).

Either the procedure will be aborted if the proposed setting conflicts with an existing one, or LCKDT will wait if necessary until any conflicting settings are removed, according to MODE.

Argument/result types:

```
dselector - DSELECTOR*
setting   - INDEX [SHARE=1/EXCLUSIVE=2]
scope     - INDEX
mode      - BOOLEAN [WAIT=TRUE/ABORT=FALSE]
dlh       - INDEX
```

ULKDT (PROC 16)
Unlocks local data store, 7e6

ULKDT (dselector, dlh)

ULKDT removes the setting with handle DLH from the local data store whose DSELECTOR is specified, and then invalidates the handle,

Argument/result types:

dselector - DSELECTOR*
dlh - INDEX

Channels 7f

RDMNU (PROC 17)
Reads local channel class menu. 7f1

RDMNU (=> menu)

RDMNU retrieves the list of channel classes supported by the local process.

Argument/result types:

menu = LIST (CLASS*, ...)

ALOPO (PROC 19)
Allocates local port for end of physical channel. 7f2

ALOPO (menu, location, mode => poh, index, port)

ALOPO allocates a local PORT to which a physical channel to a remote process can subsequently be established.

The caller specifies a MENU of channel classes supported by the remote process, and allows the callee to select one which it also supports. The selection process hinges upon whether local and remote processes reside on the same host, as revealed by the remote process' LOCATION.

The local process is to play an active or passive role, according to MODE, when the channel is established.

ALOPO returns the INDEX of the selected channel class and the handle POH for the allocated port.

Argument/result types:

```
menu     - LIST (CLASS*, ...)
location - LOCATION*
mode     - BOOLEAN [ACTIVE=TRUE/PASSIVE=FALSE]
poh      - INDEX
index    - INDEX
port     - PORT*
```

CRTCE (PROC 18)
Creates end of physical channel. 7f3

CRTCE (poh, port, psh)

CRTCE establishes a physical channel between the local port with handle POH and the specified remote PORT.

If PSH is specified, CRTCE also instates the newly-created channel as the control channel to the process with local handle PSH.

Argument/result types:

poh = INDEX
port = PORT*
psh = [INDEX]

RELPO (PROC 20)
[Deletes end of channel and] releases local port. 7f4

RELPO (poh)

RELPO releases the local port with handle POH, deleting the physical channel (if any) which emanates from it, and then invalidates the handle.

Argument/result types:

poh = INDEX

Debugging 7g

SETRC (PROC 23)
Sets DPS-10 trace word. 7g1

SETRC (value)

SETRC replaces with a new VALUE the current contents of word DTRACE in the local process.

DTRACE exists in DPS-10 processes only, where it regulates the output of trace information to the user's controlling TTY.

Argument/result types:

value - INTEGER

```
Appendix A -- PCPB36 Data Structure Format                              8

   Bit      0 If set, key data structure follows                        8a

   Bits  1-13 Unused (zero)                                             8b

   Bits 14-17 Data type                                                 8c

      EMPTY  =1  INTEGER=4  LIST=7                                      8c1
      BOOLEAN=2  BITSTR =5                                              8c2
      INDEX  =3  CHARSTR=6                                              8c3

   Bits 18-20 Unused (zero)                                             8d

   Bits 21-35 Value or its length                                       8e

      EMPTY    unused (zero)                                           8e1
      BOOLEAN  14 zero-bits + 1-bit value (TRUE=1 / FALSE=0)           8e2
      INDEX    unsigned value                                          8e3
      INTEGER  unused (zero)                                           8e4
      BITSTR   unsigned bit count                                      8e5
      CHARSTR  unsigned character count                                8e6
      LIST     unsigned element count                                  8e7

   Bits 36-?? Value                                                     8f

      EMPTY    unused (nonexistent)                                    8f1
      BOOLEAN  unused (nonexistent)                                    8f2
      INDEX    unused (nonexistent)                                    8f3
      INTEGER  two's complement full-word                              8f4
      BITSTR   bit string + zero padding to word boundary              8f5
      CHARSTR  ASCII string + zero padding to word boundary            8f6
      LIST     element data structures                                 8f7
```

```
Appendix B -- PCPB8 Data Structure Format                              9

   Byte    0  Data type (or if zero, key data structure follows)      9a

      EMPTY  =1  INTEGER=4  LIST=7                                     9a1
      BOOLEAN=2  BITSTR =5                                             9a2
      INDEX  =3  CHARSTR=6                                             9a3

   Bytes 1-?  Value                                                   9b

      EMPTY      unused (nonexistent)                                  9b1
      BOOLEAN    7 zero-bits + 1-bit value (TRUE=1 / FALSE=0)          9b2
      INDEX      2-byte unsigned value                                 9b3
      INTEGER    4-byte two's complement value                         9b4
      BITSTR     2-byte unsigned bit count + bit string
                  + zero padding to byte boundary                      9b5
      CHARSTR    2-byte unsigned character count + ASCII string        9b6
      LIST       2-byte element count + element data structures        9b7
```

Appendix C -- Summary of Op Codes

Messages

CALPE	1	Calls procedure in destination process.	
RECPE	2	Recalls procedure in destination process.	
RTNPE	3	Returns results of procedure to destination process.	
ERROR	4	Reports error to destination process.	

System Procedures

INIPS	1	Initializes local process.	
TRMPS	2	Terminates local process.	
CRTCHH	3	Creates half of logical channel.	
DELCHH	4	Deletes half of logical channel.	
OPNPK	5	Opens local packages.	
CLSPK	6	Closes local packages.	
INTPE	7	Interrupts local procedure.	
RSMPE	8	Resumes local procedure.	
ABRPE	9	Aborts local procedure.	
HLPPE	10	Solicits help from local caller.	
CRTDT	11	Creates local data store.	
DELDT	12	Deletes local data store.	
RDDT	13	Reads local data store.	
WRDT	14	Writes local data store.	
LCKDT	15	Locks local data store.	
ULKDT	16	Unlocks local data store.	
RDMNU	17	Reads local channel class menu.	
CRTCE	18	Creates end of physical channel.	
ALOPO	19	Allocates local port for end of physical channel.	
RELPO	20	[Deletes end of channel and] releases local port.	
NTEPE	21	Reports event to local caller.	
SIGPE	22	Signals local procedure.	
SETRC	23	Sets DPS-10 trace word.	

(J26282) 15-AUG-75 18:14;;;; Title: Author(s): James E. (Jim) White/JEW; Distribution: /SRI-ARC([INFO-ONLY]) ; Sub-Collections: SRI-ARC; Clerk: JEW; Origin: < JWHITE, DPSPRO,NLS;2, >, 15-AUG-75 18:12 JEW ;;;; ####;

JEW 15-AUG-75 18:17 26283

DPS-10 Version 2.5 Procedure Directory (SYSGD)

This directory describes DPS-10 as it existed when DPS was cut from NSW.

