RLL 9-NOV-74 11:25 24453

Tenex documents: to be ordered

Aslong as we have at least onecopy of each in our files (accessible by all) then I have no further request. This might lmean an additional copy over the neds of each individual. thanks 1

RLL 9-NOV-74 11:25 24453

Tenex documents: to be ordered

(J24453) 9-NOV-74 11:25;;;; Title: Author(s): Robert N. Lieberman/RLL; Distribution: /POOH([ACTION]) ; Sub-Collections: SRI-ARC; Clerk: RLL;

SLJ 10-NOV-74 00:01 24454

Let's Call A Spade A Spade ...

Just consider: we might call a common garden spade: a personalized earth-moving equipment module; a mineralogical mini-transport; a personalized strategic tellurian command and control module; an air-to-ground interface contour adjustment probe; a leveraged tactile-feedback geomass delivery system; a man-machine energy-to-structure converter; a one-to-one individualized geophysical restructurizer; a portable unitized earthwork synthesis system; an entrenching tool (Firesign Theater); a zero-sum dirt level adjuster; a feedback-oriented contour management probe and digging system; a gradient disequilibrator; a mass distribution negentroprizer; a dig-it-all system; and extra-terrestrial transport mechanism.....Spades, not words, should be used for shovelling. But words should help us unearth the truth.
--excerpt from Dream Machines/Computer Lib 1

SLJ 10-NOV-74 00:01 24454

Let's Call A Spade A Spade ...

(J24454) 10-NOV-74 00:01;;;; Title: Author(s): Sandy L. Johnson/SLJ; Distribution: /SRI-ARC([INFO-ONLY]) ; Sub-Collections: SRI-ARC; Clerk: SLJ;

JAKE 10-NOV-74 14:47 24455

The Identification Program

This is feedback for Karolyn's use that has been expanded for the record. Many thanks to Karolyn for her efforts in getting the identsystem up and running in NLS 8. Jake

The Identification Program

BUGS 1

1. The amount of space alloted to enter a literal address seems to be too small since the error message 'NLS Internal error: string too long' frequently comes up whenever a literal address is entered. Can this be lengthened (if that is what is causing the error message). 1a

2. Previously an error message of 'Exceed Capacity' was registered several times and this stopped all input of new ident data. At that time it was ascertained by JDH that the field length for all information pertaining to one ident was 400 characters. This also needs to be lengthened - might try 1000. 1b

3. There is currently no added protection against inadvertently changing an ident (possibly for NULL data). In the old system one needed to give an additional password to do this. The same protection is probably needed in the new system to prevent already assigned idents from getting changed inadvertently. 1c

4. The prompt 'Delivery' seems to be missing under GROUP entry. 1d

IMMEDIATE NEEDS 2

1. The data elements 'Network Host' and 'Network User Name' as well as ''NLS Host' and 'NLS User Name' need to be in reverse order during prompting since there is a tendency to confuse them due to the prevalence of the usual USER@HOST syntax. 2a

2. The choices of 'Network' or '<>NLS' after 'Delivery' causes many errors. First of all the use of the <SP> syntax for a prompt system o this type is very confusing, in my opinion. Secondly the same letter is needed with or without the <sp>, which is also confusing. Therefore I would reccommend changing '<>NLS' to 'Local' so that the space is not needed and one of the choices does not begin with the letter 'n'. 2b

3. It would be very convenient to have all input of ident-type entries and mailbox addresses be capitalized automatically. This would permit the typist to ignore capitalization for these fields and would save considerable time as well as help prevent errors caused by shifting. Currently they are mixed and this is confusing. Suggestions for automatic capitalization are: 2c

- First letter of both last name and first name; capitalization of middle initial (already implemented, I believe) 2c1
- All types of Idents 2c2

- All entries under membership 2c3
- Co-ordinator ident 2c4
- Network host and mail address (i.e., SMITH [@] USC-ISI 2c5
- NLS host and mail address (i.e., JAKE SRI-ARC 2c6
- All entries under groups 2c7
- Entries of organization type (USER, SERVER, TIP, etc.) 2c8

4. It would be nice to have the prompt 'Do you want to expunge' (or some such thing) come up after an update is completed. This feature would save the typist from having to go out and expunge frequently and would keep large partial copies from taking up needed disk space. 2d

5. Each time a new organization is entered it would be convenient for a Group for that organization to be set up automatically and for the IDENT of the Org. to also occur automatically under 'Groups' under the Org. Ident entry. This happens for Individuals but not for Organizations. 2e

6. There is a tendency to type CONTROL-O when one wants to get back to the beginning of a data prompt. Since there is no control character to take one back to the beginning of a data prompt, the user is taken all the way back to the beginning (IDE C:). When this happens all of the information added for that entry is lost and the user must start over. Would be useful if there was a 'Really?' prompt following each use of CONTROL-O to prevent this happening. Would also be nice if there were some way to get back to the beginning of a data prompt. (Can usually be accomplished with CONTROL-W if only one or two words have been typed.) 2f

7. I would prefer to have three commands in the ident system, namely: ADD, CHANGE, and DELETE. For a novice user it is confusing as to what to do for the latter two. (The ADD command is already there.) It would also be useful to have the 'Load Record' command be buried in two of these three so that the user only has to specify the ident of the record he wishes to CHANGE or DELETE and it would be automatically loaded. 2g

GENERAL DISCUSSION 3

In general, the ident system is very cumbersome both to run and to keep up. It now contains around 1000 entries (give or take a few) but if we continue to expand to new machines, it could contain many more entries. 3a

These are some suggestions that might be considered for future implementation (with no emphasis on whether they are or are not feasible or easily implemented): 3b

1. The use of idents should be discontinued and last names should be adopted. Now a lot of time is spent getting a person's middle initial just so he won't get an ident such as ab6. The use of last names would be more unique, more meaningful to users, more significant in journal headers, and would coincide with sndmsg thus eliminating the need for Utility users to remember both conventions and memorize NLS idents. 3b1

2. The ident system is in need of an inverted file arrangement of some sort. Currently this is all done by Content search (I believe) and is fairly slow. Also, for proofing and online reading it is time-consuming not to have all fields alphabetized automatically. 3b2

3. All data elements should remain in a given sequence no matter in what order they are edited or entered. Currently entries are organized every which way. KLM has gotten rid of some of this for new entries. 3b3

4. There should be an automatic DELETE command so that if the editor deletes an ident entry all uses of this ident throughout the current file are removed and the original ident is placed in the 'limbo' category. This is all done by hand now (that is when it is done) and is very time consuming. 3b4

5. Housekeeping for all group and organization membership lists should be automatic. That is, if an individual is eliminated from a group the ident should be eliminated from the data element 'Groups' under that individuals ident entry and vice versa. Groups for oganizations should be set up automatically (see above). Deletion of a Group or Organization ident from the file would automatically delete every use of that ident throughout the file. Likewise a change of any ident - individual, group, or orgaization - should automatically trigger its change wherever used throughout the identfile. Some of this happens now but much of it is done by hand or does not get done. 3b5

6. We should do away with the Data elements 'Network Host' and 'NetWork User Name' and replace it with 'NetWork MailboX Address' for which the expected entry would be: USERNAME@HOSTNAME 3b6

7. If we continue to produce the ARPAnet directory and

maintain the official hostname list, I would recommend expanding the infomation included under an Organization entry so that it contains most of the information now maintained in the file (feinler,hostaddr-master,). There is redundancy between these two files. To do this the current ident system might have to be split into more than one file (such as Individuals, Groups, and Organizations). 3b7

These are just some needed features that come to mind. They are not exhaustive or definitive but are rather suggestions for possible improvement of problem areas. 3c

The Identification Program

(J24455) 10-NOV-74 14:47;;;; Title: Author(s): Elizabeth J. (Jake) Feinler/JAKE; Distribution: /KLM([ACTION]) BUGS([ACTION]) NP([INFO-ONLY]) JCN([INFO-ONLY]) ; Sub-Collections: SRI-ARC BUGS NP; Clerk: JAKE; Origin: < FEINLER, IDENTS-FEEDBACK.NLS;3, >, 10-NOV-74 14:41 JAKE ;;;;####;

