1 Introduction 1

1a Data structures will be encoded according to PCPB8 when the physical channel transmits messages which are streams of 8-bit bytes. 1a

2 Data Structure Encoding 2

2a The first byte of a data structure is a type code, with the type zero having the special interpretation indicating that a key is present for this data structure, non-zero codes indicate element types. 2a

2b Key 2b

2b1 FLAG (1 byte) = 0 2b1

2b2 VALUE (any element) 2b2

2c Elements 2c

2c1 EMPTY 2c1

2c1a TYPE (1 byte) = 1

2c1b VALUE (none) empty

2c2 BOOLEAN 2c2

2c2a TYPE (1 byte) = 2

2c2b VALUE (1 byte) boolean

2c2b1 FALSE=0

2c2b2 TRUE =1

2c3 INDEX 2c3

2c3a TYPE (1 byte) = 3

2c3b VALUE (2 bytes) index

2c3b1 The value represents a positive integer in the range 1 through 2**15 - 1

2c4 INTEGER 2c4

2c4a TYPE (1 byte) = 4

2c4b VALUE (4 bytes) two's complement integer

2c5 BITSTR 2c5

2c5a TYPE (1 byte) = 5

2c5b COUNT (3 bytes)

2c5c VALUE (count bits) left adjusted in ((count+7)/8) bytes)

2c6 CHARSTR 2c6

2c6a TYPE (1 byte) = 6

2c6b COUNT (3 bytes)

2c6c VALUE (count bytes) ascii text

2c7 LIST 2c7

2c7a TYPE (1 byte) = 7

2c7b COUNT (3 bytes)

2c7b1 Note: Lists of unspecified length are specified by setting the COUNT to all ones. The end of such a list is indicated by a byte of all ones in place of a TYPE field.

2c7c REPEAT (1 byte)

2c7c1 SPECIFIEDELEMENTS=0

2c7c1a Count Data Structures

2c7c2 REPEATEDELEMENT=1

2c7c2a One Data Structure (representing count repeated instances)

2c7c3 REPEATEDVALUE=2

2c7c3a One Type (count, repeat) and count element values

3 Data Structure Format 3

3a datastruc

key	element

3a

3a1 key 3a1

3a1a key

0	element
1	x

3a2 element 3a2

3a2a empty

1
1

3a2b boolean

2	0 or 1	0 for FALSE or 1 for TRUE
1	1	

3a2c index

3	index	small positive integer
1	2	

```
3a2d          *------*----------*
integer *  4  * integer * two's complement integer
        *------*----------*
           1        4

3a2e          *------*--------*--------*
bitstr  *  5  * count * bits *
        *------*--------*--------*
           1        3      count   ((count+7)/8 bytes)

3a2f          *------*--------*--------*
charstr *  6  * count * text * Network ASCII
        *------*--------*--------*
           1        3      count

3a2g          *------*--------*---------*-------------------*
list    *  7  * count * repeat * count-structures *
        *------*--------*---------*-------------------*
           1        3         1
```

4 Examples

4a Empty

4a1

```
*------*
*  1   *
*------*
```

4b Boolean "TRUE"

4b1

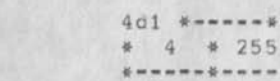
```
*------*------*
*  2   *  1   *
*------*------*
```

4c Index "7"

4c1

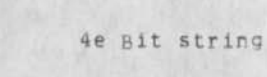
```
*------*------*------*
*  3   *  0   *  7   *
*------*------*------*
```

4d Integer "-3" 4d



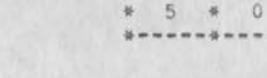
4d1

4e Bit string "10001111101011" 4e



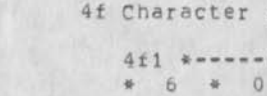
4e1

4f Character string "ABCDE" 4f

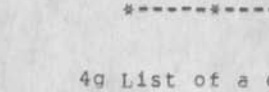


4f1

4g List of a character string "ABC" and a boolean "FALSE" 4g

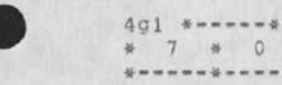


4g1



4g2

4h or 4h



4h1

```
4h2     -*------*------*------*------*------*------*
        *  A   *  B   *  C   *  2   *  0   * 255  *
        -*------*------*------*------*------*------*
```

4h2

4i List of two the booleans "TRUE", "FALSE" 4i

```
4i1 *------*------*------*------*------*------*------*------*------*
*  7  *  0  *  0  *  2  *  0  *  2  *  1  *  2  *  0  *
*------*------*------*------*------*------*------*------*------*
```

4i1

4j or 4j

```
4j1 *------*------*------*------*------*------*------*------*------*
*  7  *  0  *  0  *  2  *  2  *  2  *  1  *  0  *
*------*------*------*------*------*------*------*------*------*
```

4j1

4k Boolean "TRUE" with character string Key "X1" 4k

```
4k1 *------*------*------*------*------*------*------*------*------*
*  0  *  6  *  0  *  0  *  2  *  X  *  1  *  2  *  1  *
*------*------*------*------*------*------*------*------*------*
```

4k1

(J26152) 17-JUL-75 16:55;;;; Title: Author(s): Jonathan B.
Postel/JBP; Distribution: /NSW([INFO-ONLY]) ; Sub-Collections:
SRI-ARC NSW; Clerk: JBP; Origin: < POSTEL, PCPB8.NLS;7, >,
17-JUL-75 16:44 JBP ;;;;####;

1 26152 Distribution

1a Joseph L. Ehardt, Andy Poggio, Jan A. Cornish, Larry L. Garlick, Elizabeth J. Feinler, Kirk Sattley, Ronald P. Uhlig, James B. Lloyd, Frank J. Natoli, Peter C. Waal, Elizabeth K. Michael, William E. Carlson, Steven D. Crocker, David L. Carlstrom, Robert M. Balzer, Richard W. Watson, Lawrence A. Crain, Anthony A.L. Baggiano, Mike A. Wingfield, Jonathan B. Postel, Robert E. Millstein, Duane L. Stone, James E. (Jim) White, Albert J. Mayhan, Albert Vezza, Charles H. Irby, Eugene W. Stubbs, David L. Retz, Stephen T. Walker,

1 XXXXX JOURNAL MAIL XXXXX 1

(J26153) 18-JUL-75 18:21;;;; Title: Author(s): Andy Poggio/ANDY; Distribution: /ANDY([INFO-ONLY]) ; Sub-Collections: SRI-ARC; Clerk: ANDY;

mail from Watson

1 18-JUL-75 10:45:20-EDT,725:000000000000
Mail from BBN-TENEXB rcvd at 18-JUL-75 1045-EDT
Date: 18 JUL 1975 1043-EDT
Sender: WATSON at BBN-TENEXB
Subject: Meeting between myself and Postel to Review some things before I go away for a week From: WATSON at BBN-TENEXB
To: POSTEL
Cc: WATSON, IRBY, EHARDT, LEHTMAN, MICHAEL
Message-ID: <[BBN-TENEXB]18-JUL-75 10:43:53-EDT.WATSON> 1

2 Jon, I would like to get from you drafts of milestones, Ehardt PDP 11 memory position paper, and Lehtman NLS WM interface paper by 2:00 today so I can review them and then meet with you and others if needed starting at 3:00 as I would like them to go out either by end of today or early next week while I'm gone and want chance to make one more pass at them. Thanks Dick -------
18-JUL-75 10:58:40-EDT,1660:000000000000
Mail from BBN-TENEXB rcvd at 18-JUL-75 1058-EDT
Date: 18 JUL 1975 1056-EDT
Sender: WATSON at BBN-TENEXB
Subject: DEX
From: WATSON at BBN-TENEXB
To: BELLEVILLE, LEHTMAN
Cc: WATSON, kremers at SRI-AI, POSTEL, MICHAEL, CORNISH
Message-ID: <[BBN-TENEXB]18-JUL-75 10:56:12-EDT.WATSON> 2

3 Harvey, Bob, Doug feels pretty strongly (I think very) that the think piece on the small 11 floppy disk type Dex paper needs to sit in a paper with whole dex spectrum of which he feels thats one probable end point. If possible I would like to get together with you two at 1:00 to talk about it and anyone else that might be interested . Norton considers Dex his number one development need.
I see a staged sort of development plan.
stage 0 just getting whats there working and its requirements on hardware etc clear, stage 1 to stage n where depending on how we feel n could be small or large more editing capabilities in cassette type Dex, stage n+1 pdp 11 floppy disk dex.
Someplace in all this has to be better interface to sequential world I think or maybe that just falls out of NSW file things I do not know. Anyway in thinking about marketting over next n months I know Development has to have individuals with some time to deal with specific application areas. The DPCS area has three main components Graphics, Dex and evolution of Output Processor and Portrayal generaltion. Am assuming Bob and Elizabeth are key people to help formulate plans here awith help from others as needed. Jan K and C are involved as J K is doing something about stage 0, J C may get involved in building stage 1 with Utility funds but thats not clear yet. Dick -------
18-JUL-75 11:11:02-EDT,586:000000000000

mail from Watson

Mail from BBN-TENEXB rcvd at 18-JUL-75 1111-EDT
Date: 18 JUL 1975 1101-EDT
Sender: WATSON at BBN-TENEXB
Subject: milestones
From: WATSON at BBN-TENEXB
To: POSTEL
Cc: WATSON
Message-ID: <[BBN-TENEXB]18-JUL-75 11:01:18-EDT.WATSON> 3

4 Jon, the sub NSW areas should probably have meetings next week so everyone knows what to charge and can review his individual milestones. Besides the milestones that go out to MCA I would like more detailed ones by individual so we have a better DEWline system to see if we are sliding off the plan and need to reformulate. Thanks
Dick -------
19-JUL-75 10:44:52-EDT,347;000000000000
Mail from BBN-TENEXB rcvd at 19-JUL-75 1044-EDT
Date: 19 JUL 1975 1043-EDT
Sender: WATSON at BBN-TENEXB
Subject: distribution of PDP 11 core doc
From: WATSON at BBN-TENEXB
To: EHARDT, POSTEL
Cc: WATSON
Message-ID: <[BBN-TENEXB]19-JUL-75 10:43:42-EDT.WATSON> 4

5 Please add waal'isi and probably triolo to distribution. Thanks
Dick -------
19-JUL-75 11:05:02-EDT,1338;000000000000
Mail from BBN-TENEXB rcvd at 19-JUL-75 1105-EDT
Date: 19 JUL 1975 1103-EDT
Sender: WATSON at BBN-TENEXB
Subject: proposed structured message protocol
From: WATSON at BBN-TENEXB
To: JWHITE, POSTEL, HOPPER
Cc: WATSON
Message-ID: <[BBN-TENEXB]19-JUL-75 11:03:01-EDT.WATSON> 5

6 There is lots of momentum building up behind the protocol. Vezza has approved it, ISI is likely to use it, BBN is planning to implement it. If we are to impact it we need to reply the week after next.
The concerns I have about it are:
1) should we recommend that a second design pass be made to consider basing it on DPS entirely, 2) does it make sense for our multi host journal ideas as is, 3) does it handle NLS files with graphics and links as message bodies correctly or at all, 4) What would have to be added to DPS to handle the request reply ideas as proposed without all the introduce permanent connection approach of DPS. I guess given that if this thing goes through we are likely to have to live with it

JBP 21-JUL-75 18:35 26161

mail from Watson

Charles should make a pass through it also. There is another couple hardcopy versons on my desk. Jim can you please coordinate the views of the four of you and pass acopy of this to charles and we will get togehter a week from monday - it would even be useful if you could have a draft reply as a strawman. Thanks Dick -------

6

JBP 21-JUL-75 18:35 26161

mail from Watson

(J26161) 21-JUL-75 18:35;;;; Title: Author(s): Jonathan B. Postel/JBP; Distribution: /JBP([INFO-ONLY]) ; Sub-Collections: SRI-ARC; Clerk: JBP;

mail from Watson

1 18-JUL-75 10:45:20-EDT,725;000000000000
Mail from BBN-TENEXB rcvd at 18-JUL-75 1045-EDT
Date: 18 JUL 1975 1043-EDT
Sender: WATSON at BBN-TENEXB
Subject: Meeting between myself and Postel to Review some things before I go away for a week From: WATSON at BBN-TENEXB
To: POSTEL
Cc: WATSON, IRBY, EHARDT, LEHTMAN, MICHAEL
Message-ID: <[BBN-TENEXB]18-JUL-75 10:43:53-EDT.WATSON> 1

2 Jon, I would like to get from you drafts of milestones, Ehardt PDP 11 memory position paper, and Lehtman NLS WM interface paper by 2:00 today so I can review them and then meet with you and others if needed starting at 3:00 as I would like them to go out either by end of today or early next week while I'm gone and want chance to make one more pass at them. Thanks Dick -------
18-JUL-75 10:58:40-EDT,1660;000000000000
Mail from BBN-TENEXB rcvd at 18-JUL-75 1058-EDT
Date: 18 JUL 1975 1056-EDT
Sender: WATSON at BBN-TENEXB
Subject: DEX
From: WATSON at BBN-TENEXB
To: BELLEVILLE, LEHTMAN
Cc: WATSON, kremers at SRI-AI, POSTEL, MICHAEL, CORNISH
Message-ID: <[BBN-TENEXB]18-JUL-75 10:56:12-EDT.WATSON> 2

3 Harvey, Bob, Doug feels pretty strongly (I think very) that the think piece on the small 11 floppy disk type Dex paper needs to sit in a paper with whole dex spectrum of which he feels thats one probable end point. If possible I would like to get together with you two at 1:00 to talk about it and anyone else that might be interested . Norton considers Dex his number one development need.
I see a staged sort of development plan.
stage 0 just getting whats there working and its requirements on hardware etc clear. stage 1 to stage n where depending on how we feel n could be small or large more editing capabilities in cassette type Dex. stage n+1 pdp 11 floppy disk dex.
Someplace in all this has to be better interface to sequential world I think or maybe that just falls out of NSW file things I do not know. Anyway in thinking about marketting over next n months I know Development has to have individuals with some time to deal with specific application areas. The DPCS area has three main components Graphics, Dex and evolution of Output Processor and Portrayal generaltion. Am assuming Bob and Elizabeth are key people to help formulate plans here awith help from others as needed. Jan K and C are involved as J K is doing something about stage 0, J C may get involved in building stage 1 with Utility funds but thats not clear yet. Dick -------
18-JUL-75 11:11:02-EDT,586;000000000000

mail from Watson

Mail from BBN-TENEXB rcvd at 18-JUL-75 1111-EDT
Date: 18 JUL 1975 1101-EDT
Sender: WATSON at BBN-TENEXB
Subject: milestones
From: WATSON at BBN-TENEXB
To: POSTEL
Cc: WATSON
Message-ID: <[BBN-TENEXB]18-JUL-75 11:01:18-EDT.WATSON> 3

4 Jon, the sub NSW areas should probably have meetings next week so everyone knows what to charge and can review his individual milestones. Besides the milestones that go out to MCA I would like more detailed ones by individual so we have a better DEWline system to see if we are sliding off the plan and need to reformulate. Thanks Dick -------
19-JUL-75 10:44:52-EDT,347:000000000000
Mail from BBN-TENEXB rcvd at 19-JUL-75 1044-EDT
Date: 19 JUL 1975 1043-EDT
Sender: WATSON at BBN-TENEXB
Subject: distribution of PDP 11 core doc
From: WATSON at BBN-TENEXB
To: EHARDT, POSTEL
Cc: WATSON
Message-ID: <[BBN-TENEXB]19-JUL-75 10:43:42-EDT.WATSON> 4

5 Please add waal'isi and probably triolo to distribution. Thanks Dick -------
19-JUL-75 11:05:02-EDT,1338:000000000000
Mail from BBN-TENEXB rcvd at 19-JUL-75 1105-EDT
Date: 19 JUL 1975 1103-EDT
Sender: WATSON at BBN-TENEXB
Subject: proposed structured message protocol
From: WATSON at BBN-TENEXB
To: JWHITE, POSTEL, HOPPER
Cc: WATSON
Message-ID: <[BBN-TENEXB]19-JUL-75 11:03:01-EDT.WATSON> 5

6 There is lots of momentum building up behind the protocol. Vezza has approved it. ISI is likely to use it. BBN is planning to implement it. If we are to impact it we need to reply the week after next.
The concerns I have about it are:
1) should we recommend that a second design pass be made to consider basing it on DPS entirely. 2) does it make sense for our multi host journal ideas as is. 3) does it handle NLS files with graphics and links as message bodies correctly or at all. 4) What would have to be added to DPS to handle the request reply ideas as proposed without all the introduce permanent connection approach of DPS. I guess given that if this thing goes through we are likely to have to live with it

JBP 21-JUL-75 18:35 26162

mail from Watson

Charles should make a pass through it also. There is another couple hardcopy versons on my desk, Jim can you please coordinate the views of the four of you and pass acopy of this to Charles and we will get togehter a week from monday - it would even be useful if you could have a draft reply as a strawman. Thanks Dick -------

6

JBP 21-JUL-75 18:35 26162

mail from Watson

(J26162) 21-JUL-75 18:35;;;; Title: Author(s): Jonathan B. Postel/JBP; Distribution: /JBP([INFO-ONLY]) ; Sub-Collections: SRI-ARC; Clerk: JBP;

Current (15-JUL-75) Cassette Restrictions

1 The following restrictions currently apply to the use of the CASSETTE utility and DEX. Software modifications to be made within the next weeks will offer greater security of data capture through the network and will permit a wider variety of cassette hardware to be used, but problems with the nature of the TIP device (which was not designed with the cassette process in mind) preclude major improvements without further developments which will be outlined in later notes. 1

1a The Cassette recorder used must have the folowing remotely addressable controls: 1a

1a1 1) rewind 1a1

1a2 2) record 1a2

1a3 3) stop record 1a3

1a4 4) read 1a4

1a5 5) stop read 1a5

1b Additional cassette hardware specifications may be found in the WORKSTATION EQUIPMENT REFERENCE MANUAL, NIC # 23809. 1b

1c At present (7/15/75) the only (1) specific cassette drives supported are the following: 1c

1c1 ICP-TERMICETTE 3100 1c1

1c2 TECHTRAN 4100 1c2

1d Operation of the CASSETTE UTILITY and hence DEX itself cannot be guaranteed when any other type of drive is used. This list will be expanded in the future. 1d

1e At the present time cassette input through line processor terminals is not supported. This facility will be available in the new version of the CASSETTE UTILITY. 1e