DPS-10 Version 2,5 Procedure Directory (SYSGD)

```
(abr) %abort?%
PROCEDURE %(=> outcome)%;                                                    1

(abrtn) %abort or return?%
PROCEDURE %(=> outcome)%;                                                    2

(acqpe) %acquire control from remote callee/caller%
PROCEDURE (ch, resl REF %=> outcome, resl REF%);                             3

(addkey) %combine key and data structure%
PROCEDURE (key REF, src REF %=> dst REF%);                                   4

(aloch) %allocate call handle for remote procedure call%
PROCEDURE (psel REF, priority, entry POINTER %=> ch%);                       5

(aloecb) %allocate ECB%
PROCEDURE (length %=> ecb REF%);                                             6

(aloent) %allocate entity%
PROCEDURE (type, length %=> ent REF%);                                       7

(always) %compute UNLCKREC disposition%
PROCEDURE (disp %=> disp%);                                                  8

(blkcpe) %compare blocks%
PROCEDURE (blk1 REF, blk2 REF, length %=> outcome%);                         9

(cainit) %initialize call record%
PROCEDURE (ca POINTER);                                                      10

(calpe) %call remote procedure%
PROCEDURE (psel REF, priority, argl REF, arglmsk REF, reslmsk REF,
resl REF, entry POINTER %=> outcome, resl REF, cost%);                       11

(caterm) %terminate call record%
PROCEDURE (ca POINTER);                                                      12

(chgrec) %change record lock%
PROCEDURE (rec POINTER, type);                                               13

(chndir) %determine logical channel direction%
PROCEDURE (sg POINTER, inph %=> outph, outsh, outdead%);                     14

(clspk) %close remote packages%
PROCEDURE (ph, pkhs REF, costs REF %=> costs REF%);                          15

(cnflsh) %flush channel from system%
PROCEDURE (pch);                                                             16
```

```
(cpestruc) %compare data structures%
PROCEDURE (struc1 POINTER, struc2 POINTER %-> outcome%);                    17

(crtch) %create channel between remote processes%
PROCEDURE (ph1, ph12, ph2, ph21, scope %-> pch, poh1, poh2%);               18

(crtdt) %create remote data store%
PROCEDURE (dsel REF, value REF, scope);                                     19

(crtev) %create local event%
PROCEDURE (scope, length, psichan %-> evh%);                                20

(crtfld) %create folder%
PROCEDURE (def POINTER %-> fld POINTER%);                                   21

(crtlk) %create local lock%
PROCEDURE (scope %-> lkh%);                                                 22

(crtpr) %create local processor%
PROCEDURE (sph, stupinfo REF, scope, priority, spldr %-> pcrh%);            23

(crtps) %create remote process%
PROCEDURE (psaddr REF, userinfo REF, stupinfo REF, scope, pknames
REF, pkstupinfos REF, pkscope, pkhs REF %-> ph, pkhs REF%);                 24

(crtrec) %create record%
PROCEDURE (fld POINTER, scope %-> rec POINTER, rech%);                      25

(crtsp) %create local subprocess%
PROCEDURE (spaddr REF, stupinfo REF, scope, priority, psldr %->
sph%);                                                                      26

(decfstruc) %decode formal data structure%
PROCEDURE (type, src REF, optdst REF %-> dst REF%);                         27

(decistruc) %decode informal data structure%
PROCEDURE (type, src REF, optdst REF %-> dst REF%);                         28

(deckey) %decode data structure key%
PROCEDURE (type, src REF, optdst REF %-> dst REF%);                         29

(decpr) %decode processor parameters%
PROCEDURE (pr POINTER, acs REF, prml REF, cnt, specs REF);                  30

(decprms) %decode parameters according to specs%
PROCEDURE (prml REF, cnt, specs REF %-> prml REF%);                         31

(decstruc) %decode data structure%
PROCEDURE (type, src REF, optdst REF %-> dst REF%);                         32
```

```
(dedprs) %dispose of newly dead processors%
PROCEDURE;                                                          33

(delch) %delete channel between remote processes%
PROCEDURE (pch);                                                    34

(deldt) %delete remote data store%
PROCEDURE (dsel REF);                                               35

(delev) %delete local event%
PROCEDURE (evh);                                                    36

(delfld) %delete folder%
PROCEDURE (fld POINTER);                                            37

(dellk) %delete local lock%
PROCEDURE (lkh);                                                    38

(delpr) %delete local processor%
PROCEDURE (pcrh, okifldr %-> cost%);                                39

(delps) %delete remote process%
PROCEDURE (ph %-> cost%);                                           40

(delrec) %delete record%
PROCEDURE (rec POINTER);                                            41

(delsp) %delete local subprocess%
PROCEDURE (sph, okifldr %-> cost%);                                 42

(descrb) %construct block list descriptor%
PROCEDURE (src REF %-> descr%);                                     43

(descrl) %construct list list descriptor%
PROCEDURE (src REF %-> descr%);                                     44

(dpsmain) %main program%
PROCEDURE;                                                          45

(dpsrcvr) %recovery program%
PROCEDURE (type, message REF, addr);                                46

(drdps) %discard results of system call%
PROCEDURE (pr POINTER, acs REF);                                    47

(dtinit) %initialize data store record%
PROCEDURE (dt POINTER);                                             48
```

```
(dtterm) %terminate data store record%
PROCEDURE (dt POINTER);                                                    49

(encfstruc) %encode formal data structure%
PROCEDURE (type, src REF %-> dst REF%);                                    50

(encistruc) %encode informal data structure%
PROCEDURE (type, src REF %-> dst REF%);                                    51

(encpr) %encode processor parameters%
PROCEDURE (pr POINTER, acs REF, prml REF, cnt, specs REF);                 52

(encprms) %encode parameters according to specs%
PROCEDURE (prml REF, cnt, specs REF, dst REF %-> prml REF%);               53

(encstruc) %encode data structure%
PROCEDURE (type, src REF %-> dst REF%);                                    54

(erdps) %retrieve diagnostic message for DPS error%
PROCEDURE (pr POINTER, acs REF);                                           55

(err) %L10 run-time error%
PROCEDURE (message REF);                                                   56

(evflsh) %flush event from system%
PROCEDURE (evh);                                                           57

(evterm) %terminate event record%
PROCEDURE (ev POINTER);                                                    58

(falopo) %allocate local inter-fork port%
PROCEDURE (po POINTER, remloc REF %-> locport REF%);                       59

(fcrtce) %create end of inter-fork channel%
PROCEDURE (po POINTER, remport REF);                                       60

(fcrtps) %create fork%
PROCEDURE (po POINTER, action, host REF, intrahostaddr REF, userinfo
REF, stupinfo REF, remloc REF %-> remloc REF%);                            61

(fdelce) %delete end of inter-fork channel%
PROCEDURE (po POINTER);                                                    62

(fdelps) %delete fork%
PROCEDURE (po POINTER);                                                    63

(fndcall) %locate procedure call record%
PROCEDURE (calltype, ch, type %-> ca, otherch%);                           64
```

```
(fndcdt) %locate data store record%
PROCEDURE (dsel REF, type %-> dt POINTER%);                                    65

(fnddsd) %locate informal data structure definition%
PROCEDURE (type %-> entry POINTER%);                                           66

(fndelm) %locate element of externally-formatted list%
PROCEDURE (list REF, esel REF %-> elem REF, offset%);                          67

(fnderm) %locate error message for DPS error%
PROCEDURE (number %-> msg REF%);                                               68

(fndmsg) %locate message table entry%
PROCEDURE (number %-> entry POINTER%);                                         69

(fndpr) %locate processor with specified fork handle%
PROCEDURE (fkh %-> pcrh%);                                                     70

(fndrec) %locate record%
PROCEDURE (fld POINTER, rech, type %-> rec POINTER, rech%);                    71

(fndrps) %locate process with incoming message%
PROCEDURE %(-> ph)%;                                                           72

(fndsyp) %locate system procedure table entry%
PROCEDURE (number %-> entry POINTER%);                                         73

(frcvch) %receive data from fork%
PROCEDURE (po POINTER %-> block REF%);                                         74

(frelpo) %release local port%
PROCEDURE (po POINTER);                                                        75

(fsndch) %send data to fork%
PROCEDURE (po POINTER, block REF);                                             76

(fupdpo) %update fork state%
PROCEDURE (po POINTER %-> aliveornot%);                                        77

(gtdps) %get user call arguments from DPS%
PROCEDURE (pr POINTER, acs REF, optush, optrbaddr);                            78

(hlppe) %solicit help from remote caller%
PROCEDURE (pblm, desc REF %-> solution REF%);                                  79

(ialopo) %allocate local port%
PROCEDURE (menu REF, remloc REF, active, scope %-> poh, mnuindex,
locport REF%);                                                                 80
```

```
(iclspk) %close package%
PROCEDURE (pkh %-> cost%);                                              81

(icrtce) %create local channel end%
PROCEDURE (pqh, remport REF);                                           82

(icrtdt) %create data store record%
PROCEDURE (dsel REF, scope %-> dt POINTER%);                            83

(icrtps) %create process%
PROCEDURE (psaddr REF, userinfo REF, stupinfo REF, scope,
intrahostaddr REF %-> poh, intrahostaddr REF%);                         84

(idelce) %delete local channel end%
PROCEDURE (poh);                                                        85

(idelps) %delete process%
PROCEDURE (poh);                                                        86

(ifrtn) %compute UNLCKREC disposition%
PROCEDURE (disp %-> disp%);                                             87

(infps) %retrieve information about remote process%
PROCEDURE (ph, type %-> info%);                                         88

(inilcb) %initialize LCB%
PROCEDURE (lcb POINTER);                                                89

(inirb) %initialize resource block%
PROCEDURE (pr POINTER);                                                 90

(intpe) %interrupt remote callee%
PROCEDURE (ch);                                                         91

(iopnpk) %open package%
PROCEDURE (pkname REF, stupinfo REF, scope %-> pkh%);                   92

(ircvch) %receive data from channel%
PROCEDURE (poh %-> block REF%);                                         93

(irelpo) %release local port%
PROCEDURE (poh);                                                        94

(isndch) %send data on channel%
PROCEDURE (poh, block REF);                                             95

(itdfk) %introduce fork to DpS%
PROCEDURE (fkh, stupinfo REF, self %-> pcrh%);                          96
```

```
(itdps) %introduce remote processes%
PROCEDURE (ph1, stupinfo1 REF, ph2, stupinfo2 REF, scope, flags %->
flags+ih, ph12, ph21,%);                                          97

(itstpo) %test local port for incoming message%
PROCEDURE (poh %-> outcome%);                                     98

(iupdpo) %update port state%
PROCEDURE (poh %-> aliveornot%);                                  99

(ivdps) %invoke system call%
PROCEDURE (pr POINTER, acs REF);                                  100

(l10cal) %call L10 procedure%
PROCEDURE (entry POINTER, arg1 REF, res1 REF);                    101

(lckdt) %lock remote data store%
PROCEDURE (dsel REF, type, scope, flags %-> dtlh%);               102

(lckrec) %lock record%
PROCEDURE (rec POINTER, type);                                    103

(lkinit) %initialize lock record%
PROCEDURE (lk POINTER);                                           104

(lkterm) %terminate lock record%
PROCEDURE (lk POINTER);                                           105

(lsinit) %initialize lock set record%
PROCEDURE (ls POINTER);                                           106

(mgcover) %manager initiator%
PROCEDURE;                                                        107

(mginit) %initialize manager record%
PROCEDURE (mg POINTER);                                           108

(mgterm) %terminate manager record%
PROCEDURE (mg POINTER);                                           109

(mkarray) %construct array from list%
PROCEDURE (src REF, dst REF, xfrornot);                           110

(msgmgr) %message manager%
PROCEDURE (ph);                                                   111

(mskarlst) %mask argument/result list%
PROCEDURE (type, prm1 REF, mask REF %-> prm1 REF%);               112
```

```
(nalopo) %allocate local inter-network-process port%
PROCEDURE (po POINTER, remloc REF %-> locport REF%);                    113

(ncrtce) %create end of inter-network-process channel%
PROCEDURE (po POINTER, remport REF);                                    114

(ncrtps) %create network process%
PROCEDURE (po POINTER, action, host REF, intrahostaddr REF, userinfo
REF, stupinfo REF, remloc REF %-> remloc REF%);                         115

(ndelce) %delete end of inter-network-process channel%
PROCEDURE (po POINTER);                                                 116

(ndelps) %delete network process%
PROCEDURE (po POINTER);                                                 117

(nodedlck) %check lock request for freedom from deadlock%
PROCEDURE (lckid REF, pcrid REF, endlckid REF, endpcrid REF);           118

(nrcvch) %receive data from network process%
PROCEDURE (po POINTER %-> block REF%);                                  119

(nrelpo) %release local inter-network-process port%
PROCEDURE (po POINTER);                                                 120

(nsndch) %send data to network process%
PROCEDURE (po POINTER, block REF);                                      121

(ntepe) %make event known to remote caller%
PROCEDURE (event, desc REF);                                            122

(nupdpo) %update network process state%
PROCEDURE (po POINTER %-> aliveornot%);                                 123

(oactchn) %activate signal channel%
PROCEDURE (channel, haddr);                                             124

(oalobit) %allocate map bit%
PROCEDURE (map REF, length %-> number%);                                125

(obgnfk) %begin fork%
PROCEDURE (fkh);                                                        126

(oblkxfr) %copy block%
PROCEDURE (src REF, dst REF, length);                                   127

(ochnmsk) %construct channel mask%
PROCEDURE (channel %-> mask%);                                          128
```

```
(ocnnfk) %interconnect address spaces of two forks/files%
PROCEDURE (mainfkh, mainpg, auxfkh, auxpg, pgcnt %-> pgcnt%);                129

(ocost) %fetch current cost of fork%
PROCEDURE (fkh %-> cost%);                                                   130

(ocrtfk) %create fork%
PROCEDURE (filename REF %-> fkh%);                                           131

(odcnfk) %disconnect address spaces of two forks/files%
PROCEDURE (auxfkh, auxpg, pgcnt);                                            132

(odelfk) %delete fork%
PROCEDURE (fkh);                                                             133

(odirfk) %connect fork to directory%
PROCEDURE (fkh, directory REF, password REF);                                134

(oendfk) %end fork%
PROCEDURE (fkh);                                                             135

(ofndfk) %identify fork initiating DPS operation%
PROCEDURE %(-> fkh)%;                                                        136

(oinios) %initialize operating system%
PROCEDURE;                                                                   137

(oiptstr) %input ASCIZ to L10 string%
PROCEDURE (srcbp, dst REF %-> dst REF%);                                     138

(oitdfk) %introduce fork%