JAKE 10-NOV-74 14:52 24456

One other Ident bug

One other item I forgot is the following: when entering an ident for a group after 'type ident desired', the program waits until all information has been entered and the entry is being updated before it lets you know that this ident was already used. It would be a win if the system let you know this at the time of input rather than update. 1

JAKE 10-NOV-74 14:52 24456

One other Ident bug

(J24456) 10-NOV-74 14:52;;;; Title: Author(s): Elizabeth J. (Jake) Feinler/JAKE; Distribution: /KLM([ACTION]) ; Sub-Collections: SRI-ARC; Clerk: JAKE;

JAKE 11-NOV-74 08:17 24457

Goof in Ident Usage

The message you received yesterday was really intended for Karolyn Martin whose ident is KJM. I made an error in typing and the system accepted it. Sorry for the inconvenience - please disregard. Jake Feinler

1

JAKE 11-NOV-74 08:17 24457

Goof in Ident Usage

(J24457) 11-NOV-74 08:17;;;; Title: Author(s): Elizabeth J. (Jake) Feinler/JAKE; Distribution: /KLM([INFO-ONLY]) ; Sub-Collections: SRI-ARC; Clerk: JAKE;

JAKE 11-NOV-74 08:20 24458

Identsystem Wrap-up and Comments for Future

BUGS 1

1. The amount of space alloted to enter a literal address seems to be too small since the error message 'NLS Internal error: string too long' frequently comes up whenever a literal address is entered. Can this be lengthened (if that is what is causing the error message). 1a

2. Previously an error message of 'Exceed Capacity' was registered several times and this stopped all input of new ident data. At that time it was ascertained by JDH that the field length for all information pertaining to one ident was 400 characters. This also needs to be lengthened - might try 1000. 1b

3. There is currently no added protection against inadvertently changing an ident (possibly for NULL data). In the old system one needed to give an additional password to do this. The same protection is probably needed in the new system to prevent already assigned idents from getting changed inadvertently. 1c

4. The prompt 'Delivery' seems to be missing under GROUP entry. 1d

IMMEDIATE NEEDS 2

1. The data elements 'Network Host' and 'Network User Name' as well as ''NLS Host' and 'NLS User Name' need to be in reverse order during prompting since there is a tendency to confuse them due to the prevalence of the usual USER@HOST syntax. 2a

2. The choices of 'Network' or '<>NLS' after 'Delivery' causes many errors. First of all the use of the <SP> syntax for a prompt system o this type is very confusing, in my opinion. Secondly the same letter is needed with or without the <sp>, which is also confusing. Therefore I would reccommend changing '<>NLS' to 'Local' so that the space is not needed and one of the choices does not begin with the letter 'n'. 2b

3. It would be very convenient to have all input of ident-type entries and mailbox addresses be capitalized automatically. This would permit the typist to ignore capitalization for these fields and would save considerable time as well as help prevent errors caused by shifting. Currently they are mixed and this is confusing. Suggestions for automatic capitalization are: 2c

- First letter of both last name and first name; capitalization of middle initial (already implemented, I believe) 2c1

- All types of Idents 2c2

- All entries under membership 2c3
- Co-ordinator ident 2c4
- Network host and mail address (i.e., SMITH [@] USC-ISI 2c5
- NLS host and mail address (i.e., JAKE SRI-ARC 2c6
- All entries under groups 2c7
- Entries of organization type (USER, SERVER, TIP, etc.) 2c8

4. It would be nice to have the prompt 'Do you want to expunge' (or some such thing) come up after an update is completed. This feature would save the typist from having to go out and expunge frequently and would keep large partial copies from taking up needed disk space. 2d

5. Each time a new organization is entered it would be convenient for a Group for that organization to be set up automatically and for the IDENT of the Org. to also occur automatically under 'Groups' under the Org. Ident entry. This happens for Individuals but not for Organizations. 2e

6. There is a tendency to type CONTROL-O when one wants to get back to the beginning of a data prompt. Since there is no control character to take one back to the beginning of a data prompt, the user is taken all the way back to the beginning (IDE C:). When this happens all of the information added for that entry is lost and the user must start over. Would be useful if there was a 'Really?' prompt following each use of CONTROL-O to prevent this happening. Would also be nice if there were some way to get back to the beginning of a data prompt. (Can usually be accomplished with CONTROL-W if only one or two words have been typed.) 2f

7. I would prefer to have three commands in the ident system, namely: ADD, CHANGE, and DELETE. For a novice user it is confusing as to what to do for the latter two. (The ADD command is already there.) It would also be useful to have the 'Load Record' command be buried in two of these three so that the user only has to specify the ident of the record he wishes to CHANGE or DELETE and it would be automatically loaded. 2g

GENERAL DISCUSSION 3

In general, the ident system is very cumbersome both to run and to keep up. It now contains around 1000 entries (give or take a few) but if we continue to expand to new machines, it could contain many more entries. 3a

Identsystem Wrap-Up and Comments for Future

These are some suggestions that might be considered for future implementation (with no emphasis on whether they are or are not feasible or easily implemented): 3b

1. The use of idents should be discontinued and last names should be adopted. Now a lot of time is spent getting a person's middle initial just so he won't get an ident such as ab6. The use of last names would be more unique, more meaningful to users, more significant in journal headers, and would coincide with sndmsg thus eliminating the need for Utility users to remember both conventions and memorize NLS idents. 3b1

2. The ident system is in need of an inverted file arrangement of some sort. Currently this is all done by content search (I believe) and is fairly slow. Also, for proofing and online reading it is time-consuming not to have all fields alphabetized automatically. 3b2

3. All data elements should remain in a given sequence no matter in what order they are edited or entered. Currently entries are organized every which way. KLM has gotten rid of some of this for new entries. 3b3

4. There should be an automatic DELETE command so that if the editor deletes an ident entry all uses of this ident throughout the current file are removed and the original ident is placed in the 'limbo' category. This is all done by hand now (that is when it is done) and is very time consuming. 3b4

5. Housekeeping for all group and organization membership lists should be automatic. That is, if an individual is eliminated from a group the ident should be eliminated from the data element 'Groups' under that individuals ident entry and vice versa. Groups for oganizations should be set up automatically (see above). Deletion of a Group or Organization ident from the file would automatically delete every use of that ident throughout the file. Likewise a change of any ident - individual, group, or orgaization - should automatically trigger its change wherever used throughout the identfile. Some of this happens now but much of it is done by hand or does not get done. 3b5

6. We should do away with the Data elements 'Network Host' and 'Network User Name' and replace it with 'Network Mailbox Address' for which the expected entry would be:
USERNAME@HOSTNAME 3b6

7. If we continue to produce the ARPAnet directory and

maintain the official hostname list, I would recommend expanding the infomation included under an Organization entry so that it contains most of the information now maintained in the file (feinler,hostaddr-master,). There is redundancy between these two files. To do this the current ident system might have to be split into more than one file (such as Individuals, Groups, and Organizations). 3b7

These are just some needed features that come to mind. They are not exhaustive or definitive but are rather suggestions for possible improvement of problem areas. 3c

JAKE 11-NOV-74 08:20 24458

Identsystem Wrap-up and Comments for Future

(J24458) 11-NOV-74 08:20;;;; Title: Author(s): Elizabeth J. (Jake) Feinler/JAKE; Distribution: /KJM([ACTION]) ; Sub-Collections: SRI-ARC; Clerk: JAKE; Origin: < FEINLER, IDENTS-FEEDBACK,NLS;3, >, 10-NOV-74 14:41 JAKE ;;;;####;

JEW 22-NOV-74 13:37 24459

The Procedure Call Protocol
PCP Version 2

22-NOV-74

James E. White
Augmentation Research Center

Stanford Research Institute
Menlo Park, California 94025

PCP is an inter-process procedure call and return mechanism which provides a setting in which higher-level tools can be remotely offered and used.