1f When the CASSETTE UTILITY is used on a TIP line one must be absolutely sure that the size of the TIP buffer for the line being used is greater than the size of the longest tape record to be processed. This restriction is necessary because the cassette drive cannot be stopped while it is reading a record. If this rule

is not followed, the program will hang, drop characters and in general behave in an unpredictable and unreliable manner. In some cases, modifications to the sizes of buffers used for cassette input will have to be arranged with the Network Control Center to follow specifications of ARC. If there is difficulty getting all data through the network, ARC should be consulted with information about the TIP used, the TIP port, and the tape record size. 1f

1g When the CASSETTE UTILITY is used via a TELNET connection the TELNET escape character must be changed to something other that the default ^Z. (^W is recommended). TELNET must also be told to operate in "transparent mode" to avoid interception of control and special characters from the tape. 1g

1h We are in the process of investigating the CASSETTE problem and hope to have a somewhat more reliable software package available within the next few weeks. The above restricions, however, will remain in force. Problems and questions concerning the DEX and CASSETTE UTILITY systems should be forwarded to LEHTMAN@BBNB, or KREMERS@BBNB. 1h

Current (15-JUL-75) Cassette Restrictions

(J26163) 21-JUL-75 19:24;;;; Title: Author(s): Jan H. Kremers, Harvey G. Lehtman/JHK HGL; Distribution: /SRI-ARC([ACTION]) JHK([ACTION]) JCN([ACTION] Should this be sent to Crain or any Utility customers?) ; Sub-Collections: NIC SRI-ARC; Clerk: HGL; Origin: < LEHTMAN, CASETT.NLS;2, >, 21-JUL-75 19:20 HGL ;;;;####;

1 26163 Distribution
1a 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, Jan H. Kremers, James C. Norton,
1p Mary Ann Kellan, Buddie J. Pine, Andy Poggio, David L. Retz, Laura J. Metzger, Karolyn J. Martin, Jan A. Cornish, Larry L. Garlick, Priscilla A. Wold, Pamela K. Allen, Delorse M. Brooks, Beverly Boli, Rita Hysmith, Log Augmentation, Joseph L. Ehardt, Raymond R. Panko, Susan Gail Roetter, Robert Louis Belleville, Rene C. Ochoa, Ann Weinberg, Joan Hamilton, Adrian C. McGinnis, Robert S. Ratner, David S. Maynard, Robert N. Lieberman, Sandy L. Johnson, James H. Bair, Jeanne M. Leavitt, Rodney A. Bondurant, Jeanne M. Beck, Marcia L. Keeney, Elizabeth K. Michael, Jonathan B. Postel, Elizabeth J. Feinler, Kirk E. Kelley, N. Dean Meyer, James E. (Jim) White

Some TIP buffer sizes

1 The following list of TIP buffer assignments was provided by Joel Malman of the Network Control Center on 16 and 18 July. They should be useful in the analysis of Cassette problems. The list of TIPs was obtained from Applications; there are a few others which should also be included (such as RADC?) Numbers are octal. Input buffers are measured in words at two characters per word. Output buffers are double buffered; therefore the output buffer listed may be taken as the octal buffer size in characters. Except for AMEs and Tymshare, the list was compiled over the phone and may be incorrect in places due to my stenographic ineptitude. 1

2 Note also the fact that Joel expressed shock that we would even consider putting the TIP which was designed "for people typing at keyboards" to use in the middle of a cassette based system. There is no flow control in the TIP and buffers will indeed overflow. 2

3 AMES:

PORT	INBUF	OUTBUF	
1,2	253	527	(9600/9600 SCOPE)
3-64	34	71	(300 MODEMS)
65	1	1	(SPARE)
66	34	162	(150/1200 TTY)
67	1	1	(SPARE)
70-74	34	71	(300 MODEMS)
75	1	1	(SPARE)
76-77	34	71	(300 MODEMS) 3

4 ARPA:

PORT	INBUF	OUTBUF
1	0	173
2	244	511
3	122	164
4	122	173
5-7	122	164
10	122	173
11	104	51
12-77	44	51 4

5 KIRKLAND:

PORT	INBUF	OUTBUF
1-3	266	554
4-7	133	266
10-17	55	133
20-37	26	55
40-77	13	26 5

6 MITRE:

PORT	INBUF	OUTBUF
2-7	17	37

Some TIP buffer sizes

10	17	176
13	17	37
16,17	1	2
20-26	17	37
27	77	176
30-33	17	37
34	77	104
35-37	17	37
40,41	77	204

6

7 RML:

PORT	INBUF	OUTBUF
1	10	21
2	37	77
3-14	10	21
15	21	42
16-25	10	21
26	37	77
27-34	10	21
35	50	21
36-61	10	21
62	31	63
63-65	10	21
66	31	63
66-77	10	21

7

8 RUTGERS:

PORT	INBUF	OUTBUF
2-5	22	44
6-10	4	6
11-12	22	44
13	4	6
14	22	44
15,16	4	6
17	44	111
20-33	4	6
34,35	111	223
36-63	4	6
64	0	156
65	0	223
66-71	4	6
72	111	215
73-77	4	6

8

9 SDAC:

PORT	INBUF	OUTBUF
1	22	70
2	22	160
3-5	22	70

Some TIP buffer sizes

6	40	214		
7	22	70		
10-12	40	214		
13	22	70		
14	22	124		
17	151	16		
21,23,26		22	16	9

10 TYMSH:

PORT	INBUF	OUTBUF		
1-14	101	64	(300 MODEMS)	
15,17	202	202	(208A EXTERNAL CLOCK MODEMS)	10

HGL 21-JUL-75 20:44 26164

some TIP buffer sizes

(J26164) 21-JUL-75 20:44;;;; Title: Author(s): Harvey G. Lehtman/HGL; Distribution: /JCN([ACTION]) RWW([ACTION]) MEH([ACTION]) JHK([ACTION]) SGR([ACTION]) SRI-ARC([INFO-ONLY]) ; Sub-Collections: SRI-ARC; Clerk: HGL;

1 26164 Distribution
1a Jonathan B. Postel, Elizabeth J. Feinler, Kirk E. Kelley, N. Dean Meyer, James E. (Jim) White, Douglas C. Engelbart, Martin E. Hardy, J. D. Hopper, Charles H. Irby, Harvey G. Lehtman, James C. Norton, Jeffrey C. Peters, Dirk H. Van Nouhuys, Kenneth E. (Ken) Victor, Richard W. Watson, Don I. Andrews,
1b James C. Norton, Richard W. Watson, Martin E. Hardy, Jan H. Kremers, Susan Gail Roetter, 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 Boll, Rita Hysmith, Log Augmentation, Joseph L. Ehardt, Raymond R. Panko, Susan Gail Roetter, Robert Louis Belleville, Rene C. Ochoa, Ann Weinberg, Joan Hamilton, Adrian C. McGinnis, Robert S. Ratner, David S. Maynard, Robert N. Lieberman, Sandy L. Johnson, James H. Bair, Jeanne M. Leavitt, Rodney A. Bondurant, Jeanne M. Beck, Marcia L. Keeney, Elizabeth K. Michael

Weekly Report

1 Last Week 1

1a nsw protocols 1a

1a1 worked not at all on updating the file package document as agreed to at the june protocol meeting 1a1

1a2 worked on the "pseudo user telnet" program for old tool interaction 1a2

1b project management 1b

1b1 Task/People/Time Chart 1b1

1b2 Milestones 1b2

1b3 Budget 1b3

1c operations at isic 1c

1c1 JSYS Traps fixed (it was a hardware problem) 1c1

1d arpa protocols 1d

1d1 Put 4 RFCs online at Office-1 1d1

1d2 Read "Message protocol" report 1d2

1d3 Read INWG notes 1d3

2 Next Week 2

2a nsw protocols 2a

2a1 get the "Pseudo user telnet" program for old tool interaction to an operational state 2a1

2a2 complete updating the file package specification documents 2a2

2b project management 2b

2b1 send milestones to compass 2b1

2c operations at isic 2c

2c1 Next goal is to have NLS 8.5 as standard NLS at ISIC 2c1

2d arpa protocols 2d

2d1 Work on response to "Message Protocol" report 2d1

2d2 Read INWG notes 2d2

2d3 Read Network Measurement notes 2d3

JBP 21-JUL-75 22:47 26165

Weekly Report

(J26165) 21-JUL-75 22:47;;;; Title: Author(s): Jonathan B. Postel/JBP; Distribution: /ARC-DEV([INFO-ONLY]) ; Sub-Collections: SRI-ARC ARC-DEV; Clerk: JBP;

DVN 22-JUL-75 11:53 26166

Custody of <Arcdocumentation>

1 One of the duties you inherited in my documentation regent hat was a vague general responsibility for this directory at BBN. I have a commands branch that does a copy directory on it with getting useful information like who wrote what and when and how big. It is <dvn,docyoudear). You are welcome to it. I just did a house cleaning an got BBN back more than 1000 pages. It still has some strange stuff like Dick's messages for the month of May. 1

DVN 22-JUL-75 11:53 26166

Custody of <Arcdocumentation>

(J26166) 22-JUL-75 11:53;;;; Title: Author(s): Dirk H. Van Nouhuys/DVN; Distribution: /BEV([ACTION]) DMB([ACTION] dirt notebook please) RWW([ACTION]) DIRT([INFO-ONLY]) ; Sub-Collections: SRI-ARC DIRT; Clerk: DVN;

1 26166 Distribution
ia Beverly Boli, Delorse M. Brooks, Richard W. Watson, 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,

JBP 22-JUL-75 12:45 26167

NSW Charge Numbers

1 we will have nnew charge numbers for NSW work starting this week so dont do you time card till you are told your new charge number --jon. 1

NSW Charge Numbers

(J26167) 22-JUL-75 12:45;;;; Title: Author(s): Jonathan B. Postel/JBP; Distribution: /ARC-DEV([ACTION]) ; Sub-Collections: SRI-ARC ARC-DEV; Clerk: JBP;

1 26167 Distribution
1a 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,

DVN 22-JUL-75 12:52 26168

Next Steps in Worksmanager Documentation

1 I have been looking over <sattley,wm-help,) I feel we are to the point we can hand over the real help file to KS. I have a number of detailed suggestions which I will journalize tomorrow when I am through looking through it. I would like to see KS's file become the file in xhelp. When he has read my suggestions and acted on them or not, he might say that file is ready for review and Ann, Beverly, Kirk Kelley, and I could review it in the way we usually do documentation. It akes me feel good that the help system has got to the point someone outside ARC can work with it. 1

DVN 22-JUL-75 12:52 26168

Next Steps in Worksmanager Documentation

(J26168) 22-JUL-75 12:52;;;; Title: Author(s): Dirk H. Van Nouhuys/DVN; Distribution: /KS([ACTION]) DMB([ACTION] dirt notebook please) DIRT([INFO-ONLY]) ; Sub-Collections: SRI-ARC DIRT; Clerk: DVN;

1 26168 Distribution
ia Kirk Sattley, Delorse M. Brooks, Jonathan B. Postel, Priscilla A. Wold, Rita Hysmith, Pamela K. Allen, Delorse M. Brooks, Elizabeth F. Finney, Beverly Boli, Lawrence A. Crain, Kirk Sattley, Susan Gail Roetter, Robert N. Lieberman, Ann Weinberg, Kenneth E. (Ken) Victor, Douglas C. Engelbart, James H. Bair, Elizabeth K. Michael, Richard W. Watson, Elizabeth J. Feinler, Harvey G. Lehtman, Kirk E. Kelley, Laura E. Gould, Jeanne M. Beck, Dirk H. Van Nouhuys, James C. Norton,

JEW 22-JUL-75 15:34 26169

DPSERRS / Dps Error List

Available on-line to non-NLS users as [ISIC]<NSW-SOURCES>DPSERRS.TXT,
and periodically updated.

DPSERRS / DPS Error List

1 Introduction 1

1a This document lists the errors currently detected and reported by DPS-10: their mneumonics within the DPS source code, their numbers in decimal, and their associated diagnostic messages. Error numbers greater than 1000 may occasionally be reported, and are generated by DPS' L10 runtime environment. 1a

2 %data structures% 2

2a edddst = 1 "?." 2a

2b eddkey = 2 "Duplicate key." 2b

2c eddrky = 3 "Key has key." 2c

2d eddsto = 4 "Duplicate data store name." 2d

2e ediidx = 5 "Illegal INDEX." 2e

2f edipdl = 6 "Illegal PSEL/DSEL." 2f

2g ediuif = 7 "Illegal USERINFO." 2g

2h edmkey = 8 "Missing key." 2h

2i edodcl = 9 "LIST too long to decode." 2i

2j edolst = 10 "Maximum LIST size exceeded." 2j

2k edostr = 11 "CHRSTR too long to decode." 2k

2l edufty = 12 "Undefined data type." 2l

2m eduity = 13 "Undefined informal data type." 2m

2n edusto = 14 "Undefined data store name." 2n

2o edwesl = 15 "Non-LIST addressed by ESEL." 2o

2p edwidx = 16 "No such index." 2p

2q edwkey = 17 "No such key." 2q

2r edwtyp = 18 "Wrong data type." 2r

2s edwpmc = 19 "Incorrect number of parameters." 2s

2t edoabc = 20 "Data structure overflows source block." 2t

```
3 %errors%                                                              3
   3a eefimp =  51  "Not implemented."                                  3a
   3b eefops =  52  "Operating system error."                           3b
   3c eemerr =  53  "Unidentifiable operating system error."            3c
   3d eeuern =  54  "Undefined error number."                           3d
   3e eef110 =  55  "L10 run-time error."                               3e
4 %events%                                                              4
   4a evfacq = 101  "Won't delete ALOCH event."                         4a
   4b evfjuv = 102  "Won't delete SIPR event."                          4b
   4c evilen = 103  "Illegal event length."                             4c
   4d evoecb = 104  "Event overflow."                                   4d
5 %folders%                                                             5
   5a efodrn = 151  "Won't create record while folder drained."         5a
   5b eforun = 152  "RUNFLD overrun."                                   5b
   5c efuift = 153  "Undefined record information type."                5c
   5d efurop = 154  "Undefined RUNFLD operation."                       5d
6 %inter-process communication%                                         6
   6a ecdcce = 201  "Channel already created."                          6a
   6b ecutyp = 202  "Undefined channel type."                           6b
   6c ecwmnu = 203  "Channel type menu mismatch."                       6c
   6d ecwpkl = 204  "Inconsistent packet length."                       6d
7 %locks%                                                               7
   7a elfded = 251  "Deadlock."                                         7a
   7b elfdel = 252  "Sought lock deleted."                              7b
   7c elflck = 253  "Lock attempt failed."                              7c
```

```
    7d elmsWp = 254  "Non-existent LCB to be swapped."                    7d
    7e elmstk = 255  "Lock stack underflow."                              7e
    7f elostk = 256  "Lock stack overflow."                               7f
    7g elistk = 257  "Lock stack surplus."                                7g
8 %packages%                                                              8
    8a ekfded = 301  "Package dead."                                      8a
    8b ekupkn = 302  "Undefined package."                                 8b
9 %procedures%                                                            9
    9a epfnoh = 351  "No help available."                                 9a
    9b epfpio = 352  "No processor with sufficient priority."             9b
    9c epfsab = 353  "Won't abort system procedure."                      9c
    9d epfsin = 354  "Won't interrupt system procedure."                  9d
    9e epiacq = 355  "Context prohibits ACQPE."                           9e
    9f epiaid = 356  "Context prohibits any action by local
                      procedure."                                         9f
    9g epihlp = 357  "Context prohibits HLPPE."                           9g
    9h epiint = 358  "Context prohibits INTPE."                           9h
    9i epimsk = 359  "Illegal argument/result list mask."                 9i
    9j epinte = 360  "Context prohibits NTEPE."                           9j
    9k epiotc = 361  "Illegal system procedure outcome."                  9k
    9l epirel = 362  "Context prohibits RELPE."                           9l
    9m epirsm = 363  "Context prohibits RSMPE."                           9m
    9n epixhp = 364  "Context prohibits call to XHLPPE."                  9n
    9o epixin = 365  "Context prohibits call to XINTPE."                  9o
    9p epixnt = 366  "Context prohibits sending XNTEPE."                  9p
```

```
    9q epixrc = 367  "Context prohibits sending XRECPE."                    9q
    9r epixrm = 368  "Context prohibits call to XRSMPE."                    9r
    9s epixrn = 369  "Context prohibits sending XRTNPE."                    9s
    9t epuotc = 370  "Undefined procedure outcome."                         9t
    9u epurtn = 371  "Undefined return type."                               9u
    9v epusyn = 372  "Undefined system procedure number."                   9v
    9w epwvis = 373  "Unplanned for visit."                                 9w
    9x epfqin = 374  "Procedure can't be interrupted."                      9x
10  %processes%                                                             10
    10a esdpoh = 401  "POH already associated with process."                10a
    10b esfded = 402  "Process dead."                                       10b
    10c esipsa = 403  "Syntax error in process addr."                       10c
    10d esircv = 404  "No POH via which to receive message."                10d
    10e esisup = 405  "Not direct superior."                                10e
    10f esmpoh = 406  "No POH via which to send message."                   10f
    10g esumsg = 407  "Undefined message number."                           10g
    10h esuser = 408  "Undefined user name."                                10h
    10i esuwtd = 409  "Undefined watchdog code."                            10i
    10j eswqak = 410  "?."                                                  10j
11  %processors%                                                            11
    11a erigtd = 451  "Context prohibits GTDPS."                            11a
    11b eriptd = 452  "Context prohibits PTDPS."                            11b
    11c erisin = 453  "Not signed in."                                      11c
    11d erorsb = 454  "ABF overflow."                                       11d
    11e erosml = 455  "Small block overflow."                               11e
```

```
    11f erowin = 456  "Processor window overflow."                    11f
    11g erualo = 457  "Undefined entity type allocated."              11g
    11h eruinf = 458  "Undefined process information type."           11h
    11i eruopn = 459  "Undefined operation number."                   11i
    11j erupmi = 460  "Undefined parameter location."                 11j
    11k erupsi = 461  "Undefined PSI channel number."                 11k
    111 erurde = 462  "Undefined entity type read."                   111
    11m eruscp = 463  "Undefined scope."                              11m
    11n erusyc = 464  "Undefined system call number."                 11n
    11o eruusc = 465  "Undefined user call number."                   11o
    11p eruwre = 466  "Undefined entity type written."                11p
    11q erfded = 467  "Processor dead."                               11q
    11r erdsin = 468  "Processor already signed in."                  11r
12 %storage%                                                          12
    12a emfexh = 501  "CF storage exhausted."                         12a
    12b emient = 502  "Negative entity size."                         12b
    12c emuent = 503  "Undefined entity type."                        12c
13 %undefined handles%                                                13
    13a egmhca = 551  "Undefined call handle."                        13a
    13b egmhcn = 552  "Undefined channel handle."                     13b
    13c egmhdt = 553  "Undefined data store handle."                  13c
    13d egmhev = 554  "Undefined event handle."                       13d
    13e egmhlk = 555  "Undefined lock handle."                        13e
    13f egmhls = 556  "Undefined lockset handle."                     13f
    13g egmhmg = 557  "Undefined manager handle."                     13g
```

```
    13h eqmhpk = 558  "Undefined package handle."                    13h
    13i eqmhpo = 559  "Undefined port handle."                       13i
    13j eqmhpr = 560  "Undefined processor handle."                  13j
    13k eqmhps = 561  "Undefined process handle."                    13k
    13l eqmhsg = 562  "Undefined segment handle."                    13l
    13m eqmhsu = 563  "Undefined subprocess handle."                 13m
    13n eqmhsv = 564  "Undefined system call handle."                13n
    13o eqmhus = 565  "Undefined user call handle."                  13o
14  %no more handles%                                               14
    14a ehohca = 601  "No call handle available."                    14a
    14b ehohcn = 602  "No channel handle available."                 14b
    14c ehohdt = 603  "No data store handle available."              14c
    14d ehohev = 604  "No event handle available."                   14d
    14e ehohlk = 605  "No lock handle available."                    14e
    14f ehohls = 606  "No lockset handle available."                 14f
    14g ehohmg = 607  "No manager handle available."                 14g
    14h ehohpk = 608  "No package handle available."                 14h
    14i ehohpo = 609  "No port handle available."                    14i
    14j ehohpr = 610  "No processor handle available."               14j
    14k ehohps = 611  "No process handle available."                 14k
    14l ehohsg = 612  "No segment handle available."                 14l
    14m ehohsu = 613  "No subprocess handle available."              14m
    14n ehohsv = 614  "No system call handle available."             14n
    14o ehohus = 615  "No user call handle available."               14o
```