PROCEDURE (locfkh, targetfkh %-> fkh%);                                      139

(okwall) %check lock request for compatibility with existing locks%
PROCEDURE (lckid REF, pcrid REF %-> outcome%);                               140

(okwone) %check two locks for compatibility%
PROCEDURE (lckida REF, pcrida REF, lckidb REF, pcridb REF %->
outcome%);                                                                   141

(omovfk) %move fork%
PROCEDURE (srcfkh, dstfkh);                                                  142

(ooptstr) %output L10 to ASCIZ string%
PROCEDURE (src REF, dst REF %-> bp%);                                        143

(opnpk) %open remote packages%
PROCEDURE (ph, pknames REF, stupinfos REF, scope, pkhs REF %-> pkhs
REF%);                                                                       144
```

```
(opntstr) %return byte pointer to L10 string%
PROCEDURE (src REF %-> bp%);                                    145

(opntwrd) %return byte pointer to word%
PROCEDURE (addr, bytesize %-> bp%);                             146

(oprmgr) %operation manager%
PROCEDURE (pcrh);                                               147

(oprtstr) %print string at user's terminal%
PROCEDURE (text REF);                                           148

(ordacs) %read fork's ACs%
PROCEDURE (fkh, dst REF);                                       149

(ordbyt) %read byte%
PROCEDURE (bp REF %-> byte%);                                   150

(orelbit) %release map bit%
PROCEDURE (map REF, length, number);                            151

(orsmfk) %resume fork after DPS operation%
PROCEDURE (fkh, pcincr);                                        152

(osepfk) %separate fork%
PROCEDURE (fkh);                                                153

(osettmr) %set interval timer%
PROCEDURE (interval);                                           154

(osigerr) %operating system error%
PROCEDURE (number);                                             155

(osigfk) %signal fork%
PROCEDURE (fkh, channel);                                       156

(ostrsiz) %return length of ASCIZ string%
PROCEDURE (bp REF, boundary %-> length, overflow%);             157

(ostrxfr) %copy string%
PROCEDURE (srcbp, dstbp, length);                               158

(ostsfk) %return status of fork%
PROCEDURE (fkh %-> aliveornot%);                                159

(otest) %test for signal%
PROCEDURE (ecb REF, removeornot %-> code, newlength%);          160
```

```
(otrmos) %terminate operating system%
PROCEDURE;                                                          161

(otrpfk) %intercept DPS operations%
PROCEDURE (fkh);                                                    162

(otsttmr) %test/cancel interval timer%
PROCEDURE (cancel %-> gone, left%);                                 163

(outrfk) %release DPS operations%
PROCEDURE (fkh);                                                    164

(owait) %wait for any of a list of events%
PROCEDURE (ecbs REF, listornot %-> code, index, length%);           165

(owracs) %write fork's ACs%
PROCEDURE (fkh, src REF);                                           166

(owtsig) %wait for signal%
PROCEDURE;                                                          167

(pgdps) %return/get user call results/arguments%
PROCEDURE (pr POINTER, acs REF);                                    168

(pkflsh) %flush package from system%
PROCEDURE (pkh);                                                    169

(poflsh) %flush port from system%
PROCEDURE (poh);                                                    170

(poinit) %initialize port record%
PROCEDURE (po POINTER);                                             171

(poplcb) %pop LCB lock%
PROCEDURE;                                                          172

(porstruc) %portray data structure%
PROCEDURE (src REF, dst REF %-> dst REF%);                          173

(prflsh) %flush processor from system%
PROCEDURE (pcrh);                                                   174

(prhand) %make processor handle absolute%
PROCEDURE (pcrh %-> pcrh%);                                         175

(prinit) %initialize processor record%
PROCEDURE (pr POINTER);                                             176
```

```
(prterm) %terminate processor record%                                    177
PROCEDURE (pr POINTER);

(psflsh) %flush process from system%                                     178
PROCEDURE (ph);

(pshand) %make process handle absolute%                                  179
PROCEDURE (ph %-> ph%);

(pshlcb) %push LCB lock%                                                 180
PROCEDURE (lcb POINTER, type, handleornot);

(psterm) %terminate process record%                                      181
PROCEDURE (ps POINTER);

(ptdps) %return user call results to DPS%                                182
PROCEDURE (pr POINTER, acs REF, optush, optoutcome);

(rcvch) %receive data on local channel%                                  183
PROCEDURE (poh %-> block REF%);

(rcvps) %receive message from adjacent process%                          184
PROCEDURE (ph, msg REF %-> msg REF%);

(rddt) %read remote data store%                                          185
PROCEDURE (dsel REF %-> value REF%);

(rdpr) %read data from processor%                                        186
PROCEDURE (type, pr POINTER, src, occasionaldst %-> dst REF%);

(rdpras) %view entity in processor's address space%                      187
PROCEDURE (type, pr POINTER, praddr %-> cfaddr REF, length%);

(rdrec) %retrieve information about record%                              188
PROCEDURE (type, rec POINTER %-> value%);

(rdypr) %ready local processor for service request assignment%           189
PROCEDURE;

(rececb) %recall signal%                                                 190
PROCEDURE (ecb REF, code);

(relch) %[abort remote callee and] release call handle%                  191
PROCEDURE (ch %-> cost%);

(relent) %release entity%                                                192
PROCEDURE (type, ent REF);
```

```
(relpe) %release control to remote callee/caller%
PROCEDURE (ch, argl REF, arglmsk REF, reslmsk REF, acqevh);                193

(remlcb) %unlock LCB%
PROCEDURE (ldsc);                                                          194

(remlk) %remove local lock%
PROCEDURE (lkh, lsh);                                                      195

(rrdps) %retrieve results of system call%
PROCEDURE (pr POINTER, acs REF, optsyh);                                   196

(rsmpe) %resume remote callee%
PROCEDURE (ch);                                                            197

(rsmpr) %resume processor after DPS operation%
PROCEDURE (pr POINTER, outcome, acs REF);                                  198

(rtn) %return?%
PROCEDURE %(=> outcome)%;                                                  199

(savent) %save entity%
PROCEDURE (type, src REF %=> dst REF%);                                    200

(savstruc) %save data structure%
PROCEDURE (src REF %=> dst REF%);                                          201

(selipc) %select IPC subroutine%
PROCEDURE (po POINTER, number %=> procaddr%);                              202

(selmgr) %select next manager for dispatch%
PROCEDURE;                                                                 203

(sepfk) %separate fork from DPS%
PROCEDURE (pcrh, self);                                                    204

(sepps) %separate remote processes%
PROCEDURE (ih %=> cost1, cost2%);                                          205

(setlcb) %lock LCB%
PROCEDURE (lckid REF, ecb REF, handleornot %=> ldsc%);                     206

(setlk) %set local lock%
PROCEDURE (lkh, type, scope, flags %=> lsh%);                              207

(setmr) %set virtual interval timer%
PROCEDURE (interval, ecb REF, evh, scope %=> tmh%);                        208
```

```
(setrc) %set remote DPS trace word%
PROCEDURE (ph, setting);                                                209

(sigecb) %signal event%
PROCEDURE (ecb REF, code);                                              210

(sigerr) %DPS error%
PROCEDURE (number);                                                     211

(sigev) %signal local event%
PROCEDURE (evh, code, wheelornot);                                      212

(sigpe) %signal remote callee/caller%
PROCEDURE (ch, argl REF, arglmsk REF);                                  213

(sipr) %sign in local processor%
PROCEDURE (psname REF, pknames REF, shdpgs, psichn, flags %->
stupinfo REF, juevh, oflags%);                                          214

(sizent) %compute length of entity%
PROCEDURE (type, ent REF %-> length%);                                  215

(sizfld) %compute size of folder%
PROCEDURE (fld POINTER %-> reccount%);                                  216

(sizstruc) %compute size of data structure%
PROCEDURE (src POINTER %-> blksize, nokeysize%);                        217

(sndch) %send data on local channel%
PROCEDURE (poh, block REF);                                             218

(sndps) %send message to process%
PROCEDURE (ph, msg REF);                                                219

(sopr) %sign out local processor%
PROCEDURE;                                                              220

(spawn) %spawn manager%
PROCEDURE (procaddr, stupinfo, rscnt);                                  221

(suflsh) %flush subprocess from system%
PROCEDURE (sph);                                                        222

(suhand) %make subprocess handle absolute%
PROCEDURE (sph %-> sph%);                                               223

(suinit) %initialize subprocess record%
PROCEDURE (su POINTER);                                                 224
```

```
(suterm) %terminate subprocess record%
PROCEDURE (su POINTER);                                                    225

(swplcb) %swap LCB lock%
PROCEDURE (lcb POINTER, type);                                             226

(sysbgn) %begin remote system procedure%
PROCEDURE (ph, number, arg1, arg2, arg3, arg4, arg5, arg6, arg7 %=>
ch%);                                                                      227

(syscal) %call remote system procedure%
PROCEDURE (ph, number, arg1, arg2, arg3, arg4, arg5, arg6, arg7 %=>
EMPTY, res1, res2, res3%);                                                 228

(sysend) %resume remote system procedure%
PROCEDURE (ch %=> EMPTY, res1, res2, res3%);                               229

(syterm) %terminate system call record%
PROCEDURE (sy POINTER);                                                    230

(trcerr) %trace error%
PROCEDURE (number, msg REF);                                               231

(trcmg) %trace manager%
PROCEDURE (mode, mgh);                                                     232

(trcpo) %trace message%
PROCEDURE (mode, poh, block REF);                                          233

(trcpr) %trace user/system call arguments/results%
PROCEDURE (calltype, number, entry POINTER, prmltype, prml REF);           234

(trmlcb) %terminate LCB%
PROCEDURE (lcb POINTER);                                                   235

(trmrb) %terminate resource block%
PROCEDURE (pr POINTER);                                                    236

(tstev) %test for signalled local event%
PROCEDURE (evh %=> code, newlength%);                                      237

(tstmr) %test/cancel virtual interval timer%
PROCEDURE (tmh, flags %=> gone, left%);                                    238

(ulkdt) %unlock remote data store%
PROCEDURE (dsel REF, dtlh);                                                239

(unlckrec) %unlock record%
PROCEDURE (rec POINTER, disp);                                             240
```

```
(updtmr) %update virtual interval timers%
PROCEDURE;                                                              241

(usinit) %initialize user call record%
PROCEDURE (us POINTER);                                                 242

(usterm) %terminate user call record%
PROCEDURE (us POINTER);                                                 243

(vispe) %visit remote callee/caller%
PROCEDURE (ch, arg1 REF, arg1msk REF, res1msk REF, res1 REF %->
outcome, res1 REF%);                                                    244

(vjusr) %call user%
PROCEDURE (sph, pcrh, number %optional priority in LH%, arg1, arg2,
arg3, arg4, arg5, arg6, arg7 %-> cost, res1, res2, res3%);              245

(waicok) %wait for signal COK%
PROCEDURE (ecb REF);                                                    246

(waiecb) %wait for signal%
PROCEDURE (ecb REF %-> code%);                                          247

(waiev) %wait for signalled local event%
PROCEDURE (evhs REF %-> code, index, length%);                          248

(winscope) %verify one processor within scope of another%
PROCEDURE (pcrid1 REF, scope, pcrid2 REF %-> outcome%);                 249

(wrdt) %write remote data store%
PROCEDURE (dsel REF, value REF);                                        250

(wrpr) %write data to processor%
PROCEDURE (type, pr POINTER, src REF %-> dst%);                         251

(wrpras) %allocate entity in processor's address space%
PROCEDURE (type, pr POINTER, length %-> cfaddr, praddr%);               252

(xabrpe) %abort local procedure%
PROCEDURE (ch);                                                         253

(xalopo) %allocate local physical port%
PROCEDURE (chntypmnu REF, remloc REF, active %-> poh, mnuindex,
locport REF%);                                                          254

(xcalpe) %call local procedure%
PROCEDURE (ch, pkh, pname REF, arg1 REF, arg1msk REF, res1msk REF,
priority);                                                              255
```

```
(xclspk) %close local packages%
PROCEDURE (pkhs REF, costs REF %-> costs REF%);                          256

(xcrtce) %create local physical channel end%
PROCEDURE (poh, remport REF, ph);                                        257

(xcrtchh) %create local logical channel half%
PROCEDURE (psdesc REF, stupinfo REF, chainsh, modelsh %-> sh, ph%);      258

(xcrtdt) %create local data store%
PROCEDURE (dsel REF, value REF, scope);                                  259

(xdelchh) %delete local logical channel half%
PROCEDURE (sh %-> cost%);                                                260

(xdeldt) %delete local data store%
PROCEDURE (dsel REF);                                                    261

(xerror) %error%
PROCEDURE (number, msg REF);                                             262

(xhlppe) %solicit help from local caller%
PROCEDURE (ch, pblm, desc REF %-> solution REF%);                        263

(xinips) %initialize process%
PROCEDURE (intrahostaddr REF, userinfo REF, stupinfo REF, spsdesc
REF, pkinfo REF, pkhs REF %-> fpsdesc REF, pkhs REF%);                   264

(xintpe) %interrupt local procedure%
PROCEDURE (ch);                                                          265

(xlckdt) %lock local data store%
PROCEDURE (dsel REF, type, scope, wait %-> dtlh%);                       266

(xntepe) %make event known to local caller%
PROCEDURE (ch, event, desc REF);                                         267

(xopnpk) %open local packages%
PROCEDURE (pknames REF, stupinfos REF, scope, pkhs REF %-> pkhs
REF%);                                                                   268

(xrddt) %read local data store%
PROCEDURE (dsel REF %-> value REF%);                                     269

(xrdmnu) %read channel-type menu%
PROCEDURE %(-> menu REF)%;                                               270

(xrecpe) %recall local procedure%
PROCEDURE (ch, arg1 REF, arg1msk REF, reslmsk REF);                      271
```

```
(xrelpo) %[delete channel end and] release local port%
PROCEDURE (poh);                                                    272

(xrsmpe) %resume local procedure%
PROCEDURE (ch);                                                     273

(xrtnpe) %return to local caller%
PROCEDURE (ch, res1 REF, res1msk REF, arg1msk REF, outcome, cost);  274

(xsetrc) %set DPS trace word%
PROCEDURE (setting);                                                275

(xsigpe) %signal local procedure%
PROCEDURE (ch, arg1 REF, arg1msk REF);                              276

(xtrmps) %terminate process%
PROCEDURE %(=> cost)%;                                              277

(xulkdt) %unlock local data store%
PROCEDURE (dsel REF, dtlh);                                         278

(xwrdt) %write local data store%
PROCEDURE (dsel REF, value REF);                                    279
```