(J24459) 22-NOV-74 13:37;;;; Title: Author(s): James E. (Jim) White/JEW; Sub-Collections: SRI-ARC; Clerk: JEW; Origin: <WHITE, PCP-PCP.NLS;42, >, 22-NOV-74 11:54 JEW ;;;; ####;

ACKNOWLEDGMENTS 1

The author gratefully acknowledges the help of the following individuals, all of whom have contributed to the PCP design: 1a

Peter Deutsch (PARC-MAXC) 1a1
Charles Irby (SRI-ARC) 1a2
Jon Postel (SRI-ARC) 1a3
Rick Schantz (BBN-TENEXA) 1a4
Bob Thomas (BBN-TENEXA) 1a5
Ken Victor (SRI-ARC) 1a6

PREFACE 2

The Procedure Call Protocol (PCP) is an inter-process and/or inter-host protocol that permits a collection of processes within one or more ARPANET hosts to communicate at the procedure call level. In effect, it makes the component procedures of remote software systems as accessible to the programmer as those within his own system. PCP specifies both a virtual programming environment (VPE) in which remote procedures may be assumed to operate, as well as the inter-process exchanges that implement it. 2a

The Multi-Process Software System (MPSS) whose construction PCP makes practical and of which the NSW is an example, consists of collections of "procedures" and "data stores" called "packages", in one or more "processes", interconnected in a tree structure by "physical channels". Procedures within a process have free access to the procedures (and data stores) of each process adjacent to it in the tree structure, and may call upon them as if they were local subroutines. Superimposed upon the tree structure is a more general set of interconnections which give non-adjacent processes in the tree the same kind of access to one another. 2b

The MPSS is implemented by: 2c

1) low-level protocols which provide the basic, inter-process communicaton (IPC) facilities by which channels are implemented: an inter-host IPC protocol (PCPHST), an inter-Tenex-fork IPC protocol (PCPFRK), and data structure format specifications for both connection types (PCPFMT). 2c1

2) PCP proper, which largely defines the VPE (especially, the procedure call and return mechanism) and specifies the inter-process control exchanges required to implement it. 2c2

3) a set of system packages, implemented within each process, which augment PCP proper by providing mechanisms by which user procedures can: call remote procedures (implemented by the Procedure Interface Package, PIP), manipulate remote data stores (implemented by the PCP Support Package, PSP), and interconnect processes (implemented by the Process Management Package, PMP). 2c3

4) user packages in each process. 2c4

INTRODUCTION 3

Protocol Design Approaches 3a

Several ARPANET applications (i.e. third- and fourth-level) protocols have been designed and implemented since the Host-Host Protocol was adopted in 1970. Most have been bootstrapped from lower-level applications protocols. For example, the File Transfer Protocol (FTP) was built upon TELNET, and the Remote Job Entry Protocol (RJE) upon both TELNET and FTP. The highest-level protocol shared by all such bootstrapped protocols is TELNET. 3a1

Although the bootstrapping principle seems a sound basis for Network protocol development, we believe that TELNET, providing little more of use than a character set, is NOT the most appropriate foundation for a large class of applications protocols. 3a2

Bootstrapping at a Higher Level 3b

We contend that a Procedure Call Protocol (PCP) -- a Network-standard mechanism for invoking arbitrary named, argument-driven and result-producing procedures in a remote process -- is a much more appropriate and powerful foundation for many applications protocols. We believe that the adoption by the Network community of a PCP as the basis for most applications protocols would have at least the following effects: 3b1

1) expedite the specification of applications protocols by permitting their documentation to have a functional, rather than a syntactic orientation, 3b1a

2) largely eliminate the need for separate, application-specific user processes, 3b1b

3) reduce the cost of making large, existing software systems available as Network servers by allowing a Network interface more compatible with their internal organization, 3b1c

4) provide the basis for a more natural interface between local and remote procedures, and therefore 3b1d

5) encourage the sharing of software, by making procedures on remote hosts as accessible to the programmer as local ones. 3b1e

Document Organization 3c

At the highest level, PCP is the specification of a virtual programming environment in which remote procedures are assumed to operate. The model specifies the manner in which remote procedures gain and relinquish control, the kinds of data structures with which they can be expected to deal, and so forth. One of the tasks of the PCP implementer, therefore, is to provide a mapping between his real programming environment and the virtual one defined by PCP. The PCP virtual programming environment is the subject of the first section of this document. 3c1

At a slightly lower level, PCP is the specification of the interchanges between two connected processes which implement the virtual programming environment. This specification is given in the final section of the current document. 3c2

PCP is intended to be suitable for interlinking two processes within a single host, as well as processes on different hosts. The present document therefore gives only a functional description of PCP, common to both applications. A subsequent document will provide a detailed, syntactic description of the protocol for its Network application. 3c3

The present document is the foundation for a series of subsequent documents describing higher-level programming and applications tools designed to operate within the setting provided by PCP. 3c4

Development Plans 3d

We solicit comments on both PCP and its underlying premise. SRI-ARC will implement and employ PCP to interconnect modules of the National Software Works (NSW): the Frontend, the Works Manager, and tools. As part of this effort, SRI-ARC will make the core functions or "backend" of NLS available both as a Network server process and as a Tenex fork. The NSW Frontend will use the backend via the Network; an additional Frontend for PDP-10 Tenex will use the backend in either mode. 3d1

THE MODEL 4

Processes 4a

By definition, let a "process" consist of the following: 4a1

1) One or more procedures 4a1a

A "procedure" is a named and ordered set of primitive operations, performed in response to a "call" from another procedure. The last operation of the procedure always effects a "return" to the calling procedure. In this document, the terms "caller" and "callee" will be used to refer to the procedures which effect the call and return, respectively. 4a1a1

The behavior of a procedure is controlled by means of zero or more "arguments", passed to it by its caller. Subsequent operation of the caller may in turn be affected by zero or more "results" produced by the callee. Arguments and results are data structures. 4a1a2

A "data structure" is a unit of information having a "type" (e.g. a fixed-point integer, a character string, a list of character strings), a value (-23; "SRI-ARC"; "Hither", "Thither", "Yon"), and an optional "key" or name by which it can be identified. 4a1a3

2) Zero or more data stores 4a1b

A "data store" is a named information repository that exists throughout the process' lifetime and that can be manipulated by any procedure within the process. A data store contains a data structure. 4a1b1

3) One or more processors 4a1c

A "processor" is a mechanism which executes in proper sequence, the primitive operations from which a procedure is formed. A processor might be a physical device like a CPU, or simply the construct of a CPU's time-sharing monitor. 4a1c1

Like a Tenex fork or an OS task, both of which fit the definition, most PCP processes will reside physically within a single host machine, and each will be characterized by a single instruction set, operating system, and run-time program environment. However, the PCP definition is not meant to preclude more complex implementations. 4a2

The purpose of PCP is to provide a mechanism by which two processes, connected by a communication "channel", can call one another's procedures, manipulate one another's data stores, and coordinate processors. In the general case, all such capabilities are reciprocal. Neither process need play a purely passive role; each can call the other's procedures and manipulate its data stores. 4a3

Data Structures 4b

Data Types 4b1

Before a data structure can be transmitted between processes, its sender must convert it from his own local format to a standard format agreed upon by both sending and receiving processes. The receiver must subsequently decode the standard-format representation and from it reconstruct the data structure, but in his local format, rather that in the sender's. 4b1a

To facilitate the encoding/decoding task, a small set of data types is defined by PCP: 4b1b

CHARSTR: a character string, 4b1b1

BITSTR: a bit string, 4b1b2

INTEGER: a signed integer, 4b1b3

BOOLEAN: true or false, 4b1b4

EMPTY: null, and 4b1b5

LIST: an ordered list of data structures called "elements". This last data type provides a mechanism by which arbitrarily complex composite data structures can be constructed from the other, atomic types listed above. 4b1b6

Arguments to and results of procedures, as well as data stores, must all be modelled using these data types. Encodings for each data type listed above are specified in another document (PCPFMT). 4b1c

Other Data Structure Attributes 4b2

In addition to its value, every data structure has two additional "attributes". All the attributes of a data structure (whether it be an argument, result, or the contents of a data store), not just its value, accompany the data structure whenever it is transmitted between processes. 4b2a