DPSERRS / DPS Error List

(J26169) 22-JUL-75 15:34;;;; Title: Author(s): James E. (Jim) White/JEW; Distribution: /SRI-ARC([INFO-ONLY]) ; Sub-Collections: SRI-ARC; Clerk: JEW; Origin: < JWHITE, DPSERRS.NLS;2, >, 22-JUL-75 15:31 JEW ;;;;####;

1 26169 Distribution

1a 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,

1b Mary Ann Kellan, Buddie J. Pine, Andy Poggio, David L. Retz, Laura J. Metzger, Karolyn J. Martin, Jan A. Cornish, Larry L. Garlick, Priscilla A. Wold, Pamela K. Allen, Delorse M. Brooks, Beverly Boli, Rita Hysmith, Log Augmentation, Joseph L. Ehardt, Raymond R. Panko, Susan Gail Roetter, Robert Louis Belleville, Rene C. Ochoa, Ann Weinberg, Joan Hamilton, Adrian C. McGinnis, Robert S. Ratner, David S. Maynard, Robert N. Lieberman, Sandy L. Johnson, James H. Bair, Jeanne M. Leavitt, Rodney A. Bondurant, Jeanne M. Beck, Marcia L. Keeney, Elizabeth K. Michael, Jonathan B. Postel, Elizabeth J. Feinler, Kirk E. Kelley, N. Dean Meyer, James E. (Jim) White

Available on-line to non-NLS users as [ISIC]<NSW-SOURCES>DPSJSYS.TXT, and updated periodically.

1 Introduction

1a This document describes the internal structure of a Distributed Programming System (DPS) process on Tenex. My apologies for the terseness of the document, but time and computer resources are scarce. The reader is assumed to have as background for this present offering, a thorough understanding of the several more verbose PCP documents which have preceded it. The primary purpose of this document is to present to process implementers the details of their interface with DPS. The services provided by DPS are, roughly speaking, a superset of those described in previous documents. Comments, bug/deficiency reports, and questions are welcome.

2 Fork structure

2a A DPS process consists of a "controlling fork" (CF) containing the DPS implementation, and one or more "processing forks" (PFs) running beneath the CF, which contain user code. A PF requests the DPS services it needs by executing JSYS 400, which the CF intercepts via Tenex's JSYS trap facility and processes. A process' PFs are partitioned into one or more "subprocesses", the first called the "process leader" and created as part of the process' creation. The first PF of each subprocess is called the "subprocess leader" and is created as part of the subprocess' creation. All PFs within a subprocess execute copies of the same SAV file, and the CF routes incoming procedure calls to the apropriate subprocess and schedules its execution in an available PF within that subprocess.

3 Operations

3a Three of the five low-level "operations" implemented by the CF and accessible via JSYS 400 -- IVDPS, RRDPS, and DRDPS -- provide the PFs with access to a whole set of DPS "virtual JSYSs" (VJSYSs) implemented by the CF. Another two operations -- GTDPS and PTDPS -- provide the CF with access to a set of "virtual JUSRs" (VJUSRs; virtual jump-to-user's) implemented by the PFs. The final operation -- PGDPS -- provides a simple mechanism for VJUSR dispath for "sequential processors" (SPs) which possess the following two characteristics: they are active only while executing VJUSRs, and they execute one VJUSR at a time.

3b A PF declares itself to be a sequential processor by means of a flag bit in the SIPR VJSYS, and then invokes PGDPS. PGDPS blocks the SP until the first VJUSR is required. The SP executes the VJUSR whose number it finds in AC 0 (arguments in ACs 1-4), and then returns its results via PGDPS, which again blocks the processor. An SP need never issue either RDYPR or TSTEV/WAIEV.

3c Procedures executed by SPs can be aborted, but cannot be interrupted. To abort a procedure invoked via the PECAL VJUSR, the CF interrupts the SP on a PSI channel specified via SIPR; the SP may either ignore the interrupt, or promptly make an abort return from PECAL. SPs need not supply the PEABR, PEINT, or PERSM VJUSRs. 3c

3d Some of the arguments and results of certain operations (and VJUSRs and VJSYSs) are stored in "blocks". A block is M+1 contiguous words of memory, of which the first contains a header (XWD M,L) and the next L, data. "ABC (x)s" stands for the Address of a Block CONTAINING zero or more x's (or exactly one, if "s" is absent). "ABF (x)s" stands for the Address of a Block FOR zero or more x's (or exactly one, if "s" is absent). 3d

3e IVDPS (OP 0)
Invokes VJSYS. 3e

3e1 ACCEPTS IN 3e1

3e1a 0: XWD 3e1a

3e1a1 op [0], 3e1a1

3e1a2 ABC (3e1a2

3e1a2a XWD 3e1a2a

3e1a2a1 event handle / 0 (meaning block) 3e1a2a1

3e1a2a1a (signalled upon completion of VJSYS with completion code = XWD systemcall handle, VJSYS number), 3e1a2a1a

3e1a2a2 VJSYS number 3e1a2a2

3e1a2b XWD 3e1a2b

3e1a2b1 call handle on whose behalf the VJSYS is being executed) / 0 3e1a2b1

3e1a2b1a (meaning none), 3e1a2b1a

3e1a2b2 ABF (use by DPS in returning VJSYS results) / 0 3e1a2b2

3e1a2b2a (if not blocking operation)) 3e1a2b2a

3e1b 1-4: VJSYS arguments 3e1b

3e2 RETURNS + 3e2

3e2a systemcall handle in 0 3e2a

3e2b 1: unsuccessful, error number in 1 3e2b

3e2c 2: successful, VJSYS results in 1-4 3e2c

3f RRDPS (OP 1)
Retrieves results of VJSYS. 3f

3f1 ACCEPTS IN 3f1

3f1a 0: XWD op [1], systemcall handle 3f1a

3f1b 1: ABF (use by DPS in returning VJSYS results) 3f1b

3f2 RETURNS + 3f2

3f2a 1: unsuccessful, error number in 1 3f2a

3f2b 2: successful, VJSYS results in 1-4 3f2b

3g DRDPS (OP 2)
Discards results of VJSYS. 3g

3g1 ACCEPTS IN 3g1

3g1a 0: XWD op [2], systemcall handle 3g1a

3g2 RETURNS + 3g2

3g2a 1: unsuccessful, error number in 1 3g2a

3g2b 2: successful 3g2b

3h GTDPS (OP 3)
Gets VJUSR arguments from DPS. 3h

3h1 ACCEPTS IN 3h1

3h1a 0: XWD op [3], usercall handle 3h1a

3h1b 1: ABF (use by DPS in returning VJUSR arguments) 3h1b

3h2 RETURNS + 3h2

```
      3h2a 1:  unsuccessful, error number in 1                          3h2a
      3h2b 2:  successful,                                              3h2b
         3h2b1 XWD                                                      3h2b1
            3h2b1a VJUSR number,                                        3h2b1a
            3h2b1b requesting process handle / 0                        3h2b1b
               3h2b1b1 (meaning local DPS environment) in 0,            3h2b1b1
         3h2b2 VJUSR arguments in 1-4                                   3h2b2

   3i PTDPS (OP 4)
   Returns VJUSR results to DPS,                                        3i
      3i1 ACCEPTS IN                                                    3i1
         3i1a   0:  XWD                                                 3i1a
            3i1a1 op [4],                                               3i1a1
            3i1a2 ABC (                                                 3i1a2
               3i1a2a XWD                                               3i1a2a
                  3i1a2a1 error code / 0 (meaning successful),          3i1a2a1
                  3i1a2a2 usercall handle)                              3i1a2a2
         3i1b 1-4:  VJUSR results                                       3i1b
            3i1b1 (or, if error code specified,
            byte pointer to ASCIZ diagnostic in 1)                      3i1b1
      3i2 RETURNS +                                                     3i2
         3i2a 1:  unsuccessful, error number in 1                       3i2a
         3i2b 2:  successful                                            3i2b

   3j PGDPS (OP 5)
   [Returns previous VJUSR's results to DPS and]
   gets next VJUSR's arguments from DPS,                                3j
      3j1 ACCEPTS IN                                                    3j1
         3j1a   0: XWD                                                  3j1a
```

```
          4a1a2b ABC (byte pointers to                                  4a1a2b
            4a1a2b1 ASCIZ user name, password, and account)             4a1a2b1
        4a1a3 3: XWD                                                    4a1a3
          4a1a3a package scope / 0 (meaning open no packages),          4a1a3a
          4a1a3b process scope                                          4a1a3b
        4a1a4 4: XWD                                                    4a1a4
          4a1a4a ABC (                                                  4a1a4a
            4a1a4a1 ABC (PCPB36 package startup info) / 0               4a1a4a1
              4a1a4a1a (meaning EMPTY))s                                4a1a4a1a
            4a1a4a2 / 0 (meaning all EMPTY),                            4a1a4a2
          4a1a4b ABC (byte pointer to ASCIZ package name)s / 0          4a1a4b
            4a1a4b1 (if no packages to be opened)                       4a1a4b1
    4a1b RETURNS IN                                                     4a1b
      4a1b1 1: XWD ABC (package handle)s, process handle                4a1b1

4a2 DELPS (VJSYS 2)
Deletes previously created remote process,                              4a2
    4a2a ACCEPTS IN                                                     4a2a
      4a2a1 1: process handle / 0 (meaning all)                         4a2a1
    4a2b RETURNS IN                                                     4a2b
      4a2b1 1: cost in cents                                            4a2b1

4a3 ITDPS (VJSYS 3)
Introduces two remote processes to one another.                         4a3
    4a3a ACCEPTS IN                                                     4a3a
      4a3a1 1: XWD                                                      4a3a1
        4a3a1a ABC (PCPB36 startup info 1) / 0                          4a3a1a
          4a3a1a1 (meaning EMPTY),                                      4a3a1a1
```

4a3a1b process handle 1 4a3a1b

4a3a2 2: XWD 4a3a2

4a3a2a ABC (PCPB36 startup info 2) / 0 4a3a2a

4a3a2a1 (meaning EMPTY), 4a3a2a1

4a3a2b process handle 2 4a3a2b

4a3a3 3: XWD 4a3a3

4a3a3a flags, 4a3a3a

4a3a3a1 B0 on: logical channel only 4a3a3a1

4a3a3b scope 4a3a3b

4a3b RETURNS IN 4a3b

4a3b1 1: introduction handle 4a3b1

4a3b2 2: XWD ph12, ph21 4a3b2

4a4 SEPPS (VJSYS 4)
Separates two previously introduced remote processes. 4a4

4a4a ACCEPTS IN 4a4a

4a4a1 1: introduction handle / 0 (meaning all) 4a4a1

4a4b RETURNS IN 4a4b

4a4b1 1: cost 1 in cents 4a4b1

4a4b2 2: cost 2 in cents 4a4b2

4a5 INFPS (VJSYS 52)
Retrieves information about a remote process. 4a5

4a5a ACCEPTS IN 4a5a

4a5a1 1: XWD information type, process handle 4a5a1

4a5b RETURNS IN 4a5b

4a5b1 1: information 4a5b1

4b Packages 4b

```
4b1 OPNPK (VJSYS 5)
Opens remote packages.                                          4b1

   4b1a ACCEPTS IN                                              4b1a

      4b1a1 1:  XWD scope, process handle                       4b1a1

      4b1a2 2:  XWD                                             4b1a2

         4b1a2a ABC (                                           4b1a2a

            4b1a2a1 ABC (PCPB36 startup info) / 0               4b1a2a1

               4b1a2a1a (meaning EMPTY)                         4b1a2a1a

            4b1a2a2 )s / 0 (meaning all EMPTY),                 4b1a2a2

         4b1a2b ABC (byte pointer to ASCIZ package name)s        4b1a2b

   4b1b RETURNS IN                                              4b1b

      4b1b1 1:  ABC (package handle)s                           4b1b1

4b2 CLSPK (VJSYS 6)
Closes previously opened remote packages.                       4b2

   4b2a ACCEPTS IN                                              4b2a

      4b2a1 1:  XWD process handle, ABC (package handle)s       4b2a1

   4b2b RETURNS IN                                              4b2b

      4b2b1 1:  ABC (cost in cents)s                            4b2b1

4c Procedures                                                   4c

   4c1 CALPE (VJSYS 7)
   Calls remote procedure.                                      4c1

      4c1a ACCEPTS IN                                           4c1a

         4c1a1 1:  XWD                                          4c1a1

            4c1a1a ABC (PCPB36 result list mask) / 0            4c1a1a

               4c1a1a1 (meaning LIST (INDEX [CALLER])),         4c1a1a1

            4c1a1b addr of Tenex-format procedure selector      4c1a1b
```

```
        4c1a2 2:  XWD                                                    4c1a2

          4c1a2a ABC (PCPB36 argument list mask) / 0                     4c1a2a

            4c1a2a1 (meaning LIST (INDEX [CALLER])),                     4c1a2a1

          4c1a2b ABC (ABC (PCPB36 argument) / 0 (meaning
          EMPTY))s / 0                                                   4c1a2b

            4c1a2b1 (meaning none)                                       4c1a2b1

        4c1a3 3:  priority                                               4c1a3

      4c1b RETURNS IN                                                    4c1b

        4c1b1 1:  XWD                                                    4c1b1

          4c1b1a outcome,                                                4c1b1a

          4c1b1b ABC (ABC (PCPB36 result) / 0 (meaning EMPTY))s
          / 0                                                            4c1b1b

            4c1b1b1 (meaning none)                                       4c1b1b1

        4c1b2 2:  cost in cents                                          4c1b2

    4c2 VISPR (VJSYS 10)
    Visits remote callee/caller.                                         4c2

      4c2a ACCEPTS IN                                                    4c2a

        4c2a1 1:  XWD                                                    4c2a1

          4c2a1a ABC (PCPB36 result list mask) / 0                       4c2a1a

            4c2a1a1 (meaning LIST (INDEX [CALLER])),                     4c2a1a1

          4c2a1b call handle                                             4c2a1b

        4c2a2 2:  XWD                                                    4c2a2

          4c2a2a ABC (PCPB36 argument list mask) / 0                     4c2a2a

            4c2a2a1 (meaning LIST (INDEX [CALLER])),                     4c2a2a1

          4c2a2b ABC (ABC (pCpB36 argument) / 0 (meaning
          EMPTY))s / 0                                                   4c2a2b

            4c2a2b1 (meaning none)                                       4c2a2b1
```

4c2b RETURNS IN 4c2b

4c2b1 1: XWD 4c2b1

4c2b1a outcome, 4c2b1a

4c2b1b ABC (ABC (PCPB36 result) / 0 (meaning EMPTY))s / 0 4c2b1b

4c2b1b1 (meaning none) 4c2b1b1

4c3 ALOCH (VJSYS 11)
Allocates call handle for remote procedure call. 4c3

4c3a ACCEPTS IN 4c3a

4c3a1 1: XWD 4c3a1

4c3a1a priority, 4c3a1a

4c3a1b addr of Tenex-format procedure selector 4c3a1b

4c3b RETURNS IN 4c3b

4c3b1 1: call handle 4c3b1

4c4 RELCH (VJSYS 12)
[Aborts remote callee and] releases call handle. 4c4

4c4a ACCEPTS IN 4c4a

4c4a1 1: call handle / 0 (meaning all) 4c4a1

4c4b RETURNS IN 4c4b

4c4b1 1: cost in cents 4c4b1

4c5 ACOPE (VJSYS 13)
Acquires control from remote callee/caller. 4c5

4c5a ACCEPTS IN 4c5a

4c5a1 1: call handle 4c5a1

4c5b RETURNS IN 4c5b

4c5b1 1: XWD 4c5b1

4c5b1a outcome, 4c5b1a

4c5b1b ABC (ABC (PCPB36 result) / 0 (meaning EMPTY))s / 0 4c5b1b

4c5b1b1 (meaning none) 4c5b1b1

4c6 RELPE (VJSYS 14)
Releases control to remote callee/caller. 4c6

4c6a ACCEPTS IN 4c6a

4c6a1 1: XWD 4c6a1

4c6a1a ABC (PCPB36 result list mask) / 0 4c6a1a

4c6a1a1 (meaning LIST (INDEX [CALLER])), 4c6a1a1

4c6a1b call handle 4c6a1b

4c6a2 2: XWD 4c6a2

4c6a2a ABC (PCPB36 argument list mask) / 0 4c6a2a

4c6a2a1 (meaning LIST (INDEX [CALLER])), 4c6a2a1

4c6a2b ABC (ABC (pCPB36 argument) / 0 (meaning EMPTY))s / 0 4c6a2b

4c6a2b1 (meaning none) 4c6a2b1

4c6a3 3: event handle 4c6a3

4c6a3a (signalled upon return of remote procedure with completion code = XWD call handle, outcome) 4c6a3a