DPS-10 Version 2,5 Procedure Directory (SYSGD)

(J26283) 15-AUG-75 18:17;;;; Title: Author(s): James E. (Jim) White/JEW; Distribution: /SRI-ARC([INFO-ONLY]) ; Sub-Collections: SRI-ARC; Clerk: JEW; Origin: < JWHITE, TOC,NLS;2, >, 15-AUG-75 18:15 JEW ;;;;####;

26283 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

reply to Steve Warshall's note on Frontend Requirements

Introduction 1

After we sent out the recent look at PDP-11 memory requirements there was a flurry of activity and comments. One being the note by Steve Warshall. Steve's note was clearly designed to stimulate discussion and help get the issues and requirements clarified. Since the requirements for various pieces of the NSW system framework have never been really made explicit, there is clearly a need for dialog in this area. Toward this end I put together some comments and economic perspective on some of his statements which in no sense are complete. 1a

I had intended to do more but events have been moving so fast that it does not seem useful elaborating or polishing this document given the fact we are looking in considerable detail at the whole spectrum of possible Frontend facilities that could be provided and their memory and other requirements. In any case the comments below may still have some use so I pass them along now. 1b

As is pointed out below, the relative cost of a PDP-11 system with or without a disk, is insignificant. A similar argument can be made for the 11/70, that has a cost for comparable configurations about $20k higher than a 11/40. There may be factors other than economics arguing against the use of a disk, or an 11/70 such as possible procurement regulation problems etc., and we would find it useful to have them brought explicitly to the surface. 1c

There are important cases where what Steve's model yields is economically disadvantageous and expensive where our model yields significant savings. 1d

What we hope to show in the economic argument below is that even if you assume Steve presented a workable or complete Frontend model for meeting his requirements, and accept essentially intact his resource estimates, that simple economic analysis shows that the cost saving of Steve's model, over the one we have been working on in the best cases where Steve's model yields cost savings, is very small and seems essentially down in the noise. 1e

The Frontend Requirements 2

Steve lists some requirements in the user interface domain he thinks should probably be met as a minimum. 2a

1) A help capability 2a1

2) Intracommand editing 2a2

3) Command completion 2a3

4) Common command completion character or signal and other miscellaneous control characters 2a4

5) Ability to escape to the NSW Exec(Works manager and Frontend commands) 2a5

6) Ability to filter or transform certain character sequences. 2a6

He expressed some reservations about number 3, but anyway we can think of achieving number 1 yields number 3 also. Before going on to say a few words about why we agree that the above requirements are important, let us discuss some other requirements that we think Steve would probably agree are important and which impact decisions about a Frontend model. 2b

7) Minimize communication costs. Most users of the ARPANET presently treat it as a free resource. Billing is likely to begin soon on a tariff basis charging by the packet as do the present commercial packet switched services such as Telenet. As we show below, communication costs to operate character-at-a-time full duplex tools are likely to be high. If we want to provide the good human engineering only possible in a character-at-a-time full duplex environment, communication costs must be considered. 2b1

The alternative is to restrict NSW tools to operate in a line-at-a-time half duplex environment. It is because of communication costs and CPU cycle costs that so many line-at-a-time half duplex tools presently exist. The Frontend concept allows us to break out of such a restrictive environment while yielding the cost savings of the line-at-a-time half duplex approach. The FE should work equally well with both tools of line-at-a-time half duplex, and character-at-a-time full duplex capabilities. 2b1a

A future NSW system architecture that might utilize local clusters of TBH's for running the most frequently used tools of a given installation would clearly allow low cost local communications. This possibility could alter significantly the considerations and arguments for various Frontend approaches. 2b1b

In summary choices made in the total NSW system design that impact on communication costs have very significant impact on Frontend design and we need early input from the Steering Committee about their assumptions and requirements in this area. 2b1c

8) Remove from the Tool Bearing Hosts (TBH) frequently

performed tasks, minimize TBH character handling, and the cost of frequent process activation presently being paid in character-at-a-time full duplex systems. 2b2

The economic savings here and in the requirement above can be substantial as shown below. We believe the savings in communication costs and TBH cycles in these two requirements possible through use of local computing are such that there will be a steady trend to move functions out toward locality with the user. 2b2a

9) Provide a responsive human engineered environment for the user. 2b3

Delays introduced by the network and TBH activations are unnecessarily frustrating to the user during command specification. Acceptance of the system and his productivity are increased if we can improve responsiveness wherever possible at reasonable cost. We do not emphasize this point because it seems obvious to us from our thousands and thousands of console hours of interactive experience, and most importantly the environment to satisfy it falls out from meeting the other requirements, if the Frontend is not overloaded with too many terminals. We assume as does Steve the desirability to support 20 users on an 11/40 FE. 2b3a

10) Allow an environment that effectively supports both simple and more sophisticated tools. The distinction involves factors such as richness of capability and richness of the command language/user interface. While there are presently many useful tools that perform a small number of functions with very simple command languages, and we know there will be more in the future, there are also in existence and evolving more sophisticated tools with the rich command languages needed in such cases. NLS is an example of a more sophisticated tool. The direction that mail systems are evolving with their rich command languages to handle sending, receiving/reading, filing, and eventually retrieval is another example. Further, if one looks at the very long list of tool capabilities useful in the NSW that J.C.R. Licklider passed out at the last ARPA Principle Investigator meeting, one sees both the extent of new tool developments likely in the future and the number of application tool domains that will yield more sophisticated tools. One only has to read some of the recent NSW Steering Committee planning documents to see the range and extent of tools desired. 2b4

Steve listed three motivations he understood were the main ones behind our approach. 2c

1) Reduce communication costs because the users errors are caught locally. 2c1

2) Simplify and reduce new tool building cost by providing a virtual terminal and set of user interface services to new or old tool builders. 2c2

