JMB 13-SEP-75 02:20 26465

Moving toward freedom of information

References: <33442,> <26448,> <33469,> <33470,> <26455,> <33476,>

JMB 13-SEP-75 02:20 26465

Moving toward freedom of information

I strongly feel that now is the right time to move in the direction of encouraging a more open kind of clientele than NSA, CIA, DIA, etc, The ethical issues have been at hand a long time; now we have more of an opportunity to choose the way our work is used than we have ever had before. It is in the interests of our long-term survival as well as personal work satisfaction to stop courting users (or funding sources) devoted to hiding information and to begin to take advantage of the increasing number of other interested groups having the potential of broadening the base of distributed information. 1

How can it support the philosophy behind a "public" journal and a "distributed" knowledge community to provide tools for the kind of "intelligence" that is information hoarded and hidden from others, including groups who want their own walled-off NLS or Journal or refuse to use the public journal. How can this serve to encourage the long-range goal of a network of distributed knowledge? There is an unmistakable trend in this society toward openness in the distribution of information; DVN describes this feeling well in <33442,>. 2

That is our long-term interest as i have interpreted it from our various position papers, and from the strong feelings of people in arc. It may also be dangerous to our goals in the long run to lose control of the development of our tools and community to groups having their own imperatives inimical to our goals and, as has been demonstrated in the case of cia, imperatives inimical to their own initially stated intentions. 3

I feel a sense of excitement in the opportunities mushrooming in Applications for a broader variety of interested users. It seems to me it would be a mistake not to move right now to develop this potential clientele with a spurt of imagination and skillful marketing effort. From this ambitious viewpoint, can we get more information about our various alternatives for funding Development work and Applications, and about our freedom to choose among various prospective clients (i.E, freedom to say "no" to some)? 4

Moving toward freedom of information

(J26465) 13-SEP-75 02:20;;;; Title: Author(s): Jeanne M. Beck/JMB; Distribution: /SRI-ARC([INFO-ONLY]) ; Sub-Collections: SRI-ARC; Clerk: JMB;

26465 Distribution

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, Susan K. Ocken, Raphael Rom, David C. Smith, 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, 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

PKA 13-SEP-75 17:01 26466

Weekly Report for Aug 4-8

USER SERVICES WEEKLY REPORT for Week of AUG 4 - 8: 1

from JMB 1a

Sat on pins and needles all week while AMC training was on-again, off-again. Now looks like maybe week of 18th. Spent most of the week getting materials and myself prepared for trip: 1a1

--trying to find out wich MSG document went with which version of MSG. It turns out we had an older verson of MSG on the system and a doc for a newer version. By the end of the week I convinced people to get the newest version of MSG up at O-1 and it works very much better than the old one. Consulted with PKA on cleaing up the MSG document; it's all ready. 1a1a

--thought about how to teach TENEX, reviewed the users guide with LJM (who's revising it), and later proofed it. Waiting to receive final copy from JHB. 1a1b

DEX Users Guide: made a few revisions & rewordings. HGL made some changes in DEX itself; so am in process of changing the document to reflect those. 1a2

Talked with JHB about revising the Lineprocessor Users Guide. Marking up a draft. Did a lot of research on new setups for the lineprocessors the clients will soon be getting: new lights & buttons and ELF startup procedures. We will be directing the new version of the users guide to ELF users as well as TIP users. 1a3

Tried out XED a little. It's difficult to figure out how to use it from the documentation that comes with it. 1a4

from SGR--on travel at Gunter; see trip report 1b

from RH 1c

I went over the Vela file with Clements earlier in the week. I showed him how to browse through it using different print commands, viewspecs, etc. I started giving Kathy Rigsbee the first course, which I will complete upon my return to Washington. I also spent additional time with her practicing what she had learned so far. She is very bright and shows a lot of promise. I also spent many hours this week working with Kathy Milks of MATS who is also very bright. I helped Bill Bangert fix up some of his old files, mainly reformatting them. I also gave him additional Output Processor training and DNLS training. Spoke with Betty Finney a couple of times helping them with minor problems. I also fixed up the structure problems with the MRAO's which Bill Carlson had requested. Got ready for my trip to California. Answered a few questions for Maria Scott of the SRI grup. 1c1

from PKA 1d

Experimented with MSG and read various versions of the documentation to see which document went with the version online. Finally decided that Office-1 did not have the most current version. Jeff fixed it. Made some edits to the MSG document by John Vittal. Essentially it is the same with only typo corrections. I also made some revisions to the DEX document for Jeanne. She will possibly need this for her upcoming trip. On Wednesday I worked with Sandy all day to learn Feedback. Read aout DNLS on Thurs and prepared to teach Dee. She has to do some tables for a document and it would be much easier for her to do it with DNLS. So spent a few hours on Fri. working with Dee. Organized my course notebook on Fri. afternoon. 1d1

from PAW2--on travel at Gunter; see trip report 1e

Weekly Report for Aug 4-8

(J26466) 13-SEP-75 17:01;;;; Title: Author(s): Pamela K. Allen/PKA; Distribution: /US([INFO-ONLY]) JCN([INFO-ONLY]) JHB([INFO-ONLY]) ; Sub-Collections: SRI-ARC US; Clerk: PKA; Origin: < ROETTER, AUGUST4-8.NLS;1, >, 12-SEP-75 19:10 SGR ;;;; ####;

26466 Distribution

Susan Gail Roetter, Priscilla A, Wold, Jeanne M. Beck, Pamela K. Allen, Rita Hysmith, Sandy L. Johnson, James C. Norton, James H. Bair,

Telephone Calls Charge Number

When making telephone calls on NSW project business please tell the SRI operator the charge number is 4417-900. Sub 900 is for telephone calls, the other subs are for labor only. (if you only give the operator 4417 it end up as an "invalid charge" that we have to fix up later.) --jon. 1

Telephone Calls Charge Number

(J26467) 14-SEP-75 22:26;;;; Title: Author(s): Jonathan B. Postel/JBP; Distribution: /ARC-DEV([INFO-ONLY]) ; Sub-Collections: SRI-ARC ARC-DEV; Clerk: JBP;

26467 Distribution

Susan K, Ocken, Raphael Rom, David C. Smith, Mary Ann Kellan, Andy Poggio, David L. Retz, Jan A. Cornish, Larry L. Garlick, Delorse M. Brooks, Beverly Boli, James E. (Jim) White, Ann Weinberg, Kenneth E. (Ken) Victor, Dirk H. Van Nouhuys, Jonathan B. Postel, Elizabeth K. Michael, David S. Maynard, Karolyn J. Martin, Harvey G. Lehtman, Kirk E. Kelley, Charles H. Irby, Joseph L. Ehardt, Robert Louis Belleville, Don I. Andrews, Richard W. Watson, Douglas C. Engelbart, JBP 14-SEP-75 23:18 26468

DPS Documents

Mil: Here are the citations to the latest documents on our
"distributed programming system" (nee Procedure Call Protocol).
--jon. 1

JEW 15-AUG-75 18:22 26285
DPS-10 Notes
Location: (MJOURNAL, 26285, 1:w) 2

JEW 13-AUG-75 18:31 26271
DPS-10 Version 2.5 Programmer's Guide
Location: (LJOURNAL, 26271, 1:w) 3

JEW 15-AUG-75 18:14 26282
DPS Version 2.5 Implementer's Guide
Location: (MJOURNAL, 26282, 1:w) 4

JEW 15-AUG-75 18:17 26283
DPS-10 Version 2.5 Procedure Directory [SYSGD]
Location: (MJOURNAL, 26283, 1:w) 5

JEW 13-AUG-75 14:08 26267
DPS-10 Version 2.5 Source Code
Location: (LJOURNAL, 26267, 1:w) 6

Think about this one: it is at least 200 pages ! 6a

(J26468) 14-SEP-75 23:18;;;; Title: Author(s): Jonathan B. Postel/JBP; Distribution: /MEJ([INFO-ONLY]) ; Sub-Collections: SRI-ARC; Clerk: JBP;

26468 Distribution
   Mil E, Jernigan,

Colonel Brunner Falls into a PIT

Demo for the Big Colonel 1

This morning I was scheduled to give a demonstation for Col Brunner who is the commanding officer of all of Gunter Air Force Base. It was scheduled at 9:00 and at 8:30, this place starting jumping..everyone cleaning up etc. I was sitting quietly at the terminal getting things ready., A few minutes before 9, the local imp died three times in five minutes. I began to think this Colonel had more power than I could imagine...but it turned out that someone was using the vacuum cleaner and had plugged it into the outlet next to where the imp was and the imp couldn't take all that cleanliness. 1a

With that solved, I was ready to begin...well Col Brunner didn't show up until 10:00 and by that time, even I was getting a little nervous. 1b

Well, he came and I started from scratch with a little talk. I explained alot about documentation especially how we were doing the rewrites of 66-1. 1c

I showed him some editing things and then explained how the formatting was done..he wanted to see it so I ran afmformat (after remembering to load the right ouputprocessor) and ran it to the printer..it went fine. 1c1

Then I talked about the journal and linking and linked to EKM for a quick demo of that.. 1c2

I mentioned some special programs that we use for editing such as as Modify and Publish and also the Format subsystem and format library. 1c3

I discussed Com, showed him so COM proofs and discreetly handed him my business card. (He wanted to know where the logo was). 1c4

Then it was on to graphics and he asked me to draw a flowchart right then and there.. With a little fumbling, it all went well. He felt that drawing things like that was pretty complicated and not as easy for people to learn as the text part was. He wanted to be able to draw right on the screen. 1c5

Then he began to talk about how he had used a computer once to play tic tac toe and how much he liked that.. Well, there was my cue, I mentioned we had that and we also had better games.. Of course he wanted to play and how could I resist but to get Wumpus going.. All of the other bigshots that were with him became very interested. (I think they liked the idea that it was sort of like a war game Ugg). Well, he played for about 10 minutes and then unfortunately...FELL INTO a PIT.. But...when he heard that I take a terminal home at night..he asked if I could bring one over to his office and show him how to use it and then he could take it home at night and show his family how to play. (I think he wants to play himself). Well, at least the wumpus is in the front door..will keep you informed as to how things go from here, but in general i think it went well. 1d

Colonel Brunner Falls into a PIT

(J26469) 15-SEP-75 10:20;;;; Title: Author(s): Ann Weinberg/POOH; Distribution: /SRI-ARC([INFO-ONLY]) ; Sub-Collections: SRI-ARC; Clerk: POOH;

26469 Distribution

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, Susan K. Ocken, Raphael Rom, David C. Smith, 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, 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

DVN 15-SEP-75 18:08 26470

Do you want more seminar stuff?

I had a set of note for what I said in the ARC Seminar. Something or other crashed as I was putting it online and I never finished. I could finish and supply it if you think it useful? 1

DVN 15-SEP-75 18:08 26470

Do you want more seminar stuff?

(J26470) 15-SEP-75 18:08;;;; Title: Author(s): Dirk H. Van Nouhuys/DVN; Distribution: /JHB([ACTION]) RLL([INFO-ONLY]) ; Sub-Collections: SRI-ARC; Clerk: DVN;

26470 Distribution
James H, Bair, Robert N, Lieberman,

Pit Proves Telepathy

keep up the good work

Pit Proves Telepathy

This morning I was working att office-1 and checking in RSEX once ina a while to see if BBNB was up yet. One time I blundered and by chance displayed the list of user on SRI-AI. I said to Jake, "there's SRI in Wumpus...It's probalby Anne entertaing some big shot at Gunter." (or words to that effect. So help me DEC I did. Ask Jake if I didn't. 1

Pit Proves Telepathy

(J26471) 15-SEP-75 18:17;;;; Title: Author(s): Dirk H. Van Nouhuys/DVN; Distribution: /POOH([INFO-ONLY]) JAKE([INFO-ONLY]) ; Sub-Collections: SRI-ARC; Clerk: DVN;

26471 Distribution
   Ann Weinberg, Elizabeth J. Feinler,

DVN 15-SEP-75 18:33 26472

Command Words for Readmail

Comment on 26464.

DVN 15-SEP-75 18:33 26472

Command Words for Readmail

In general I like the Readmail design. It would be nice to keep the verbal quality in initial NLS command words. Howabout "Scan" for "Brief" and "File" for "Catagory". 1

Command Words for Readmail

(J26472) 15-SEP-75 18:33;;;; Title: Author(s): Dirk H. Van Nouhuys/DVN; Distribution: /FEEDBACK([ACTION]) RA3Y([ACTION]) DIRT([INFO-ONLY]) ; Sub-Collections: SRI-ARC FEEDBACK DIRT; Clerk: DVN;

26472 Distribution

Special Jhb Feedback, Raymond R, Panko, 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, the HELP group: re your <26463,>

I'd like to be in the Help group; I have worked on Hep files in the past and expect to be in the future. Thanks.....jeanne 1 the HELP group: re your <26463,>

(J26473) 15-SEP-75 18:34;;;; Title: Author(s): Jeanne M. Beck/JMB; Distribution: /DAV([ACTION]) ; Sub-Collections: SRI-ARC; Clerk: JMB;

26473 Distribution
David C, Smith,

My feeling on the prevailing trend.