4c7 INTPE (VJSYS 15)
Interrupts remote callee. 4c7

4c7a ACCEPTS IN 4c7a

4c7a1 1: call handle / 0 (meaning all) 4c7a1

4c8 RSMPE (VJSYS 16)
Resume previously interrupted remote callee. 4c8

4c8a ACCEPTS IN 4c8a

4c8a1 1: call handle / 0 (meaning all) 4c8a1

4c9 NTEPE (VJSYS 17)
Makes event known to remote caller. 4c9

4c9a ACCEPTS IN 4c9a

4c9a1 1: XWD 4c9a1

4c9a1a ABC (PCPB36 event description) / 0 4c9a1a

4c9a1a1 (meaning EMPTY), 4c9a1a1

4c9a1b event code 4c9a1b

4c10 HLPPE (VJSYS 20)
Solicits help from remote caller. 4c10

4c10a ACCEPTS IN 4c10a

4c10a1 1: XWD 4c10a1

4c10a1a ABC (PCPB36 problem description) / 0 4c10a1a

4c10a1a1 (meaning EMPTY), 4c10a1a1

4c10a1b problem code 4c10a1b

4c10b RETURNS 4c10b

4c10b1 1: ABC (PCPB36 solution) / 0 (meaning EMPTY) 4c10b1

4d Data Stores 4d

4d1 CRTDT (VJSYS 21)
Creates remote data store. 4d1

4d1a ACCEPTS IN 4d1a

4d1a1 1: XWD 4d1a1

4d1a1a scope, 4d1a1a

4d1a1b addr of Tenex-format data store selector 4d1a1b

4d1a2 2: ABC (PCPB36 initial value) / 0 4d1a2

4d1a2a (meaning EMPTY) 4d1a2a

4d2 DELDT (VJSYS 22)
Deletes previously created remote data store. 4d2

```
   4d2a ACCEPTS IN                                                    4d2a

      4d2a1 1:  addr of Tenex-format data store selector              4d2a1

4d3 RDDT  (VJSYS 23)
Reads remote data store.                                              4d3

   4d3a ACCEPTS IN                                                    4d3a

      4d3a1 1:  addr of Tenex-format data store selector              4d3a1

   4d3b RETURNS IN                                                    4d3b

      4d3b1 1:  ABC (PCPB36 value) / 0 (meaning EMPTY)                4d3b1

4d4 WRDT  (VJSYS 24)
Writes remote data store.                                             4d4

   4d4a ACCEPTS IN                                                    4d4a

      4d4a1 1:  XWD                                                   4d4a1

         4d4a1a ABC (PCPB36 value) / 0 (meaning EMPTY),               4d4a1a

         4d4a1b addr of Tenex-format data store selector              4d4a1b

4d5 LCKDT (VJSYS 25)
Locks remote data store.                                              4d5

   4d5a ACCEPTS IN                                                    4d5a

      4d5a1 1:  XWD                                                   4d5a1

         4d5a1a scope,                                                4d5a1a

         4d5a1b addr of Tenex-format data store selector              4d5a1b

      4d5a2 2:  XWD                                                   4d5a2

         4d5a2a flags,                                                4d5a2a

            4d5a2a1 B0 on:  abort if lock not settable
            immediately                                               4d5a2a1

         4d5a2b lock type                                             4d5a2b

   4d5b RETURNS IN                                                    4d5b

      4d5b1 1:  datalock handle                                       4d5b1
```

4d6 ULKDT (VJSYS 26)
Unlocks previously locked remote data store. 4d6

4d6a ACCEPTS IN 4d6a

4d6a1 1: XWD 4d6a1

4d6a1a datalock handle, 4d6a1a

4d6a1b addr of Tenex-format data store selector 4d6a1b

4e Channels 4e

4e1 CRTCH (VJSYS 27)
Creates channel between two remote processes. 4e1

4e1a ACCEPTS IN 4e1a

4e1a1 1: XWD process handle 1, process handle 2 4e1a1

4e1a2 2: scope 4e1a2

4e1b RETURNS IN 4e1b

4e1b1 1: channel handle 4e1b1

4e1b2 2: XWD port handle 1, port handle 2 4e1b2

4e2 DELCH (VJSYS 30)
Deletes previously created channel between two remote processes. 4e2

4e2a ACCEPTS IN 4e2a

4e2a1 1: channel handle / 0 (meaning all) 4e2a1

5 VJSYSs for manipulating local process 5

5a Subprocesses 5a

5a1 CRTSP (VJSYS 31)
Creates local subprocess. 5a1

5a1a ACCEPTS IN 5a1a

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

5a1a2 2: XWD 5a1a2

5a1a2a scope, 5a1a2a

5a1a2b ABC (PCPB36 startup info) / 0 5a1a2b

5a1a2b1 (meaning EMPTY) 5a1a2b1

5a1a3 3: priority 5a1a3

5a1b RETURNS IN 5a1b

5a1b1 1: subprocess handle 5a1b1

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

5a2a ACCEPTS IN 5a2a

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

5a2b RETURNS IN 5a2b

5a2b1 1: cost in cents 5a2b1

5b Processors 5b

5b1 CRTPR (VJSYS 33)
Creates local processor. 5b1

5b1a ACCEPTS IN 5b1a

5b1a1 1: XWD scope, subprocess handle 5b1a1

5b1a2 2: XWD 5b1a2

5b1a2a ABC (PCPB36 startup info) / 0 (meaning EMPTY), 5b1a2a

5b1a2b priority 5b1a2b

5b1b RETURNS IN 5b1b

5b1b1 1: processor handle 5b1b1

5b2 DELPR (VJSYS 34)
Deletes local processor. 5b2

5b2a ACCEPTS IN 5b2a

5b2a1 1: processor handle / 0 5b2a1

5b2a1a (meaning all within subprocess but leader) 5b2a1a

5b2b RETURNS IN 5b2b

5b2b1 1: cost in cents 5b2b1

5b3 SIPR (VJSYS 35)
Signs in local processor. 5b3

5b3a ACCEPTS IN 5b3a

5b3a1 1: byte pointer to ASCIZ process name 5b3a1

5b3a1a (ignored except from first process-leader processor) 5b3a1a

5b3a2 2: XWD 5b3a2

5b3a2a flags 5b3a2a

5b3a2a1 B0 on: auto processor creation 5b3a2a1

5b3a2a1a (CF to create/delete processors as required; ignored except from first process-leader processor) 5b3a2a1a

5b3a2a2 B1 on: sequential processor 5b3a2a2

5b3a2a2a (processor wil use the PGDPS operation as its VJUSR dispatch mechanism) 5b3a2a2a

5b3a2a3 B2 on: auto ready 5b3a2a3

5b3a2a3a (CF will simulate a call to RDYPR after SIPR and after each PTDPS; processor need never invoke RDYPR explicitly) 5b3a2a3a

5b3a2b ABC (byte pointer to ASCIZ package name)s / 0 5b3a2b

5b3a2b1 (meaning none; list index serves as an "internal package handle") 5b3a2b1

5b3a3 3: QWD 5b3a3

5b3a3a first page of subprocess-global storage 5b3a3a

5b3a3a1 (ignored except from subprocess leader), 5b3a3a1

5b3a3b last page of subprocess-global storage 5b3a3b

5b3a3b1 (ignored except from subprocess leader; first greater than last implies none), 5b3a3b1

5b3a3c 0, 5b3a3c

5b3a3d PSI channel / -1 (meaning none) 5b3a3d

5b3a3d1 (either for VJUSR request event, or, for sequential processors, to abort a procedure) 5b3a3d1

5b3b RETURNS IN 5b3b

5b3b1 1: XWD 5b3b1

5b3b1a ABC (PCPB36 [sub]process[or] startup info) / 0 5b3b1a

5b3b1a1 (meaning EMPTY), 5b3b1a1

5b3b1b event handle 5b3b1b

5b3b1b1 (signalled by CF to request a VJUSR with completion code = XWD usercall handle, VJUSR number), 5b3b1b1

5b3b2 2: flags 5b3b2

5b3b2a B0 on: local process is at root of tree 5b3b2a

5b3b2b B1 on: local subprocess is process leader 5b3b2b

5b3b2c B2 on: local processor is subprocess leader 5b3b2c

5b4 RDYPR (VJSYS 36)
Readys local processor for next service request (INIPK / TRMPK / PECAL / LRDDT / LWRDT). 5b4

5c channels 5c

5c1 SNDCH (VJSYS 37)
Outputs portion of PCPB36 data structure on local channel, 5c1

5c1a ACCEPTS IN 5c1a

5c1a1 1: XWD ABC (portion), port handle 5c1a1

5c2 RCVCH (VJSYS 40)
Inputs next portion of PCPB36 data structure from local channel, 5c2

5c2a ACCEPTS IN 5c2a

5c2a1 1: port handle 5c2a1

5c2b RETURNS IN 5c2b

5c2b1 1: ABC (portion) 5c2b1

5d Locks 5d

5d1 CRTLK (VJSYS 41)
Creates local lock. 5d1

5d1a ACCEPTS IN 5d1a

5d1a1 1: scope 5d1a1

5d1b RETURNS IN 5d1b

5d1b1 1: lock handle 5d1b1

5d2 DELLK (VJSYS 42)
Deletes local lock. 5d2

5d2a ACCEPTS IN 5d2a

5d2a1 1: lock handle / 0 (meaning all) 5d2a1

5d3 SETLK (VJSYS 43)
Sets local lock. 5d3

5d3a ACCEPTS IN 5d3a

5d3a1 1: XWD 5d3a1

5d3a1a scope (value ALL illegal), 5d3a1a

5d3a1b lock handle 5d3a1b

5d3a2 2: XWD 5d3a2

5d3a2a flags, 5d3a2a

5d3a2a1 B0 on: abort if lock not settable immediately 5d3a2a1

5d3a2b lock type 5d3a2b

5d3b RETURNS IN 5d3b

5d3b1 1: lockset handle 5d3b1

5d4 REMLK (VJSYS 44)
Unsets local lock. 5d4

5d4a ACCEPTS IN 5d4a

5d4a1 1: XWD 5d4a1

5d4a1a lock handle, 5d4a1a

5d4a1b lockset handle 5d4a1b

5e Events 5e

5e1 CRTEV (VJSYS 45)
Creates local event. 5e1

5e1a ACCEPTS IN 5e1a

5e1a1 1: XWD 5e1a1

5e1a1a scope (value ALL illegal), 5e1a1a

5e1a1b PSI channel to be interrupted when event signalled / -1 5e1a1b

5e1a1b1 (meaning none) 5e1a1b1

5e1a2 2: max length 5e1a2

5e1b RETURNS IN 5e1b

5e1b1 1: event handle 5e1b1

5e2 DELEV (VJSYS 46)
Deletes local event. 5e2

5e2a ACCEPTS IN 5e2a

5e2a1 1: event handle / 0 (meaning all) 5e2a1

5e3 SIGEV (VJSYS 47)
Signals a local event. 5e3

5e3a ACCEPTS IN 5e3a

5e3a1 1: event handle 5e3a1

5e3a2 2: completion code (non-zero) 5e3a2

5e4 TSTEV (VJSYS 50)
Tests for and clears signalled local event. 5e4

5e4a ACCEPTS IN 5e4a

5e4a1 1: event handle 5e4a1

5e4b RETURNS IN 5e4b

5e4b1 1: completion code / 0 (meaning unsignalled) 5e4b1

5e4b2 2: new length 5e4b2

5e5 WAIEV (VJSYS 51)
Waits for to be signalled and clears [any] one of a list of local events. 5e5

5e5a ACCEPTS IN 5e5a

5e5a1 1: ABC (event handle)s 5e5a1

5e5b RETURNS IN 5e5b

5e5b1 1: XWD 5e5b1

5e5b1a block offset to left-most signalled event handle, 5e5b1a

5e5b1b new total length 5e5b1b

5e5b2 2: completion code for left-most signalled event 5e5b2

6 VJUSRs implemented by (every processor in) every subprocess 6

6a Processors 6a

6a1 SOPR (VJUSR 1)
Signs out local processor. 6a1

6b Packages 6b

6b1 INIPK (VJUSR 2)
Initializes local package for subprocess. 6b1

6b1a ACCEPTS IN 6b1a

6b1a1 1: internal package handle 6b1a1

6b1b RETURNS IN 6b1b

6b1b1 1: package version number 6b1b1

6b2 TRMPK (VJUSR 3)
Terminates local package for subprocess. 6b2

6b2a ACCEPTS IN 6b2a

6b2a1 1: internal package handle 6b2a1

6c procedures 6c

6c1 PECAL (VJUSR 4)
Calls local procedure on behalf of remote caller. 6c1

6c1a ACCEPTS IN 6c1a

6c1a1 1: internal package handle 6c1a1

6c1a2 2: byte pointer to ASCIZ procedure name 6c1a2

6c1a3 3: XWD 6c1a3

6c1a3a call handle, 6c1a3a

6c1a3b ABC (6c1a3b

6c1a3b1 ABC (PCPB36 argument) / 0 (meaning EMPTY) 6c1a3b1

6c1a3b2)s / 0 (meaning none) 6c1a3b2

6c1b RETURNS IN 6c1b

6c1b1 1: XWD 6c1b1

6c1b1a outcome, 6c1b1a

6c1b1b ABC (6c1b1b

6c1b1c ABC (PCPB36 result) / 0 (meaning EMPTY) 6c1b1c

6c1b1d)s / 0 (meaning none) 6c1b1d

6c2 PEINT (VJUSR 5)
Interrupts previously called local procedure
on behalf of remote caller. 6c2

6c2a ACCEPTS IN 6c2a

6c2a1 1: call handle 6c2a1

6c3 PERSM (VJUSR 6)
Resumes previously interrupted local procedure
on behalf of remote caller. 6c3

6c3a ACCEPTS IN 6c3a

6c3a1 1: call handle 6c3a1

6c4 PEABR (VJUSR 7)
Aborts previously called local procedure
on behalf of remote caller. 6c4

6c4a ACCEPTS IN 6c4a

6c4a1 1: call handle 6c4a1

6c5 PENTE (VJUSR 10)
Makes ^Vent detected by remote callee known to local caller. 6c5

6c5a ACCEPTS IN 6c5a

6c5a1 1: XWD 6c5a1

6c5a1a call handle for local caller, 6c5a1a

6c5a1b call handle for remote callee 6c5a1b

6c5a2 2: XWD 6c5a2

6c5a2a ABC (PCPB36 event description) / 0 (meaning
EMPTY), 6c5a2a

6c5a2b event code 6c5a2b

6c6 PEHLP (VJUSR 11)
Solicits help from local caller on behalf of remote callee. 6c6

6c6a ACCEPTS IN 6c6a

6c6a1 1: XWD 6c6a1

6c6a1a call handle for local caller, 6c6a1a

6c6a1b call handle for remote callee 6c6a1b

6c6a2 2: XWD 6c6a2

6c6a2a ABC (PCPB36 problem description) / 0 (meaning EMPTY), 6c6a2a

6c6a2b problem code 6c6a2b

6c6b RETURNS IN 6c6b

6c6b1 1: ABC (PCPB36 solution) / 0 (meaning EMPTY) 6c6b1

6d Data Stores 6d

6d1 LVRDT (VJUSR 12)
Verifies existence of local data store. 6d1

6d1a ACCEPTS IN 6d1a

6d1a1 1: internal package handle 6d1a1

6d1a2 2: byte pointer to ASCIZ data store name 6d1a2

6d2 LRDDT (VJUSR 13)
Reads local data store on behalf of remote process. 6d2

6d2a ACCEPTS IN 6d2a

6d2a1 1: XWD 6d2a1

6d2a1a ABC (PCPB36 element selector) / 0 6d2a1a

6d2a1a1 (meaning whole data store), 6d2a1a1

6d2a1b internal package handle 6d2a1b

6d2a2 2: byte pointer to ASCIZ data store name 6d2a2

6d2b RETURNS IN 6d2b

6d2b1 1: ABC (PCPB36 value) / 0 (meaning EMPTY) 6d2b1

6d3 LWRDT (VJUSR 14)
Writes local data store on behalf of remote process. 6d3

6d3a ACCEPTS IN 6d3a

6d3a1 1: XWD 6d3a1

6d3a1a ABC (PCPB36 element selector) / 0 6d3a1a

6d3a1a1 (meaning whole data store), 6d3a1a1

6d3a1b internal package handle 6d3a1b

6d3a2 2: byte pointer to ASCIZ data store name 6d3a2

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

7 VJUSRs implemented by process leader 7

7a Processes 7a

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

7a1a ACCEPTS IN 7a1a

7a1a1 1: XWD 7a1a1

7a1a1a ABC (PCPB36 startup info) / 0 7a1a1a

7a1a1a1 (meaning EMPTY), 7a1a1a1

7a1a1b new process handle 7a1a1b

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

7a2a ACCEPTS IN 7a2a

7a2a1 1: old process handle 7a2a1

7a3 NTDPS (VJUSR 24)
Notes impending deletion of local process. 7a3

7b Packages 7b

7b1 OKOPK (VJUSR 17)
OKs opening of [and initializes] local package by remote process. 7b1

7b1a ACCEPTS IN 7b1a

7b1a1 1: XWD scope, new package handle 7b1a1

7b1a2 2: byte pointer to ASCIZ package name 7b1a2

7b1a3 3: XWD 7b1a3

7b1a3a internal package handle (meaning INIPK too) / 0, 7b1a3a

7b1a3b ABC (PCPB36 startup info) / 0 (meaning EMPTY) 7b1a3b

7b1b RETURNS IN 7b1b

7b1b1 1: package version number / 0 7b1b1

7b1b1a (if no internal package handle specified) 7b1b1a

7b2 OKCPK (VJUSR 20)
OKs closing of [and terminates] local package by remote process. 7b2

7b2a ACCEPTS IN 7b2a

7b2a1 1: XWD 7b2a1

7b2a1a internal package handle (meaning TRMPK too) / 0, 7b2a1a

7b2a1b old package handle 7b2a1b

7b2a2 2: byte pointer to ASCIZ package name 7b2a2

7c Channels 7c

7c1 OKCCH (VJUSR 21)
OKs creation of channel to local process. 7c1

7c1a ACCEPTS IN 7c1a

7c1a1 1: new port handle 7c1a1

7c2 OKDCH (VJUSR 22)
OKs deletion of previously created channel to local process. 7c2

7c2a ACCEPTS IN 7c2a

7c2a1 1: old port handle 7c2a1

7c3 NTLCH (VJUSR 23)
Notes loss of channel to remote process. 7c3

7c3a ACCEPTS IN 7c3a

7c3a1 1: XWD 7c3a1

7c3a1a flags, 7c3a1a

7c3a1a1 B0 on: process, rather than port handle 7c3a1a1

7c3a1b handle 7c3a1b

8 Data Type Assignments 8

8a Argument list mask LIST (INDEX [CALLER=1] / DSELECTOR*, ...) 8a

8b Code INDEX 8b

8b1 (event, problem, error) 8b1

8c Completion code INTEGER (non-zero) 8c

8d Cost INTEGER 8d

8e Data store selector 8e

8e1 LIST (%ph% INDEX, %pkh% INDEX, %data store% CHARSTR, %element% ESELECTOR*1) 8e1