3) Enforce stylistic uniformity of the user interface on tool builders. 2c3

Let us consider our views on these. Number 1, above, while attainable as a by-product, we would agree is not a significant economic factor and we are not sure why Steve thought we considered it significant. 2d

Number 2, above is economically and conceptually important and is discussed in the next section, but we would only consider it a secondary benefit of our approach to Frontend design. 2e

Number 3, above has been somewhat muddled in Steve's discussion. We do feel that in an environment with large numbers of tools, as we expect over time to be the case for the NSW, users who are experts or skilled in some tools or subsets of commands of tools are going to be novices in others they use less frequently or approach for the first time. We feel an environment can be provided whereby certain stylistic, syntactic features across tools can have some coherence and consistency. We feel more consistency will be helpful to the user. We believe this user interface environment can also be rich enough so that tool builders do not feel restricted. 2f

Our experience in using both NLS and a range of tools across ARPANET hosts show the clear value more consistency has for the user. We have no interest in enforcement per se, in fact, we feel adequate escape hatches need to exist wherever normal usage might restrict some functions or new approach. We do think however, that if a set of services and facilities can be provided (an environment) that would encourage more consistency wherever it makes little real difference, then tool builders will take advantage of it, appreciate it and both they and the user will benefit. 2f1

In summary, then, the three justifications listed by Steve as his understanding of our motivations for our approach are not the main ones. Instead, our goal is to meet the five requirements he listed in a way that also satisfies the four additional ones above. 2g

Let us now say a few words about why we agree that Steve's list contains functions that should be listed as minimal requirements. 2h

1) A help capability. 2h1

We believe that in the NSW that will mature there will be many tools of both simple and sophisticated types. There will be many kinds of users, secretaries, managers, writers, programmers, analysts, etc, who will span the spectrum from "every-day" expert users of computer based tools to infrequent or casual users of computer based tools. Even the expert users will occassionally need to use tools or functions in tools that they are not familiar with and need refreshing about what can be done next or how to learn to accomplish a certain function. For example, we find ourselves reasonably frequently using the ? facility in NLS when using an infrequently used but needed command. Or using the help facilities available in FTP Mailsys, or Telnet when we need some infrequently used function or to find out how to perform a function we are sure must exist, but have never needed before. These facilities are used both at the beginning and in the middle of commands. Our frustration cost is the inconsistency in how help is invoked. 2h1a

The value of these facilities cannot be given a dollar value by us, but common usage has clearly established their need. 2h1b

Steve mentioned ? type help, but others are possible as part of the user interface design such as noise words, (extra words echoed to the user to help keep him oriented about what is expected next) command completion, prompts, etc. These generally are easily provided if a full ? type facility can be provided. 2h1c

Entry to more extensive online documentation help or tutorial data bases can and probably should be provided to ease training and hand holding costs over time. These are not discussed much here as we do not want to stir side arguments except to point out that our approach to the Frontend provides the capability to more precisely determine the initial entry point to such facilities and provides a central point for data gathering that could be used by more active tutor processes. 2h1d

2) Intracommand editing 2h2

The simplist such facility is the backspace character facility to erase the last character typed. Similarly the

last word or line can be erased or the edited line retyped freshly on the page, Deleting the entire command is also commonly found, Backspace word can be given extended meaning to backout a whole field during command specification, Backspacing a screen selection is also useful on displays. 2h2a

One can imagine more extensive editing facilities for editing arbitrary words or hunks of text during literal typein, Our approach can provide a full set of such facilities, All such editing capabilities that minimize pain when an error has occurred are useful, The richer the set of capabilities, the more convenient for the user, 2h2b

Command Completion 2i

Steve indicates his doubt about the need here, Our experience indicates clearly the utility and importance of command completion, Associated with command completion is usually a set of "noise words" that are also fed back to aid the user in knowing where he is and what to expect next, This philosphy is used throughout NLS and by BBN in their systems, for example, we have a large user group using Office-1 made up of secretaries, managers, technical people, people from a variety of professions, We are extensively involved in training, handling feedback and so forth with them, Again and again the importance of command completion and noise words is mentioned for aiding their use of the Tenex and NLS systems and how much they like it, Our User Options capability allows people to choose the mode of command recognition, the amount of completion and so forth they desire based on their terminal type and past background, It can be completely turned off, However, given reasonable response from the host and network even highly skilled and experienced people tend to prefer to utilize such facilities. 2i1

We personally use it extensively and find it valuable when using new or infrequently used commands. 2i2

Common Command Completion and Common Escape to NSW Exec and other Signals 2j

some tools terminate commands with control characters of various flavors, others use carriage returns, and other special signals, On the present ARPANET a variety of characters and mechanisms are used to signal the system executive or to stop a printout etc, This variety leads to confusion and in the ARPANET has been the cause of a lot of criticisms. 2j1

Agreeing on some standard characters for these functions is clearly useful, 2j2

Filtering out or Transforming Characters 2k

We are not quite sure what Steve has in mind here, 2k1

Finally one function that may not be in the minimal required set, but is easy to implement with any reasonable Frontend model is a User Option capability to allow the user to set standard characters to ones convenient from his terminal, adjust command recognition mode, amount of feedback etc. We have found such a capability settles a lot of arguments and allows stylistic adjustments that users seem to need, 21

Alternative Models to Meet the Above Requirements 2m

The essence of the line of thought that will evolve below is that to meet the above described requirements of Steve and the desire to minimize communication costs, that we think he would probably agree with, will require an approach essentially equivalent to or very close to what we have built. 2m1

Steve thought about 10k of code was needed for the user interface functions. Depending on how one counts the code needed to interface to the protocols, as being part of the protocols or user interface code, we can come up with different numbers for our approach. Our user CLI code is under 12k with about another 2k required for the protocol interface. The only real issue is the size of per user context required. We are looking at how to minimize this. 2m2

The argument goes roughly as follows, 2m3

1) Single character interactions can be very expensive in a packet communication system. If the reader has not been aware of this, we recommend that he turn to the next section where some examples and sample numbers are computed. This implies to us that one wants to collect as much of the command as possible before transmission to the tool. As indicated earlier alternative NSW system designs that utilize extensive local communication capabilities would alter the argument. 2m3a

Either the approach of just collecting the command string as Steve would advocate or to parse the command or just collect the string as our approach could do would meet this important need to cut the very expensive potential communication cost. 2m3b

reply to Steve Warshall's note on Frontend Requirements

2) If we are going to be able to do intracommand editing at the Frontend more extensive than just backspace character or word, then the input material and state of things during command specification must remain at the Frontend until the command is completely specified. 2m3c

3) Now we want to provide ? and possibly other simple help capabilities like prompts. Noise words and command completion are also desirable. 2m3d

The only way the Frontend can know when to expect a command word or what types of things are legal in an argument (clearly some classes of arguments can not be interpreted by the FE and must just be collected as strings for the tool to analyze) is that it must be sent ahead of time some sort of representation of the commands that are legitimate at that point. This representation must indicate what the field separators are so that the Frontend will know when a field has been completed, and the characters to look for before performing command completion, and it must be sent the completion and noise word characters to echo back. 2m4

If the command languages are stylistically restricted to being very simple such as of form <command word> <arg>---<arg> <command completion> with agreed upon ahead of time characters for field separators, this information can be sent to the Frontend as the simplest kind of table and a set of simple table lookup procedures will suffice to provide help and command completion information. 2m5

But since we do not want to set such severe stylistic syntactic restrictions on tool builders, and because we would like to have the Frontend be useful with tools having richer languages, this very simple approach won't work. 2m6

Even taking simple command languages where command words could appear at some point in the command like the middle or even the end such as the RAND editor would cause problems for such an approach. 2m7

For command languages with multiple command words and variations on the argument types and sequences possible, or having optional fields, or context dependent fields of other types, the above approach clearly will not work. 2m8

Steve may have had in mind a more field dependent approach, namely after each field is completed it is shipped off to the tool for parsing and back comes a table of what things could be legal next and other relevant status information. We would

agree that it would allow a simpler Frontend and provide a way to do command completions and help of various sorts, but we would argue it could be expensive of the communication medium, probably provide jerky slow response to the user, and only allow the simplest within-field form of command editing. 2m9

The only alternative that we can see to handle these more sophisticated languages and collect up the whole command in the Frontend would require some sort of structured representation of the command language being sent to the Frontend. 2m10

To interact with the user and keep track of where it is in this structure and be able to backup based on intracommand editing requires the ability on the part of the Frontend code to traverse such data structures. 2m11

Once this is done it is a small step indeed to drop out a completed parse at the end of the command specification. If you have the completed parse, you can save the generally higher cost, CPU cycles at the TBH, by passing a representation of the parsed command across rather than passing a collected character string that just has to be reparsed. 2m12

It is this line of reasoning that lead us to our approach of having a Command Language Interpreter (CLI) as the vehicle for meeting the type of requirements outlined above. The important point is that we have a CLI in the Frontend not because we require it per se, but because when examining the mechanisms available to meet the requirements above, we feel it meets them at low cost and provides important secondary benefits such as taking some load off the TBH and other benefits to tool builders and the user interface. Such benefits include a central statistics gathering point useful for researchers on the user interface, or for an adaptive tutorial tool, and data from the parse useful in determining an initial entry point to a help online documentation database. 2m13

We indicated earlier the cost in memory consumption for the PDP-11 code for our CLI approach. It is a tightly coded approach that should be easy to understand and maintain. We did not mention the per-user context required. Steve mentioned 1k as his guess at a reasonable number. We have been using 2k on the PDP-10 for a stack and have had no problems with overflow of status information. Perhaps this allocation can be reduced, and the slightly larger estimate (because of smaller word size) we made for the PDP-11 can also be reduced some. Other context information than just that for the CLI, possibly also needed by Steve's suggestion, is also needed, and we are also looking at how to minimize that. 2m14

reply to Steve Warshall's note on Frontend Requirements

Disk Requirements 2n

We have assumed all along that a disk would be required for the Frontend to support its maximum number of terminals. To the best of our knowledge we have been stating this verbally all along. We had hoped to support the initial six DSDC terminals stand alone in memory, based on our initial estimates. It is limitation here that was the only change to the best of our knowledge. 2n1

Estimation without a detailed model or design is a dangerous and tricky business. Once we get the PDP-10 system up and running we intend to go back over all the code and approaches to look for places to cut, save, reorganize, etc. We welcome Steve's and others participation in such optimization. 2n2