The following data structure attributes are defined: 4b2b

1) Type: as explained above 4b2b1

2) Value: the contents of the data structure (e.g. -23, "SRI-ARC"), which must be consistent with its type. 4b2b2

3) Key: a secondary data structure (with its own type and value, but not key), which can be used to select the data structure from among others within a list, or within the process as a whole. 4b2b3

Syntax conventions 4b3

In this and subsequent, PCP-related documents, it will often be necessary to describe specific data stores, and the arguments required and results returned by specific procedures. In some cases it will suffice to state only the data type required (or a set of several allowed types); in others it will be necessary to specify not only the type of the argument, result, or data store, but also a range of allowed values. In a few situations, some discussion of keys will be required as well. 4b3a

In all PCP-related documents, data structure descriptions will have the following form: 4b3b

datastruc ::= ['< key '>] typechoices 4b3b1

If the description is that of a data store, KEY will necessarily be specified; if it is that of a procedure argument or result, the whole description will be preceeded by the name of the argument or result to which it applies and a dash ('-'). 4b3c

The non-terminals KEY and TYPECHOICES in the BNF above are described below. Throughout the BNF, the meta-linguistic

brackets and slash denote optional and alternative elements
of the description, respectively: 4b3d

```
key          ::= typechoices / name                                  4b3d1

   name ::= the VALUE of a STRING                                    4b3d1a

   Because keys of type STRING are so common, "name" is
   admitted as shorthand for a key of type STRING and
   value "name".                                                     4b3d1b

typechoices ::= type ['/ typechoices]                                4b3d2

   type ::= typename ['[ valuechoices ']]                            4b3d2a

      typename ::= "CHARSTR" / "BITSTR" / "INTEGER" /
                   "BOOLEAN" / "EMPTY"  /
                   "LIST" ['( list ')] / "any"                       4b3d2a1

         list ::= (datastruc / "...") [', list]                      4b3d2a1a

      valuechoices ::= value ['/ valuechoices]                       4b3d2a2

         value ::= [symbolicvalue] ['= absolutevalue]                4b3d2a2a
```

The syntax above permits one to specify the allowed values (in both a symbolic and an absolute form), types ("any" implies that any data type is permitted), and keys of the data structure being described. It also allows specification of the same information for each element of a list ("..." implies an arbitrary number of elements like the previous one). Comments enclosed in percent signs ('%) may be inserted anywhere in the description, between terminal elements. 4b3e

As an example (oriented toward the Tenex programming environment), the following notation might be employed to describe a procedure argument CONNECTION which identifies an ARPANET connection in terms either of its JFN (if the connection is already established); or the host address and socket number to which a request for connection is to be addressed, and the kind of buffering to be done for the connection: 4b3f

```
   connection= %jfn% INTEGER / LIST (%hostaddr% INTEGER,
               %socket% INTEGER, %buffering% INTEGER [NONE=0
               / TILLCRLFSENT=1 / TILLBUFFERFULL=2])                 4b3f1
```

Calling Procedures in Another Process 4c

Introduction 4c1

By definition, a process must provide, as two of the primitive operations available for procedure construction, mechanisms for calling another procedure (and for passing its arguments to it) and for returning to a caller (with the results of the procedure). When both caller and callee reside within the same process, as is the case with most existing software systems, each part of this "call-return mechanism" (CRM) is usually just a few machine instructions. One of PCP's tasks is to specify an alternate CRM to be used when the two procedures reside in different processes. 4c1a

Transfer of Control 4c2

The simplest kind of interaction between a procedure and its caller is that in which the procedure is called, runs to completion, and then returns to the caller. Such interactions are the most frequent ones, supported by every process' native CRM, and PCP provides for them. A more complicated control transfer discipline is sometimes required, however, and it too is supported by PCP (even though, in some cases, it may not be supported by the process' native CRM). 4c2a

In the most general case supported by PCP, a procedure is permitted to return "temporarily" to its caller, via a coroutine linkage, an arbitrary number of times before returning to it "permanently". Temporary returns lead to suspension of the callee (and the saving of whatever state information is required to resume it), further execution of the caller, and then "resumption" of the callee. 4c2b

Arguments and results can be exchanged between caller and callee in a coroutine linkage, just as in a call and (permanent) return. 4c2c

The caller distinguishes a final return from a temporary one by means of a "return type" provided by the callee. The following return types are defined: 4c2d

PERMANENT 4c2d1

As its name suggests, this type of return is permanent. No state information has been saved by the callee, and it therefore cannot be resumed. 4c2d1a

TEMPORARY 4c2d2

This return type signifies a coroutine linkage. The caller may perform any actions it wishes provided it eventually resumes the callee, whose state has been saved. In particular, the caller may call additional procedures in the other process, or even (another instance of) the same procedure. 4c2d2a

INPROGRESS 4c2d3

This return type is reserved for use by the Procedure Interface Package, described in another document. 4c2d3a

As indicated above, PCP permits a caller whose callee has made a temporary return, to re-call the same procedure one or more times before resuming the original instance of it. So that several calls to a single procedure can be distinguished, the caller's CRM associates a unique "call handle" (CH) with every procedure called. The association is made when the procedure is called, and is maintained until the procedure makes a permanent return. The CH, rather than the procedure name, is then used in specifying the procedure to be resumed. 4c2e

Certain "subtypes" of temporary and permanent returns are sufficiently common to warrant formalization, and are described at appropriate points in this document. Upon such returns, PCP places special constraints upon the kinds of parameters that can be exchanged between caller and callee, and (for temporary returns) upon the actions the caller is required to perform before resuming the callee. 4c2f

Specifying Arguments and Results 4c3

Introduction 4c3a

The caller of a procedure might use its results in any of a variety of ways, depending upon the context of the call: as the basis for a branch decision, as an argument to a second procedure, or not at all (i.e. the result might be ignored). 4c3a1

When caller and callee reside in different processes, certain inefficiencies can result, in the last two cases described above, if the CRM, unaware of the result's intended use, blindly returns it to the caller. In the second of the two cases, the data structure is shipped back to the caller's process only to be ignored; in the other, it is returned to the caller and then with a subsequent call shipped back to the process from which it originated. 4c3a2

Argument and Result List Masks 4c3b

To help eliminate such inefficiencies, PCP permits a caller to provide an argument list "mask", in addition to the argument list itself, which specifies the source of each argument; and a result list mask which specifies the destination of each result. 4c3b1

The argument list mask permits the caller to supply each argument either from the argument list itself, or from a data store in the callee's process (in which case, the corresponding element of the argument list should contain EMPTY). For each result, the result list mask instructs the CRM to either return the result to the caller via the result list, deposit it in a specified data store in the callee's process, or discard it. In the last two cases, the corresponding element of the result list contains EMPTY. 4c3b2

To use the result of one procedure as an argument to a second (in the same process), the caller need only appropriately set corresponding elements of the result list mask in the first call and the argument list mask in the second. 4c3b3

The PCP Support Package, described in another

document, provides a means by which temporary data stores within the callee's process can be created for such use. 4c3b3a

Data Structure Addressing 4c3c

As has already been pointed out, the LIST data type permits the construction of arbitrarily complex data structures. PCP permits the programmer, by means of the argument and result list masks, to direct arguments and results from and to individual elements of a complex data store. 4c3c1

The pseudo data type DSELECTOR*, when it appears in an argument or result list mask, selects a particular data structure from among a process' data stores, and is simply shorthand for the following: 4c3c2

```
DSELECTOR* ==> LIST (%ph% INTEGER, %pkh% INTEGER,
                %data store key% any, <BOOLEAN
                [KEY=TRUE / INDEX=FALSE]> %element% any
                / INTEGER, ...)
```
4c3c2a

PH is the process handle of the process, and PKH the package handle of the package, which contains the data store which is to serve as the source or destination for the argument or result. Process handles and packages are constructs of the Process Management (PMP) and PCP Support (PSP) Packages, respectively, described in other documents. When PCP is used outside of the higher-level framework provided by these packages, PH and PKH should be set to zero. 4c3c3