8f Depth INTEGER 8f

8g Diagnostic CHARSTR 8g

8h Element selector 8h

8h1 LIST (<BOOLEAN [KEY=TRUE / INDEX=FALSE]> %element% any/INDEX, ...) 8h1

8i Handle INDEX 8i

8i1 (systemcall, usercall, process [SELF=1/SUPER=2], subprocess [SELF=1/LEADER=2], processor [SELF=1/LEADER=2], package, internal package, call, introduction, channel, port, lock, lockset, datalock, event) 8i1

8j Lock type INDEX [SHARE=1/EXCLUSIVE=2] 8j

8k Login parameter CHARSTR 8k

8k1 (user, password, account) 8k1

8l Name CHARSTR 8l

8l1 (process, package, data store) 8l1

8m Number INDEX 8m

8m1 (VJSYS, VJUSR) 8m1

8n Outcome INDEX [VISIT=1 / SUCCESS=2 / FAILURE=3] 8n

8o Priority INDEX 8o

8p Procedure selector 8p

8p1 LIST (%ph% INDEX, %pkh% INDEX, %pname% CHARSTR) 8p1

8q Process address CHARSTR 8q

8q1 <action> [<SP> <host address>] <SP> <intrahost address> 8q1

8q1a Action is either "CRT", meaning create a new process, or "SPL", meaning splice to an existing process. 8q1a

8q1b Host address is a decimal host addr or standard host name (defaulting to that of the local host). 8q1b

8q1c Intrahost address is a SAV filename on Tenex (for CRT), or a decimal ICP contact socket number (for SPL). 8q1c

8r Process information type INDEX [HOSTADDR=1] 8r

8s Result list mask LIST (INDEX [CALLER=1/DISCARD=2] / DSELECTOR*, ...) 8s

8t scope INDEX [PROCESSOR=1/SUBPROCESS=2/PROCESS=3/ALL=4] 8t

8u Startup info any 8u

8v Subprocess address CHARSTR 8v

8v1 <intrahost address> 8v1

9 PCPB36 Data Structure Format 9

9a Bit 0 If set, key data structure follows 9a

9b Bits 1-13 Unused (zero) 9b

9c Bits 14-17 Data type 9c

9c1 EMPTY =1 INTEGER=4 LIST=7 9c1

9c2 BOOLEAN=2 BITSTR =5 9c2

11 Appendix -- Change Summaries 11

11a Summary of 16-JUL-75 Changes (26100,) 11a

11a1 1) Operation PGDPS has been added to simplify VJUSR dispatch for "sequential" processors (see discussion) who so identify themselves via a new SIPR flag bit. 11a1

11a2 2) If B2 of AC 2 is raised in SIPR, PTDPS (and the SIPR itself) will thereafter be understood to imply RDYPR. 11a2

11a3 3) The calling sequences for OKOPK/OKCPK have been modified to allow calls to INIPK/TRMPK to be piggybacked on them for packages in the subprocess leader. 11a3

11a4 4) The calling sequence for CRTPS has been modified to allow a call to OPNPK to be piggybacked on it, with a savings of two inter-process messages. 11a4

11a5 5) The event provided in IVDPS is signalled with the following completion code (formerly 1): 11a5

11a5a XWD systemcall1 handle, VJSYS number 11a5a

11a6 6) The event provided in RELPE is signalled with the following completion code (formerly 1): 11a6

11a6a XWD call handle, outcome 11a6a

11a7 7) The following entities are converted to upper-case by DPS whenever accepted from the user (in one process), and therefore will appear in upper-case if ever presented to the user (in another process): 11a7

11a7a process addresses and names; package, procedure, and data store names; and elements of user information (i.e. user, password, and account). 11a7a

11a8 8) Subprocesses with no packages may present a zero, rather than the address of a zero-length list of package names, in the RH of AC 2 in SIPR. 11a8

11a9 9) The "all" option has been deleted from CLSPK. 11a9

11a10 10) GTDPS returns the VJUSR number in the LH of AC 0. 11a10

11a11 11) Clarification: To indicate to SIPR that no (zero) address space pages are to be shared by processors within the subprocess, make the "first" page greater than the "last".

Setting both to zero implies that one page (namely, page 0) is to be shared. 11a11

11a12 12) Clarification: Whenever a byte pointer contains -1 as its left half, the usual Tenex default (namely 440700) is assumed. 11a12

11a13 13) Clarification: Whenever an address is presented to DPS in a full word, the LH is ignored and may (as far as DPS is concerned) contain trash. 11a13

11a14 14) Clarification: If in doubt, set scope to ALL=4, priorities to 1, and version numbers (INDEXs) to 1. 11a14

11b Summary of 10-JUN-75 Changes 11b

11b1 1) Claimed change to PECAL (see 30-MAY # 11), ommitted by mistake, actually made. 11b1

11c Summary of 30-MAY-75 Changes 11c

11c1 1) pF requests CF services via JSYS 400, rather than JSYS. Sorry for the flipflop, but note that it's a one-instruction change. 11c1

11c2 2) The length (L) field of all ABF's should be zero when supplied by the programmer. 11c2

11c3 3) In GTDPS, the usercall handle and ABF are placed in the ACs, rather than in an ABC. 11c3

11c4 4) INFPS (VJSYS 52) has been added for use by the NVT package in locating a process within the network. 11c4

11c5 5) The priority argument has been moved from VISPE and RELPE to ALOCH. 11c5

11c6 6) The event handle argument has been moved from ALOCH to RELPE. 11c6

11c7 7) INTPE and RSMPE accept zero as a call handle, meaning all. 11c7

11c8 8) The call handle argument has been deleted from NTEPE and HLPPE, being redundant (i.e. supplied in IVDPS). 11c8

11c9 9) Abort provisions have been eliminated from SNDCH and RCVCH. 11c9

11c10 10) CRTEV requires an additional "length" argument specifying the number of completions simultaneously storable within the event. TSTEV and WAIEV return the number of completions which remain stored in the specified event(s) after the operation. 11c10

11c11 11) Wherever DPS communicates an ASCIZ string to the PF, whether as a VJSYS result or as a VJUSR argument, it supplies a byte pointer to the ASCIZ string, rather than an ABC. Specifically, this change is made to the calling sequences of PECAL (procedure name); LVRDT, LRDDT, LWRDT (data store name); and OKOPK, OKCPK (package name). 11c11

11c12 12) The subprocess handle result has been deleted from OKOPK. 11c12

11c13 13) NTDPS (VJUSR 24) has been added to inform the process leader of the process' termination, prior to initiation of signout. 11c13

11c14 14) Process addresses must contain one of the following verbs: "CRT", meaning create a new process; or "SPL", meaning splice to an existing process. 11c14

11c15 15) The Tenex format for a data store selector contains ABC (PCPB36 element selector), rather than "zero or more ABC (element of PCPB36 element selector)". 11c15

11d Summary of 27-APR-75 Changes 11d

11d1 1) PF requests CF services via HALTF, rather than JSYS. 11d1

11d2 2) PF provides a single block of storage for use by DPS in returning parameters to the PF, rather than a separate block for each VJSYS result / VJUSR argument. If insufficient storage is provided, the operation is aborted and the supplied block's L set to the size required; a second call to RRDPS (providing increased storage) may be employed to recover from the error. 11d2

11d3 3) CALPE, INTPE replaced by CALPE, VISPE, ALOCH, RELCH, ACQPE, RELPE, INTPE, RSMPE, NTEPE, HLPPE. 11d3

11d3a - CALPE is basically shorthand for the sequence ALOCH - VISPE - RELCH. 11d3a

11d3b - VISPE is basically shorthand for the sequence RELPE - ACQPE. 11d3b

11d3c - Help (HLPPE) and note (NTEPE) returns have been factored out into separate VJSYSs. 11d3c

11d3d - Return type and subtype have been combined into a single "outcome", except that abort returns are represented as VJSYS / VJUSR failures (i.e. return +1). 11d3d

11d3e - The EVH supplied to ALOCH designates an event to be signalled whenever the remote caller/callee returns to the local procedure and therefore acts as a cue to issue ACQPE. 11d3e

11d3f - INTPE suspends the remote caller and must be followed by RSMPE/RELCH. 11d3f

11d3g - LCAPE, LINPE replaced by PECAL, PEINT, PERSM, PEABR, PENTE, PEHLP. 11d3g

11d4 4) LVRDT VJUSR added to verify existence of local data store. 11d4

11d5 5) SOPR changed from a VJSYS to a VJUSR; signout event eliminated from SIPR. 11d5

11d6 6) CALPE, VISPE, RELPE, CRTSP, CRTPR allow priority. 11d6

11d7 7) LCKDT, SETLK allow abort if lock can't be set immediately. 11d7

11d8 8) CRTLK, CRTPS, ITDPS, CRTCH require scope. 11d8

11d9 9) REMLK requires lock handle. 11d9

11d10 10) ULKDT requires data store selector, rather than process handle. 11d10

11d11 11) SIPR allows automatic processor creation. 11d11

11d12 12) WAIEV returns number of signalled events. 11d12

11d13 13) OPNPK returns package handles (omitted by accident). 11d13

11d14 14) VJSYSs, VJUSRs renumbered. 11d14

DPSJSYS / DPS-10 Programmer's Guide

(J26170) 22-JUL-75 15:40;;;; Title: Author(s): James E. (Jim) White/JEW; Distribution: /SRI-ARC([INFO-ONLY]) ; Sub-Collections: SRI-ARC; Clerk: JEW; Origin: < JWHITE, DPSJSYS,NLS;2, >, 22-JUL-75 15:27 JEW ;;;;####;

1 26170 Distribution
1a 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,
1b Mary Ann Kellan, Buddie J. Pine, Andy Poggio, David L. Retz, Laura J. Metzger, Karolyn J. Martin, Jan A. Cornish, Larry L. Garlick, Priscilla A. Wold, Pamela K. Allen, Delorse M. Brooks, Beverly Boli, Rita Hysmith, Log Augmentation, Joseph L. Ehardt, Raymond R. Panko, Susan Gail Roetter, Robert Louis Belleville, Rene C. Ochoa, Ann Weinberg, Joan Hamilton, Adrian C. McGinnis, Robert S. Ratner, David S. Maynard, Robert N. Lieberman, Sandy L. Johnson, James H. Bair, Jeanne M. Leavitt, Rodney A. Bondurant, Jeanne M. Beck, Marcia L. Keeney, Elizabeth K. Michael, Jonathan B. Postel, Elizabeth J. Feinler, Kirk E. Kelley, N. Dean Meyer, James E. (Jim) White

1 26170 Distribution
1a 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,
1b Mary Ann Kellan, Buddie J. Pine, Andy Poggio, David L. Retz, Laura J. Metzger, Karolyn J. Martin, Jan A. Cornish, Larry L. Garlick, Priscilla A. Wold, Pamela K. Allen, Delorse M. Brooks, Beverly Boli, Rita Hysmith, Log Augmentation, Joseph L. Ehardt, Raymond R. Panko, Susan Gail Roetter, Robert Louis Belleville, Rene C. Ochoa, Ann Weinberg, Joan Hamilton, Adrian C. McGinnis, Robert S. Ratner, David S. Maynard, Robert N. Lieberman, Sandy L. Johnson, James H. Bair, Jeanne M. Leavitt, Rodney A. Bondurant, Jeanne M. Beck, Marcia L. Keeney, Elizabeth K. Michael, Jonathan B. Postel, Elizabeth J. Feinler, Kirk E. Kelley, N. Dean Meyer, James E. (Jim) White

KIRK 22-JUL-75 16:44 26171

Journalized help transfer procedures

Files in xhelp should have default directory for links: xhelp. Files in help should have defalt directory for links: help. When 8,5 is brought up, we should put a command branch in <xhelp>helpd.nls that 1) deletes directives 2) sets the default directory for links 3) updates to a new name in help and 4) ftp's a copy to isic if that's possible

KIRK 22-JUL-75 16:44 26171

Journalized help transfer procedures

1 All the files in <help> directory have been moved to <xhelp> so that they can be de-bugged while 8.5 is still experimental. 1

2 When 8.5 is brought up as the running system, all output processor directives should be deleted from a tool description file and copied over to the <help> directory if it is a tool in 8.5. NSW tool description files should stay in xhelp until NSW is the running system. 2

3 After 8.5 is brought up, no changes should be made in the help directory. All changes should be made in xhelp and then a new version copied to help. 3

4 No help files should be updated in the xhelp directory at isic. When 8.5 is brought up at isic, all the files in the help dir there should be copied from xhelp at bbnb as outlined above for bbnb. 4

5 Sorry about any inconvenience this may cause. Let me know of any problems with these new procedures. 5

KIRK 22-JUL-75 16:44 26171

Journalized help transfer procedures

(J26171) 22-JUL-75 16:44;;;; Title: Author(s): Kirk E. Kelley/KIRK;
Distribution: /DMB([ACTION] dirt) DIRT([INFO-ONLY]) ;
Sub-Collections: SRI-ARC DIRT; Clerk: KIRK;

1 26171 Distribution

1a Delorse M. Brooks, Jonathan B. Postel, Priscilla A. Wold, Rita Hysmith, Pamela K. Allen, Delorse M. Brooks, Elizabeth F. Finney, Beverly Boli, Lawrence A. Crain, Kirk Sattley, Susan Gail Roetter, Robert N. Lieberman, Ann Weinberg, Kenneth E. (Ken) Victor, Douglas C. Engelbart, James H. Bair, Elizabeth K. Michael, Richard W. Watson, Elizabeth J. Feinler, Harvey G. Lehtman, Kirk E. Kelley, Laura E. Gould, Jeanne M. Beck, Dirk H. Van Nouhuys, James C. Norton,

Fe Meeting Report

1 This note summarizes a series of meetings about the NSW Front End (FE) implementation. 1

2 The attendees were : rww dir jle chi jew llg jbp andy dia 2

3 The main issues is space: the pieces of code and data that need to be present in the FE wont fit together in the address space of the pdp11. 3

4 Our current estimates of the sizes of the various code and data modules are given in the file <ehardt, fe-memory-consumption,>. 4

5 The following issues and tasks have evolved over the series of meetings, under each issue or task is the current status or resolution. 5

5a (1) CHI - reduce to size of the CLI context: goal 8K 5a

5a1 8K is ok 5a1

5b (2) CHI JEW LLG - combine DPS and CLI into a single process 5b

5b1 case 1 - B4700 code also works with DPS 5b1

5b2 case 2 - only CLI uses DPS 5b2

5b2a This appears not to be promising, rather a design utilizing a shared page between DPS and CLI containing both code and data, and using signals or EMTs to communicate events to each other looks more interesting. (see--number 11) 5b2a

5c (3) DIA - fix L1011 compiler to parse large segments of code to generate more optimal code (this version of the compiler will not run under nls but from the exec) 5c

5c1 This looks to be more work that previously discussed, but is still being explored. (See--number 18) 5c1

5d (4) JEW - reduce the size of the DPS context: goal 2K 5d

5d1 2K appears to be fine. 5d1

5e (5) JLE DLR - investigate the buffer and memory magement in ELF, investigate buffer and code optimization in exec and telnet. 5e

5e1 Suggested that demand paging system would be a win, but this appears to be far more work than we want to take on. 5e1

5e2 For system with no normal TELNET, only special TELNET for CLI old tool access the TELNET code could be reduced from 1000 to 600 words for a savings of 400 words. In such a system if the CLI were to play the role of the EXEC the 3000 word EXEC would be eliminated. 5e2

5f (6) JLE - talk to DEC about prices for disks, memory etc, for 11/40, 11/45, and 11/70. 5f

5f1 Price List Obtained 5f1

5g (7) CHI - figure out the size of the code for CLI mannaged context switching and for display featurets 5g

5g1 Display Code on 10 takes 4K, so estimate 6-8K on 11. 5g1

5g2 Context switch code should be small a few hundred words at the most. 5g2

5h (8) LLG - investigate the ELF facilities that can be used to reduce the size of the DPS code 5h

5h1 There is not very much to be saved here, but the ELF primitives will be used where appropriate. An estimate of 20-24K for the DPS code is reasonable. 5h1

5h2 This goes with (2) above. 5h2

5i (9) DLR JEW CHI - investigate eliminating ELF. 5i

5i1 There is not much to be saved, but it would be useful to list the primitives along with the amount of space that would be saved if the primitive were deleted and the programs that would be impacted if the primitive were deleted. 5i1

5j (10) JLE DLR - Evaluate the relative merits of ELF, UNIX and RSX11 for our needs. 5j

5j1 prepare a note for our internal use that compares these systems and our requirements especially noting their facilities relating to: 5j1

5j1a real and virtual address space management 5j1a

5j1b disk management 5j1b

5j1c process structure 5j1c

5j1d interprocess communication 5j1d

5k (11) LLG - DPS-CLI communication 5k

5k1 prepare a note on the design for the DPS - CLI communication considering both the shared page and message mapping techniques, incorporating comments by CHI and JEW. 5k1

51 (12) JLE - Prepare order for disk 51

511 ??? 511

5m (13) RWW - Send note stating our general problem to nsw steering committee. 5m

5m1 A note was sent (see -- 26099,). 5m1

5n (14)CHI JEW LLG - Review of the proposed DPS/CLI interface design with attention to the single vs, multi- process alternatives. 5n

5n1 Done (see-- number 11) 5n1

5o (15) DLR JLE - Investigate alternative memory management schemes 5o

5o1 Rough estimate is that 4 man months would be required to revise ELF to use variable size real pages, and that this would be desirable for other reasons (eg input output management). To revise ELF to allow variable size virtual pages might be accomplished with 1 man month of effort. 5o1

5o1a [to be scheduled] note on implications and effort involved if this task is undertaken. 5o1a

5p (16) JLE - Continue developing the position paper for the steering committee, deliver a draf version to COMPASS. 5p

5q (17) CHI - Breakdown of CLI code. 5q

5q1 Done (see--26144) 5q1

5r (18) DJA - L1011 status report: what remains to be done, current expectations of savings. 5r

5r1 Done (see--andrews,l1011status,) 5r1

5s (19) [to be scheduled] Study the relative efficiency of L1011 vs assembly code. 5s

5t (20) LLG DLR - Storage Management 5t

5t1 prepare a note on storage management techniques, there must be a choice between the following: 5t1