We are now ready to move onto the next section where we show that the FE costs we are talking about are relatively very small when looked at, as we believe they should be, in the context of the total NSW system. 2n3

Our Economic perceptions 2o

There are at least two ways to evaluate the cost of a component, one is to look at its stand-alone cost, and the other is to examine its cost relative to other parts of the total system or the total system cost. 2o1

One part of the relative cost approach is to determine which costs or benefits are most sensitive to the base assumptions. What we hope comes clear below is that the system is quite insensitive to various assumptions about FE costs. It is in the area of communication costs and savings of TBH costs that there seems to be considerable sensitivity. 2o2

It is this fact that leads us to feel that this whole discussion is not worth much investment if economics are the main bone of contention. If there are others, hopefully they will come forth, but my reading of the strawman document is that it is economics we are to concentrate on and are considered most troubling. 2o3

The various facets we are aware of on the economics of the FE and various approaches to its construction are now outlined below. The whole operational system includes such factors as the users time and costs (salary, overhead, etc), training and learning time, the cost of the FE, the cost of communications, the cost of time on the host running the tool (the TBH), the cost of running the Works Manager and its associated

communication costs. In order to make clear the bones of the argument we will not consider all possible cases, but look at two FE configurations and two classes of tools. The two FE configurations will be: 2o4

a) Steve's estimate taken essentially at face value for the minimum requirements (We have upped his memory requirement to 96k as more reasonable and detailed design might show it considerably higher).

b) The configuration on an 11/40 required to run the approach as previously estimated. 2o4a

The tool classes are a line-at-a-time half duplex tool, a simple full duplex character-at-a-time tool and a more sophisticated full duplex character-at-a-time tool. 2o5

We assume a tool bearing host (TBH) priced such as the BBN systems at $600,000/year. 2o6

For communication costs we make assumptions about ARPANET costs. These costs are lower than commercial network rates. 2o7

The assumption we make about communication costs is 40 cents per 1,000 packets. 2o8

We do not know how to estimate WM costs so assume them zero for this argument. 2o9

In summary the economic arguments for our estimated approach are: 2o10

1) It can reduce the load on the tool bearing host of the character by character handling and parsing required to provide a well human engineered set of user interface services. This cost saving can vary from small to substantial as shown below. 2o10a

2) It reduces the communication cost of handling the character by character interaction of full duplex character at-a-time tools and still allows for a full range of user interface services, command completion, prompts, ?, noise words, full intracommand editing facilities etc. 2o10b

3) It provides increased responsiveness, and thus hard to define but real cost to the user during command specifications or requests for help over what would be possible in an ARPANET environment if the user interface services were to be done in the TBH. 2o10c

4) It reduces the cost of new tool creation by removing from the tool builder the task of building user interface machinery. We have considerable experience to indicate that for sophisticated tools this capability yields significant cost savings, and even for simple tools is very valuable. The ability this facility provides for low cost experimentation and tuning of the user interface is also significant. 2o10d

5) It does provide an environment that would encourage more stylistic consistency which we have no desire to force on builders, but, if it should occur is of value to the many users who we see in the NSW are likely to use a wide variety of tools. 2o10e

6) Provides a central point for collecting statistics that may be valuable for system evaluation researchers, or could be used by experimental adaptive tools. 2o10f

7) Provide information that give a more accurate entry point to the help and tutorial facilities that we see will exist or evolve to aid the training and refresh of the user. 2o10g

The cost of the various PDP-11 configurations are computed from the latest DEC price list available to us, these calculations do not include discounts as we are not sure whether such are available to NSW users. 2o11

The argument below is staged as follows: 2o12

1) First we look at the relative cost of the FE to the total system for several cases. This stage demonstrates the relative insensitivity to cost savings in the total system from changes in FE configurations. 2o12a

2) We look at cost trends in the computer industry as a whole, which shows that it is such as to reinforce the above argument. 2o12b

3) We look at harder to define savings in system building costs. 2o12c

4) We look at savings to the user in having the more extensive user services available. 2o12d

Economic Analysis of Cost Savings and Dissavings from Steve's Frontend (SteveFE), as Compared to the SRI Frontend (SRIFE) 2o13

Steve claims that his FE will be considerably less expensive

than the proposed SRI FE. At first blush, this seems to be true. While we estimate that the proposed SRI FE will cost $127,670, we estimate that Steve's FE could be as cheap as $96,570 if one accepts Steve's assumptions. So, at least superficially, Steve's FE seems to give a cost savings of 24.4 percent. 2o13a

The purpose of this analysis is to examine Steve's FE savings more rigorously. The analysis shows that in terms of total system cost, that is the cost to users for typical tools (NLS and XED, in our case studies), Steve's FE gives a negligible one to three percent savings. Furthermore, this small best-case savings may be swamped by losses for many tools, because the SRI FE allows more efficient use of back ends (BE's) and produces communication savings that Steve's system may not meet. 2o13b

The First Cut Using NLS as an Example 2o14

As Table I indicates, Frontend cost, whether Steve's FE or the SRI FE is considered, is a small fraction of total tool usage costs for NLS; between five and six percent. As a result, the 24 percent savings in Frontend cost which Steve's FE claims to bring gives a small (1.5 percent) reduction for total costs. For a less Backend-intensive tool, XED, the savings are slightly more; 3.4 percent, but even this seems negligible. 2o14a

All costs are expressed in annual costs per user for convenience and comparibility. The user, after all, is interested in how much a given service is costing each year, and if savings are reported to the user, he or she will rightly interpret these savings in terms of their impact on total costs. 2o14b

Frontend costs in Tables I and II were calculated this way. Capital costs for Steve's FE ($96,570) and the SRI FE ($127,670) were divided by 20 to obtain capital cost per user. This assumes, of course, that both systems will support 20 users. Next, capital cost per user was divided by four to obtain annual cost per user; dividing by four assumes that obsolescence, physical wear and the cost of interest will combine to make the effective life of either FE four years. We neglected FE maintenance or operator costs, which if included, only strengthens our argument. 2o14c

To calculate backend (BE) costs per user per year, we assumed that a PDP-10 would be used, and that the rent would be $600,000 per year. This cost figure may be somewhat low

or high depending on algorithms used for equipment amortization, operation support, etc. We take the $600,000 figure as that is the current BBN rate. We know from our experience that a PDP-10 KA can support 25 NLS users, so that annual BE cost per NLS user will be $24,000 ($600,000/25). For XED, it seems that a PDP-10 might support 60 users, so that annual cost per XED user could be $10,000 ($600,000/60). 2o14d

The latter seems high as we have data that indicate Tenex would have a hard time supporting even 60 Telnet users, but we use 60 to give an extreme datapoint. 2o14e

reply to Steve Warshall's note on Frontend Requirements

Table I 2o15

First Cut* Cost Comparison Between Steve's FE** and the SRI FE*** for NLS 2o16

	Annual Cost Per User	Percent	2o17
SRI FE	$ 1,596	6.2%	
Backend (BE)	24,000	93.8	
Total	$25,596	100.0	2o18
Steve's FE	$ 1,207	4.8%	
Backend (BE)	24,000	95.2	
Total	$25,207	100.0	2o19
Steve's FE Savings/Year/User	$389		2o20
Steve's FE Percent Savings	1.5%		2o21

*Best case for Steve's FE
**Steve Warshall's Frontend
***SRI Frontend 2o22

reply to Steve Warshall's note on Frontend Requirements

Table II 2023

First Cut Cost Comparison Between Steve's FE and the SRI FE for XED 2024

	Annual Cost Per User	Percent	
			2025
SRI FE	$ 1,596	13.8%	
Backend (BE)	10,000	86.2	
Total	$11,596	100.0	2026
Steve's FE	$ 1,207	10.8%	
Backend (BE)	10,000	89.2	
Total	$11,207	100.0	2027
Steve's FE Savings/Year/User	$389		2028
Steve's FE Percent Savings	3.4%		2029

reply to Steve Warshall's note on Frontend Requirements

The Second Cut: Backend Efficiency 2o30

The Frontend can do more than communicate with users. If it fully handles the user interface and does parsing, it will take some load off the backend. For a service like NLS, between 10 and 20 percent of the BE load might be removed by an SRI FE. We will use 10 percent in our calculations, to be conservative. As a result, the BE will be 10 percent less expensive if the SRI FE is used. We do not see comparable cost savings coming from Steve's FE. 2o30a

Table III shows what will happen in the case of NLS if the SRI FE takes 10 percent of the load off the backend, while Steve's FE has no impact on the BE. A 10 percent cost savings would make BE cost per user $21,600 (see the preceding section for details of FE and BE cost calculations). The table shows that the small 1.5 percent savings for Steve's FE is swamped by adverse impacts on the BE, so that total cost to the user will actually rise 8.7 percent! 2o30b

Because the BE impact is so large, under our assumptions, Steve's FE could be a loss for a tool like NLS. For a system like XED, a very fine analysis would be necessary to determine whether Steve's FE produces a net gain or loss when the impact of the FE on BE performance is considered. The 3.4 percent savings calculated in the first cut analysis could easily be offset by even tiny changes in BE efficiency. 2o30c

RWW 15-AUG-75 18:20 26284

reply to Steve Warshall's note on Frontend Requirements

Table III 2o31

Second Cut Cost Comparison* Between Steve's FE and the SRI FE for NLS 2o32

	Annual Cost Per User	Percent	
			2o33
SRI FE	$ 1,596	6,9%	
Backend (BE)	21,600	93,1	
Total	$23,196	100,0	2o34
Steve's FE	$ 1,207	4,8%	
Backend (BE)	24,000	95,2	
Total	$25,207	100,0	2o35

Steve's FE (dis)savings -$2,011 2o36

Steve's FE Percentage (dis)savings 8,7% 2o37

*Includes BE Impacts 2o38

The Third Cut: Communication Costs 2o39

In the preceding section, we illustrated how small impacts of the FE design on BE efficiency could change total costs much more than Steve's FE savings. This section treats a variable that may be even more important: communication costs. 2o39a

As discussed below, communication costs for character-at-a-time full duplex tools like NLS or XED could be at least $6,000 per user each year if there is no frontend. The FE, by performing user interface functions and eliminating the need to echo typed-in-text can reduce communication cost from $6,000 to $120 per year. This fifty-fold reduction in communication costs, $5880, is three to four times the annual FE cost for the typical user. Unless an alternative FE can provide comparable cost savings by only interacting with the BE on a command at-a-time basis, claimed savings could turn into large losses. For example, as shown in Table IV, the use of a FE reduces total NLS cost per user (including communication costs) by 18.6 percent. For simpler tools like XED, (which like NLS has character echoing) the impact is even larger: 33.4 percent (see Table V). Compared to these figures, the two to three percent best case savings from Steve's FE seem insignificant. 2o39b