The more general data structure addressing scheme supported by PMP permits references to data structures in arbitrary processes. 4c3c3a

DATA STORE KEY identifies by key a particular data store (within the specified package and process, if appropriate). Each ELEMENT selects an element of the previous data structure (assumed to be a LIST), either by specifying its index (beginning with one), or its key (if the key is ambiguous, the element with lowest index is selected). 4c3c4

As an example, given the following data store in package 4 of process 3: 4c3c5

```
<hosts> LIST (<tenex> LIST (CHARSTR [="BBN-TENEX"],
        CHARSTR [="SRI-ARC"]), <non-tenex> LIST
```

(CHARSTR [="UCSB-MOD75"], CHARSTR
[="AMES-TIP"])) 4c3c5a

the selector: 4c3c6

LIST (INTEGER [=3], INTEGER [=4], CHARSTR [="hosts"],
<BOOLEAN [=TRUE]> CHARSTR [="non-tenex"], <BOOLEAN
[FALSE]> INTEGER [=2]) 4c3c6a

appearing in an argument list mask would cause the data structure CHARSTR [="AMES-TIP"] to be supplied as an argument to the procedure. 4c3c7

Reporting the Outcome of a Procedure 4c4

Many procedures attempt well-defined tasks at which they either succeed or fail. The outcome of such a procedure might be communicated to the caller in a variety of ways: returned as a formal result of the procedure, deposited in a data store, or by any other mechanism agreed upon by both caller and callee. Although it cannot prevent the use of such ad hoc schemes, PCP provides and encourages the use of a more standard mechanism (even though the process' native CRM may not), by defining the following subtypes of permanent returns: 4c4a

SUCCESS: the procedure has succeeded at its task (or the concept of success or failure is not meaningful for this procedure), 4c4a1

FAILURE: the procedure has failed at its task, and 4c4a2

ABORTED: the procedure has encountered an unexpected and irrecoverable error. For this subtype, the results returned by the callee consist of a program-readable error code and an optional, human-readable diagnostic message, 4c4a3

Summoning Assistance from a Caller 4c5

A procedure occasionally reaches a point in its execution beyond which it cannot proceed without help. It might need additional buffer space, another parameter from the human user upon whose behalf it is executing, etc. The procedure may have invested considerable real and/or processing time before the need was discovered; if it has to abort, the investment is lost. In most cases, other procedures along the thread of control will also be forced to abort, and a higher-level procedure will finally locate the buffer space or obtain the parameter from the user, and then re-attempt the aborted chain of procedure calls. 4c5a

To help minimize such inefficiencies, PCP permits the procedure encountering the problem to solicit help from any of the procedures along the thread of control, by means of the temporary return subtype HELP. The callee supplies a program-readable code which identifies the problem, and any other parameters it considers appropriate. 4c5b

The caller may perform any operations it wishes provided it eventually resumes the callee, whose state has been saved, with an indication of whether or not the requested help has been provided. If the caller cannot provide help, it must propagate the request up the thread of control, by returning to ITS caller via a coroutine linkage of subtype HELP, and then relay to the callee whatever help (if any) the higher-level procedure has provided. 4c5c

Reporting Events to a Caller 4c6

A procedure occasionally witnesses (or causes) an event about which other procedures along the thread of control may wish to know. The event might be the start or completion of some major step in the procedure's execution (e.g. the opening of a network connection), the reaching of a breakpoint, etc. In such circumstances, PCP permits the callee, by means of the temporary return subtype NOTE, to notify each of the procedures along the thread of control of the event. 4c6a

The callee supplies a program-readable code which identifies the event which has occurred, and any other parameters it considers appropriate. The caller may perform any operations it wishes provided it eventually resumes the callee, whose state has been saved. Before doing so, however, it must propagate the notice up the thread of control, by returning to ITS caller via a coroutine linkage of subtype NOTE, so that all higher-level procedures receive word of the event. 4c6b

Priviledged Procedures 4c7

In a multi-process system, those system procedures which effectively manipulate several processes, must be implemented as a main, user-callable procedure, and one or more internally-callable procedures (one in each process) which the main procedure invokes. 4c7a

So the programmer can prevent such procedures from being called directly by the user and thus misused, PCP permits a procedure to be marked "priviledged", and prevents such procedures from being called in any but priviledged procedure call requests. It is left to each process to properly restrict the class of procedures which can make priviledged calls to remote procedures. 4c7b

An exclamation mark ('!) following the name of a procedure indicates that the procedure being described or called must or is being called with a priviledged procedure call request. 4c7c

Interrupting a Procedure 4d

It is sometimes desireable to be able to "interrupt" a long procedure, either to handle a higher-priority request or to stop an undebugged and looping procedure; PCP therefore provides such a mechanism. The interruption is transparent to the callee, but appears to the caller as a temporary return of subtype INTERRUPTED. The caller is then free to perform any operations it wishes, provided it eventually resumes the callee, whose state has been saved. 4d1

Aborting a Procedure 4e

An undebugged, looping procedure, interrupted as described above, must eventually be aborted. PCP therefore permits a caller to "abort", rather than resume, any callee which has made a temporary return (not necessarily of subtype INTERRUPTED). The callee's processor may perform as many more operations within the callee as are required to leave the process in a self-consistent state, but must then abort the procedure and return to the caller as if the procedure had voluntarily initiated a permanent return of subtype ABORTED. 4e1

Processor Assignment 4f

Whenever a remote procedure is called, it is assigned a processor by the remote process. When the procedure returns to its caller temporarily, the processor halts, the procedure's state is saved by the process, and the processor is released. Resumption of the procedure by the caller necessitates reassignment of a processor, restoration of the callee's state, and resumption of the callee. When the procedure returns permanently, the processor halts, no state information is saved, and the processor is released a final time. 4f1

A procedure call is said to be "outstanding" whenever the caller awaits a return. The number of calls to procedures within the remote process that may be simultaneously outstanding is equal to the number of processors with which the remote process is equipped. 4f2

THE IMPLEMENTATION 5

Messages 5a

Introduction 5a1

The PCP CRMs of connected processes communicate with one another by means of "messages" they exchange via the channel that connects them. 5a1a

Messages are data structures. Using the LIST data type, PCP's CRM combines the various message parameters, each of which is itself a data structure, to form a larger data structure which it transmits as a message to the other process. Data structures are thus used by PCP to express not only the values of data stores and procedure arguments and results, but also inter-process requests. 5a1b

Although they appear in messages, process (PHs) and package (PKHs) handles are not used by PCP proper. Process handles and packages are constructs of the Process Management (PMP) and the PCP Support (PSP) Packages, respectively, described in other documents. When PCP is used outside of the higher-level framework provided by these packages, PHs and PKHs should be set to zero wherever they are required. 5a1c

PH and CH, which are the first parameters of most messages, identify, respectively, the process which is the intended recipient of the message (used by PMP to route messages between processes), and the procedure call to which the message pertains (a newly assigned CH appears in the CALPRO message; a previously assigned one in all other types of messages). 5a1d

Thoughout the descriptions below, the terms "sending process" and "receiving process" denote, respectively, the process that sends the message being described, and the process that is its intended recipient. 5a1e

Messages 5a2

Call procedure 5a2a

CALPRO (ph, ch, pkh, pname, args, arglmsk, reslmsk, priviledged) 5a2a1

This message effectively calls with arguments ARGS, procedure PNAME in package PKH in the receiving process. PRIVILEDGED must be true to successfully call a priviledged procedure. 5a2a2

ARGLMSK and RESLMSK are the argument and result list masks, respectively. 5a2a3

RESLMSK is applied only when the call makes a permanent return of subtype SUCCESS or FAILURE. For subtype ABORTED, the result list mask is overridden and the procedure results (which consist of diagnostic information) are returned to the caller. RESLMSK has no effect upon the results of a temporary return which, like diagnostic information, are always returned to the caller. 5a2a3a