The prospect of ARC dealing at all with the CIA or NSA depresses me. Aside from the illegal acts the CIA has apparently committed against the American people, even their legal activites do not excite me. I would much rather do work with the EPA, Coastal and other planning commissions and so forth, all of whom stand equally to benefit from the sort of system Doug enthusiastically presented today, not to mention our current systems. 1

DeBono, in his book NEW THINK, which I picked off our shelf, has a statement which summarizes what I see as the crucial issue before us: "You don't dig new holes by digging the same holes deeper." 2

Perhaps we are ignoring one important resource in our search for both appealing and challenging applications. Other divisions at SRI, say Urban and Social Systems, must have many contacts in the EPA et al. where we might begin to dig our new holes. 3

My feeling on the prevailing trend,

(J26474) 15-SEP-75 18:35;;;; Title: Author(s): Jan A. Cornish/JAC3;
Distribution: /SRI-ARC([INFO-ONLY]) ; Sub-Collections: SRI-ARC;
Clerk: JAC3; Origin: < CORNISH, MYRESPONSE.NLS;2, >, 15-SEP-75
18:31 JAC3 ;;;;####;

26474 Distribution

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, Susan K. Ocken, Raphael Rom, David C. Smith, 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, 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

PAW2 15-SEP-75 19:11 26475

SRI Course Report

SRI (2 mandays) 1

1. Persons (users or not) contacted [use uppercase if they have a directory] 1a

Dan LYNCH (SRI-AI), Geoff GOODFELLOW (SRI-AI), and Lance Murphy (SRI-Operations Evaluation) using KRUZIC directory. 1a1

2. COURSE: Basic and Second Course material covered. 1b

Attending the Monday morning session were Dan, Geoff, and Lance. Dan and Geoff met again Tuesday morning, completing both courses. 1b1

In order to complete the Basic and Second Course I met with Lance individually for four different sessions. 1b1a

Monday morning the Basic course was covered with the exception of all TENEX material. Both Dan Lynch and Geoff Goodfellow knew TENEX material thouroughly. Met with Lance that afternoon to cover TENEX material as well as to review the morning session. 1b2

Dan and Geoff met again Tuesday morning covering the Second Course, again with the exception of TENEX material. 1b3

I met with Lance Friday morning for the Second Course, getting as far as the Sendmail section. We resumed our session a week later, reviewing and completing the Second Course. 1b4

4. APPLICATION (# slots) 1c

Dan and Geoff both from SRI-AI had played with NLS but were interested in getting a structured introductory course covering basic NLS commands and capabilities. They are both well entrenched in various TENEX subsystems and thus wanted further exposure to NLS. 1c1

Lance Murphy, in Operations Evaluation is working part-time for Pam Kruzic and will be using her directory for his work. He will be creating files and generally inserting and editing text. 1c2

5. ISSUES : 1d

Classes went well and all three members seemed pleased with NLS. All were interested in getting the sample sessions for independent practice when they become available. 1d1

(J26475) 15-SEP-75 19:11;;;; Title: Author(s): Priscilla A. Wold/PAW2; Distribution: /US([INFO-ONLY]) DCE([INFO-ONLY]) JCN([INFO-ONLY]) JHB([INFO-ONLY]) ; Sub-Collections: SRI-ARC US; Clerk: PAW2;

26475 Distribution

Susan Gail Roetter, Priscilla A, Wold, Jeanne M, Beck, Pamela K, Allen, Rita Hysmith, Sandy L. Johnson, Douglas C. Engelbart, James C, Norton, James H, Bair,

Big Numbers

He says there are some numbers that are so large or so small that they are never seen, because they refer to nothing. "You never see a number larger than ten to the hundred and twenty-fourth, for example."
"Why not?"
"Because there is nothing bigger than that. That is the volume of the known universe in cubic fermis. A fermi is the smallest dimension that makes any sense to talk about--ten to the minus thirteen centimetres. That's about the diameter of an electron. Nothing we know of is smaller than that." 1

-- John McPhee in "The Curve of Binding Energy" 1a

(J26476) 16-SEP-75 01:54;;;; Title: Author(s): Jonathan B. Postel/JBP; Distribution: /SRI-ARC([INFO-ONLY]) MAP([INFO-ONLY]) ; Sub-Collections: SRI-ARC; Clerk: JBP;

26476 Distribution

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, Michael A. Padlipsky, Susan K. Ocken, Raphael Rom, David C. Smith, 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, 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 so that was you snooping around and not the wumpus

we must think along the same wavelengths. well whatever it was, the demo went well and i can imagine it was pretty funny when you discovered thaat i was really doing what you thought i might be doing. anyway, things are going better in some ways but there are some new problems. i think they never end. the head colonel has made an appointment with me for wednesday to come teach him wumpus.. oh well.. see you soon..ann 1 so that was you snooping around and not the wumpus

(J26477)   16-SEP-75 05:19;;;;   Title: Author(s): Ann Weinberg/POOH; Distribution: /DVN([INFO-ONLY]) ; Sub-Collections: SRI-ARC; Clerk: POOH;

26477 Distribution
Dirk H. Van Nouhuys,

BEV KIRK 16-SEP-75 11:37 26478

Documentation Weekly Report

Software Technology transfer for Help, User programs completed; Sec, Func, Guide to press; more AFFormat and Xhelp, Base.

BEV KIRK 16-SEP-75 11:37 26478

Documentation Weekly Report

Week ending 9/12/75 1

Bev 1a

This Week 1a1

Continued working on Xhelp, Base. 1a1a

Journalized Secretarial Functions Guide. Found out from LAC how many copies he wants for Gunter. Sent Sec. Func. Guide to SRI press. 1a1b

Set documentation priorities with Kirk. 1a1c

Next Week 1a2

Complete first pass at Xhelp, Base. 1a2a

Check progress of Sec. Func. Guide through printing. 1a2b

Attend WP Conference 1a2c

Check progress of Glossary through last stages of printing. 1a2d

Kirk 1b

Done 1b1

Taught JAC3 and DAV how to use NDDT 1b1a

Introduced DAV to Help code, taught him how to bring up a new system. 1b1b

Told JAC3 all about user programs including AFMFormat. Passed the userprograms book of source code and documentation over to him. 1b1c

Helped Glenn Sherwood and Jan debug their SRI subsystem. 1b1d

Hassled with lost AFM Chapters, Volumes, and Programs. 1b1e

Do 1b2

Vacation for 3.5 weeks. Available via sndmsg. 1b2a

Documentation Weekly Report

(J26478) 16-SEP-75 11:37;;;; Title: Author(s): Beverly Boli, Kirk E, Kelley/BEV KIRK; Distribution: /SRI-ARC([INFO-ONLY]) DIRT([INFO-ONLY]) DMB([INFO-ONLY] for dirt notebook) ; Sub-Collections: SRI-ARC DIRT; Clerk: BEV;

26478 Distribution

James C. Norton, Delorse M. Brooks,
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, 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 Susan K. Ocken, Raphael Rom, David C. Smith, 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, 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

Termination Loop

Today I ran twice my commands branch (vannouhuys,dvn,readus). Each time I tried to stop it with <CTRL-O>. Each time it stopped and "user terminated process" began to flash in my TTY window in cycles of about ten seconds and continued to flash until I hit <CTRL-C>. 1

Termination Loop

(J26479) 16-SEP-75 11:49;;;; Title: Author(s): Dirk H. Van Nouhuys/DVN; Distribution: /FEEDBACK([ACTION]) ; Sub-Collections: SRI-ARC FEEDBACK; Clerk: DVN;

26479 Distribution
Special Jhb Feedback,

USER SERVICES REPORT: COURSE FOR MARK MICHAEL - Sept. 9-12

Independent (2 mandays) 1

1. Persons (users or not) contacted [use uppercase if they have a directory] 1a

Mark Michael 1a1

2. COURSE: 1b

Training was conducted Tuesday through Friday mornings. 1b1

Tuesday - DNLS training began with a discussion of how to read mail (Sendmail). This consisted of the Jump to BUG, Jump to Link, Jump to Successor, and Jump to File Return commands. We then discussed editing and used the Final Report Chapter (26386,) as practice. We started to try the Modify Substitute Spaces command but BBNB crashed. 1b2

Wednesday - We used the Modify command on the Final Report Chapter and then Jeanne Beck took over as I didn't feel well. 1b3

Jeanne covered: 1b3a

How to send a journal item, the capabilities of the journal, journal citations, journal catalogs, locator; 1b3b

The rest of the Modify commands, a little about Format subsystem, what Output Processor directives are, the Letter & Publish subsystems; 1b3c

In answer to his questions I went over plex, branch, group, end of branch, tail, head, etc.; 1b3d

Use of CTRL-E and Level-adjust to compose the outline of a document he began to write. 1b3e

Thursday - We reviewed how to send a Journal item and discussed sndmsg and linking in TENEX, how to use HELP, statement names, and links in NLS. He spent about an hour with Bob Belleville getting a demo of Graphics and another hour with Dirk talking about document production. 1b4

Friday - This morning Mark told me that he was given a TI735 by OSI and that he needed to know TNLS. After a crash-TNLS course covering most of the first three courses, including Output Processor and user programs, we also covered process branches. 1b5

3. APPLICATION 1c

Mark represents a potential client-community within the law profession, He is acting as consultant to a Hawaiian law firm that is interested in automating the paperwork that is necessary for them to do, He will be learning NLS as well as Super-Wilbur, For more information see Elizabeth Michael, 1c1

4, ISSUES 1d

In the description above about what was covered there was some repetition from one day to the next, I believe this was because we covered such a large amount of material that Mark was able to absorb capabilities and concepts but wasn't able to remember all the details, I feel confident that he will be able to pick up what he needs but I would not suggest a similar schedule for most DNLS learners, 1d1

USER SERVICES REPORT: COURSE FOR MARK MICHAEL - Sept, 9-12

(J26480) 16-SEP-75 12:37;;;; Title: Author(s): Susan Gail Roetter, Jeanne M, Beck/SGR JMB; Distribution: /SRI-ARC([INFO-ONLY]) ; Sub-Collections: SRI-ARC; Clerk: SGR;

26480 Distribution

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,

Susan K. Ocken, Raphael Rom, David C. Smith, 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, 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

Smalltalk, SIMULA and distributed processing

It strikes me that there is a strong correlation between the distributed processing notions of NSW/AKW and two programming languages with which I am very familiar: Smalltalk and SIMULA. SIMULA is widely known and there are many documents on it. Smalltalk is a new language developed by Alan Kay's Learning Research Group at Xerox PARC. It is based on the SIMULA notions of "class" and "instance", but it is very cleanly organized internally, giving the system a great deal more power.

Briefly, there is only one kind of entity in Smalltalk: an instance of a class. Classes are instances of class CLASS, which is an instance of itself. Instances can remember state, so they can simulate coroutines. Communication between instances is via "messages", which are streams of tokens. Classes need not know HOW another class will respond to a message; it need know only what forms the messages can have and what the response will be. For example, the class NUMBER responds to the message "+" by doing an addition, whereas STRING responds to the same message by concatenating. So the interpretation of the expression "X + Y" depends on the class of the instance bound to the variable "X". In this sense semantics are "intrinsic" to each class, rather than being "extrinsically" defined in symbols like "+". Note that the response of an instance of a class to a message may involve any kind of computing, perhaps even communication with a remote host. The internal workings necessary to generate the response can also be modified or even rewritten without affecting the overall program logic. For this reason Smalltalk is an excellent simulation language, permitting one to simulate other (including future) implementations using current ones.

I would be willing to give a seminar on Smalltalk and/or SIMULA, perhaps including a visit to PARC if we can arrange it, if there is sufficient interest. 1

Smalltalk, SIMULA and distributed processing

(J26482) 16-SEP-75 16:05;;;; Title: Author(s): David C. Smith/DAV; Distribution: /ARC-DEV([ACTION]) ; Sub-Collections: SRI-ARC ARC-DEV; Clerk: DAV;

26482 Distribution

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

when i tried to load message at isic, i got the error message "loading user programformat error, word 0 block type 0 can't load frontend for this subsys"
  %larry 1

message subsys problem

(J26483) 17-SEP-75 06:05;;;; Title: Author(s): Lawrence A. Crain/LAC; Distribution: /FEED([ACTION]) NDM([ACTION]) ; Sub-Collections: NIC; Clerk: LAC;

26483 Distribution
   Special Jhb Feedback, N, Dean Meyer,

LETTER: To Krasan of FNCB, Re: Literature request

This letter and enclosed documentation was sent on 12 Sept 75 to the First City National Bank. JCN will visit there soon.

LETTER: To Krasan of FNCB, Re: Literature request

Dr, Robert N, Lieberman
Stanford Research Institute
Augmentation Research Center
333 Ravenswood Avenue
Menlo Park, California 94025
415-326-2600 x4119

17 SEP 75

Mr, Victor Krasan
First National City Bank
Group Management Office
19th Floor
111 Wall Street
New York, NY 10015

Dear Mr, Krasan:

Jim Norton has requested me to send the enclosed documents, If you have any questions or in need of additional materials please contact me or Jim Norton (who will be on vacation most this month), 1

I might also help if there are any questions on possible communications to our computer service, Thank you, 2

Sincerely,

Robert N, Lieberman

ENCLOSURE:

"The Augmented Knowledge Workshop," Douglas C, Engelbart, Richard W, Watson, and James C. Norton, 1-MAR-73, (ijournal,14724,)
"Coordinated Information Services for a Discipline- or Mission-Oriented Community," Douglas Engelbart, 12-DEC-72, (mjournal,12445,)
"Seminar on the AUGMENTED KNOWLDEDGE WORKSHOP," SRI-ARC, 12-Aug-75, <journal,33193,>

"The SRI-ARC Workshop Utility Service: What and Why," James Norton, 10-SEP-75, (jjournal,26368,)

LETTER: To Krasan of FNCB, Re: Literature request

(J26484) 17-SEP-75 14:32;;;; Title: Author(s): Robert N. Lieberman/RLL; Distribution: /DCE([INFO-ONLY]) JCN([INFO-ONLY]) ARC-LOG([INFO-ONLY]) RA3Y([INFO-ONLY]) NDM([INFO-ONLY]) ; Sub-Collections: SRI-ARC ARC-LOG; Clerk: RLL; Origin: < LIEBERMAN, KRASAN,NLS;4, >, 11-SEP-75 18:20 RLL ;;;; ####;

26484 Distribution

Douglas C, Engelbart, James C, Norton, James C. Norton, Log Augmentation, Raymond R, Panko, N, Dean Meyer, RLL 17-SEP-75 15:07 26485

LETTER: To Charles Engle of FAA, Sent 16 Sept 75, Re: document request

This letter and enclosures were sent to Federal Aviation Agency per Mel Draper's request, Mel (of AFAA) met Engle.