Note that such savings are only present when character echoing systems like NLS or XED are used. Clearly just knowing a field was for literal typein and collecting and echoing this locally would provide a significant savings even if a Backend did command completion etc.Half-duplex, line-at-a-time systems have little or no communication savings when the FE is attached. However, if an NSW client uses character-echoing systems even a portion of the time, savings from the SRI FE could be quite large. 2o39c

Communication costs were based on an NLS sample session, in which a two-page item was composed, edited and sent to another user. The session took about 20 minutes. Although the sample was not selected rigorously, it seems typical of NLS composition/editing loads. For a system like XED, communication loads would be about the same as in NLS, perhaps higher, since XED is less powerful and the user must specify more information to do a task. 2o39d

In the NLS sample session, over 5,000 packets were sent. Most of these packets arose because typed-in text characters were echoed through the net, which would not happen if a FE

were present. With a FE, only 100 packets would have been needed in the session. The sample session was twenty minutes, so hourly use would have been 15,000 packets without a FE, 300 with an SRI FE. At $0.40 per 1,000 packets (which is substantially below what commercial networks charge and what the ARPA net would probably have to charge to cover costs), a one hour session would cost $6.00 without a FE, $0.04 with a FE for communications. 2o39e

To put figures on a yearly basis, we assumed (to understate the importance of the communication savings of the FE) that NLS or XED would be used 20 hours each week, 50 weeks per year. So annual charges without a FE would be $6,000, or $120 with an FE, for each user. 2o39f

Again, we cannot tell whether or how much Steve's FE would reduce communication costs. We suspect there would be some reduction relative to a full grammar case, as in the current CLI. We do point out, however, that even small reductions of communication cost could wash out cost gains of FE's that are more expensive of communication. 2o39g

reply to Steve Warshall's note on Frontend Requirements

Table IV

Third Cut Cost Comparison* Between Steve's FE and the SRI FE NLS Case

With No FE Communication Interaction

	Annual Cost Per User	Percent
SRI FE	$ 1,596	5.0%
Backend (BE)	24,000	76.0
Communication	6,000	19.0
Total	$31,596	100.0%

With FE Communication Interaction

	Annual Cost Per User	Percent
SRI FE	$ 1,596	6.2%
Backend (BE)	24,000	93.3
Communication	120	0.5
Total	$25,716	100.0%
SRI FE Savings	$5,880	
Percentage Savings	18.6%	

*Includes FE Communication Interactions

reply to Steve Warshall's note on Frontend Requirements

Table V 2o51

Third Cut Cost Comparison* Between Steve's FE and the SRI FE XED Case 2o52

With No FE Communication Interaction 2o53

	Annual Cost Per User	
SRI FE	$ 1,596	
Backend (BE)	10,000	
Communication	6,000	
Total	$17,596	2o55

With FE Communication Interaction 2o56

	Annual Cost Per User	
		2o57
SRI FE	$ 1,596	
Backend (BE)	10,000	
Communication	120	
Total	$11,716	2o58
SRI FE Savings	$5,880	2o59
Percentage Savings	33.4%	2o60

*Includes FE Communication Interactions 2o61

Trends in Computer Cost 2p

Various numbers get thrown around in the computer industry about the rate at which the cost of computing is coming down. The number of a 20% decrease in cost per year is one frequently heard. For minicomputer systems such as the PDP-11 the decrease rate may be greater as there is much more vigorous competition in the mini computer area than in the large main frame domain. 2p1

For example, memory costs from DEC have gone from $17.6k for 16k four years ago to $6k today. Its even less for the 11/70. One can obtain 16k of PDP-11 memory from Cambridge Memories for $2.3k. 2p2

While this means TBH costs will be coming down also, the

previous tables show that the cost differences between Frontend approaches, small to start with, will become even more insignificant. 2p3

System Building Cost 2q

The FE approach we are taking allows a tool installer or builder to express the user interaction discipline for his tool, display or typewriter oriented, in a high order language we call Control Meta Language (CML). This statement then compiles into a data structure we call a grammar that drives the FE command language interpreter (CLI). To change an interface takes a few simple edits to the CML program and a recompile. 2q1

We do not have any controlled experiments to show the differences in productivity between creating a user interface using the CML approach and more conventional approaches, but the following anecdotes illustrate the types of savings possible. In general, one can do in hours what would take days or weeks in conventional approaches. We have non-professional programmers quickly creating sophisticated display based user interfaces for their small application programs using this approach. 2q2

An example of a very very simple user interface is the interface to the NSW PDP-10 Encapsulator that took a few minutes to write using CML and produced a 50 word grammar and probably would have taken an hour or two conventionally. 2q3

An example of a large sophisticated tool is the NLS Base subsystem which took about 4 hours of an experienced CML user to create the user interface for; it would have required, based on pre CML experience, about two-three weeks to have created with conventional approaches. 2q4

Another example shows that during the development of the powerful graphics line drawing capability within NLS we found that in the first design of the user interface that its command word terminology and syntax of many commands to be unsatisfactory. With about six hours work it was completely changed and working again. We expect the graphics user interface to continue to change as we get outside user feedback. Possibly more importantly, all the ? help facilities, and hardcopy command summary documentation were automatically up-to-date in the process. This rarely happens with conventional approaches as they generally require extra steps for such documentation to be produced. 2q5

It is improvements in productivity in all aspects of the software system building process like this that the NSW is all about. 2q6

Value to the User 2r

Value in terms of cost to the user of increased responsiveness and improved services during command specifications are clearly hard to define. 2r1

Clearly one can expect factors such as better system acceptance, less handholding and direct consulting, and some increase in productivity to have value. 2r2

Conclusion 2s

When Steve's proposed scheme is compared against the one we are providing in a whole picture context, ours appears to be quite economically attractive when all factors are considered. 2s1

Even if one just looks at the stage 1 argument and is not willing to give us any points for the later ones, in the cases where Steve's configuration yields less expense, the percent improvement offered is small and close to the noise level. 2s2

Further, it has to be conceded that many useful benefits over and above absolute minimal ones come from the approach we are taking. 2s3

However, to try and put everyones mind at ease we are looking at how our previous memory estimate can be reduced and what the tradeoffs are in memory required as service provided. We hope to have more to say in early September. 2s4

For example, had DPS continued to be used we could have folded its function directly into the CLI for very significant memory savings. 2s5

Dave Retz is also looking at ELF to see where it might be made smaller. I should point out we have no real intent in which operating system is used as long as it meets FE needs. ELF was chosen by the Steering Committee. 2s6

RWW 15-AUG-75 18:20 26284

reply to Steve Warshall's note on Frontend Requirements

(J26284) 15-AUG-75 18:20;;;; Title: Author(s): Richard W. Watson/RWW; Distribution: /MAW([INFO-ONLY]) DLS([INFO-ONLY]) WEC([INFO-ONLY]) LAC([INFO-ONLY]) SRI-ARC([INFO-ONLY]) ; Sub-Collections: SRI-ARC; Clerk: RWW; Origin: < WATSON, NEWREPLY.NLS;1, >, 15-AUG-75 18:09 RWW ;;;;####;

26284 Distribution

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,
Mike A. Wingfield, Duane L. Stone, William E. Carlson, Lawrence A. Crain, 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

JEW 15-AUG-75 18:22 26285

DPS-10 Notes

These notes pertain to DPS-10 as it existed when DPS was cut from NSW.

DPS-10 Notes

```
Loading RUNFIL                                                  1
   <arcsubsys>tenldr                                            1a
   /s                                                           1b
   <rel-nls>xl10data                                            1c
   dpsdta                                                       1d
   dps                                                          1e
   dpsosi                                                       1f
   dpstbl                                                       1g
   dpstrc                                                       1h
   <rel-nls>xl10runtime                                         1i
   <rel-nls>l10lrp                                              1j
   <nsw-sources>festgmgt                                        1k
   <subsys>stenex                                               1l
   symbol                                                       1m
   l10ddt                                                       1n
   debug                                                        1o
                                                                1p
   ddt                                                          1q
   1[400000                                                     1r
   2[1,,dps                                                     1s
   sevecx                                                       1t
   pushj s,stop<st:                                             1u
   pushj s,sback<sb:                                            1v
   pushj s,spar<sp:                                             1w
   pushj s,srec<sr:                                             1x
```

- Code inter-host IPC, including splice. 3c1

- Support unsolicited signout of root process. 3c2

- Change costs from milliseconds CPU to cents, with considerable help from Tenex. 3c3

- Truely connect inferior fork to USER INFO specified directory, rather than just verifying password. This task requires help from Tenex. 3c4

Miscellaneous 3d

- Merge RECPE and RTNPE messages, adding direction BOOLEAN. 3d1

- Code user-program-specific mini-subsystems which merge DPS into their address space and store the canned user program's filename in DPS' ACs. 3d2

DPS-10 Notes

(J26285) 15-AUG-75 18:22;;;; Title: Author(s): James E. (Jim) White/JEW; Distribution: /SRI-ARC([INFO-ONLY]) ; Sub-Collections: SRI-ARC; Clerk: JEW; Origin: < JWHITE, DPSMISC,NLS;2, >, 15-AUG-75 18:20 JEW ;;;;####;

26285 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

BEV KIRK 18-AUG-75 00:28 26286

More Airforce format, secretarial functions guide, future help needs, etc.

Documentation weekly report for week ending 8/15/75

BEV KIRK 18-AUG-75 00:28 26286

More Airforce format, secretarial functions guide, future help needs, etc.

Kirk 1

Done 1a

Re-did and un-did various AFMFormat program directives. 1a1

Updated the manual for formatting Airforce Manuals. 1a2

Wrote a table of contents generator and a volume formatter to augment the AFM chapter formatting command. 1a3

Updated <xhelp,helpd,> some. 1a4

Discussed future help needs with JAC3 and BEV questions about help code and descriptions. 1a5

Do 1b

Finish getting Help files in order for bringing up 8.5. 1b1

Write the step by step DPCS procedures for the Air Force Manual. 1b2

Debug the Volume formatter and add bells, whistles, and necessary changes. 1b3

Decide for sure whether or not to punt the Letter Program. 1b4

Work with Bev to review her help writing. 1b5

Continue transfer to Jan of Help and Class-I user programs. 1b6

Put the current list of development documentation on line. 1b7

Bev 2

This Week 2a

Began edits on Xhelp, Base. Have notes on changes that are needed in other Xhelp files as they show up by working on this file in Bev,hellnotes. 2a1

Made master copies for printing of the Sec. Func. Guide. Gave one set of masters to Jim B. Still need Format SS from Dirk, and Preface from Ann. 2a2

Went over future needs to fix up help with Kirk and Jan. 2a3

More Airforce format, secretarial functions guide, future help needs, etc.