ARGLMSK and RESLMSK are LISTs whose i-th elements corresponds to the procedure's i-th argument and result, respectively. If the mask has fewer elements than the number of arguments or results, its last element is applied to all that remain; excess elements are ignored. A one-element result list mask, for example, can be used to specify a single disposition for all results. 5a2a3b

The receiving process is expected to execute the indicated procedure and respond with a RTNPRO message (specifying the same CH), indicating either a temporary or permanent return. 5a2a4

Format: 5a2a5

LIST (op, ph, ch, pkh, pname, args, arglmsk, reslmsk, priviledged) 5a2a5a

op - INTEGER [CALPRO=0] 5a2a5b
ph - INTEGER 5a2a5c
ch - INTEGER 5a2a5d

```
        pkh          - INTEGER                                          5a2a5e
        pname        - CHARSTR                                          5a2a5f
        args         - LIST                                             5a2a5g
        arglmsk      - LIST (INTEGER [ARGLIST=0] / DSELECTOR*,
                       ...)                                             5a2a5h
        reslmsk      - LIST (INTEGER [RESLIST=0 / DISCARD=1] /
                       DSELECTOR*, ...)                                 5a2a5i
        priviledged- BOOLEAN                                            5a2a5j

    Return from procedure                                               5a2b

      RTNPRO (ph, ch, type, subtype, results)                           5a2b1

      This message acknowledges the return (type TYPE and
      subtype SUBTYPE) with results RESULTS, within the sending
      process, of the procedure identified by CH, previously
      called or resumed by the receiving process,                       5a2b2

      Format:                                                           5a2b3

        LIST (op, ph, ch, type, subtype, results)                       5a2b3a

        op      - INTEGER [RTNPRO=1]                                    5a2b3b
        ph      - INTEGER                                               5a2b3c
        ch      - INTEGER                                               5a2b3d
        type    - INTEGER [PERMANENT=0 / TEMPORARY=1 /
                  INPROGRESS=2]                                         5a2b3e
        subtype- %permanent% INTEGER [SUCCESS=0 / FAILURE=1 /
                   ABORTED=2] /
                  %temporary% INTEGER [GENERALCOROUTINE=0 /
                   NOTE=1 / HELP=2 / INTERRUPTED=3]
                  %inprogress% EMPTY                                    5a2b3f
        results- %success or failure or general coroutine% any
                   /
                  %aborted% LIST (%errcode% INTEGER, %errmsg%
                   CHARSTR / EMPTY)
                  %note or help% LIST (%event or problem%
                   INTEGER, %parameter% any) /
                  %interrupted or detached% EMPTY                       5a2b3g
```

Resume procedure 5a2c

RSMPRO (ph, ch, args) 5a2c1

This message effectively resumes with new arguments ARGS, the procedure identified by CH awaiting resumption in the receiving process after a temporary return. 5a2c2

The receiving process is expected to resume the indicated procedure and respond with a RTNPRO message (specifying the same CH), indicating either a permanent or another temporary return. 5a2c3

Format: 5a2c4

LIST (op, ph, ch, args) 5a2c4a

op - INTEGER [RSMPRO=2] 5a2c4b
ph - INTEGER 5a2c4c
ch - INTEGER 5a2c4d
args- LIST 5a2c4e

Interrupt procedure 5a2d

INTPRO (ph, ch) 5a2d1

This message effectively interrupts the procedure identified by CH, in the receiving process. The receiving process is expected to acknowledge the interruption with a RTNPRO message which indicates a permanent return of subtype INTERRUPTED. 5a2d2

Format: 5a2d3

LIST (op, ph, ch) 5a2d3a

op- INTEGER [INTPRO=3] 5a2d3b
ph- INTEGER 5a2d3c
ch- INTEGER 5a2d3d

Abort procedure 5a2e

ABRPRO (ph, ch) 5a2e1

This message effectively aborts the procedure identified by CH awaiting resumption in the receiving process after a temporary return. 5a2e2

The receiving process is expected to abort the indicated procedure and respond with a RTNPRO message (specifying the same CH), indicating a permanent return of subtype ABORTED. 5a2e3

Format: 5a2e4

LIST (op, ph, ch) 5a2e4a

op= INTEGER [ABRPRO=4] 5a2e4b
ph= INTEGER 5a2e4c
ch= INTEGER 5a2e4d

Note protocol violation 5a2f

PCPERR (ph, ch, errcode, errmsg) 5a2f1

This message notifies the receiving process that the sending process has witnessed it violate the Procedure Call Protocol. ERRCODE and ERRMSG (which is optional) identify the error in program- and human-readable form, respectively. 5a2f2

The receiving process should at least log the error report, and one or both processes should probably break off communication with the other. 5a2f3

Format: 5a2f4

LIST (op, ph, ch, errcode, errmsg) 5a2f4a

op = INTEGER [PCPERR=5] 5a2f4b
Ph = INTEGER / EMPTY 5a2f4c
ch = INTEGER / EMPTY 5a2f4d
errcode= INTEGER 5a2f4e
errmsg = CHARSTR / EMPTY 5a2f4f

No oeration 5a2g

NOP () 5a2g1

This message requests no operation, and may be discarded by the receiving process without action. 5a2g2

Format: 5a2g3

LIST (op) 5a2g3a

op= INTEGER [NOp=6] 5a2g3b

Inter-Process Communication 5b

Introduction 5b1

PCP assumes that an inter-process communication channel, which it can use to transmit the messages described above, can be established between processes which implement PCP; and that each process provides its procedures with appropriate mechanisms for manipulating it. The inter-process communication (IPC) primitives described below are thus assumed, not provided by PCP. 5b1a

Since PCP is designed to be useful for mediating communication between processes linked by different kinds of channels, the details of the channel implementation are assumed hidden in the IPC procedures. The IPC primitives may be implemented by a process either as lower-level procedures or as the kind of primitive operations from which procedures are BUILT. In the descriptions below, the former approach has been assumed for convenience. 5b1b

In the procedure descriptions in this and other, PCP-related documents, the term "local process" denotes the process in which the procedure being described resides. 5b1c

Channels 5b2

A channel is a full-duplex communication device which connects a "port" in one process with a port in another. Each port is known to its local process via a "port handle" (POH). 5b2a

NOTE: The Process Management Package, described in another document, destinguishes between physical and logical channels; the IPC primitives discussed in the current document implement physical channels. 5b2a1

The following shorthands are currently defined to denote, respectively, a channel suitable for interconnecting two processes either within the same job, host, or network; a port to which such a channel can be attached; and the location of the process which contains the port: 5b2b

```
CHNTYPSEL* ==> LIST (%chntyp% INTEGER [INTERHOST=0 /
                   INTERJOB=1 / INTERPRC=2], %channel width%
                   INTEGER)                                  5b2b1

PORT*      ==> %channel-implementation dependent% any        5b2b2

PRCLOC*    ==> LIST (%hostaddr% INTEGER, %jobno% INTEGER,
                   %prcno% INTEGER)                          5b2b3
```

Primitives 5b3

Create process 5b3a

CRTPRC (prcaddr => poh, prcname) 5b3a1

This procedure creates and initializes an instance of the process whose address is PRCADDR, allocates a local port which it makes known to the local process via POH, and constructs a channel between the local and newly-created processes. It also returns the generic name PRCNAME of the process created (see PMP). The specified process is assumed to be one which, like the local process, implements PCP, and it too, as a result of this procedure, receives a port handle, by which it can effectively reference the local process. 5b3a2

PMP uses CRTPRC to create inferior processes and specifies legal formats for PRCADDR. When PCP is used outside of the higher-level framework provided by PMP, however, PRCADDR is subject to purely local interpretation. 5b3a3

Argument/result types: 5b3a4

prcaddr- CHARSTR 5b3a4a
poh - INTEGER 5b3a4b
prcname- CHARSTR 5b3a4c

Delete process 5b3b

DELPRC (poh => cost) 5b3b1

This procedure terminates and deletes the process attached to the port known to the local process via POH (which must have been created with CRTPRC), invalidating the handle. COST is the cost in cents of the process' use. 5b3b2