LETTER: To Charles Engle of FAA, Sent 16 Sept 75, Re: document request

Dr. Robert N. Lieberman
Stanford Research Institute
Augmentation Research Center
333 Ravenswood Avenue
Menlo Park, California 94025
415-326-6200 x4119

17 SEP 75

Mr. Charles Engle
ARD 141, Building 7
DOT/FAA
NAFEC
Atlantic City, New Jersey 80024

Dear Mr. Engle:

Mel Draper of AFAA called me about your interest in our developments and service. Enclosed you will find several documents that I hope will be useful. 1

As I am sure Mel mentioned, the documentation does not give a good picture of our capabilities. Thus, we would be pleased to show the system and talk with you at your convenience. 2

I have also included a brochure on a seminar we give. It is an intensive week of system assessment for those who wish to learn more about our developments. The cost is expected to be $750 including computer time. I have included the announcement for the last one and will send you the next one as soon as it is available. 3

Please feel free to call me if you have any questions. 4

Sincerely,

Robert N. Lieberman

ENCLOSURES:

"The Augmented Knowledge Workshop," Douglas C. Engelbart, Richard W. Watson, and James C. Norton, 1-MAR-73, (ijournal,14724,)

"Coordinated Information Services for a Discipline- or Mission-Oriented Community," Douglas Engelbart, 12-DEC-72, (mjournal,12445,)

"Workstation Equipment Reference Manual," SRI-ARC, 10-DEC-74, <journal,23809,>

"The SRI-ARC Workshop Utility Service: What and Why," James Norton, 10-SEP-75, (jjournal,26368,)

"Seminar on the AUGMENTED KNOWLDEDGE WORKSHOP," SRI-ARC, 12-Aug-75, <journal,33193,>

"Announcement of AKW Workshop for August 1975," SRI-ARC, 15-AUG-75, <journal,26274,>

LETTER: To Charles Engle of FAA, Sent 16 Sept 75, Re: document request

(J26485) 17-SEP-75 15:07;;;; Title: Author(s): Robert N. Lieberman/RLL; Distribution: /DCE([INFO-ONLY]) JCN([INFO-ONLY]) ARC-LOG([INFO-ONLY]) ; Sub-Collections: SRI-ARC ARC-LOG; Clerk: RLL; Origin: < LIEBERMAN, DRAPER.NLS;4, >, 11-SEP-75 17:36 RLL ;;;; ####;

26485 Distribution
Douglas C, Engelbart, James C, Norton, James C, Norton, Log Augmentation,

1st rough draft of the Programmer's Guide to the Debugger Dispatcher

comments, etc. invited and desired; now that the 3 main programmers' guides exist in rough form, i intend to combine them into one coherent programmers' guide to the debugger

1st rough draft of the Programmer's Guide to the Debugger Dispatcher

INTRODUCTION 1

The debugger dispatcher (DD) is that module of the debuger that is responsible for communication with the debugger frontend and for dispatching user requests (made via the debugger frontend) to the appropriate routines in language and/or operating system modules (LMs and OSMs). 1a

When a user requests a specifc debugging function (though her interaction with the CLI (Command Language Interpretter) and the debugger grammar), the CLI translates the user's request to a request on the debugger backend. The debugger dispatcher is that module in the debugger backend that will receive this request. The DD will then call the appropriate routine(s), in the proper LM, OSM, and/or DD, to perform the specific action and possibly to obtain results that will then be returned to the CLI to be presented to the user. 1b

(The debugger dispatcher is coded to run on a TENEX system, but its design is not dependent on TENEX. It is written in a high level language, and those portions of the DD that depend on the DD running under TENEX are well isolated. Therefore, it is theoretically possible to run the DD on systems other than TENEX by only recoding those portions that are TENEX dependent and then recompiling the DD.) 1c

(This assumes, of course, that these other systems support the high level language used by the DD.) 1c1

GROSS STRUCTURE AND OPERATION OF THE DEBUGGER DISPATCHER 2

The debugger dispatcher consists basically of a dispatch table, routines that will be called by the CLI, routines and data structures that will be called and referenced by other modules of the debugger (hereafter refered to as external routines and data structures), and any routines and data structures (hereafter refered to as support routines and data structures) needed for the support of the above routines and data structures. 2a

(The DD also contains the code for the support of the communication protocols for frontend-backend communication, the runtime environment for the highlevel language used by the debugger itself, and certain global data declarations that are always available to all modules of the debugger.) 2a1

User requests fall roughly into the following catagories: 2b

1) requests for actions that are both language and operating system independent that cause a state change in the debugger,

e.g. I had been debugging using language X, and hence the X LM had been loaded, and I now wish to use language Y, and thus the Y LM must be loaded (after giving the X LM a chance to clean up); 2b1

2) requests that are both language and operating system independent, e.g. what is my current debugging state; and 2b2

3) requests that are language and/or operating system dependent, e.g. display the contents of cell N of the current target process in the current high level language. 2b3

Each of these types of requests are handled by the DD in a slightly different manner. 2c

(There is no formalization within the debugger with respect to request types. This discussion of request types is merely a mechanism for documentation communication purposes to give a feeling for how the debugger works.) 2c1

(In fact what happens is the following: a user makes a request via the CLI, and the CLI then calls the relavent procedure(s) in the debugger backend. The backend procedures then dispatch the request in one of the following manners. Various user requests may call the same backend procedure, or a single user request may call several backend procedures.) 2c1a

When the DD receives a type 2 request from the CLI, it performs whatever action is necessary to satisfy the request and then returns to the CLI, potentially returning strings to be displayed to the user. By the very nature of type 2 requests, the DD is able to satisfy these requests without calling any routines in a LM or OSM. After the DD has returned to the CLI, the type 2 request can be considered to have been completely satisfied. 2c2

When the DD receives a type 3 request from the CLI, the following typical sequence of events will occur: 2c3

the DD will look in the current appropriate (LM or OSM) dispatch table to determine if this type of request is supported by the current LM or OSM. 2c3a

If this request is not supported, an appropriate error message string will be generated and returned to the CLI to be presented to the user, and this request can be considered over. 2c3a1 the DD will perform some syntacical and semantical checks on the arguments for this request and will convert them to internal debugger format. 2c3b

If the arguments are invalid or illegal, then the DD will either generate an appropriate error message string and return to the CLI (and this request can be considered over), or the DD will interact with the user (via the CLI) to get valid arguments (and the request will procede normally). 2c3b1

the DD will then invoke the appropriate LM or OSM routine, whose address was obtained from the appropriate dispatch table, to satisfy the request. (see the discussion elsewhere for invoking sequences.) 2c3c

the invoked routine will perform its function and then return to the DD, potentially returning strings of information (which may be error messages) to be passed on to the user. 2c3d

the DD will then return to the CLI, passing along any strings generated by the invoked routine, and this request can be considered over. 2c3e

(In fact, if the invoked routine is a coroutine, the above 2 steps may be repeated a number of times before the request is finished.) 2c3e1

When the DD receives a type 1 request from the CLI, a combination of the above is likely to happen. A typical type 1 request might be that the user wishes to change the high level language that is being used as the current implementation language. This request would cause the current LM to be unloaded, and a new LM to be loaded. But before a LM can be unloaded, its termination routine (if one exists) must be called; and as soon as a new LM is loaded, its initialization (if one exists) routine must be called. However, the actual loading and unloading of the LMs are functions handled entirely by the DD. 2c4

An important part of the debugger dispatcher is its dispatch table. The dispatch table contains: 2d

addresses of external routines, and 2d1

addresses of external data structures, and 2d2

in some instances, a dispatch table entry is itself an external data structure. (A dispatch table entry that is itself a data structure will be called a simple data structure.) 2d3

To perform its function, a LM or OSM routine may find it necessary to call routines provided by the DD, or to refernce data structures in the DD. To do so, the LM or OSM routine will use the DD dispatch table and can thus call or reference routines and/or data structures that it does not provide itself. 2e

GENERATION OF THE DEBUGGER DISPATCHER 3

The following discussion is specific for generating the debugger dispatchere module designed to run under TENEX. However, the principles involved are the same regardless of what operating system the DD will be run under. 3a

A DD designed to run under TENEX lives in the address space of the debugger in pages 0(octal) - 237(octal). 3b

A DD is a TENEX SSAVE file that is started like any other normal tool in the NSW. 3c

To generate a DD, the debugger loader must be used. The debugger loader contains: 3d

debugger-wide definitions, 3d1

the L10 runtime environment (for the debugger dispatcher and any other modules written in L10), and 3d2

the debugger frontend to backend communication package. 3d3

The following are the current TENEX and debugger loader commands to generate a DD module (comments are bracketed by percent signs; atsign is the TENEX prompt character indicating willingness to accept a TENEX command; asterick is the debugger loader prompt character indicating willingness to accept a debugger loader command; upper case refers to primitives that are discussed below): 3e

```
@get <nsw-debugger>beldr.sav      % get the debugger loader %   3e1

@reenter                          % start it properly %         3e2

*FILE1                                                          3e3

...                                                             3e4

*FILEn                                                          3e5
```

```
        *<ALTMODE>          % done loading command to the loader %          3e6

        @OPTIONAL                                                           3e7

        @SSAVE (pages from) 0 (to) 237 (on) <nsw-debugger>FILENAME.sav      3e8
```

FILE1 ... FILEn are the rel files that comprise the DD. 3f

OPTIONAL is an oppurtunity for the DD to perform some pre-initialization. 3g

FILENAME conforms to NSW tool naming conventions.(????) 3h

THE DISPATCH TABLE 4

Many of the entries in the DD's dispatch table are copies of entries in the OSM's dispatch table, or pointers into the OSM's dispatch table. These entries are copied from the OSM's dispatch table after the OSM is initialized. This is done so that LM routines do not have to know about the OSM dispatch table and thus only have to deal with one dispatch table, i.e., the DD's dispatch table. 4a

The symbolic offset names for the entries in the DD dispatch table are contained in the file <nsw-debugger>ddtdsp.nls. Also, the debugger loader contains these definitions. (Note that an offset of 0 refers to the first entry in the dispatch table.) 4b

decimal offset	symbolic offset name	meaning	4c
0	ddiosi	address of OSM initialization procedure	4c1
1	ddsymp	symbol table pointer for this module	4c2
2	ddbpte	address OSM procedure to call when a breakpoint is hit	4c3
3	ddbptl	address OSM procedure to call prior to resuming from a breakpoint	4c4
4	ddmsta	address OSM routine for "memstat" function	4c5
5	ddrd1w	address OSM routine to read 1 word from target process' address space	4c6

6	----	RESERVED FOR FUTURE USE	4c7
7	ddrdnw	address OSM routine to read 1 or more words from target process' address space	4c8
8	----	RESERVED FOR FUTURE USE	4c9
9	ddwr1w	address OSM routine to write 1 word in the target process' address space	4c10
10	----	RESERVED FOR FUTURE USE	4c11
11	ddwrnw	address OSM routine to write 1 or more words in the target process' address space	4c12
12	----	RESERVED FOR FUTURE USE	4c13
13	ddsrcm	address OSM routine for searching memory	4c14
14	----	RESERVED FOR FUTURE USE	4c15
15	ddgpfs	address OSM routine to get target process' state vector	4c16
16	ddspfs	address OSM routine to set target process' state vector	4c17
17-26	----	RESERVED FOR FUTURE USE	4c18
27	ddcfrk	address of internal currrent target process handle	4c19
28	ddsvec	address of current target process' state vector	4c20
29	ddland	address of the current LM's dispatch table	4c21
30	ddadro	maximum offset value for symbolic address displays	4c22
31	ddlste	value of the last evaluated address range element	4c23
32	dddcda	address of DD's decode address range routine	4c24
33	ddchrt	address of GFS data structure	4c25

34	ddchrs	address DD routine for changing GFS data structure	4c26
35	ddarng	address DD routine to manage LSTADIS data structure	4c27
36	ddldval	last displayed value	4c28
37	dddfmk	default search mask	4c29
38-50	----	RESERVED FOR FUTURE USE	4c30

DETAILED DISCUSSION OF EACH ENTRY IN THE DEBUGGER DISPATCHER'S DISPATCH TABLE 5

This section will discuss in detail each entry in the DD's dispatch table. Each entry will be discussed under its symbolic offset name. 5a

ddiosi 5b

entry type - OSM procedure address 5b1

procedure function (brief) - 5b2

perform operating system module initialization 5b2a

discussion - 5b3

this entry is a copy of entry osini in the OSM's dispatch table. (for a detailled discussion see -- xxx.) 5b3a

ddsymp 5c

entry type - symbol table pointer 5c1

discussion - 5c2

This entry is a symbol table pointer for the symbol table for the DD. (For most languages running on a TENEX this consists of the lefthalf of the word being a negative count of the number of words in the symbol table and the righthalf of the word being the address of the first word of the symbol table.) This entry is not used by the debugger, but is merely a convience to aid in the debugging of the DD itself. 5c2a