5t1a current ELF primitives 5t1a

5t1b L10 runtime 5t1b

5t1c new (or modified) routines 5t1c

5t2 if the l10 routines were chosen and compiled by l1011 the savings to be made by hand coding are estimated to be 20% by CHI and 50% by JLE 5t2

5u (21) JLE - B4700 code 5u

5u1 Estimate received from Triolo that the "null-IP" will take 7000-8000 words. 5u1

5v (22) ??? - Coordinate KEV's debugger with PP-11 testing. 5v

6 It was also resolved at the second meeting that we should proceed to construct a test version of the system even tho it may only support one or two users. 6

6a This requires several peices to be operational: 6a

6a1 VM ELF - DLR 6a1

6a2 Loader - ANDY 6a2

6a3 Flea DDT - DLR 6a3

6a4 CLI-11 - JLE & ANDY 6a4

6a5 DPS-11 - LLG 6a5

7 The next meeting is scheduled for 10 am Tuesday 29th July. 7

Fe Meeting Report

(J26172) 22-JUL-75 18:04;;;; Title: Author(s): Jonathan B. Postel/JBP; Distribution: /RWW([INFO-ONLY]) DLR([INFO-ONLY]) JLE([INFO-ONLY]) CHI([INFO-ONLY]) JEW([INFO-ONLY]) LLG([INFO-ONLY]) JBP([INFO-ONLY]) ANDY([INFO-ONLY]) DIA([INFO-ONLY]) ; Sub-Collections: SRI-ARC; Clerk: JBP; Origin: < POSTEL, FE-PROBLEMS,NLS;8, >, 22-JUL-75 17:59 JBP ;;;;####;

1 26172 Distribution
1a Richard W. Watson, David L. Retz, Joseph L. Ehardt, Charles H. Irby, James E. (Jim) White, Larry L. Garlick, Jonathan B. Postel, Andy Poggio, Don I. Andrews,

NSW Protocols Weekly Status Report: 22-JUL-75

1 NSW Protocols Weekly Status Report: 22-JUL-75 1

1a JIM WHITE 1a

1a1 Major Responsibility: DPS-10 1a1

1a2 Accomplished Last Week 1a2

1a2a - Continued debugging CLI/DPS/NLS system with DSM/CHI. 1a2a

1a2b - Reviewed LLG's DPS-11 user interface spec. 1a2b

1a2c - Moved base of operations to ISIC, now that the JSYS trap feature is operational there. 1a2c

1a2d - Reviewed report of Message Transmission Protocol Subcommittee at request of RWW. 1a2d

1a2e - Documented all errors currently detected and reported by DPS-10, giving the error number and diagnostic associated with each (Journal and [ISIC]<NSW-SOURCES>DPSERRS.TXT). 1a2e

1a2f - Designed several new DPS features: 1a2f

1a2f1 > a signout VJSYS which will enable the FE to return the user to the EXEC in response to QUIT (filling a hole in the DPS-10 design). 1a2f1

1a2f2 > two new VJSYSs by which processor-created forks can be granted access to DPS primitives (for indirect use, initially, by the encapsulator). 1a2f2

1a2f3 > a signin option by which a subprocess can encapsulate itself (with respect to its use of the file system). 1a2f3

1a2f4 > a new DPS operation which returns the diagnostic message associated with a specified DPS error number. 1a2f4

1a2f5 > a new VJSYS by which a procedure can return parameters to its remote caller/callee without relinquishing control (for use where NOTE is unsatisfactory). 1a2f5

1a2f6 > a new VJSYS by which a group of remote procedures can be executed in series, with a single pair of inter-process messages. 1a2f6

1a2g - No debugging of CLI/DPS/WM system with Stu Schaffner

this past week; understand from CHI that Schaffner's busy checking out DPs HELP, etc. features in the shared-page mode of interaction with the FE. 1a2g

1a3 Scheduled Next Week 1a3

1a3a - Continue debugging CLI/DPS/NLS system with DSM/CHI. 1a3a

1a3b - Continue debugging CLI/DPS/WM system with Stu Schaffner, who will be exercising ITDPS and the data store primitives for the first time (carry over from last week). 1a3b

1a3c - Review second draft of LLG's DPS-11 user interface spec. 1a3c

1a3d - Draft strawman critique of report of Message Transmission Protocol Subcommittee at request of RWW. 1a3d

1a3e - Review Journal-related final report section at request of RWW. 1a3e

1a3f - Begin implementing newly-designed DPS features. 1a3f

1a4 Queued 1a4

1a4a - Code inter-host inter-process communication (this is not critical path). 1a4a

1b LARRY GARLICK 1b

1b1 Major Responsibility: DPS-11 1b1

1b2 Accomplished Last Week 1b2

1b2a - Completed second draft of DPS-11 user interface document. Expect one more pass to add an alternate message passing scheme -- message mapping. 1b2a

1b2b - DLR and I are evaluating storage management requirements using ELF primitives rather than L10 storage management. Will prepare a note for the 29-JUL meeting. 1b2b

1b2c - Coding DPS-11 event and manager management. 1b2c

1b3 Scheduled for Short Term 1b3

1b3a - Prepare final draft of DPS-11 user interface document. 1b3a

1b3b - Continue DPS-10 => DPS-11 conversion: 1b3b

1b3b1 > Install new VJSYS/VJUSR call mechanisms. 1b3b1

1b3b2 > Remove subprocess and processor management. 1b3b2

1b3b3 > Code data format conversion routines for PCPB8. 1b3b3

1b3c - Shooting for a test of DPS-11 with CLI-11 by 1-SEP. 1b3c

1b4 Scheduled for Long Term 1b4

1b4a - Install DPS-11. 1b4a

NSW Protocols Weekly Status Report: 22-JUL-75

(J26173) 22-JUL-75 18:33;;;; Title: Author(s): James E. (Jim) White/JEW; Distribution: /SRI-ARC([INFO-ONLY]) ; Sub-Collections: SRI-ARC; Clerk: JEW; Origin: < JWHITE, PROSTS.NLS;6, >, 22-JUL-75 18:31 JEW ;;;;####;

1 26173 Distribution
1a 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,
1b Mary Ann Kellan, Buddie J. Pine, Andy Poggio, David L. Retz, Laura J. Metzger, Karolyn J. Martin, Jan A. Cornish, Larry L. Garlick, Priscilla A. Wold, Pamela K. Allen, Delorse M. Brooks, Beverly Boli, Rita Hysmith, Log Augmentation, Joseph L. Ehardt, Raymond R. Panko, Susan Gail Roetter, Robert Louis Belleville, Rene C. Ochoa, Ann Weinberg, Joan Hamilton, Adrian C. McGinnis, Robert S. Ratner, David S. Maynard, Robert N. Lieberman, Sandy L. Johnson, James H. Bair, Jeanne M. Leavitt, Rodney A. Bondurant, Jeanne M. Beck, Marcia L. Keeney, Elizabeth K. Michael, Jonathan B. Postel, Elizabeth J. Feinler, Kirk E. Kelley, N. Dean Meyer, James E. (Jim) White

JBp 22-JUL-75 21:18 26174

1 RLL has brought to my attention the fact that a "short course" on NLS is being planned for late August, i see this as a worthy undertaking and wish it to be maximally successfull, however i an concerned that we perhaps do not have the resources (computer cycles, terminals, meeting rooms, etc.) to devote to the task with out generating serious conflicts within ARC. --jon. 1

(J26174) 22-JUL-75 21:18;;;; Title: Author(s): Jonathan B.
Postel/JBP; Distribution: /DCE([INFO-ONLY]) JCN([INFO-ONLY])
RWW([INFO-ONLY]) RLL([INFO-ONLY]) JHB([INFO-ONLY]) ;
Sub-Collections: SRI-ARC; Clerk: JBP;

1 26174 Distribution
1a Douglas C. Engelbart, James C. Norton, Richard W. Watson, Robert N. Lieberman, James H. Bair,

Quarterly Report Input

1 Quarterly Report for period ending 17 July 75 1

1a File Package 1a

1a1 The set procedures to be implemented in each host to facilitate the movement of files between workspaces in either the same computer or different computers is called the "File Package". 1a1

1a2 The File Package specification was greatly rewritten and enlarged to include specifics on file types and scenarios for file movement. 1a2

1b Protocol Format 1b

1b1 The communication between the Distributed Programming System modules in each host is in accordance with either a 36 bit format or a 8 bit format. The 8 bit format, called "pCpB8", was documented. 1b1

1c Protocol Meeting 1c

1c1 SRI hosted a meeting to discuss the protocols needed in the NSW. The major topic of the meeting was the Distributed Programming System designed by SRI to meet the needs of NSW and perhaps other large multi-computer programming systems. Also discussed were specifics of the communication protocol format (PCPB8) and the file moving procedures (the File Package). 1c1

JBP 22-JUL-75 22:25 26175

Quarterly Report Input

(J26175) 22-JUL-75 22:25;;;; Title: Author(s): Jonathan B. Postel/JBP; Distribution: /BEV([INFO-ONLY]) ; Sub-Collections: SRI-ARC; Clerk: JBP;

1 26175 Distribution
1a Beverly Boli,

L1011 Code Improvement Status

1 Current status of L1011 improvement effort. 1

1a NOTE: Since memory on the NSW frontends is precious, it appears worthwhile to spend some time shrinking the amount of code produced by L1011. This is a survey of methods and status of improvements made so far. 1a

1b These are the code improvement areas: 1b

1b1 register allocation: 1b1

1b1a This involves modifying every rule that uses/requires/releases a register (i.e. produces code that uses a register). This improvement is necessary because the current algorithm is not only inefficient but full of holes. (Some have been stuffed with thumbs, some not). A large coding and debugging effort (compared to following improvements), but will result in better and less code produced by compiler. 1b1a

1b2 BR to a JMP: 1b2

1b2a The branch forward relative instruction is a bugaboo for L1011 since it is a one pass compiler. Also, it is the only way a test of the condition code can be made. The obvious thing to do is put out a BR (of appropriate condition testing) around a JMP to the desired place (result= 3 words where 1 might be enough). BR to a JMP means that the compiler remembers where it produced a JMP forward and, if possible, when it needs to produce a BR to the same label, it produces a BR to the previouslyproduced JMP instruction. 1b2a

1b3 Shrink INVOKE: 1b3

1b3a Special cases of the INVOKE command and perhaps some LIST procedure calls can be made shorter. (Primarily here because DPS does many INVOKEs). 1b3a

1b4 Short JMP fwd: 1b4

1b4a Another way to reduce branch forward code: Where it is known that a forward jump is going over a single simple statement, demand that that statement fit within 256 words and do a branch relative (by fixing it up later). 1b4a

1b5 Destination Scheme: 1b5

1b5a Another way to reduce jumps: Each control flow

statement (e.g. WHILE, CASE, LOOP), knows where the exit and looping locations are, only with respect to the start and end of the control flow statement itself. Code could be improved if each statement-compiling rule knew the 'destination' of that statement -- i.e. what was going to happen when control exits that statement, such as branching back to a loop, going to next statement, or going around an ELSE part, etc. This improvement will reduce the number of jmps produced and replace some forward jumps with backward jumps. 1b5a

1b5b NOTE: The destination cannot be passed inside the last statement of a block since the compiler does not know it was the last until AFTER it is compiled. See large parse tree plan below. 1b5b

1b6 Record defs/Field refs: 1b6

1b6a This involves making the compiler smart about records that are declared in the file that is being compiled. Fields that are same pattern of bits in different records can be implemented as the same field reference code to save space. Also, fields that are words, bytes or bits can be referenced by fewer instructions than the current 'blind' field reference technique. 1b6a

1b7 Parse large tree: 1b7

1b7a The compiler currently parses one statement and then produces code for it. The following things could be done if large portions of procedures were parsed, then compiled: 1b7a

1b7a1 Better register allocation (by counting refs and keeping often used things in registers). 1b7a1

1b7a2 Counting instructions that a branch represents so that a branch forward can be done where possible. 1b7a2

1b7a3 Better destination scheme by passing the destination inside the last statement of a block. 1b7a3

1c As of 7/22 AM, this is the status of the L1011 improvements: 1c

1c1 register allocation: implemented, partly debugged. 1c1

1c2 record definition/field reference: implemented, partly debugged. 1c2

1c3 BR to a JMP: implemented, partly debugged. 1c3

1d Estimate of time to implement (man days) and very rough estimate of typical average code reduction percentages. 1d

	impl	debug*	% reduction		
1d1 improvement					1d1
1d2 reg allocation:		done	3	1-3%	1d2
1d3 shrink invoke:		.5	.5	1-5% (depends on use)	1d3
1d4 BR to a JMP:		done	.5	1-2%	1d4
1d5 short JMP fwd:		1	1.5	1-4%	1d5
1d6 dest scheme:	.5	1	1-3%		1d6
1d7 records/fields:		done	1	1-3% (depends on use)	1d7
1d8 large parse:	3	3	- (not recommended)		1d8
1d9 and reg alloc: (not recommended)		1	2	.5-2%	1d9
1d10 and fwd bran: (not recommended)		?	?	.5-2%	1d10
1d11 and dest plan: (not recommended)		1	1	.5-2%	1d11
1d12 SUM (recommended)		2	7.5	6-20%	1d12

1d12a *initial test-case debugging before compiler is available for general use. 1d12a

1e We will proceed with the above items (but excluding those that are not recommended). 1e

L1011 Code Improvement Status

(J26176) 22-JUL-75 11:30;;;; Title: Author(s): Don I. Andrews/DIA;
Distribution: /RWW([ACTION]) JBP([ACTION]) JEW([ACTION])
DIA([ACTION]) CHI([ACTION]) LLG([ACTION]) ANDY([ACTION])
JLE([ACTION]) KEV([ACTION]) ; Sub-Collections: SRI-ARC; Clerk:
DIA; Origin: < ANDREWS, L1011STATUS.NLS;4, >, 22-JUL-75 11:26
DIA ;;;;####;

1 26176 Distribution
1a Richard W. Watson, Jonathan B. Postel, James E. (Jim) White, Don I. Andrews, Charles H. Irby, Larry L. Garlick, Andy Poggio, Joseph L. Ehardt, Kenneth E. (Ken) Victor,

RLL 23-JUL-75 14:56 26177

Response to KWAC Fall 75 meeting questionnaire

1 FROM: Robert Lieberman (RLL) 1

2 Will you be able to attend? 2

2a The Friday 12 Sept 75 is personnally bad and thus I would have to leave Thursday in order to be home for Friday AM. This leaves me only three days for that week. 2a

2b Monday 15 September is a Holiday for me and I would not be able to travel until Tuesday, again leaving only three days for the week. Since I don't know of anyone else effected by Yom Kippur, this might not be too bad for most. 2b

2c If I had to choose between the weeks, I would take the week beginning the 15th. 2c

2c1 I could possibly leave Monday night; arriving in Boston Tuesday AM. 2c1

2d Week beginning: 2d

2d1 1 Sept: Labor day is the Monday; Saturday is a holiday (Rosh Hashanah) 2d1

2d2 8 Sept: Friday personnally bad; would miss Thursday also for flying. 2d2

2d3 15 Sept: Monday is holiday, otherwise OK 2d3

2d4 22 sept: OK 2d4

3 Would you like to have the meeting in Cambridge? 3

3a yes 3a

4 Do you prefer a 4 or 5 day meeting? 4

4a sightseeing can be done on weekends, but a mid week break is not too bad. Would think night sessions would be neat if we had Wednesday off. 4a

5 What is you choice of dates? 5

5a NOT the week of 8 Sept. 5a

5b FAIR for week of 15 Sept. 5b

5c OK for rest of month and October. 5c

Response to KWAC Fall 75 meeting questionnaire

6 Would you like me to arrange your hotel/motel room? 6

6a Probably SRI-ARC contingent will book its rooms as a block via SRI. 6a

7 Would you like a list of hotels and prices in the Boston-MIT area? 7

7a Yes, but only the most convenient. 7a

RLL 23-JUL-75 14:56 26177

Response to KWAC Fall 75 meeting questionnaire

(J26177) 23-JUL-75 14:56;;;; Title: Author(s): Robert N. Lieberman/RLL; Distribution: /RMS2([ACTION]) JCN([INFO-ONLY]) ; Sub-Collections: SRI-ARC; Clerk: RLL; Origin: < LIEBERMAN, KWAC75.NLS;2, >, 23-JUL-75 14:49 RLL ;;;;#####;

1 26177 Distribution
1a Robert M. Sheppard, James C. Norton,

RLL 12-AUG-75 15:13 26178
14 AUG 75 7:36PM 26178

This is covering letter to OSIS (NSF) for the standard Utility proposal, A copy of the OSHA proposal was enclosed. We expect to draft a NSF proposal in Septemeber.