PMP uses DELPRC to terminate and delete inferior processes. 5b3b3

Argument/result types: 5b3b4

poh - INTEGER 5b3b4a
cost- INTEGER 5b3b4b

Send message to process 5b3c

SNDMSG (poh, message) 5b3c1

This procedure transmits the message MESSAGE to and interrupts the process connected by a channel to the port known to the local process via POH. PCP requires that the receiving process promptly accept delivery of the message via its RCVMSG procedure. 5b3c2

Argument/result types: 5b3c3

poh - INTEGER 5b3c3a
message- any 5b3c3b

Accept message from process 5b3d

RCVMSG (poh -> message) 5b3d1

This procedure accepts delivery of the next message MESSAGE transmitted by the process connected by a channel to the port known to the local process via POH. 5b3d2

IPC prevents the transmission of a second message from the sending process before the first has been accepted. However, PCP requires that the local process promptly invoke RCVMSG after being interrupted by the message's arrival. 5b3d3

Argument/result types: 5b3d4

poh - INTEGER 5b3d4a
message- any 5b3d4b

Create end of inter-process channel 5b3e

CRTCHNEND (poh, remport) 5b3e1

This procedure, which implements PMP's CRTPHYCHNEND procedure, creates one end of (i.e. helps to create) a channel of the type specified when the local port was allocated, between the port known to the local process via POH, and remote port REMPORT. 5b3e2

Argument/result types: 5b3e3

poh - INTEGER 5b3e3a
remport- PORT* 5b3e3b

Delete end of inter-process channel 5b3f

DELCHNEND (Poh) 5b3f1

This procedure, which implements PMP's DELPHYCHNEND procedure, deletes one end of (i.e. helps to delete) the previously-created channel attached to the port known to the local process via POH. 5b3f2

Argument/result types: 5b3f3

poh- INTEGER 5b3f3a

Allocate local port 5b3g

ALOPOR (chntypmnu, remloc => chntypsel, port, poh) 5b3g1

This procedure, which implements PMP's ALOPHYPOR procedure, allocates a port PORT within the local process and known to it via POH, for subsequent use in establishing a channel of one of the types CHNTYPSEL permitted by CHNTYPMNU, between the local process, and the remote process at location REMLOC. 5b3g2

Argument/result types: 5b3g3

chntypmnu- LIST (CHNTYPSEL*, ...) 5b3g3a
remloc - PRCLOC* 5b3g3b
chntypsel- CHNTYPSEL* 5b3g3c
port - PORT* 5b3g3d
poh - INTEGER 5b3g3e

Release local port 5b3h

RELPOR (poh) 5b3h1

This procedure, which implements PMP's RELPHYPOR procedure, releases the previously-allocated port known to the local process via POH. 5b3h2

Argument/result types: 5b3h3

poh- INTEGER 5b3h3a

Implementation 5b4

To completely specify the interconnection via PCP of any specific pair of processes, one must more precisely specify the IPC primitives described above. In particular, one must specify an encoding for MESSAGE (i.e. for data structures) and the manner in which the IPC primitives are constructed from still more primitive operations. Both tasks, as previously pointed out, are channel-type dependent, and will be undertaken in subsequent documents for at least those channels appropriate for interconnecting: 5b4a

1) processes in different ARPANET hosts, and 5b4a1

2) forks in the same Tenex job. 5b4a2

The Procedure Interface Package
PIP Version 2

22-NOV-74

James E. White
Augmentation Research Center

Stanford Research Institute
Menlo Park, California 94025

PIP is an interface to the Procedure Call Protocol (PCP == 24459,), with which the reader of the present document is assumed familiar.

JEW 22 NOV 74 7:46PM The Procedure Interface Package

(J24460) 22-NOV-74 14:28;;;; Title: Author(s): James E. (Jim) White/JEW; Sub-Collections: SRI-ARC; Clerk: JEW; Origin: < WHITE, PCP-PIP,NLS;10, >, 22-NOV-74 12:19 JEW ;;;; ####;

PREFACE 1

The Procedure Call Protocol (PCP) is an inter-process and/or inter-host protocol that permits a collection of processes within one or more ARPANET hosts to communicate at the procedure call level. In effect, it makes the component procedures of remote software systems as accessible to the programmer as those within his own system. PCP specifies both a virtual programming environment (VPE) in which remote procedures may be assumed to operate, as well as the inter-process exchanges that implement it. 1a

The Multi-Process Software System (MPSS) whose construction PCP makes practical and of which the NSW is an example, consists of collections of "procedures" and "data stores" called "packages", in one or more "processes", interconnected in a tree structure by "physical channels". Procedures within a process have free access to the procedures (and data stores) of each process adjacent to it in the tree structure, and may call upon them as if they were local subroutines. Superimposed upon the tree structure is a more general set of interconnections which give non-adjacent processes in the tree the same kind of access to one another. 1b

The MPSS is implemented by: 1c

1) low-level protocols which provide the basic, inter-process communicaton (IPC) facilities by which channels are implemented: an inter-host IPC protocol (PCPHST), an inter-Tenex-fork IPC protocol (PCPFRK), and data structure format specifications for both connection types (PCPFMT). 1c1

2) PCP proper, which largely defines the VPE (especially, the procedure call and return mechanism) and specifies the inter-process control exchanges required to implement it. 1c2

3) a set of system packages, implemented within each process, which augment PCP proper by providing mechanisms by which user procedures can: call remote procedures (implemented by the Procedure Interface Package, PIP), manipulate remote data stores (implemented by the PCP Support Package, PSP), and interconnect processes (implemented by the Process Management Package, PMP). 1c3

4) user packages in each process. 1c4

INTRODUCTION 2

The Procedure Interface Package (package name = PIP) provides the local process with the necessary tools for using the procedures of a remote process. PIP is primarily an interface to PCP and includes procedures for calling, resuming, interrupting, and aborting remote procedures. 2a

In- and Out-of- Line Execution 2b

PIP permits a local procedure to execute a remote procedure in either series or parallel with its own processor's continued execution. 2b1

Whenever a local procedure requests "in-line" execution of a remote procedure, PIP waits for the callee to return before returning to the caller. While the remote procedure is being executed, the caller's processor is suspended and thus incapable of performing useful work. By selecting in-line execution of a remote procedure, the caller treats the callee as if it were a local procedure with which it must share a single processor. 2b2

Whenever "out-of-line" execution is requested, PIP initiates the call (i.e, sends the CALPRO message) and returns (type INPROGRESS) immediately to the caller, rather than waiting for the procedure's return (i.e, for the RTNPRO message). The caller's processor is thus available for useful work while the remote procedure is being executed. By selecting out-of-line execution of a remote procedure, the caller can exploit the fact that the callee resides in a different process (and processor), capable of independent execution. 2b3

PIP will eventually acknowledge the return of a procedure called out-of-line, by signalling locally an event specified by the caller. The caller may specify either a different event for each outstanding procedure call, or a single event for all. In the latter case, it must, after being notified of the event, examine PIP's CHLIST data store to determine which callee has returned. In either case, the caller must resume (or abort) the callee via PIP's RSMPRO procedure, to obtain its results. 2b4

Remote procedures may be resumed, as well as called, in either of the two "modes" described above, and the mode selection is independent of that of the initial call or the most recent resumption. 2b5

Call Handle Assignment 2c

PIP assumes responsibility for allocating (via CALPRO) and releasing (via RSMPRO and ABRPRO) call handles (CHs) within the local process. Each CH is unique within the local process, and thus serves to identify both the callee and its process. 2c1

Suggested Shorthands 2d

PCP requires that every process provide, as one of the primitive operations available for procedure construction, a mechanism for returning to a caller with the results of the procedure. In its general form, this primitive might look like the following: 2d1

```
RETURN (type, subtype, results => args)                          2d1a
```

The following shorthands are also suggested: 2d2

Terminate a procedure normally 2d2a

```
EXIT (SUCCESS/FAILURE, results) ==>                              2d2a1
   RETURN (PERMANENT, SUCCESS/FAILURE, results)                  2d2a1a
```

Terminate a procedure abnormally 2d2b

```
ABORT (errcode, errmsg) ==>                                      2d2b1
   RETURN (PERMANENT, ABORTED, LIST (errcode, errmsg))           2d2b1a
```