(Note that the DD symbol table must reside in the same part of the debugger address space allocated to the DD.) 5c2a1 ddbpte 5d

entry type - OSM procedure address 5d1

procedure function (brief) - 5d2

Perform any operating system and/or OSM specific action required at breakpoint hit time. 5d2a

discussion - 5d3

this entry is a copy of entry osbpte in the OSM's dispatch table. (for a detailled discussion see -- xxx,) 5d3a

ddbptl 5e

entry type - OSM procedure address 5e1

procedure function (brief) - 5e2

address of OSM procedure to call prior to resuming from a breakpoint or tracepoint 5e2a

discussion - 5e3

this entry is a copy of entry osbptl in the OSM's dispatch table. (for a detailled discussion see -- xxx,) 5e3a

ddmsta 5f

entry type - OSM coroutine address 5f1

coroutine function (brief) - 5f2

The function of this OSM coroutine is to build strings (using the current output mode) for the display of the address space of the target process. (This coroutine implements the TENEX "MEMSTAT" command.) 5f2a

discussion - 5f3

this entry is a copy of entry osmsta in the OSM's dispatch table. (for a detailled discussion see -- xxx,) 5f3a

ddrdiw 5g

entry type - OSM procedure address 5g1

procedure function (brief) - 5g2

This OSM procedure is used to read one word in the address space of the target process. 5g2a

discussion - 5g3

this entry is a copy of entry osrdlw in the OSM's dispatch table. (for a detailled discussion see -- xxx.) 5g3a

ddrdnw 5h

entry type - OSM procedure address 5h1

procedure function (brief) - 5h2

This OSM procedure is used to read one or more words in the address space of the target process. 5h2a

discussion - 5h3

this entry is a copy of entry osrdnw in the OSM's dispatch table. (for a detailled discussion see -- xxx.) 5h3a

ddwr1w 5i

entry type - OSM procedure address 5i1

procedure function (brief) - 5i2

This OSM procedure is used to write one word in the address space of the target process. 5i2a

discussion - 5i3

this entry is a copy of entry oswr1w in the OSM's dispatch table. (for a detailled discussion see -- xxx.) 5i3a

ddwrnw 5j

entry type - OSM procedure address 5j1

procedure function (brief) - 5j2

This OSM procedure is used to write one or more words in the address space of the target process. 5j2a

discussion - 5j3

this entry is a copy of entry oswrnw in the OSM's dispatch table. (for a detailled discussion see -- xxx.) 5j3a ddsrcm 5k

entry type - OSM procedure address 5k1

procedure function (brief) - 5k2

This OSM procedure will search the target process' address space between 2 passed addresses (inclusively) for cells that contain the passed value (after both have been masked appropriately). 5k2a

discussion - 5k3

this entry is a copy of entry ossrcm in the OSM's dispatch table. (for a detailled discussion see -- xxx.) 5k3a

ddgpfs 5l

entry type - OSM procedure address 5l1

procedure function (brief) - 5l2

This OSM procedure is used to obtain the state vector of the target process and to write the obtained state vector in the cells allocated for the state vector in the OSM's dispatch table. 5l2a

discussion - 5l3

this entry is a copy of entry osgpfs in the OSM's dispatch table. (for a detailled discussion see -- xxx.) 5l3a

ddspfs 5m

entry type - OSM procedure address 5m1

procedure function (brief) - 5m2

This OSM procedure is used by LM or DD routines to modify the state vector of a target process. 5m2a

discussion - 5m3

this entry is a copy of entry osspfs in the OSM's dispatch table. (for a detailled discussion see -- xxx.) 5m3a

ddcfrk 5n

entry type - address of an OSM data structure 5n1 data structure name - CURFORK 5n2

data structure meaning - 5n3

this is a debugger internal name for the target process that is currently being debugged. 5n3a

data structure type - 5n4

this data structure is a single word 5n4a

discussion - 5n5

this entry is the address of entry oscfrk in the OSM's dispatch table. (for a detailled discussion see -- xxx.) 5n5a

ddsvec 5o

entry type - address of an OSM data structure 5o1

data structure name - PFSTATE 5o2

data structure meaning - 5o3

this OSM data structure contains the OS state of the current target process. 5o3a

data structure type - 5o4

this data structure is composed of 50 words (under TENEX, the first 16 of these words represent the registers of the target process; the meaning of the rest of the words will be specified later.) 5o4a

discussion - 5o5

this entry is the address of entry ossvec in the OSM's dispatch table. (for a detailled discussion see -- xxx.) 5o5a

ddland 5p

entry type - address of a data structure 5p1

data structure name - LANDSP 5p2

data structure meaning - 5p3

this is the address of the language module's dispatch table 5p3a data structure type - 5p4

this data structure is composed of 50 words (for a detailed discussion se -- xxx,) 5p4a

ddadro 5q

entry type - simple data structure 5q1

data structure name - MAXOFFSET 5q2

data structure meaning - 5q3

if addresses are being displayed as a symbol plus an offset, then if the offset is greater than the value of this cell, the address should be displayed numerically. 5q3a

data structure type - 5q4

this data structure consists of the single word in the DD dispatch table 5q4a

ddlste 5r

entry type - simple data structure 5r1

data structure name - LSTEADR 5r2

data structure meaning - 5r3

the value of this data structure is the value of the last completely evaluated address range element. 5r3a

data structure type - 5r4

this data structure consists of the single word in the DD dispatch table 5r4a

discussion - 5r5

This data structure should be updated by the LM every time it evaluates an address range element for which it is meaningfull to update this cell (e.g. it is not meaningful to update this cell after the evaluation of an ARE that corresponds to the target process' signal status). 5r5a

dddcda 5s

entry type - procedure address 5s1

procedure function (brief) - 5s2

This procedure is used to determine the gross type of AREs. 5s2a

when called - 5s3

This procedure will be called by any DD, LM, or OSM routine that evaluates AREs to determine the gross type of the ARE before it is evaluated. 5s3a

arguments - 5s4

1st argument: the address of the first ARE string 5s4a

2nd argument: the address of the corresponding second ARE string 5s4b

results - 5s5

1st result: 5s5a

a value indicating the gross type of the address range 5s5a1

error conditions - 5s6

this procedure will return an illegal gross type on any errors that it detects 5s6a

discussion - 5s7

Before any DD, LM, or OSM routine completely evaluates an ARE, this routine must be called to determine the gross type of the ARE. This procedure thus provides for uniform interpretation of AREs. If an ARE is illegal or invalid, then this procedure will return a gross type indicating this. 5s7a

ddchrt 5t

entry type - address of a data structure 5t1

data structure name - GFS 5t2

data structure meaning - 5t3

this is the DD Generic Function String 5t3a

data structure type - 5t4

this is a 128 character L10 string 5t4a

discussion - 5t5

(for a detailed discussion see -- xxx,) 5t5a

ddchrs 5u

entry type - procedure address 5u1

procedure function (brief) - 5u2

This procedure is used to modify the GFS. 5u2a

when called - 5u3

If a LM wishes to change which character will be used for which generic function (e.g. at initialization time), the LM must use this procedure and NOT modify the GFS directly. 5u3a

arguments - 5u4

1st argument: the address of a string containing as its first character the ascii character that is to perform a generic function 5u4a

2nd argument: the generic function the character is to serve 5u4b

3rd argument: zero or the address of a result list which will be filled in withthe first element of the list being set to the GFS 5u4c

results - 5u5

if a third argument is specified, then on success the first element of the result list (whose address is passed as the non-zero third argument) will get a copy of the GFS; if the third argument is zero, then nothing will be returned on success. if a third argument is passed, and this routine detects bad first or second arguments, then this routine will generate a L10 HELP signal in an attemp to get correct arguments; if no help is avaiable, or if no third argument is passed, then a L10 ABORT signal will be generated. 5u5a

error conditions - 5u6

this procedure will generate an L10 ABORT signal if it receives bad input. (details to be specified later.) 5u6a discussion - 5u7

This procedure is used to modify the GFS data structure. 5u7a

ddarng 5v

entry type - procedure address 5v1

procedure function (brief) - 5v2

This procedure is used for reading or writing the DD data structure LSTADIS (which contains the address of the last n displayed cells). 5v2a

when called - 5v3

This procedure will be called whenever any DD, LM, or OSM routine wishes to read or write the LSTADIS data structure. 5v3a

arguments - 5v4

1st argument: 5v4a

FALSE to indicate read an entry from the LSTADIS data structure; TRUE to make a new entry in LSTADIS. 5v4a1

2nd argument: 5v4b

if this is a read operation, then this argument is the index of the last displayed address desired, e.g. the most recently displayed address has an index of 0, the address displayed before that has an index of 1, etc.; if this is a write operation, then this is the new address to add to LSTADIS 5v4b1

results - 5v5

for write operations - 5v5a

NONE 5v5a1

for read operations - 5v5b

the index-th (mod n, where n is the number of entries maintained, and is currently set to 4) last displayed address 5v5b1

error conditions - 5v6

NONE 5v6a

discussion - 5v7

this procedure is used to maintain the LSTADIS data structure. 5v7a

ddldval 5w

entry type - simple data structure 5w1

data structure name - LSTVDIS 5w2

data structure meaning - 5w3

this is the value of the last displayed cell 5w3a

data structure type - 5w4

this data structure consists of the single word in the DD dispatch table 5w4a

discussion - 5w5

this data structure should be maintained by LM routines whenever they display cells to the user. 5w5a

dddfmk 5x

entry type - simple data structure 5x1

data structure name - DEFMASK 5x2

data structure meaning - 5x3

this is the debugger default mask for content searches and memory setting commands 5x3a

data structure type - 5x4

this data structure consists of the single word in the DD dispatch table 5x4a

discussion - 5x5

The LM lnmass routine is responsible for setting this cell; and the LM routines lnfmem and lnmems may use this cell 5x5a

GENERAL DISCUSSION OF ROUTINES CALLABLE FROM THE CLI 6

The CLI will in resspnose to user interactions, call procedures in the debugger backend. This document will not go into any detail about the communication protocol used for this purpose. (see --xxx.) Suffice it to say, that the communication part of the DD is well isolated and easily changable. Calls from the CLI get translated into debugger formatted calls, with debugger formatted arguments, by this communication code within the DD. Calls from the debugger backend to the CLI, and results from calls on the debugger backend have a similar inverse translation applied to them. 6a

The following section will discuss the routines in the DD that are callable from the CLI in terms of the debugger call/return mechanisms. 6b

(Note that all routines that are called by the CLI accept as their last argument the address of a result list. It is this list that gets the results discussed below; i.e. the first result is actually returned as the first element of the passed result list, the second result is actually returned as the second element of the passed result list, etc.) 6b1

DETAILED DISCUSSION OF ROUTINES CALLABLE FROM THE CLI 7

procedure name - iniddt 7a

procedure function (brief) - 7a1

the function of this routine is to initialize the debugger 7a1a

arguments - 7a2

1st argument: 7a2a

the value of this argument indicates which operating system module should be loaded 7a2a1

2nd argument: 7a2b

the value of this argument indicates which language module should be loaded 7a2b1

3rd argument: 7a2c

the address of the result list 7a2c1

results - 7a3

1st result: 7a3a

the GFS is returned to the CLI 7a3a1

error conditions - 7a4

If this procedure cannot perform its functions it will generate an ABORT to the CLI. 7a4a

discussion - 7a5

This procedure performs all first time initialization required to run the debugger. This includes the initialization of the input and output mode records, the initialization of the GFS, the loading and initialization of the initial LM and OSM, and the establishment of a communication path from the debugger backend to the CLI. 7a5a

Upon successful completion of its tasks, it will return a copy of the GFS to the CLI. 7a5b

procedure name - xdbgp 7b

procedure function (brief) - 7b1

the function of this procedure is to establish the state necessary for the debugging of a process, i.e, to make the passed process the target process. 7b1a

arguments - 7b2

1st argument: 7b2a

process name (more detail on this later) 7b2a1

2nd argument: 7b2b

the address of the result list 7b2b1

results - 7b3

NONE 7b3a

error conditions - 7b4

This procedure will generate an ABORT (with an appropriate message) to the CLI if it is not possible to establish the debugging state necessary to debug the passed process name. 7b4a

discussion - 7b5

this procedure will first convert the passed process name into the internal debugger handle for the process. 7b5a

Next it will call the terminate debug process routines in the OSM and then in the LM. 7b5b

And finally, it will call the breakpoint enter routines in the OSM and then the LM, indicating that a new process has been specified. 7b5c

The called routines are expected to perform whatever cleanup, and/or initialization is required to perform debugging. 7b5d

procedure name - xdchrd 7c

procedure function (brief) - 7c1

arguments - 7c2

1st argument: 7c2a

the address of the result list 7c2a1

results - 7c3

NONE 7c3a

error conditions - 7c4

NONE 7c4a

discussion - 7c5

this procedure will interpret the GFS and generate a number of lines of information. Each line consists of user readable information pertinent to successive characters in the GFS. After each line has been formatted a call on the utility routine pntstr will be made to present the line to the user. A normal return will be given to the CLI when all such lines have been formatted and presented to the user. 7c5a

procedure name - xdchrs 7d

procedure function (brief) - 7d1

to modify the GFS 7d1a

arguments - 7d2

see entry ddchrs in the detailed discussion of the DD's dispatch table 7d2a

results - 7d3

see entry ddchrs in the detailed discussion of the DD's dispatch table 7d3a

error conditions - 7d4

see entry ddchrs in the detailed discussion of the DD's dispatch table 7d4a

discussion - 7d5

This is the procedure whose address is contained in entry ddchrs of the DD's dispatch table. This procedure can be called by the CLI in response to user request to modify the GFS, as well as being able to be called by LM initialization routines. 7d5a

procedure name - xdtypd 7e

procedure function (brief) - 7e1

to interpret and display to the user the permanent output mode record 7e1a

arguments - 7e2

1st argument: 7e2a

the address of the result list 7e2a1

results - 7e3

NONE 7e3a

error conditions - 7e4

NONE 7e4a

discussion - 7e5

this procedure will interpret the permanent output record and generate lines of information (reflecting this interpretation), and make use of the utility routine pntstr to present these lines to the user. 7e5a

procedure name - xdtyps 7f

procedure function (brief) - 7f1

to modify the permanent output mode record 7f1a

arguments - 7f2

1st argument: 7f2a

a value representing the main output mode keyword specified by the user 7f2a1

2nd argument: 7f2b

a value representing the secondary output mode keyword specified by the user if the main output mode has a secondary mode; FALSE otherwise 7f2b1