Next Week 2b

Get Sec. Func. Guide printed, 2b1

Contin ue on Xhelp, Base. 2b2

Review ideas for needs in doc. for next proposal with Kirk and Dirk and Ann if she is available, 2b3

Get glossary printed, 2b4

BEV KIRK 18-AUG-75 00:28 26286

More Airforce format, secretarial functions guide, future help needs, etc.

(J26286) 18-AUG-75 00:28;;;; Title: Author(s): Beverly Boli, Kirk E. Kelley/BEV KIRK; Distribution: /DMB([ACTION] dirt) DIRT([INFO-ONLY]) ; Sub-Collections: SRI-ARC DIRT; Clerk: KIRK;

26286 Distribution

Delorse M. Brooks, 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, 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,

BEV KIRK 18-AUG-75 00:28 26287

More Airforce format, secretarial functions guide, future help needs, etc.

Documentation weekly report for week ending 8/15/75

BEV KIRK 18-AUG-75 00:28 26287

More Airforce format, secretarial functions guide, future help needs, etc.

Kirk 1

Done 1a

Re-did and un-did various AFMFormat program directives. 1a1

Updated the manual for formatting Airforce Manuals. 1a2

Wrote a table of contents generator and a volume formatter to augment the AFM chapter formatting command. 1a3

Updated <xhelp,helpd,> some. 1a4

Discussed future help needs with JAC3 and BEV questions about help code and descriptions. 1a5

Do 1b

Finish getting Help files in order for bringing up 8.5. 1b1

Write the step by step DPCS procedures for the Air Force Manual. 1b2

Debug the Volume formatter and add bells, whistles, and necessary changes. 1b3

Decide for sure whether or not to punt the Letter Program. 1b4

Work with Bev to review her help writing. 1b5

Continue transfer to Jan of Help and Class-I user programs. 1b6

Put the current list of development documentation on line. 1b7

Bev 2

This Week 2a

Began edits on Xhelp, Base. Have notes on changes that are needed in other Xhelp files as they show up by working on this file in Bev,hellnotes. 2a1

Made master copies for printing of the Sec. Func. Guide. Gave one set of masters to Jim B. Still need Format SS from Dirk, and Preface from Ann. 2a2

Went over future needs to fix up help with Kirk and Jan. 2a3

More Airforce format, secretarial functions guide, future help needs, etc.

Next Week 2b

Get Sec. Func. Guide printed, 2b1

Contin ue on Xhelp, Base, 2b2

Review ideas for needs in doc. for next proposal with Kirk and Dirk and Ann if she is available, 2b3

Get glossary printed, 2b4

BEV KIRK 18-AUG-75 00:28 26287

More Airforce format, secretarial functions guide, future help needs, etc.

(J26287) 18-AUG-75 00:28;;;; Title: Author(s): Beverly Boli, Kirk E. Kelley/BEV KIRK; Distribution: /DMB([ACTION] dirt) DIRT([INFO-ONLY]) ; Sub-Collections: SRI-ARC DIRT; Clerk: KIRK;

26287 Distribution

Delorse M. Brooks, 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, 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,

SGR 18-AUG-75 11:52 26288

New Home for User Services Trainers

The following people will begin using bbnb for their primary source of computer power: Susan Roetter, Jeanne Beck, Rita Hysmith, Priscilla Wold, and Pam Allen (directory is PALLEN). Please direct sndmsg's to us at bbnb. We will have directories at office-1 for a time to avoid missing mail but will be reading mail more frequently at bbnb. 1

SGR 18-AUG-75 11:52 26288

New Home for User Services Trainers

(J26288) 18-AUG-75 11:52;;;; Title: Author(s): Susan Gail Roetter/SGR; Distribution: /SRI-ARC([ACTION]) KWAC([ACTION]) ; Sub-Collections: SRI-ARC KWAC; Clerk: SGR;

26288 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, Elizabeth F. Finney, Lawrence A. Crain, E. S. VonGehren, Glenn A. Sherwood, Kathey L. Mabrey, Jeanne M. Beck, David A. Potter, Robert N. Lieberman, Terry H. Proch, Ronald P. Uhlig, Susan Gail Roetter, Michael A. Placko, Stanley M. (Stan) Taylor, Elizabeth J. Feinler, Rudy L. Ruggles, Frank G. Brignoli, Robert M. Sheppard, Richard W. Watson, Douglas C. Engelbart, James C. Norton, James H. Bair, Duane L. Stone, Inez M. Mattiuz, Connie K. McLindon
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

DVN 18-AUG-75 12:22 26289

Break in Record of Help Reads

Recording of the occasion on which the Help Data Based is opened for the first time by a user in a given session stoped May 30 when help was moved from <Documentation> to <Userguides>. I have asked Jeff Peters, who wrote the programs, to get it started again. We will journalize a report on use between March and May shortly. 1

DVN 18-AUG-75 12:22 26289

Break in Record of Help Reads

(J26289) 18-AUG-75 12:22;;;; Title: Author(s): Dirk H. Van Nouhuys/DVN; Distribution: /DMB([ACTION] dirt notebook please) DIRT([INFO-ONLY]) JCP([INFO-ONLY]) ; Sub-Collections: SRI-ARC DIRT; Clerk: DVN;

26289 Distribution

Delorse M. Brooks, 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, 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, Jeffrey C. Peters,

Messages from Watson

(work) temporary work space 1

13-AUG-75 12:20:24-EDT,484;000000000000
Date: 13 AUG 1975 1220-EDT
From: WATSON
Subject: Future NSW Proposal Areas
To: irby, michael, postel, jwhite, Boli
cc: watson 1a

I have started a file [bbnb] (watson,ideas,) that should expand into a shopping list for next NSW Proposal. It needs to be filled out and words added to motivate the tasks and show relevancY to NSW. . Would like to do some work on it at this afternoons NSW meeting at 3:00, Please take a look at it so you cAn contribute, Thanks Dick ------- 1b

14-AUG-75 16:53:28-EDT,1412;000000000000
Date: 14 AUG 1975 1653-EDT
From: WATSON
Subject: Getting Shopping List Together for Next NSW Proposal To: postel
cc: vannouhuys, irby, jwhite, boli, michael, belleville 1c

Jon, as you know next NSW Proposal is your ball. I will be helping but you need to get it together. The first step is to get shopping list together of things ordered in our estimate on NSW interest and distribute to RADC, Gunter Carlson Pentagon etc. I have an initial set of notes of ideas in [bbnb] <watson> ideas that can serve as start. What is needed is to take each idea and add a few motivating sentences or paragraphs of why NSW should be interested or why there is need. We need at least for our use some rough estimate of manpower. I would appreciate your pulling this together in fairly final form while I'm gone and after I have a chance to look at it when I get back we can send it out first week in Sept. My assumption is that we will make an extended tour East stopping at RADC Gunter etc before meeting Carlson to begin talks. Will probably go to MCA for FE talks also on tour. You should not feel you have to write it all, we need to get wider participation in the proposal thinkpiece generation business. Since there is heavy emphasis on documentation, Dirk should be involved also at least knowing what we are doing. Doug may also want to get involved at some level and point. See you Sept 2 Dick ------- 1d

15-AUG-75 18:31:44-EDT,244;000000000000
Date: 15 AUG 1975 1831-EDT
From: WATSON
Subject: nsw distribution lists
To: postel
cc: watson 1e

Messages from Watson

Jon, please up date your lists with hignett etc so we can keep using them without stuff going to lloyd all the time. Thanks Dick -------

1f

15-AUG-75 18:33:38-EDT,349;000000000000
Date: 15 AUG 1975 1833-EDT
From: WATSON
Subject: invite to join auto din 2 design
To: postel
cc: watson

1g

While I am away you are likely to get asked to spend a couple days in washington reviewing autodin 2 design on nsw funds, its ok with me if you want to do it as long as everything else like nsw shopping list gets done, Dick -------

1h

15-AUG-75 18:36:31-EDT,486;000000000000
Date: 15 AUG 1975 1836-EDT
From: WATSON
Subject: nsw things
To: postel
cc: watson

1i

Jon besides shopping list having high priority the other area is getting think piece on alternative FE designs costs and hoe to reduce elf costs, Charles dave and andy primed on these but you should peek in from time to time, Kremers should be getting into helping getting nls 9 up and larry should be jumping in there also, Hope to find the world still here when I get back, Dick -------

1j

15-AUG-75 19:29:30-EDT,879;000000000000
Date: 15 AUG 1975 1929-EDT
From: WATSON
Subject: conversation with carlson
To: engelbart at OFFICE-1, norton at OFFICE-1, postel
cc: watson

1k

Talked with Bill today and among subjects that came up were follow on funding, He hopes to have two pots one a combined pot for Nsw system building operation etc, and then side pots from services for applications, He thought total from both available to us would be under $500K, which is about 6-7 peoples worth, He said congress cut 1,5 out of 6 million fiscal 76 budget, He says our current level of funding way too high, He indivated that command and control program or vldb programs might be good other places for funds and that DCA may fund some protocol work, I really hope we can get the 8 peop;es wotrh of NSW funding I have been planning

Messages from Watson

on, but its clear we will have to hastle this fall for funding.
Dick ------- 1l

17-AUG-75 17:41:02-EDT,383;000000000000
Mail from BBN-TENEXB rcvd at 17-AUG-75 1741-EDT
Date: 17 AUG 1975 1737-EDT
Sender: WATSON at BBN-TENEXB
Subject: DCA review
From: WATSON at BBN-TENEXB
To: POSTEL
Cc: WATSON
Message-ID: <[BBN-TENEXB]17-AUG-75 17:37:22-EDT,WATSON> 1m

Jon, the DCA review could be important as start of funding source
to continue our protocol work. So hope you take part. thanks dick 1n

------- 1o

Messages from Watson

(J26290) 18-AUG-75 12:29;;;; Title: Author(s): Jonathan B. Postel/JBP; Distribution: /JBP([INFO-ONLY]) ; Sub-Collections: SRI-ARC; Clerk: JBP;

26290 Distribution
Jonathan B. Postel,