RLL 12-AUG-75 15:13 26178
14 AUG 75 7:36PM 26178

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

Dr. Harold Bamford
National Science Foundation
Office of Scientific Information Services
Washington, DC 20550

Dear Dr. Bamford:

I felt that it would be best to send you a copy of a recent proposal instead drafting one especially for you, since it is not yet known what our standard proposal will be for January 1976. 1

Changes in our pricing of the Workshop Utility Services are likely to depend upon new hardware configurations and the availability of a new "pie slice scheduler." 2

The new scheduler makes it possible to dynamically allocate computer cycles to groups on a percentage basis. Thus, one user group can be guaranteed a minimun of X percent of the computer system (based on CPU cycles) no matter how many other users are logged on the computer. 2a

This will now allow more than one person at time to be logged on in a 'slot' (currently 3% of the available CPU cycles) if they are willing to share the fixed percentage of the machine with other users in their group. 2b

We do not believe, however, that these changes will account for much of a difference in the per share cost we are currently charging. 3

TERMINALS 4

I have also enclosed a report on various terminals that can be used in conjunction with our system. It is somewhat out-of-date since this technology has changed rapidly in the last year. 4a

With regard to the Hewlitt Packard 2640A: 4b

The Lineprocessor does not support this terminal although it looks like it could with about one man-month of effort. 4b1

The major drawback would be the 9 characters needed to position our pointing device, called a mouse. The Data Medias require only 2 characters. This could make the movement of the cursor appear jerky on the HP-2640A screen. 4b2

The Line drawing and symbol options for the HP-2640A are not supported by our system. No estimate is available on how much effort it would take, assuming it could be done. 4b3

At last check this terminal only supported 2400 baud lines; for display NLS a 4800 baud terminal is most desirable. 4b4

COMMUNICATIONS 5

We believe that the new managers of the ARPANET, DCA, are allowing any government agency the use of the net. Please contact Bob Brownfield, DCA Code 531 at 202-692-7583, for detail information. 5a

Another alternative would be WATS telephone service (band 5 - full country) which would cost $1675 plus 7% excise tax per month for two incoming lines (240 hours per month with $4.65 for each additional hour of connect time). 5b

WATS service will limit use to voice grade lines. Display terminals could be used with VADIC modems which cost $975 each to purchase. (This runs at 1200 baud as opposed to a recommended 4800 baud for display terminals) 5b1

Wide band leased lines from ATT would cost about $2300 per month. 5c

Private communication companies using Satellite or special ground lines would cost about $1200 to $1300 per month for wide band service. 5d

The new commercial network, TELENET, would cost anywhere from $8000 to $10,000 per year for an active user on a display terminal. 5e

I hope this information is helpful. Please contact me if you have any questions. 6

Sincerely,

Robert N. Lieberman

Enclosures

SRI proposal No. ISU 75-116
NLS Workshop Support for OSHA

Workstation Equipment Reference Manual

(J26178) 12-AUG-75 15:13;;;; Title: Author(s): Robert N. Lieberman/RLL; Distribution: /JCN([INFO-ONLY]) ARC-LOG([INFO-ONLY]) DCE([INFO-ONLY]) ; Sub-Collections: SRI-ARC ARC-LOG; Clerk: RLL; Origin: < ARC-LOG, NSF,NLS;6, >, 25-JUL-75 17:07 RLL ;;;;
####;

Possible XL10 Bug

1 In the course of debuggong NLS-9 last week, Karolyn and I came accross a bug in code in FILMNP which I think reflects a bug in the XL10 compiler. 1

2 By mistake, there were multiple invocations of the same catchphrase (global to the file, if it makes any difference) in a single procedure without dropping the first invocation. There were also three other catchphrases invoked (which were local to the procedure). Upon exit (before returning or terminating) the catchphrases were explicitly dropped except for the second invocation of the multiply invoked catchphrase. Upon execution, the system blew up at the return. By removing the redundant invocation, we got the code to work. 2

Possible XL10 Bug

(J26179) 23-JUL-75 11:40;;;; Title: Author(s): Harvey G. Lehtman/HGL; Distribution: /DIA([ACTION]) NPG([INFO-ONLY]) ; Sub-Collections: SRI-ARC NPG; Clerk: HGL;

1 26179 Distribution

1a Don I. Andrews, 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,

RLL 23-JUL-75 19:34 26180

QED and recognition modes.

1 Your idea of a variant of the demand mode is appreciated. Thanks it will be put in with other suggestions. 1

2 In speaking to you about the recognition modes I did not mention the secondary mode one can select when TERSE is picked as the primary mode. In essence, after typing <SP> while in terse, you can select any of the four modes again as the secondary mode. Terse in this case will look like demand except will not permit first level command words as alternatives. Anticipatory and fix will behave the same as usual. In this way you can have the most frequently used command recognized by one letter and the other commands recognized in any of the modes. 2

3 About the XED edit controls, I don't know if it is easily changed but will find out and let you know. 3

4 Installing another editor on OFFICE-1 is VERy unlikely. First, we don't want another system (editor or not) running (it can have serious effects on system response for all). Second, I believe having another editor would only serve to confuse people (of course not for those who know the other editor). It is our intention to have an integrated and coherent interface. Our aim should be to improve NLS in every way possible. We mentioned some while you were here, I believe. Ron T sent me a copy of the codes. I will send him a note thanking him. 4

5 Thanks again; please keeping suggesting. Send them to FEEDBACK@office-1 or FEED via the journal system. Copies to those you think are appropriate. 5

RLL 23-JUL-75 19:34 26180

QED and recognition modes.

(J26180) 23-JUL-75 19:34;;;; Title: Author(s): Robert N. Lieberman/RLL; Distribution: /JGN([ACTION]) ; Sub-Collections: SRI-ARC; Clerk: RLL;

1 Introduction 1

1a This document describes the current problems that result from the limited memory resources of the PDP-11. Since there is difficulty in trying to fit into the PDP-11 all those functions that are deemed desirable by the NSW Steering Committee, some consideration of the various issues seems useful toward achieving some compromise solution. 1a

1b Therefore this memorandum considers various strategies for the implementation of the NSW Frontend (FE) code. It should be noted that the strategies discussed here are not mutually exclusive and that the final solution will result from the selection of the most complementary set of strategies described. 1b

1c The memorandum first discusses 3 implementation domains where alternative strategies are being developed: 1) the implementation language, 2) process interaction, and 3) main memory resource management. This material represents the framework within which solutions have been sought during the design phase. Then the memorandum describes in detail the current status of PDP-11 resource management. Finally, it makes some general observations about the performance of the PDP-11 as a consequence of this limited resource situation. 1c

1d We have not made any final decisions yet. This document is a status report of our current thinking and is intended to communicate to the members of the NSW Steering Committee (and other relevant parties) generally complete information about the substantial issues that have been identified. If we come to a place that might require NSW management attention, this background information should aid them in arriving at a decision. We welcome any questions (or any other form of dialogue, for that matter) regarding the information presented in this memorandum. 1d

1e Some readers may prefer to skip to the section of the memorandum that deals with the current resource requirements. This is fine, but we encourage the reader to return to this part later and read the document sequentially. 1e

2 Various Implementation Strategies 2

2a Implementation Language Strategy 2a

2a1 Whenever main memory resources are perceived to be limited during the course of the implementation of a system, the question of whether or not a high-level language should be used as the implementation language naturally arises. The responses to this question are well known, but a quick summarization of them as they apply to the NSW Frontend effort is reasonable. 2a1

2a2 Often it is generally desirable to implement systems in a high-level language. Doing so significantly reduces the costs associated with the development, debugging, and maintenance phases of any piece of software. This situation is further enhanced in the case of the Frontend software since various pieces of code that are being written by SRI will run in both the PDP-10 and the PDP-11. This approach makes it possible for terminals connected to a PDP-10 via some TIP to obtain most of the advantages of the Frontend that terminals directly coupled to the NSW PDP-11 will receive. The use of a high-level language whose compiler(s) can produce machine code for either machine greatly simplifies this task. 2a2

2a3 However, two costs are exacted upon the system implementors when writing in a high-level language: 1) increased storage requirements, and 2) some performance degradation. There can be no argument that these costs hurt in a system with limited physical resources. The significant question remains as to whether or not these costs override the notable advantages of writing in high-level languages. 2a3

2a4 It is our observation that our L10/11 compiler is currently generating code that requires about 25-30% more storage than hand-written assembly code. So that we can try to minimize the storage penalties associated with L10/11 code, we are currently re-working the compiler to improve its code generation up to 15% over what we are experiencing to-date. We feel that code efficiency gains beyond what we expect from our current plans can only be had at costs

beyond what the benefits can justify in a developmental project. 2a4

2a5 We think that the implementation of the Frontend code in a high-level language is the correct approach when considering all costs, and we plan no changes in strategy in this area. Since the target date for an operational (as opposed to developmental) NSW is 1978, the final choice of the Frontend machine is not confined to the PDP-11 and writing assembly language code at this point would only require re-implementation at that time. 2a5

2b Process Interaction Strategy 2b

2b1 In this category, the issues are particularly complex and therefore more difficult to clearly present. Due to the extremely tight fit in the PDP-11, considerable attention is focused at this time in the design strategies affecting the interaction between the Distributed Programming System (DPS), Command Language Interpreter (CLI), and Display Terminal Handler (DTH). We expect to achieve a solution where the different processes can interact as efficiently as possible with one another--thereby reducing overall storage requirements--while not greatly impairing their generality. 2b1

2b2 The use of shared pages is one such strategy that is being explored as one mechanism for interaction between processes. Another strategy is to make maximum use of the operating system facilities for main memory management within and between processes. Undoubtedly other strategies will be developed as we proceed further in the implementation of the Frontend. 2b2

2b3 But to return the discussion to DPS, we have been reluctant to "strip" it of features (and thereby reduce memory requirements) since packages other than the CLI are presumed to require its services. The B4700 code is an example of such a package. Even though we are experiencing limited memory resources in the 11/40 and the possibility exists of not being able to put the B4700 code in the same machine as the Frontend code, it is still our hope that the adoption of an appropriate memory management strategy will permit us to retain the B4700 package in the Frontend processor. 2b3

2c Main Memory Resource Management Strategy 2c

2c1 General 2c1

2c1a The discussion that follows requires that the reader have some knowledge of the memory management facilities of the PDP-11. The maximum amount of physical main memory that can be configured on the 11/40 or 11/45 is 124K words (128K less the 4K reserved for the device registers). Because of the 16-bit word architecture of the PDP-11, individual processes possess a virtual address space that may contain up to 32K words on the 11/40. (On the 11/45 and 11/70, code and data have separate virtual address spaces of 32K each.) Memory relocation registers provide the hardware mechanism for mapping a virtual address space into the physical memory of the PDP-11.

2c1b Given this memory management philosophy, there are a number of strategies that can be used in the implementation of the Frontend. What follows, then, is a discussion of some of the ways in which we might "fold" the various pieces of Frontend code into the available physical memory of the 11/40. (This is necessary since it is not possible to have all the code and data for 20 terminals actually reside permanently in main memory.) The preferred strategy undoubtedly will be one that can be implemented within certain cost constraints, but also achieves some of our performance goals.

2c2 1) Swap Virtual Address Space 2c2

2c2a This strategy simply swaps the entire contents of a virtual address space between main memory and disk storage. This strategy seemingly would make it quite easy for the job scheduler to automatically and entirely control the swapping activity. Moreover, this strategy permits the maximum re-use of the available physical memory.

2c2b However, the amount of information that must be swapped (up to 32K words) can cause considerable delay--even when a very high-speed disk is used as the swapping device. Additionally, it requires a large area of contiguous physical storage to be available before the swap can be performed.

2c3 2) Swap Data Context 2c3

2c3a This strategy assumes that re-entrant code permanently resides in main memory and that only the writable data unique to a given terminal or process is swapped between main memory and disk storage when needed. Its chief virtue is that this strategy reduces to an absolute minimum the amount of main memory that is swapped. This results in the least opportunity for time being lost due to disk transfer time. On the other hand, it only permits re-use of that area of main memory actually used for swapable data. Currently, this only represents a little over 10% of physical storage.

2c4 3) Independently Swap Data Context and Overlay Code 2c4

2c4a This strategy also assumes that the writable data unique to a given terminal is swapped between main memory and disk storage when needed. Additionally, this strategy includes the notion that code is overlaid by the operating system at the explicit request of user code. Its chief advantage is that it reduces the amount of physical storage actually required by that code, thereby permitting a larger number of swapable data contexts to be resident in storage at any one time. Its disadvantages include the increased amount of swapping between main memory and disk storage, and the increased complexity of the user code that must now specifically preside over its code residency.

2c5 4) Demand Paging 2c5

2c5a This strategy assumes that only referenced pages need to be swapped between disk and main memory. This philosophy is attractive as long as the system is careful to avoid the classical deadlock and thrashing conditions that can arise. Clearly some responsibility lies with the implementors of user code if page faults are to be held to a minimum. Failure to do so can result in particularly poor performance of the overall system.

2c5b A special footnote should be made about the demand paging strategy: the 11/45 and 11/70 can implement this strategy with little difficulty. However, the 11/40 does not officially possess a page faulting capability. In spite of this, investigation has shown that demand paging can be partially implemented on the 11/40 as long as certain instruction forms are absent from Frontend code.

2c6 Operating System Selection 2c6

2c6a Concurrent with our consideration of main memory management stategies, we been investigating the various operating systems currently available for implementing these strategies.

2c6b UNIX implements the first strategy of swapping the entire contents of an address space. Although UNIX possesses considerable attributes (such as an operational file system and demonstrated reliability in supporting multiple terminals), its swapping strategy introduces too much delay for it to be practical as the PDP-11 Frontend operating system.

2c6c RSX11D implements a strategy similar to but not exactly like the third one described above. However, RSX11D also possesses limitations that are slowly emerging. There is a municipal project in another group at SRI that is currently engaged in overseeing the installation of a multi-terminal 11/45 system running under RSX11D. It has been determined that design decisions internal to RSX11D have resulted in a strategy that is biased against high-speed display terminals in favor of slow-speed TTYs. Since this is not compatible with long-range NSW goals, the selection of this operating system is suspect.

2c6d Since ELF was developed under ARPA sponsorship, however, the source code is readily available to permit the introduction of new facilities useful to the entire ARPA community. Thus it seems most likely that a modified ELF system will be used that implements an efficient main memory management scheme as defined by NSW Frontend requirements.

3 Current Resource Management Estimates 3

3a General 3a

3a1 This part brings together information concerning the use of memory resources within the 11/40. It contains our most recent estimates--estimates that result from compilation of appropriate Frontend modules or careful estimation where compilation is not yet possible. Care is taken to identify the requirments of resident code, resident data, and swapable data so that the effects of increasing or decreasing the number of users can be more closely observed. 3a1

3b Disk Estimates 3b

3b1 The following map describes the disk storage requirments of a Frontend supporting 20 users. If our estimates are correct, 20 users should fit comfortably on a single 512K word disk drive. 3b1

3b2 Disk needs: 20 users 3b2

3b2a Frontend code	112K	
CLI state @ 8K	160K	
DPS state @ 2K * 2 events	80K	(worst case)
Display data @ 2K	40K	
Grammars @ 2K	40K	
User/node profiles @ 1K	20K	
TOTAL	452K	

3b2b NOTE: This assumes that all users have different grammars and that all terminals are high-speed displays.

3c Main Memory Estimates 3c

3c1 General 3c1

3c1a The following map describes the main memory requirements of the Frontend 11/40 as required by strategy 2 defined above (i.e., code and certain data are resident, while swapping is performed on the remaining terminal-dependent data). The reader will note that memory is portrayed as a number of virtual address

spaces. This is done to improve comprehension since it more clearly presents how code and data might be mapped into the physical and virtual address spaces of the 11/40.

3c1b This strategy assumes that each terminal (or job) will have several processes associated with it--the CLI and DPS are examples of such processes. The operating system is responsible for activating the appropriate process when information is waiting to be consumed. The use of pages shared between different address spaces permits these processes to exchange information. The use of semaphores and/or signals permit these processes to coordinate their activities in response to some user or network introduced event.

3c1c The reader is cautioned that the information presented here is simply a working hypothesis. It does not necessarily portray the actual approach that will be used in the 11/40 Frontend, since that has not been completely decided at this time.

3c2 Address Space 0: 3c2

3c2a ELF KERNEL Code (includes estimates for swapping disk device driver, swapping primitives, & look-ahead swap scheduler)	12K	Resident
3c2b UTILITY Page Code (includes VM ELF Loader, ELF Kernel Debugger)	4K	Resident
3c2c KERNEL data for 20 users Data	8K	Resident
TOTAL	24K	

3c3 Address Space 1: 3c3

3c3a NCP	4K	Resident Code
EXEC/TELNET Modified for NSW	4K	Resident Code
Printer process	.5K	Resident Code
User Crossnet debugger	3.5K	Resident Code
User Address Space Loader	1K	Resident Code
3c3b NCP buffers	6K	Resident Data
EXEC buffers	5K	Resident Data
TELNET buffers	7K	Resident Data
User Crossnet debugger data	.75K	Resident Data
printer buffers (2)	.25K	Resident Data
TOTAL	32K	

3c4 Address Space 2: 3c4

3c4a Display Terminal Handler	6K	Resident Code
DEX support	2K	Resident Code
3c4b L10 runtime	3K	Shared w/ DPS & CLI
Frontend Storage Management	.2K	
CLI/DPS communication buffer	.8K	
3c4c CLI/DTH communication buffer	4K	Shared w/ CLI
TOTAL	16K	

3c5 Address Space 3: 3c5

3c5a DPS	23K	Resident Code
L10 List runtime	.5K	Resident Code
DPS data	.5K	Resident Data
3c5b DPS disk window	2K	Swapable Data (includes stack=1.25K, l10runtime data=.75K
3c5c L10 runtime	3K	Shared w/ CLI & DTH
Frontend Storage Management	.2K	
CLI/DPS communication buffer	.8K	

TOTAL	30K

3c6 Address Space 4: 3c6

3c6a CLI Code	14K	Resident
3c6b CLI shared grammar Data	2K	Swapable
CLI disk window (includes stack=2,75K, L10runtime data=.75K, FE data=.5K, FE free storage zone=2K, CLI/OSI data=.75K, writable grammar=.25K, user/node profile=1K)	8K	Swapable Data
3c6c L10 runtime	3K	Shared w/
Frontend Storage Management	.2K	DPS & DTH
CLI/DPS communication buffer	.8K	
3c6d CLI/DTH communication buffer	4K	Shared w/ DTH
TOTAL	32K	

3c7 Summary 3c7

3c7a These estimates are relatively complex (because of the use of shared pages and such), and its seems worthwhile to attempt to develop some perspective regarding the way in which this information reflects a coherent strategy.

3c7b First of all, the subtotal that one finds at the end of the estimate for each address space defines how much storage is used in that address space. This subtotal includes any information stored in shared pages since we are defining the requirements of that address space and it is important that the number not exceed 32K words. But this means that one cannot simply accumulate the sizes of each address space so as to determine the physical memory requirements of the Frontend code in the 11/40.

3c7c For the reader's convenience, we offer the

following summarization of physical main memory requirements:

3c7d	Resident		Swapped		Shared	Total
Address Space 0	24K					24K
Address Space 1	32K					32K
Address Space 2	8K			8K		16K
Address Space 3	24K		2K	4K		30K
Address Space 4	14K		10K	8K		32K
	102K	+	12K	8K	=	122K

3c7d1 [Note: Since ELF currently performs memory management using 4K word pages, we actually require 124K words for the 11/40 Frontend code.]

3c7e A point of particular concern is the scarcity of main memory resources and the possible effects of that condition. For example, the above table indicates that 12K words of storage are required to store the swappable data that is associated with an individual user terminal. We would much prefer to have several instances of swappable data resident in memory at one time so that context switching between users would be improved (from the point of view of time). However, this clearly is not possible given the physical constraints of the 11/40 physical memory and a swapping philosophy that includes only swapping of unique terminal data. For this reason, it is almost certain that some strategy which includes the swapping of code as well as data will be adopted in the Frontend.

3d Some Observations regarding these Estimates 3d

3d1 Disk Limitations 3d1

3d1a Performance Characteristics of Candidate Swap Devices

3d1a1 There exist several mass storage devices that are candidates as a swapping peripheral. The following information portrays the basic characteristics of each of these devices:

3d1a1a Characteristics	RS04	RP03	RP04	
type device:	fixed-	moving-	moving-head	
price of 1 unit:	$19200	$33500	$35000	
words per disk:	512K	20480K	44000K	
word tfr rate:	4.0	7.5	2.5	usec
time for 1/2 rev:	8.5	12.5	8.3	msec
next track seek:	-	7.5	7.0	msec

3d1a2 Although a fixed-head disk like the RS04 is regarded historically as superior to a moving-head disk like the RP04 as a swapping peripheral, there are circumstances under which a moving-head disk can be very competitive. This situation can arise when head motion is held to a minimum (thereby minimizing the penalty of seek time) and transfers between disk and main memory are accomplished using large block lengths (thus maximizing the effectiveness of high-speed transfer).