3rd argument: 7f2c

the address of the result list 7f2c1

results - 7f3

NONE 7f3a

error conditions - 7f4

if this routine is passed illegal or invalid arguments it will do nothing 7f4a

discussion - 7f5

this procedure is called by the CLI in response to a user's request to change the permanent output mode 7f5a

procedure name - xdinpd 7g

procedure function (brief) - 7g1

to interpret and display to the user the permanent input mode record 7g1a

arguments - 7g2

1st argument: 7g2a

the address of the result list 7g2a1

results - 7g3

NONE 7g3a

error conditions - 7g4

NONE 7g4a

discussion - 7g5

this procedure will interpret the permanent input mode record and generate lines of information (reflecting this interpretation), and make use of the utility routine pntstr to present these lines to the user. 7g5a

procedure name - xdinps 7h

procedure function (brief) - 7h1

to modify the permanent input mode record 7h1a

arguments - 7h2

1st argument: 7h2a

a value representing the main input mode keyword specified by the user 7h2a1

2nd argument: 7h2b

a value representing the secondary input mode keyword specified by the user if the main input mode has a secondary mode; FALSE otherwise 7h2b1

3rd argument: 7h2c

the address of the result list 7h2c1

results - 7h3

NONE 7h3a

error conditions - 7h4

if this routine is passed illegal or invalid arguments it will do nothing 7h4a

discussion - 7h5

this procedure is called by the CLI in response to a user's request to change the permanent input mode 7h5a

procedure name - xdouts 7i

procedure function (brief) - 7i1

to allow displayed output to be directed to the user (via his terminal) and/or to a file. 7i1a

arguments - 7i2

the first 3 arguments comprise a state table as follows (see discussion below for the meanings of each of the states): 7i2a

arg 1	arg 2	arg 3	state	7i2a1
FALSE	FALSE	--	1	7i2a2
FALSE	TRUE	FALSE	2	7i2a3
FALSE	TRUE	"FILE" (*)	3	7i2a4
FALSE	TRUE	"TERMINAL" (**)	4	7i2a5
file (***)	FALSE	--	5	7i2a6
file (***)	TRUE	--	6	7i2a7

footnotes: 7i2a8

(*) - this argument is the value for the keyword "FILE" 7i2a8a

(**) - this argument is the value for the keyword "TERMINAL" 7i2a8b

(***) - this argument is the address of a string containing the name of the file the user wishes to have his output printed on 7i2a8c

4th argument: 7i2b

the address of the result list 7i2b1

results - 7i3

1st result: 7i3a

for requested states 5 or 6, the full file name will be returned to the CLI; for requested states 1-4, there are no results 7i3a1

error conditions - 7i4

if states 5 or 6 are requested and the first argument is invalid or illegal, a HELP will be generated to the CLI to attempt to make it all ok; if no help is available an ABORT will be generated. 7i4a

this routine assumes that it will be called by the CLI and that only valid combinations of the arguments will be passed. 7i4b

discussion - 7i5

this routine provides the user with control over where his output will be presented. Output can be displayed at her terminal and/or written in a file. (Note that if output is only being written in a file, then certain commands are no longer available to the user.) 7i5a

The first 3 arguments specify which state the user wishes to be in as follows: 7i5b

state meaning 7i5b1

1 get back to the default state, i.e. display
output
on the user's terminal and close any open
output files 7i5b2

2 an output file has been previously specified,
and
the user wishes output to be both displayed
at his
terminal and written in the output file 7i5b3

3 an output file has been previously specified,
and
the user wishes output to be written in the
output
file only 7i5b4

4 an output file has been previously specified, and the user wishes output to be displayed only on his terminal 7i5b5

5 the first argument specifies the name of an output file that the user wishes to have his output written on in addition to being displayed on her terminal; if the named file already exists, then the new output will be written on the end of the file 7i5b6

6 the first argument specifies the name of a new output file that the user wishes to have his output written on in addition to being displayed on her terminal 7i5b7

procedure name - xdmasd 7j

procedure function (brief) - 7j1

to display the default debugger mask to the user in the permanent output mode radix 7j1a

arguments - 7j2

1st argument: 7j2a

the address of the result string 7j2a1

results - 7j3

NONE 7j3a

error conditions - 7j4

NONE 7j4a

discussion - 7j5

this routine will use the utility routine pntstr to display the default mask (DEFMASK) to the user as a number in the permanent output mode radix 7j5a procedure name - xdmass 7k

procedure function (brief) - 7k1

to set up the default mask (DEFMASK) 7k1a

arguments - 7k2

1st argument: 7k2a

FALSE meaning to use the permanent input mode record to interpret the third argument; or the value of the keyword the user specified for the main input mode to be used to interpret the third argument 7k2a1

2nd argument: 7k2b

only has meaning if the first argument is not FALSE; in this case it is the value of the secondary input mode to be used for the interpretation of the third argument (or it can be FALSE if the main input mode does not require a secondary mode) 7k2b1

3rd argument: 7k2c

the address of a string to be evaulated, according to the current input mode, to become the new default debugger mask (DEFMASK) 7k2c1

4th argument: 7k2d

the address of the result list 7k2d1

results - 7k3

NONE 7k3a

error conditions - 7k4

This routine will generate an ABORT (with an appropriate message) to the CLI if the current LM does not provide the lnmass routine. 7k4a

Other error conditions will be displayed to the user as strings via the pntstr utility routine. 7k4b

discussion - 7k5

This routine makes use of the LM lnmass routine to evaluate the third argument and to set up DEFMASK. If lnmass returns a non-null string it will be presented to the user via the utility routine pntstr. 7k5a

procedure name - sadrl 7l

procedure function (brief) - 7l1

to display, and optionally to assign to, address lists 7l1a

arguments - 7l2

1st argument: 7l2a

the address of a string containing the address list to be displayed and optionally assigned to 7l2a1

2nd argument: 7l2b

FALSE; or value of CML keyword that user used to terminate the specified address list; or the value of the CML keyword to be used as the main output mode 7l2b1

3rd argument: 7l2c

FALSE; or the value of the secondary output mode keyword 7l2c1

4th argument: 7l2d

FALSE; or TRUE indicating that this should be an assignment as well as a display operation 7l2d1

(this will be true if the user uses the 2nd and 3rd arguments for specifying output mode, and he wants to do an assignment; otherwise this will most likely be false and assignment will be indicated, if desired, by the address list terminating character.) 7l2d1a

5th argument: 7l2e

the address of the result list 7l2e1

results - 7l3

NONE 7l3a

error conditions - 7l4

any error conditions detected by this routine will be handled internally and an appropriate error message will be displayed to the user via the utility routine pntstr 714a

discussion - 715

this is the main routine used for displaying and assigning to address lists. it breaks down the passed address list into address ranges and then uses the LM lnsadr routine to obtain strings to be presented to the user; it makes use of the utility routine pntstr to display strings to the user for the display only case, and uses the utility routine pasnstr to display strings to the user and get new values for the display and assignment case. 715a

procedure name - xdtadrl 7m

procedure function (brief) - 7m1

to display an address list only on the user's terminal (and optionally to assign to the address list) independent of whether or not the user had specified that his output should go to a file and/or his terminal 7m1a

arguments - 7m2

see sadrl procedure 7m2a

results - 7m3

see sadrl procedure 7m3a

error conditions - 7m4

see sadrl procedure 7m4a

discussion - 7m5

This procedure uses sadrl to do most of its work. What it does is temporarily (for the course of displaying and assigning to the address list) modify the output state of the debugger so that output goes only to the users terminal. 7m5a

procedure name - xdpadrl 7n

procedure function (brief) - 7n1

to print an address list in the user's output file 7n1a

arguments - 7n2

1st argument: 7n2a

see sadrl procedure 7n2a1

2nd argument: 7n2b

see sadrl procedure 7n2b1

3rd argument: 7n2c

see sadrl procedure 7n2c1

4th argument: 7n2d

the address of a string containing the name of an output file; or FALSE meaning to use the existing output file 7n2d1

5th argument: 7n2e

see sadrl procedure 7n2e1

results - 7n3

see sadrl procedure 7n3a

error conditions - 7n4

see sadrl procedure 7n4a

discussion - 7n5

This procedure uses sadrl to do most of its work. What it does is temporarily (for the course of displaying to the address list) modify the output state of the debugger so that output is written only on the file specified as the 4th argument, 7n5a

procedure name - xdtab 7o

procedure function (brief) - 7o1

to use the value of the data structure LSTVDIS (the last value displayed to the user) as an address list to be displayed, using the same output mode used in the previous command 7o1a

arguments - 7o2

1st argument: 7o2a

the address of the result list 7o2a1

results - 7o3

see sadrl procedure 7o3a

error conditions - 7o4

see sadrl procedure 7o4a

discussion - 7o5

This procedure uses sadrl to do most of its work. What it does is generate an address list string from the value in the LSTVDIS data structure and then call sadrl to do its work. It is the respnsibility of the LM to maintain LSTVDIS. 7o5a

procedure name - xdpound 7p

procedure function (brief) - 7p1

to use the value of the first element of the LSTADIS data structure (the last N addresses displayed to the user) as an address list to be displayed, using the same output mode used in the previous command 7p1a

arguments - 7p2

1st argument: 7p2a

the address of the result list 7p2a1

results - 7p3

see sadrl procedure 7p3a

error conditions - 7p4

see sadrl procedure 7p4a

discussion - 7p5

This procedure uses sadrl to do most of its work. What it does is generate an address list string from the value of the first element of the LSTADIS data structure and then call sadrl to do its work. It is the respnsibility of the LM to maintain LSTADIS. 7p5a

(It gets the value of the first element on the LSTADIS data structure by using the external routine ddarng.) 7p5a1

Basically, this routine provides an inverse for the xdtab routine; e,g, if the user had displayed n, and the contents of cell n were m, and then the user gave the command that caused the xdtab routine to be called, then cell m would be displayed; if the user then gave the command to cause the xdpound routine to be called, then cell n would be displayed again, 7p5b

procedure name - xdlnfd 7q

procedure function (brief) - 7q1

to display the next cell in the memory of the target process, using the same output mode used in the previous command 7q1a

arguments - 7q2

1st argument: 7q2a

the address of the result list 7q2a1

results - 7q3

see sadrl procedure 7q3a

error conditions - 7q4

see sadrl procedure 7q4a

discussion - 7q5

This procedure uses sadrl to do most of its work, What it does is call the LM xxx routine to generate an address list and then this address list is passed to sadrl, 7q5a

procedure name - xdupar 7r

procedure function (brief) - 7r1

to display the previous cell in the memory of the target process, using the same output mode used in the previous command 7r1a

arguments - 7r2

1st argument: 7r2a

the address of the result list 7r2a1

results - 7r3

see sadrl procedure 7r3a

error conditions - 7r4

see sadrl procedure 7r4a

discussion - 7r5

This procedure uses sadrl to do most of its work. What it does is call the LM xxx routine to generate an address list and then this address list is passed to sadrl. 7r5a

procedure name - xdasgn 7s

procedure function (brief) - 7s1

to assign to the address list used in the previous command 7s1a

arguments - 7s2

1st argument: 7s2a

the address of the result list 7s2a1

results - 7s3

see sadrl procedure 7s3a

error conditions - 7s4

see sadrl procedure 7s4a

discussion - 7s5

This procedure uses sadrl to do most of its work. What it does is call sadrl, passing the address list used in the previous command and simulating the user terminating the address list with the LARROWCHAR. 7s5a

procedure name - xdeql 7t

procedure function (brief) - 7t1

to obtain the value of the address list used in the previous command 7t1a

arguments - 7t2

1st argument: 7t2a

the address of the result list 7t2a1

results - 7t3

see sadrl procedure 7t3a

error conditions - 7t4

see sadrl procedure 7t4a

discussion - 7t5

This procedure uses sadrl to do most of its work. What it does is call sadrl, passing the address list used in the previous command and simulating the user terminating the address list with the EQUALCHAR. 7t5a

procedure name - xdexcm 7u

procedure function (brief) - 7u1

to display the address list used in the previous command in ascii output mode 7u1a

arguments - 7u2

1st argument: 7u2a

the address of the result list 7u2a1

results - 7u3

see sadrl procedure 7u3a

error conditions - 7u4

see sadrl procedure 7u4a

discussion - 7u5

This procedure uses sadrl to do most of its work. What it does is call sadrl, passing the address list used in the previous command and simulating the user terminating the address list with the EXCMARKCHAR. 7u5a

procedure name - xdslsh 7v

procedure function (brief) - 7v1

to display the address list used in the previous command in symbolic output mode 7v1a

arguments - 7v2

1st argument: 7v2a

the address of the result list 7v2a1

results - 7v3

see sadrl procedure 7v3a

error conditions - 7v4

see sadrl procedure 7v4a

discussion - 7v5

This procedure uses sadrl to do most of its work. What it does is call sadrl, passing the address list used in the previous command and simulating the user terminating the address list with the SLASHCHAR. 7v5a

procedure name - xdbslsh 7w

procedure function (brief) - 7w1

to display the address list used in the previous command in string output mode 7w1a

arguments - 7w2

1st argument: 7w2a

the address of the result list 7w2a1

results - 7w3

see sadrl procedure 7w3a

error conditions - 7w4

see sadrl procedure 7w4a

discussion - 7w5

This procedure uses sadrl to do most of its work. What it does is call sadrl, passing the address list used in the previous command and simulating the user terminating the address list with the BSLASHCHAR. 7w5a

procedure name - xdlsq 7x

procedure function (brief) - 7x1

to display the address list used in the previous command in numeric output mode 7x1a

arguments - 7x2

1st argument: 7x2a

the address of the result list 7x2a1

results - 7x3

see sadrl procedure 7x3a

error conditions - 7x4

see sadrl procedure 7x4a

discussion - 7x5

This procedure uses sadrl to do most of its work. What it does is call sadrl, passing the address list used in the previous command and simulating the user terminating the address list with the LSQUARECHAR. 7x5a

procedure name - xdrsq 7y

procedure function (brief) - 7y1

to display the address list used in the previous command in record output mode 7y1a

arguments - 7y2

1st argument: 7y2a

the address of the result list 7y2a1

results - 7y3

see sadrl procedure 7y3a

error conditions - 7y4

see sadrl procedure 7y4a

discussion - 7y5

This procedure uses sadrl to do most of its work. What it does is call sadrl, passing the address list used in the previous command and simulating the user terminating the address list with the RSQUARECHAR. 7y5a

procedure name - xdqmark 7z

procedure function (brief) - 7z1

to display the blocknames for the symbols used in the address list used in the previous command in string output mode; i.e. this routine provides a mechanism to determine in which blocks symbols are defined 7z1a

arguments - 7z2

1st argument: 7z2a

the address of the result list 7z2a1

results - 7z3

see sadrl procedure 7z3a

error conditions - 7z4

see sadrl procedure 7z4a

discussion - 7z5

This procedure uses sadrl to do most of its work. What it does is call sadrl, passing the address list used in the previous command and simulating the user terminating the address list with the QMARKCHAR. 7z5a

procedure name - xdfind 7a@

procedure function (brief) - 7a@1

to display, and optionally assign to, those cells within the bounds of an address list that meet specfic requirements 7a@1a

arguments - 7a@2

1st argument: 7a@2a

the value of the CML keyword indicating whether this is to be a search for references to the passed 2nd argument, or a search for cells whose value is specified by the 2nd argument, or a search for cells whose value is not equal to the 2nd argument 7a@2a1

2nd argument: 7a@2b

the address of a string to be evaluated, according to the current input mode, to serve as the search argument 7a@2b1

3rd argument: 7a@2c

FALSE or a value representing the main current input mode keyword specified by the user 7a@2c1

4th argument: 7a@2d

FALSE or a value representing the secondary current input mode keyword specified by the user if the main input mode has a secondary mode 7a@2d1

5th argument: 7a@2e

mask specification as follows: if FALSE then use default mask (DEFMASK); otherwise the address of a string to be evaluated, according to the current input mode, to be used as the mask 7a@2e1

6th argument: 7a@2f

the address of a string containing the address list to be searched, displayed, and optionally assigned to 7a@2f1

7th argument: 7a@2g

FALSE; or value of CML keyword that user used to terminate the specified address list; or the value of the CML keyword to be used as the main output mode 7a@2g1

8th argument: 7a@2h

FALSE; or the value of the secondary output mode keyword 7a@2h1

9th argument: 7a@2i

FALSE; or TRUE indicating that this should be an assignment as well as a display operation 7a@2i1

10th argument: 7a@2j

the address of the result list 7a@2j1

results - 7a@3

NONE 7a@3a

error conditions - 7a@4

any error conditons detected by this routine will be handled internally and an appropriate error message will be displayed to the user via the utility routine pntstr 7a@4a

discussion - 7a@5

This routine breaks down the passed address list into address ranges and then uses the LM lnfmem routine to obtain strings to be presented to the user; it makes use of the utility routine pntstr to display strings to the user for the display only case, and uses the utility routine pasnstr to display strings to the user and get new values for the display and assignment case, 7a@5a

procedure name - xdmems 7aa

procedure function (brief) - 7aa1

to set all cells in a specified address list to a specfic value 7aa1a

arguments - 7aa2

1st argument: 7aa2a

the address of a string to be evaluated, according to the current input mode, to serve as the value to set memory to 7aa2a1

2nd argument: 7aa2b

FALSE or a value representing the main current input mode keyword specified by the user 7aa2b1

3rd argument: 7aa2c

FALSE or a value representing the secondary current input mode keyword specified by the user if the main input mode has a secondary mode 7aa2c1

4th argument: 7aa2d

FALSE to indicate no mask should be used when modify memory; TRUE means get mask specification from 5th argument 7aa2d1

5th argument: 7aa2e

mask specification as follows: if FALSE then use default mask (DEFMASK); otherwise the address of a string to be evaluated, according to the current input mode, to be used as the mask 7aa2e1

6th argument: 7aa2f

the address of a string containing the address list whose memory is to be modified 7aa2f1

7th argument: 7aa2g

the address of the result list 7aa2g1

results - 7aa3

NONE 7aa3a

error conditions - 7aa4

any error conditons detected by this routine will be handled internally and an appropriate error message will be displayed to the user via the utility routine pntstr 7aa4a

discussion - 7aa5

This routine breaks down the passed address list into address ranges and then uses the LM lnmems routine to set the specified memory to the specified value. 7aa5a

procedure name - xdhigs 7ab

procedure function (brief) - 7ab1

to change the high level language (and hence the LM) being used to debug the target process 7ab1a

arguments - 7ab2

to be specified later 7ab2a

nth argument: 7ab2b

the address of the result list 7ab2b1

results - 7ab3

to be specified later 7ab3a

error conditions - 7ab4

to be specified later 7ab4a

discussion - 7ab5

to be specified later 7ab5a

procedure name - xdstad 7ac

procedure function (brief) - 7ac1

to display the current state of the debugger to the user 7ac1a

arguments - 7ac2

1st argument: 7ac2a

the address of the result list 7ac2a1

results - 7ac3

to be specified later 7ac3a

error conditions - 7ac4

to be specified later 7ac4a

discussion - 7ac5

to be specified later 7ac5a

procedure name - xdspds 7ad

procedure function (brief) - 7ad1

to modify the execution speed of the target process 7ad1a

arguments - 7ad2

to be specified later 7ad2a

nth argument: 7ad2b

the address of the result list 7ad2b1

results - 7ad3

to be specified later 7ad3a

error conditions - 7ad4

to be specified later 7ad4a

discussion - 7ad5

to be specified later 7ad5a

procedure name - xdbrkc 7ae

procedure function (brief) - 7ae1

to remove a breakpoint 7ae1a

arguments - 7ae2

to be specified later 7ae2a

nth argument: 7ae2b

the address of the result list 7ae2b1

results - 7ae3

to be specified later 7ae3a

error conditions - 7ae4

to be specified later 7ae4a

discussion - 7ae5

to be specified later 7ae5a

procedure name - xdbrkd 7af

procedure function (brief) - 7af1

to display the status of a breakpoint 7af1a

arguments - 7af2

to be specified later 7af2a

nth argument: 7af2b

the address of the result list 7af2b1

results - 7af3

to be specified later 7af3a

error conditions - 7af4

to be specified later 7af4a

discussion - 7af5

to be specified later 7af5a

procedure name - xdbrks 7ag

procedure function (brief) - 7ag1

to set a breakpoint 7ag1a

arguments - 7ag2

to be specified later 7ag2a

nth argument: 7ag2b

the address of the result list 7ag2b1

results - 7ag3

to be specified later 7ag3a

error conditions - 7ag4

to be specified later 7ag4a

discussion - 7ag5

to be specified later 7ag5a

procedure name - xdtrcc 7ah

procedure function (brief) - 7ah1

to remove a tracepoint 7ah1a

arguments - 7ah2

to be specified later 7ah2a

nth argument: 7ah2b

the address of the result list 7ah2b1

results - 7ah3

to be specified later 7ah3a

error conditions - 7ah4

to be specified later 7ah4a

discussion - 7ah5

to be specified later 7ah5a

procedure name - xdtrcd 7ai

procedure function (brief) - 7ai1

to display the status of a tracepoint 7ai1a

arguments - 7ai2

to be specified later 7ai2a

nth argument: 7ai2b

the address of the result list 7ai2b1

results - 7ai3

to be specified later 7ai3a

error conditions - 7ai4

to be specified later 7ai4a

discussion - 7ai5

to be specified later 7ai5a

procedure name - xdtrcs 7aj

procedure function (brief) - 7aj1

to set a tracepoint 7aj1a

arguments - 7aj2

to be specified later 7aj2a

nth argument: 7aj2b

the address of the result list 7aj2b1

results - 7aj3

to be specified later 7aj3a

error conditions - 7aj4

to be specified later 7aj4a

discussion - 7aj5

to be specified later 7aj5a

procedure name - xdcontinue 7ak

procedure function (brief) - 7ak1

to resume target process execution after encountering a breakpoint or a tracepoint or to start target process execution after specifying a target process 7ak1a

arguments - 7ak2

to be specified later 7ak2a

nth argument: 7ak2b

the address of the result list 7ak2b1

results - 7ak3

to be specified later 7ak3a

error conditions - 7ak4

to be specified later 7ak4a

discussion - 7ak5

to be specified later 7ak5a

procedure name - xdexecute 7al

procedure function (brief) - 7al1

to execute a single instruction on behalf of the target process 7al1a

arguments - 7al2

to be specified later 7al2a

nth argument: 7al2b

the address of the result list 7al2b1

results - 7al3

to be specified later 7al3a

error conditions - 7al4

to be specified later 7al4a

discussion - 7al5

to be specified later 7al5a

UTILITY ROUTINES 8

Many of the routines called from the CLI make use of the following two utility routines for transmitting strings to the CLI to be displayed to the user, 8a

procedure name - pntstr 8b

procedure function (brief) - 8b1

to display a string on the user's terminal and/or to write the string in the current output file 8b1a arguments - 8b2

1st argument: 8b2a

the address of the string to be displayed and/or written 8b2a1

results - 8b3

NONE 8b3a

error conditions - 8b4

NONE 8b4a

discussion - 8b5

The first thing this routine does is to append a carriage-return, linefeed sequence to the end of the passed string. 8b5a

Then, if output is currently being displayed on the user's terminal, then this routine will call the SHOW procedure in the CLI to display the passed string to the user; if output is being written in an output file, then this procedure will (next) write the passed string in the output file. 8b5b

procedure name - pasnstr 8c

procedure function (brief) - 8c1

to display a string on the user's terminal and/or to write the string in the current output file, and to obtain a new value from the user (and optionally new current input mode parameters) 8c1a

arguments - 8c2

1st argument: 8c2a

the address of the string to be displayed and/or written 8c2a1

2nd argument: 8c2b

the address of a string to get written with the new value string specified by the user 8c2b1

results - 8c3

1st result: 8c3a

FALSE or a value representing the main input mode keyword specified by the user 8c3a1

2nd result: 8c3b

a value representing the secondary input mode keyword specified by the user if the main output mode has a secondary mode; FALSE otherwise 8c3b1

error conditions - 8c4

NONE 8c4a

discussion - 8c5

The purpose of this routine is to display a string to the user which represents the displaying of some entity, e.g. a cell, in the target process, and then to obtain from the user a new value string for this entity. The user may specify a new current input mode to be used to interpret the new value string. 8c5a

The first thing this routine does is to append several spaces, followed by a LARROWCHAR, followed by a space, to the passed string. 8c5b

Then, if output is being sent to the user's terminal, this routine will generate a HELP to the CLI. This HELP will cause the string to be displayed and then the execution of a CML rule that will enable the user to (optionally specify a new current input mode and to) specify a new value to replace the value just displayed. The new value specified by the user, which may be a null string, will be written in the string whose address was passed as the second argument. 8c5c

Then, if output is being written to an output file, the passed string, followed by the same appended characters mentioned above, followed by the new value string, followed by a carriage-return and a linefeed, will be written on the output file. 8c5d

Finally, this routine will return; returning the values of the keywords specified by the user for the new input mode (these values will be FALSE if the user did not specify them). 8c5e

(J26486) 17-SEP-75 13:24;;;; Title: Author(s): Kenneth E. (Ken)
Victor/KEV; Distribution: /NPG([ACTION]) RWW([INFO-ONLY]) ;
Sub-Collections: SRI-ARC NPG; Clerk: KEV; Origin: <
NSW-DEBUGGER, !DOC/DISPATCHER-MODULE.NLS;1, >, 17-SEP-75 13:21 KEV
;;;;####;

26486 Distribution

Susan K. Ocken, Raphael Rom, David C. Smith, Andy Poggio, David L. Retz, Jan A. Cornish, Larry L. Garlick, Robert Louis Belleville, Elizabeth J. Feinler, Joseph L. Ehardt, Jonathan B. Postel, Kirk E. Kelley, Karolyn J. Martin, David S. Maynard, Kenneth E. (Ken) Victor, James E. (Jim) White, Elizabeth K. Michael, Don I. Andrews, J. D. Hopper, Charles H. Irby, Harvey G. Lehtman, Richard W. Watson,

marcia,
could you please add jan kremers, and the other new programmers to the npg group,
thanx
1

(J26487) 17-SEP-75 14:10;;;; Title: Author(s): Kenneth E. (Ken) Victor/KEV; Distribution: /MLK([ACTION]) ; Sub-Collections: SRI-ARC; Clerk: KEV;

26487 Distribution
Marcia L, Keeney,

DMB 17-SEP-75 17:07 26488

ARC Meeting

Tomorrow, 9/18/75, the continuation of the ARC meeting will be held in the conference room at 11:00. Please plan to attend. Thanks-Dee 1

ARC Meeting

(J26488) 17-SEP-75 17:07;;;; Title: Author(s): Delorse M. Brooks/DMB; Distribution: /SRI-ARC([ACTION]) ; Sub-Collections: SRI-ARC; Clerk: DMB;

26488 Distribution

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,
Susan K. Ocken, Raphael Rom, David C. Smith, 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, 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

DVN 17-SEP-75 15:05 26489

DRAFT Proposal For Slot-Based Documentation Community

See beginning of text for comment.