3d1b Effects of Swap Length on Performance

3d1b1 Since these different mass storage devices have different transfer rates, it is useful to measure the effect of transfer length on their performance. Though it is self-evident that the smallest elapsed time is required to transfer a swap block when it is done as a single physical I/O transfer, other considerations sometimes preclude that the swap be performed in that manner. Therefore we submit a table that describes the time required to transfer a storage block of specified length:

3d1b1a Length	RS04	RP03	RP04	
1K words	12.5	32.0	22.5	msec
2K words	16.5	40.0	25.0	msec
4K words	24.5	55.5	30.0	msec
8K words	40.5	85.0	40.0	msec
16K words	72.5	145.0	60.0	msec
24K words	104.5	205.0	80.0	msec
32K words	136.5	265.0	100.0	msec

3d1b2 Several assumptions were made in the above calculations. The moving-head disk columns contain an extra 11.7 msec (for the RP04) or 12.5 msec (for the RP03) to allow for the appropriate head positioning

time. All columns contain the appropriate average latency between transfers--thereby requiring 2 transfers per revolution of the disk if the figures are to be accurate.

3d1c Observations on Swap Performance

3d1c1 An initial observation is that there are relatively few swaps/second possible given the size of the CLI disk window. The number of CLI disk window swaps is even further reduced when we introduce the CLI shared grammar or DPS disk window swaps of 2K words each, which are assumed to be independent from the CLI disk window swaps. Given that the CLI disk window is 8K words and it will be transferred as 2 blocks of 4K words, we will be able to perform about 20 swaps per second (ignoring the issue of whether or not other swaps occur during that time). Since a context switch amounts to swapping someone out to disk as well as swapping another in from disk, there can be on the order of 10 process activations per second before the disk completely saturates with I/O traffic.

3d1c2 One conclusion is that some way must be developed that permits faster activation of the CLI process. For example, a smaller swap length would have an immediately beneficial effect upon performance. Unfortunately, the various disk windows currently seem to be as small as possible since they have already undergone as much reduction in size as seems reasonable at this point.

3d1c3 However, there does appear to be a favorable solution to this process activation situation. It concentrates on the perception that CLI process activations are in either of 2 categories: 1) collecting keystrokes that further the specification of a command, and 2) collecting those keystrokes that are arguments (the textual ones in particular) of the command. In the first case, it is necessary to activate the CLI process for each keystroke so that it can perform the command parsing function for which it is designed. In the second case, however, control can be given to a smaller process that can collect literal text keystrokes on behalf of the CLI until an activation event occurs--whereupon it can return control to the CLI with everything that has been previously entered.

3d1c4 Given that this strategy is implemented, the number of times that process activation must occur is considerably reduced. As a consequence of this, performance as perceived by the user should be improved considerably when compared with our current mode where users are TELNETed to some host running NLS. We would like to be able to report what performance will be like for perhaps 10 or 20 users based upon some absolute scale. Regretfully, it is beyond our power to produce such an estimate at this time. Rather, it can be said more simply that performance will degrade as the number of users increases. In any case, it is our hope that degradation in the 11/40 Frontend as a result of a large number of users will not cause delay that is perceived by the users.

3d2 Future Considerations 3d2

3d2a There are a number of considerations that suggest that future Frontend systems should employ a processor other than the 11/40. For example, if the 11/45 were used, address space limitations would be alleviated considerably since an address space can contain up to 32K words in code and 32K words in data. In contrast, the 11/40 permits 32K words for the storage of both code and data. Moreover, the 11/45 permits a demand paging strategy to be cleanly implemented (unlike the 11/40) and this strategy seems to be the one that we consider the most promising for main memory management in the Frontend.

3d2b Long-range consideration should be given to the 11/70 as an eventual successor to the 11/40 since we currently require the maximum memory size on the 11/40 (and even the 11/45) as the minimum Frontend configuration. With the opportunity to configure more main memory on the 11/70, higher performance can be achieved since less swapping is required. The cache store of the 11/70 would increase the effective instruction execution rate, thereby indefinitely forestalling the time when there are inadequate CPU resources as a result of system demands. Finally, the RS04 swapping disk almost doubles in performance when attached to the 11/70 since I/O is done on a 32-bit (rather than a 16-bit) bus.

3d2c Perhaps more controversial, but a long-range

consideration nonetheless, would be the use of some new computer mainframe whose architecture is based upon a 32-bit word. An example of such a system might be the Interdata 8/32 or perhaps a DEC product that might be introduced at some later time. Although this is not realistic within a short timeframe because of the need for ARPANET-related hardware and software, it is clearly within the timeframe circumscribed by long-range planning.

3d2d In conclusion, some computer possessing characteristics superior to the 11/40 would permit more capabilities to be installed in the Frontend--currently desirable capabilities that are best postponed and future capabilities that will eventually come under consideration.

Issues in PDP-11 Frontend Development

(J26181) 23-JUL-75 20:09;;;; Title: Author(s): Joseph L. Ehardt/JLE;
Distribution: /SRI-ARC([INFO-ONLY]) ; Sub-Collections: SRI-ARC;
Clerk: JLE; Origin: < EHARDT, NSW-POSITION-PAPER.NLS;3, >,
23-JUL-75 20:06 JLE ;;;;

####;####;

NSW Milestones

1 Introduction 1

1a This is a statement of milestones for SRI-ARCs NSW effort, these are our current best estimates, we expect that as we get into the various tasks our estimates may change and this note will be updated. 1a

1b Terms 1b

1b1 NLS 8.0 1b1

1b1a The existing NLS at Office-1 and BBNB. 1b1a

1b2 NLS 8.5 1b2

1b2a The NLS with the file structure and Graphics initial capabilities. 1b2a

1b3 NLS 9.0 1b3

1b3a The NLS with the front-end back-end split, using the CLI and DPS for communications, and integrated into the NSW. 1b3a

1b4 CLI 1b4

1b4a Command Language Interpreter 1b4a

1b5 DPS 2.5 1b5

1b5a The Distributed Programming System as defined in June 1975. 1b5a

1b6 DPS 3.0 1b6

1b6a The Distributed Programming System as revised and redesigned by April 1976. 1b6a

1b7 Applications Packages 1b7

1b7a Sets of related procedures for carrying out system applications, for example the File Package. 1b7a

1b8 FE-10 1b8

1b8a The Front-End on the PDP-10 Tenex: this is principally the CLI, but is dependent on DPS and Terminal Control modules as well as the operating system. 1b8a

1b9 FE-11 1b9

1b9a The Front-End on the PDP-11: this is principally the CLI, but is dependent on DPS and Terminal Control modules as well as the operating system. 1b9a

1b10 "char tty" 1b10

1b10a This characterizes the character at a time interaction using hardcopy terminals, withn echoing and prompting provided by the CLI. 1b10a

1b11 "half duplex tty" 1b11

1b11a This characterizes the interaction where the terminal echos the characters actually typed, and the CLI has no information about how much has been typed (ie, line at a time or not) thus the CLI does no echoing or prompting. 1b11a

1b12 "display" 1b12

1b12a This characterizes a full duplex two dimentional alphanumeric display work station incorporating a line processor. 1b12a

1b13 Base 1b13

1b13a The central core editing commands of NLS 1b13a

2 Protocols 2

2a	Distributed Programming system 2.5 code-10 debugged	15-Sep-75	2a
2b	Distributed Programming System 2.5 code-11 debugged	15-Sep-75	2b
2c	Distributed Programming System ver 2.5 documentation	15-Oct-75	2c
2d	Applications Packages version 2.5 documentation	15-Oct-75	2d
2e	Tool Bearing Host version 2.5 documentation	15-Nov-75	2e
2f	Measurement and Efficiency study report	15-Dec-75	2f
2g	Distributed Programming System ver 3.0 documentation	1-Apr-76	2g
2h	Applications Packages version 3.0 documentation	1-Apr-76	2h

3 Front End 3

3a	FE-10 (char tty, new tool) runs	15-Aug-75	3a

3b	FE-10 (char tty, old tool) runs	15-Oct-75	3b
3c	FE-10 (half duplex tty, new & old tool) runs	15-Oct-75	3c
3d	FE-10 (display, new tool) runs	15-Oct-75	3d
3e	FE-11 (char tty, new tool) runs	1-Jan-76	3e
3f	FE-11 (char tty, old tool) runs	1-Jan-76	3f
3g	FE-11 (display, new tool) runs	15-Feb-76	3g
3h	Measurement and Efficiency study report	1-Apr-76	3h
3i	Command Meta Language Evaluation report	1-Apr-76	3i
3j	New Front End features study report	1-Apr-76	3j
4	NLS		4
4a	WM-NLS file interaction design document	1-Aug-75	4a
4b	NLS 8.5 ready for experimental use	15-Aug-75	4b
4c	Identification system specification document	15-Aug-75	4c
4d	Base in NSW accessed in char tty mode	21-Aug-75	4d
4e	Journal design document	1-Sep-75	4e
4f	Programs subsystem	1-Oct-75	4f
4g	Cobol Output subsystem in 8.0 or 8.5	1-Oct-75	4g
4h	Base in NSW accessed in display mode	15-Nov-75	4h
4i	modify, format,letter, publish subsystems in 8.5	15-Nov-75	4i
4j	Calculator subsystem	1-Dec-75	4j
4k	NLS 8.5 ready to be used by the utility	1-Jan-76	4k
4l	Output Processor modifications	1-Mar-76	4l
4m	modify, format,letter, publish subsystems in 9.0	1-Mar-76	4m
4n	Sendmail and Journal	1-Apr-76	4n
4o	Graphics in NLS 9.0	1-Apr-76	4o

4p LSI 11 terminal controller	1-Apr-76	4p
5 Documentation		5
5a NLS Base subsystem documentation	1-Oct-75	5a
5b NLS Programs subsystem documentation	1-Nov-75	5b
5c NLS Modify, Format, Letter, Publish subsystem doc	15-Dec-75	5c
5d NLS Calculator subsystem documentation	15-Dec-75	5d
5e Front End user documentation (on line)	1-Jan-76	5e
5f Front End user documentation (off line)	1-Jan-76	5f
5g Front End system documentation	15-Jan-76	5g
5h Front End tool suppliers information document	1-Feb-76	5h
5i Debugger Document	1-Feb-76	5i
5j NLS Journal documentation	1-Apr-76	5j

NSW Milestones

(J26183) 24-JUL-75 12:42;;;; Title: Author(s): Jonathan B. Postel/JBP; Distribution: /RWW([INFO-ONLY]) CHI([INFO-ONLY]) JEW([INFO-ONLY]) EKM([INFO-ONLY]) BEV([INFO-ONLY]) JBP([INFO-ONLY]) ; Sub-Collections: SRI-ARC; Clerk: JBP; Origin: < POSTEL, MILESTONES.NLS;9, >, 24-JUL-75 12:39 JBP ;;;;####;

1 26183 Distribution
1a Richard W. Watson, Charles H. Irby, James E. (Jim) White,
Elizabeth K. Michael, Beverly Boli, Jonathan B. Postel,

DVN 24-JUL-75 13:47 26184

Two New Idents, One Real and One Fake

1 Marcia, would you create two idents for me. One is Beth Sherman, her address is room K1096 at SRI, extension 2633, she is part of the Information Sciences Lab. Delivery hardcopy. She wants to be BS, if that is taken BETH, if that is taken ELS. The other is a fake ident for use in user documentation. Lets make him Hypothetical (Hy) O User, ident HOU, Address is Augmentation Reserach Center. No delivery....Thanks. 1

DVN 24-JUL-75 13:47 26184

Two New Idents, One Real and One Fake

(J26184) 24-JUL-75 13:47;;;; Title: Author(s): Dirk H. Van Nouhuys/DVN; Distribution: /MLK([ACTION]) PWO([INFO-ONLY]) POOH([INFO-ONLY]) BEV([INFO-ONLY]) JHB([INFO-ONLY]) KIRK([INFO-ONLY]) ; Sub-Collections: SRI-ARC; Clerk: DVN;

1 26184 Distribution
1a Marcia L. Keeney, pat Whiting O'Keefe, Ann Weinberg, Beverly Boli,
James H. Bair, Kirk E. Kelley,

response to your cobol memo

1 After reading your memo <iijournal,32066> and large parts of the B4700 COBOL manual I have several comments to make and questions to ask: 1

2 Comments 2

2a On page 56-5 of the manual AFC 171-121(c1) vol III, it states that items which begin in area A start in col 8 and items which begin in area B start in col 12. Hence your positioning strategy is invalid. Here is the strategy Elizabeth and I propose for the first source line of an NLS statement. 2a

2a1 Level 1 statements will start in column 8. Except within the data division, all other statements will start in column 12. Within the data division, statements in the file section are either (my terminology follows) FD statements or SD statements or grouped in record descriptions. 2a1

2a1a All FD and SD statements will be handled similarly to statements outside the data division. The first statement of a record description should be level 1 and as such will begin in column 8. Subsequent statements in the same record description will be indented two columns per level. The COBOL level number will be inserted using the formula 2a1a

2a1a1 COBOL level = 2*(NLS level) - 1 2a1a1

2a2 Statements in the data division which are not in the file section are in the working-storage section. Statements here are either 66, 77 or 88 statements or grouped in record descriptions. 2a2

2a2a All 66,77 and 88 statements will be handled similarly to statements outside the data division. Statements in record descriptions will be handled similarly to record descriptions in the file section. 2a2a

2b Here is our strategy for continuations lines. Our philosophy is that the programmer will not think in terms of continuations at all; rather, continuations are handled solely by COBOLAID. In other words, no NLS statement will be a continuation of the preceding NLS statement. The programmer will be able to think solely in terms of NLS statements. Thus what follows is a description of what COBOLAID will do, not what the programmers must do. 2b

2b1 Comments will be broken at word boundary into several comments. Note that for COBOLAID, a comment is any NLS statement, regardless of level, whose first non-blank character

is "*". A word which is too big for one source line will be simply split across comments. 2b1

2b2 All other continuation circumstances involve a continuation character,"-", being placed in column 7 and the continuation commencing in column 12. 2b2

3 Questions 3

3a Do you still wish to handle the identification area as you originally stated ? 3a

3b Do you still wish to handle the sequence numbers as you originally specified ? 3b

3b1 I think it is sufficient to have the sid of the corresponding NLS statement appear in columns 1-6 right justified with zero fill as the sequence number for a source card. It seems superfluous to me to have column 6 essentially be a counter for continuations, thus placing an implicit limit on the number of source cards an NLS statement can generate. 3b1

3c Do you still want page throws to be implicitly generated before the data and procedure divisions ? 3c

3d In your discussion of sequence numbers, I do not understand what you mean by "A card inserted between statements(on the direction of the programmer ..)". 3d

response to your cobol memo

(J26185) 24-JUL-75 17:24;;;; Title: Author(s): Jan A. Cornish/JAC3;
Distribution: /LAC([ACTION]) HGL([INFO-ONLY]) EKM([INFO-ONLY]
) ; Sub-Collections: SRI-ARC; Clerk: JAC3; Origin: < CORNISH,
RESPONSE.NLS;2, >, 24-JUL-75 14:07 JAC3 ;;;;####;

1 26185 Distribution
1a Lawrence A. Crain, Harvey G. Lehtman, Elizabeth K. Michael,

JAC3 24-JUL-75 17:27 26186

response to your cobol memo

1 After reading your memo <ijournal,32066> and large parts of the B4700 COBOL manual I have several comments to make and questions to ask: 1

2 Comments 2

2a On page 56-5 of the manual AFC 171-121(c1) vol III, it states that items which begin in area A start in col 8 and items which begin in area B start in col 12. Hence your positioning strategy is invalid. Here is the strategy Elizabeth and I propose for the first source line of an NLS statement. 2a

2a1 Level 1 statements will start in column 8. Except within the data division, all other statements will start in column 12. Within the data division, statements in the file section are either (my terminology follows) FD statements or SD statements or grouped in record descriptions. 2a1

2a1a All FD and SD statements will be handled similarly to statements outside the data division. The first statement of a record description should be level 1 and as such will begin in column 8. Subsequent statements in the same record description will be indented two columns per level. The COBOL level number will be inserted using the formula 2a1a

2a1a1 COBOL level = 2*(NLS level) - 1 2a1a1

2a2 Statements in the data division which are not in the file section are in the working-storage section. Statements here are either 66, 77 or 88 statements or grouped in record descriptions. 2a2

2a2a All 66,77 and 88 statements will be handled similarly to statements outside the data division. Statements in record descriptions will be handled similarly to record descriptions in the file section. 2a2a

2b Here is our strategy for continuations lines. Our philosophy is that the programmer will not think in terms of continuations at all; rather, continuations are handled solely by COBOLAID. In other words, no NLS statement will be a continuation of the preceding NLS statement. The programmer will be able to think solely in terms of NLS statements. Thus what follows is a description of what COBOLAID will do, not what the programmers must do. 2b

2b1 Comments will be broken at word boundary into several comments. Note that for COBOLAID, a comment is any NLS statement, regardless of level, whose first non-blank character

is "*". A word which is too big for one source line will be simply split across comments. 2b1

2b2 All other continuation circumstances involve a continuation character,"-", being placed in column 7 and the continuation commencing in column 12. 2b2

3 Questions 3

3a Do you still wish to handle the identification area as you originally stated ? 3a

3b Do you still wish to handle the sequence numbers as you originally specified ? 3b

3b1 I think it is sufficient to have the sid of the corresponding NLS statement appear in columns 1-6 right justified with zero fill as the sequence number for a source card. It seems superfluous to me to have column 6 essentially be a counter for continuations, thus placing an implicit limit on the number of source cards an NLS statement can generate. 3b1

3c Do you still want page throws to be implicitly generated before the data and procedure divisions ? 3c

3d In your discussion of sequence numbers, I do not understand what you mean by "A card inserted between statements(on the direction of the programmer ..)". 3d

JAC3 24-JUL-75 17:27 26186

response to your cobol memo

(J26186) 24-JUL-75 17:27;;;; Title: Author(s): Jan A. Cornish/JAC3;
Distribution: /LAC([ACTION]) ; Sub-Collections: SRI-ARC; Clerk:
JAC3; Origin: < CORNISH, RESPONSE,NLS;2, >, 24-JUL-75 14:07 JAC3
;;;;####;

1 26186 Distribution
1a Lawrence A. Crain,