DRAFT Proposal For Slot-Based Documentation Community

COMMENT: At the stage of (25867,) and (26132,) the effort to prepare something to hand to customers began to concetrate on a document that would promote design and analysis. This is the other fork, a proposal for community organization. If there is to be a community something like this should be sent to the KWACS before their meeting in Boston October 13 and discussed with them there. If you have read my comments on the Advanced Automation community(33456,) you will see I have borrowed many of its ideas. The cost section is most rough. Some one who understands pricing (JCN?) should go over it. 1

Summary 2

The Augmentation Research Center (ARC) is seeking technology transfer and development support though establishment of communities of users sharing common activities. Many users of the Online System (NLS) developed at ARC are employing it to produce documents, and explore new methods of document production. Production includes original typing, editing and review, printing, distribution, and subsequent updating. Sharing what they have learned and cost of new developments would benefit them all as well as new users interested in NLS for this purpose. SRI propose to organize a community of users based on communication facilities, shared information and development costs, and sharing of certain facilities. 2a

Problems 3

The Augmentation Research Center Developed NLS in the expectation that users would group into communities with common interests, but so far only one general community exists. Many many members now employ the system to produce documents. They range from thousands of pages with complex format and distributed production cycle to simple reports written by one author. It is easy for most of the people working with NLS to see refinements that would improve its effectiveness in this area or allow them to develop the effectiveness of thier organization. However, resources for development are limited and concentrated in certain areas. Much creative activity is going on in user organizations but communication is hit-or-miss. Considerable expertise in analysis and design of computer-based aid to publication in general exists at SRI outside of ARC, but it has not been readily available to NLS users. In a period of rising cost and technical confusion some organizations not now using NLS are looking for opportunities to keep abreast of developing technology or for a flexible medium of exploratory development. 3a

Objectives 4

Our general goal is to create a community of organizations

interested in development of computer aids to publication who will experience and explore advanced applications through use of NLS for day to day work, innovation, and communication among themselves.
To that end SRI will: 4a

Facilitate communication by means of online media, a newsletter, meetings, and periodic reports.
Provide consulting services to assist in transfering systems developments to using organizations and solving users special problems.
Undertake development work related to computer aids to document production. 4a1

Methods 5

Liaison 5a

Membership in the community includes subscription to a general description of all Publications work going on in NLS, to shared online file where users can report their activities and scan the activities of others and subscription to a printed newsletter derived from changes in that file. SRI will issue a report annually on development funded by the community. Ten copies of each will be issued to each participating client. 5a1

Membership includes participation in meetings to exchange information and guide development work in connection with the meetings of the knowledge workshop architects. 5a2

A principal role for the participating organizations is for them to acquaint the systems developers with the real problems to be solved, and to guide and assist them developments and procedures that make publication effective. 5a3

Development 5b

Beyond the cost of organization, consultation, information exchange, and resource management, SRI will devote approximately 3/4 of the funds from community subscriptions to development work in the area of applications of NLS or other computer aids to document production, initially at SRI but with with the expectation that other organizations may take part in the development work in the future. 5b1

Consulting 5c

Subscription includes up to one weeks services of an appropriate SRI professional at the subscriber's site for training or other consulting in the use of NLS as a publications tool or in the general area of publications systems design and analysis. 5c1

Subscription also includes the opportunity to visit SRI for up to a week per year for training and consultation with supporting office services provided by SRI. 5c2

Further consulting work may be arranged on an individual contract basis. 5c3

Management of Common Resources 5d

SRI offers the service of managing resources common to the DPCS community, for example transfering files and handling billing of service bureaus that make printing copy from NLS files or preparation of graphics proof copies on uncommon hardware such as high-resolution terminals or electrostatic printers. 5d1

Promotion 5e

It is our plan to seek at our cost participation from as broad a representative cross section of organizations interested in advanced development in publication as possible. Each participating organization will be expected to designate one key individual (the knowledge workshop architect) who would be directly concerned with monitoring our work and who will participate in the sessions at KWAC conferences that guide DPPCS community development and liaison. Key element of the on-going program is the participation by diverse organizations both in funding and in providing technical support. Such participation is intended to aid in attaining a major goal--i.e., to help guide the research so that it results in early and effective technological transfer to users.
Proprietary Development
Interests and objectives specific to participating clients may be handled on a confidential basis. Should a client desire to sponsor a special investigation in the course of this project, such sponsorship can be arranged on the basis of a separate contract, with complete confidentiality maintained as to purpose, scope, and results. During the course of the program, the Institute reserves the right to respond to individual requests from government agencies or commercial and industrial clients for specific studies or development work in 5e1

Proprietary Development 6

Interests and objectives specific to participating clients may be handled on a confidential basis. Should a client desire to sponsor a special investigation in the course of this project, such sponsorship can be arranged on the basis of a separate contract, with complete confidentiality maintained as to purpose, scope, and results. During the course of the program, the Institute reserves the right to respond to individual requests from government agencies or commercial and industrial clients for specific studies or development work in the general area of computer aids to publication. All such requests will be carefully screened to avoid direct conflict of interest with the proposed program. 6a

Background 7

The Augmentation Research Center (ARC) has existed for several years as a group researchers and system developers within SRI. The activities of ARC are aimed at exploring the possibilities for augmenting individuals and groups in the performance of knowledge work with the help of computer aids.
ARC's Initial Research and Development Strategy was for researchers within ARC do as much of their work as possible using the range of capabilities offered. Thus they have served not only as researchers, but also as the subjects for the analysis and evaluation of the augmentation systems for its first ten years. The Next Stage in ARC's Research and Development Strategy has now begun. They are involving a wider group of people so that they can begin to transfer the fruits of their past work to others, and so that they can obtain feedback needed for further evolution from a wider spectrum of applications than occurs in the Center alone. Thes outsiders subscribe to ARC services as to an information utility. ARPA and other government agencies have provided a considerable amount of funding for the development of the ARC technology during the past ten years. The Workshop Utility Service provides an effective medium for transfer of this technology to government, commercial, and educational organizations, thereby returning useful results from the investment. As the community of using organizations grows, this return will become increasingly more significant. It is ARC's goal that these effects will be widespread in our society, both through direct use of the Workshop Utility and from use of related systems incorporating some aspects of the technology being developed here. 7a

Establishing a Workshop Utility and providing the type of service discussed here are part of ARC's long-term commitment to pursue the continued development of Augmented Knowledge Workshops in a pragmatic, evolutionary manner. 7b

Documentation in NLS 7c

NLS provides the basis for flexible systems of creating, modifying, disseminating, and controlling documentation. NLS has particular advantages in easy modification of master copies, large-scale modification and reorganization of documents either as initial drafts or later for revision after publication, facile detailed editing, and flexibility of printed output, including line drawings. NLS is used as a medium to make printed or microfilm versions of files that are primarily intended for reading online an to publish material that would not otherwise be online.
NLS has been used for over six years to produce, reports, small users' guides, proposals, and other technical documents for the Augmentation Research Center. Beginning in 1974 it has been used for publication in other organizations including Airforce documents in the range of 1000's of pages. 7c1

Input:
Input into NLS is through typing directly online at a display terminal or typewriter-like terminals, or offline onto a magnetic medium that is later read into the computer, or through copying online files from other computer systems.
To put text directly online, NLS users employ group of commands beginning with "Insert" in the Editor Subsystem.
Input to magnetic media, on the other hand, is normally through the DEX (Deferred Execution) system. The present DEX system can operate through several terminals and digital cassette recorders. It is possible to record limited editing during input.
NLS has programs to accept input from several existing systems. Other systems may require special-purpose translations programs to format the text into ASCII TENEX files. Insert Sequential Commands in the Editor subsystem convert such files to NLS files with options to preserve their format and/or translate it into NLS hierarchy. 7c2

Draft Development:

All NLS files are organized in outline form. A group of commands in the Editor subsystem can rearrange and reorder these outlines more rapidly and flexible than is the case with paper copy or online online systems that address text line by line. This facility is particularly useful the initial stages of creating a document. Similar commands can transfer or copy files or parts of files according to their outline position or content.
Editing:
Copying transfer, and replacement commands that operate on small units of text can greatly increase the productivity of editors. Automatic editing facilities are found in the

Publish, Modify and Format Subsystem. The Publish Subsystem contains, for example, a command to generate a table of contents. The Modify subsystem contains a command to correct the number of spaces between sentences, and the Format subsystem a command to set up an online file for printing in one of several standard formats.
Draft Control:
Present controls include the date and user's identity of the last change recorded with every NLS statement and automatic dating and identification of printed drafts. Automatic queing and access controls for drafts could easily be built.
Illustration:
The Graphics subsystem allows you to draw and edit simple illustrations, e.g. organization or flow charts, that are part of NLS files. Text and graphics are fully integrated. Users with screens of sufficient resolution may view and edit such drawings and print them through appropriate printers. In the case of half tones and complex line drawings, the user must set aside white space with format directives and strip in the the illustrations during printing in the manner normal to photo offset publication. 7c3

Output:
Commands in the Editor subsystem allow printing text in a simple draft form (Output Quickprint), or a format with headers, footers, control of top and side margins, etc., in a monospace font on a local printer or terminal (Output Printer), or via output to microfilm and offset plates with a variety of type sizes, fonts, and columnation (Output COM). Coded directives, visible online but not printed, control format via Output Printer or Output COM. Such directives are most often inserted automatically by use of the Format subsystem or the Sendmail subsystem, but may also be inserted by users with special training. 7c4

Post publication control:
The Automatic numbering and indexing services of the NLS Sendmail subsystem provide a medium for freezing, cataloging, and identifying documents, and recording their standing with respect to updates. Systems to automatically reformat an print changed pages could easily be built.
Procedures:
NLS offers new freedom to the publications process. Procedures that have in the past been forced on us by the medium, for example limited distribution of drafts, become matters of option. As a result introduction of NLS into a publications operation on more than an occasional basis requires careful planning. 7c5

Finally, we should point out that some industrial sectors have realized extraordinary success in making the transition to automated publication aids. Examples are the newspaper publication industry, large book publishers, phone directory publishers, and others. This segment of the industry has met common problems with a strength gained from cooperation and pooled resources to meet their shared needs. The results have been commendable and attest to the cost savings and production streamlining that can be accomplished by procedure rationalization and judicious application of new technology tools. 7d

Rough Costs: 8

We suggest a subscription price of $13,000 per year on the model of a similar community function in the area of advanced automation. In addition $2000 of the price of a slot would be paid to the community in the belief community services would replace a small part of slot services. The first $58,000 of community income would go for community services. Above 58,000, $10,000 dollars per subscriber would go for development work in this area.
Breakdown of the $58,000 cost of operation
1500 $/slot for the week of consulting including travel that means when the 4th subscriber joins we get 3 person months of development work, and about 4 person moth per subscriber thereafter.
$1000 $/per slot for service community member visiting SRI. One professional full time doing information facilitation, organization, planning.: 40K (my salary for half or three quarter time, some support)
1/2 person time for clerical support 8K
Several organizations have bought more than one slot. Only one subscription would be appropriate per organization. 8a

DRAFT Proposal For Slot-Based Documentation Community

(J26489) 17-SEP-75 15:05;;;; Title: Author(s): Dirk H. Van Nouhuys/DVN; Distribution: /NRN([ACTION]) DRB([ACTION]) DCE([ACTION]) KLM([ACTION] Docplan notebook please) DOCPLAN([INFO-ONLY]) BEV([INFO-ONLY]) POOH([INFO-ONLY]) ; Sub-Collections: SRI-ARC DOCPLAN; Clerk: DVN;

26489 Distribution

Norman R. Nielsen, David R. Brown, Douglas C. Engelbart, Kathey L. Mabrey, Raymond R. Panko, James H. Bair, David R. Brown, Glenn A. Sherwood, N. Dean Meyer, Kathey L. Mabrey, Norman R. Nielsen, Thomas L. Humphrey, Robert Louis Belleville, Elizabeth K. Michael, Richard W. Watson, James C. Norton, Robert N. Lieberman, Pat Whiting O'Keefe, Douglas C. Engelbart, Dirk H. Van Nouhuys, Beverly Boli, Ann Weinberg,

JEW 17-SEP-75 17:50 26490

Another Man's Views on the Intelligence Community

See also (33442,) (26448,) (33469,) (26455,) (33476,) (26456,) (26465,) and (33485,).

JEW 17-SEP-75 17:50 26490

Another Man's Views on the Intelligence Community

The following are my present feelings about some of the issues raised so far regarding receiving funding from the military and/or intelligence communities. These thoughts are offered because I believe, with Charles, first that one SHOULD bring his personal values to work with him, and second that the personal goals of each member of the group do indeed have a bearing upon the goals which the group as a whole can achieve. 1

Do the military and government agencies like the CIA and NSA serve a useful (i.e. necessary) function? 2

Although I have not always felt this way, I believe that they ARE necessary. The necessity arises because men simply do not in practice adhere to their own moral standards, either when acting as individuals (giving rise to police forces) or as nations (giving rise to intelligence groups and armies). I think history gives us every reason to believe that if US military and intelligence agencies were disbanded, we wouldn't long exist as a free nation. 2a

I think the real source of horror and shame which the thought of these groups tends to evoke in me, and in most of us, is not that the groups exist, but first that they NEED to exist and second that they themselves are so often corrrupt. Although I oppose many present activities of the government in general and the intelligence community in particular (e.g. direct intervention in the affairs of other nations, the gathering of information unrelated to their mission, and unlawful acts of various kinds), it's important for me to keep sight of the fact that it is these corruptions that I oppose, and not the groups themselves. 2b

Is it healthy to spend one's every day writing about ICBMs? 3

I don't believe it is, and therefore have great empathy with Ann and Kirk, who have been placed in precisely that position. Even if one believes, as I do, that a soldier must sometimes shoot people, it doesn't follow that it's healthy for the general citizenry to spend all its time reading war novels. 3a

The psychological health of the soldier or general is, of course, even more severely threatened, which perhaps helps explain the corruption which so inevitably arises in such domains. 3a1

Is the augmentation of groups who refuse to work openly and publicly an appropriate task for ARC? 4

I believe the answer to this question is yes. I have never understood our goal to be the construction of a single, mammoth knowledge workshop (OFFICE-1 through OFFICE-10,000) where all information within the system is accessible to all. 4a

The Utility already has clients whose Journal inputs have restricted access; if the AKW notion ever really catches on, there will be many more such user groups. The CIA is, of course, one example of a group whose knowledge must largely be kept secret; but there are numerous other examples (e.g. the Social Security administration) which don't have such malevolent connotations. 4b

I think what we're really interested in selling is the AKW approach; many only loosely affiliated AKWs will probably result. Although a user will surely be able, in general, to disseminate information to users in other workshops, he cannot reasonably be required to do so. If it's really our goal to create one single knowledge store, I don't think it's a realistic one. 4c

Does involvement with the military and the intelligence community pose a threat to ARC's image? 5

Yes, I think it does, in the minds of some. I have a friend who recently told me that, as someone who works with computers, my job is killing people. But in the long run, I don't think one acquires a good image by consciously seeking one. 5a

Are we missing real opportunities elsewhere by accepting funding from the intelligence community? 6

I don't know the answer to this question, and therefore tend to trust Doug's judgment here. Although I believe Doug can be wrong in his judgment, I don't believe he would intentionally misrepresent the funding climate. 6a

Assuming that he's right and that the funding obtainable elsewhere would be insufficient to support us all, there are, I think, legitimate arguments for going that route anyway, one being that the CIA will probably never lack the manpower required to carry out the tasks it undertakes, while the same cannot be said for many non-defense-related agencies. But before choosing such a course, the possibility of staffing cuts must be squarely faced. 6b

Having said all this let me say also that a few years ago I would have answered each of these questions in a completely different way than I do now, as the Marine Corps would attest. I therefore have some understanding of, and great respect for the kind of objections being raised. I think it's very important to ARC's internal health that guidelines in this area be formulated, whether by popular vote or management fiat, and then explicitly stated, so we don't continue feeling that the issue's been sidestepped, and so we all have a basis for making decisions about our own role within the group. 7

JEW 17-SEP-75 17:50 26490

Another Man's Views on the Intelligence Community

(J26490) 17-SEP-75 17:50;;;; Title: Author(s): James E. (Jim) White/JEW; Distribution: /SRI-ARC([INFO-ONLY]) ; Sub-Collections: SRI-ARC; Clerk: JEW; Origin: < JWHITE, ARCMSG.NLS;4, >, 17-SEP-75 17:37 JEW ;;;;####;

26490 Distribution

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,
Susan K. Ocken, Raphael Rom, David C. Smith, 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, 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

CHI 17-SEP-75 15:35 26491

NSW Protocol Thoughts -- Interhost Communication

The following thoughts were sent in an earlier message. However, since many people have told me they did not receive the original, I am resending the ideas. 1

Interhost protocol thoughts: 2

Process identification: 2a

Processes can be uniquely identified by their host "connection". The NCP already provides this level of multiplexing and we should use it. The problems this generates for the Works Manager should be delt with as an exceptional case and should not impact the design of all communication. The creation and destruction of NSW processes on remote hosts is discussed below. 2a1

Basic message forms: 2b

PCPB8 data structures should be the allowed data for these connections. 2b1

The following lists should be used for inter-process communication (with semantics as per current MSG usage): 2b2

INVOKE-FUNCTION: 2b2a

LIST(type=1, tid, function-name, arguments) 2b2a1

ACKNOWLEDGEMENT: 2b2b

LIST(type=2, tid, error-code, results) 2b2b1

PROTOCOL-ERROR: 2b2c

LIST(type=3, format-error-code, error-string) 2b2c1

Used to report to a process that the last message sent by it was uninterpretable (not of legal format). This would NOT be used to report that a specified function was not found or that the form of the arguments to a function were wrong. It is only used to report illegal data types or other syntactic errors. 2b2c1a

where 2b3

type is INDEX in range 1 to 3. 2b3a

tid (transaction identifier) is EMPTY OR INDEX for INVOKE and INDEX for ACKNOWLEDGE, EMPTY meaning no acknowledgement expected. The tid used in an acknowledgement must correspond to the tid used in the invoke that is being acknowledged. 2b3b

function-name is uppercase CHARSTR. 2b3c

arguments and results are EMPTY OR LIST(elements), where EMPTY denotes no arguments/results and elements are any valid PCPB8 data elements. 2b3d

error-code is EMPTY OR INDEX, EMPTY meaning successfully processed. If INDEX then first result should be an error-string. 2b3e

format-error-code is INDEX. 2b3f

error-string is CHARSTR to be logged and/or presented to user. 2b3g

examples: 2b4

invoke: 2b4a

LIST(1, 1, "WMRENAME", LIST("oldfilename", "newfilename")) 2b4a1

acknowledge: 2b4b

LIST(2, 1, EMPTY, LIST("fulloldfilename", "fullnewfilename")) 2b4b1

protocol-error: 2b4c

LIST(3, 23, "INVALID PCPB8 DATA ELEMENT ENCOUNTERED") 2b4c1

Creating/destroying processes: 2c

Below I discuss two ways of creating/destroying NSW processes on remote hosts. The simplest form to implement unfortunately suffers from the deficiency that a process must be asked to destroy itself. This is obviously difficult if it is looping! Since processes correspond roughly to tools and since tools will be added forever, this seems fairly serious (there will always be processes that are barely running and will loop). 2c1

Method A (conventional method): 2c1a

Each TBH will provide a contact socket for creating NSW processes (ala TELNET, FTP). An ICP will be done to this socket and will yield another socket (pair). The process that is then listening to the new socket is capable of receiving a message asking it to create a process named P using login info L. Assuming the correctness of P and L, the sockets are then connected to the desired process. To destroy the process later, it is necessary to send it a message asking it to kill itself. All messages referred to are of the type described above. 2c1a1

Method B: 2c1b

Each TBH provides a contact socket that expects a new flavor of ICP, which names includes specification of the process to create P and the login parameters L. It returns the sockets to use subsequently to communicate with the newly created process. To destroy the process, a message is sent to the contact socket rather than the socket allocated for the process. 2c1b1

Thus, this contact socket can be thought of as a process with two functions 2c1b1a

CREATE-PROCESS(process-name, login-parameters -> sockets) and 2c1b1a1

DESTROY-PROCESS(login-parameters, sockets -> cost). 2c1b1a2

login-parameters are repeated as verification that the requestor has the right to destroy the specified process. 2c1b1a2a

process-name is a CHARSTR. 2c1b1b

login-pararmeters is LIST(%user-name% CHARSTR, %password% CHARSTR, %password% (EMPTY/CHARSTR)). 2c1b1c

sockets is an INTEGER representing an even/odd socket pair. 2c1b1d

This scheme has the advantage that a process is killed by a higher level, debugged piece of code, rather than asking the process to commit suicide. It might be harder to implement on some hosts because it does not use traditional ICP. (The same effect could be attained by using ICP to a contact socket which returned the socket number of a process that could create/destroy NSW processes and allocate new sockets for newly-created processes. The extra ICP and sockets is a little embarrassing, however.) 2c1b2

I personally find method B more appealing because it addresses NSW needs more directly and seems much more reliable. The nature of the processes that are being created/destroyed are quite different than traditional server processes that are one-per-host, system modules that can be expected to function properly. These NSW processes are many-per-host, application programs. They must be expected to mal-function and there must be provision for killing them when they do. 2c2

NSW Protocol Thoughts -- Interhost Communication

(J26491) 17-SEP-75 15:35;;;; Title: Author(s): Charles H. Irby/CHI; Distribution: /ARC-DEV([INFO-ONLY]) ; Sub-Collections: SRI-ARC ARC-DEV; Clerk: CHI; Origin: < IRBY, CHI.NLS;1, >, 17-SEP-75 15:32 CHI ;;;;####;

26491 Distribution

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

JEW 17-SEP-75 19:40 26492

AKW System Design Note 1

Very rough, rudimentary thoughts -- largely derived from DCE's 1973 paper -- which hopefully will evolve into a more detailed AKW system design.

AKW System Design Note 1

This memo begins to summarize the system characteristics necessary to construct the Augmented Knowledge Workshop (AKW) described by ARC in such papers as "The Augmented Knowledge Workshop" (14724,), which DCE, RWW, and JCN wrote for the 1973 NCC. Throughout this memo, the AKW system "framework" will be distinguished from the collection of tools which this framework contains or makes available to its users. 1

Tool Types 1a

- The AKW will provide access to a family of "core" tools characterized by their intrinsic importance and heavy use in all areas of knowledge work. 1a1

The following classes of core tools have been identified; others probably exist: 1a1a

1) those for studying and manipulating on-line documents (e.g. the NLS editor). 1a1a1

2) those for conducting real-time and less-than-real-time dialog with other AKW users (i.e. teleconferencing and message exchange; e.g. the NLS shared screen and Journal facilities). 1a1a2

Whether core tools are in any way special-cased by the framework remains to be determined. 1a1b

- The AKW will provide access to other, more "specialized" tools which will be used less frequently than core tools, and in general by only a portion of the AKW user population. 1a2

- The AKW must be capable of assimilating new, even undebugged tools (both core and specialized) without jeopardizing the integrity of the framework. 1a3

Although a fairly comprehensive initial set of tools can probably be identified and made part of the AKW's initial configuration, new tools will be needed, built, and installed long after the system programmers who implement the framework have completed their task. 1a3a

- The AKW must provide limited access to "non-AKW" tools, as well as to those tools either created or modified (with varying degrees of effort) expressly for AKW use. 1a4

Tool Relationships 1b

- The AKW will be embedded in a communications network (e.g. the ARPANET), constitute a distributed system, and therefore require appropriate interfacing software and protocols for binding its various modules together. 1b1

These facts arise because the AKW will provide geographically separated users with access to a potentially large and diverse set of tools, constructed by many independent software manufacturers, each of whom implements the tools he vends using the programming language, operating system, and host computer he deems most appropriate to the task. 1b1a

A number of approachs toward interconnecting modules of a large, distributed system have already been explored. Work in this area includes the Resource Sharing Executive (RSEXEC) and Route Oriented Simulation System (McROSS), both developed by BBN; the Distributed Programming System (DPS) developed by ARC for NSW; and the MSG program being developed by MCA. 1b1b

The most appropriate vehicle for AKW may be a composite of several of these systems, or an entirely new protocol which learns from their mistakes. 1b1b1

- The AKW will, by means of a common command language interpreter (CLI), provide a user interface which is consistent in form across all tools. "A user will learn to use additional functions by increasing vocabulary, not by having to learn separate 'foreign' languages." 1b2

The CLI and associated Control Meta Language (CML) developed by ARC for NLS and adapted for use in NSW exemplify this aspect of the AKW. 1b2a

The existence of a common CLI implies the absence of a specialized CLI within the tool itself. An AKW tool (in contrast to a non-AKW tool) is thus a set of functions which can be invoked programatically by the CLI; a non-AKW tool contains its own peculiar CLI, and therefore cannot be used with the AKW CLI. 1b2b

- The AKW will facilitate the application of several tools to a single task by allowing the user to supply results produced by one tool as arguments to another. 1b3

Of most practical importance here is the ability to couple core tools to specialized tools, so that, for example, an editor can be employed to prepare input for or manipulate output from a data management system. However, it seems quite probable that implementing this special case implements the general case as well. 1b3a

This is one of the areas in which NSW is very weak. 1b3b

- The AKW must permit tools (both core and specialized) of the same general class (e.g. editors) to compete for the user's attention. 1b4

This requirement constrains the extent to and manner in which core tools can be special-cased by the framework. 1b4a

- The AKW will present a unified file system from which all tools take their input and deposit their output. 1b5

Operational Facilities 1c

- The AKW framework will provide run-time handles for extracting use and performance data which will help identify areas of the system which, because of their high utilization or unsatisfactory performance, demand the attention of system programmers. 1c1

Such information will be both provided upon specific request, or logged periodically at a central "Workshop Control Center" (WCC). 1c1a

- The AKW will provide sophisticated interactive debugging facilities for use by system programmers in shaking down new areas of the framework. 1c2

Programming Capabilities 1d

- The AKW must provide a user programming capability. 1d1

The rate at which users will require very specialized tools tailored to their particular tasks will exceed the rate at which applications programmers can supply them. 1d1a

- The AKW framework will provide powerful interactive debugging facilities which tool suppliers and users can employ to debug their tools and user programs, respectively. 1d2

Management Facilities 1e

- The AKW will provide administrative controls which managers can employ to regulate use of the workshop by users in their charge. 1e1

- The AKW will provide centralized billing for all tools it offers the user. 1e2

AKW System Design Note 1

(J26492) 17-SEP-75 19:40;;;; Title: Author(s): James E. (Jim) White/JEW; Distribution: /SRI-ARC([INFO-ONLY]) ; Sub-Collections: SRI-ARC; Clerk: JEW; Origin: < JWHITE, AKWMSG.NLS;5, >, 17-SEP-75 19:33 JEW ;;;;####;

26492 Distribution

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,
Susan K. Ocken, Raphael Rom, David C. Smith, 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, 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

RLL 17-SEP-75 19:53 26493

Photographer coming Thursday.

SRI commanders have decided to have each suborganizational group prepare a slide show about themselves. Wonderful, wonderful. Our own show and tell story has been placed in the inexperienced hands of myself. Well, fret not The only part you all might play is being yourselves tomorrow (hard to do??) But I thought I would warn you of this gross production that is pending tomorrow. The conference room will be the photo center except for the time our meeting is on. thank you for your cooperation. Rob 1

Photographer coming Thursday.

(J26493) 17-SEP-75 19:53;;;; Title: Author(s): Robert N. Lieberman/RLL; Distribution: /SRI-ARC([ACTION]) ; Sub-Collections: SRI-ARC; Clerk: RLL;

26493 Distribution

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, Susan K. Ocken, Raphael Rom, David C. Smith, 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, 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