Summon assistance from a caller 2d2c

```
HELP (problem, parm => args) ==>                                 2d2c1
   RETURN (TEMPORARY, HELP, LIST (problem, parm) => args)        2d2c1a
```

Report event to caller 2d2d

```
NOTE (event, parm) ==>                                           2d2d1
   RETURN (TEMPORARY, NOTE, LIST (event, parm))                  2d2d1a
```

PROCEDURES 3

Call remote procedure 3a

CALPRO (ph, pkh, pname, args, arglmsk, reslmsk, mode, cmplevnt
=> type, subtype, results, ch) 3a1

This procedure calls with arguments ARGS, remote procedure PNAME in the process and package known to the local process via PH and PKH, respectively. If MODE specifies that the remote procedure is to be executed out-of-line, CMPLEVNT is the name of the local event to be signalled by PIP when the callee returns. 3a2

Process (PHs) and package (PKHs) handles are constructs of the Process Management and PCP Support Packages, respectively, described in other documents. When PIP is used outside of the higher-level framework provided by these packages, PH and PKH should be set to zero. 3a2a

ARGLMSK, RESLMSK, TYPE, SUBTYPE, RESULTS, and CH denote, respectively, the argument and result list masks for the call, the type and subtype of return made by the callee, the results it returns, and the call handle (EMPTY for permanent returns) via which the procedure call will be known to the local, invoking processor; all of which are described in the PCP document. 3a3

Argument/result types: 3a4

```
ph       - INTEGER                                                  3a4a
pkh      - INTEGER                                                  3a4b
pname    - CHARSTR                                                  3a4c
args     - LIST                                                     3a4d
arglmsk  - LIST (INTEGER [FROMCALLER=0] / %from data store%
             DSELECTOR*, ...)                                       3a4e
reslmsk  - LIST (INTEGER [TOCALLER=0 / DISCARD=1] / %to data
             store% DSELECTOR*, ...)                                3a4f
mode     - BOOLEAN [INLINE=TRUE / OUTOFLINE=FALSE]                  3a4g
cmplevnt- CHARSTR / EMPTY                                           3a4h
type     - INTEGER [PERMANENT=0 / TEMPORARY=1 / INPROGRESS=2]       3a4i
subtype  - %permanent% INTEGER [SUCCESS=0 / FAILURE=1 /
             ABORTED=2] /
           %temporary% INTEGER [GENERALCOROUTINE=0 / NOTE=1 /
             HELP=2 / INTERRUPTED=3] /
           %inprogress% EMPTY                                       3a4j
```

```
      results - %success or failure or general coroutine% any /
                %aborted% LIST (%errcode% INTEGER, %errmsg%
                  CHARSTR / EMPTY) /
                %note or help% LIST (%event or problem% INTEGER,
                  %parameter% any) /
                %interrupted or inprogress% EMPTY                    3a4k
      ch      - INTEGER / EMPTY                                      3a41
```

Resume remote procedure 3b

RSMPRO (ch, args, mode, cmplevnt => type, subtype, results) 3b1

This procedure resumes with new arguments ARGS, the remote procedure known to the local, invoking processor via CH, which previously made a temporary return. If MODE specifies that the remote procedure is to be resumed out-of-line, CMPLEVNT is the name of the local event to be signalled by PIP when the callee next returns. 3b2

TYPE, SUBTYPE, and RESULTS denote, respectively, the type and subtype of return made by the callee, and the results it returns; all of which are described in the PCP document. 3b3

Argument/result types: 3b4

```
      ch      - INTEGER                                              3b4a
      args    - LIST                                                 3b4b
      mode    - BOOLEAN [INLINE=TRUE / OUTOFLINE=FALSE]              3b4c
      cmplevnt- CHARSTR / EMPTY                                      3b4d
      type    - INTEGER [PERMANENT=0 / TEMPORARY=1 / INPROGRESS=2]   3b4e
      subtype - %permanent% INTEGER [SUCCESS=0 / FAILURE=1 /
                  ABORTED=2] /
                %temporary% INTEGER [GENERALCOROUTINE=0 / NOTE=1 /
                  HELP=2 / INTERRUPTED=3] /
                %inprogress% EMPTY                                   3b4f
      results - %success or failure or general coroutine% any /
                %aborted% LIST (%errcode% INTEGER, %errmsg%
                  CHARSTR / EMPTY) /
                %note or help% LIST (%event or problem% INTEGER,
                  %parameter% any) /
                %interrupted or inprogress% EMPTY                    3b4g
```

Interrupt remote procedure 3c

INTPRO (ch) 3c1

This procedure interrupts the previously called or resumed remote procedure known to the local, invoking processor via CH. 3c2

Argument/result types: 3c3

ch- INTEGER 3c3a

Abort remote procedure 3d

ABRPRO (ch) 3d1

This procedure aborts the remote procedure known to the local, invoking processor via CH, which must have previously made a temporary return. 3d2

Argument/result types: 3d3

ch- INTEGER 3d3a

DATA STORES 4

CHLIST List of outstanding remote procedure calls 4a

This read-only data store specifies the call handle CH for and status STATUS of each currently outstanding procedure call originating from the local process. 4b

Data structure type: 4c

<chlist> LIST (<%ch% INTEGER> %status% BOOLEAN [INPROGRESS=TRUE / AWAITINGRESUME=FALSE], ...) 4c1

The PCP Support Package
PSP Version 2

22-NOV-74

James E. White
Augmentation Research Center

Stanford Research Institute
Menlo Park, California 94025

PSP is a procedure call support tool that operates within the setting provided by the Procedure Call Protocol (PCP -- 24459,), with which the reader of the present document is assumed familiar.

(J24461) 22-NOV-74 14:52;;;; Title: Author(s): James E. (Jim)
White/JEW; Sub-Collections: SRI-ARC; Clerk: JEW; Origin: <
WHITE, PCP-PSP,NLS;7, >, 22-NOV-74 13:07 JEW ;;;; ####;

PREFACE 1

The Procedure Call Protocol (PCP) is an inter-process and/or inter-host protocol that permits a collection of processes within one or more ARPANET hosts to communicate at the procedure call level. In effect, it makes the component procedures of remote software systems as accessible to the programmer as those within his own system. PCP specifies both a virtual programming environment (VPE) in which remote procedures may be assumed to operate, as well as the inter-process exchanges that implement it. 1a

The Multi-Process Software System (MPSS) whose construction PCP makes practical and of which the NSW is an example, consists of collections of "procedures" and "data stores" called "packages", in one or more "processes", interconnected in a tree structure by "physical channels". Procedures within a process have free access to the procedures (and data stores) of each process adjacent to it in the tree structure, and may call upon them as if they were local subroutines. Superimposed upon the tree structure is a more general set of interconnections which give non-adjacent processes in the tree the same kind of access to one another. 1b

The MPSS is implemented by: 1c

1) low-level protocols which provide the basic, inter-process communicaton (IPC) facilities by which channels are implemented: an inter-host IPC protocol (PCPHST), an inter-Tenex-fork IPC protocol (PCPFRK), and data structure format specifications for both connection types (PCPFMT). 1c1

2) PCP proper, which largely defines the VPE (especially, the procedure call and return mechanism) and specifies the inter-process control exchanges required to implement it. 1c2

3) a set of system packages, implemented within each process, which augment PCP proper by providing mechanisms by which user procedures can: call remote procedures (implemented by the Procedure Interface Package, PIP), manipulate remote data stores (implemented by the PCP Support Package, PSP), and interconnect processes (implemented by the Process Management Package, PMP). 1c3

4) user packages in each process. 1c4

INTRODUCTION 2

The PCP Support Package (package name=PSP) provides remote processes with the necessary tools for using the local process conveniently. PSP includes procedures for obtaining access to groups of local procedures and data stores, manipulating local data stores, creating temporary ones, and a NOP. 2a

Packages 2b

The procedures and data stores within a process are partitioned by function into one or more "packages". Packages are referred to initially (in the OPNPKS procedure) by name, and thereafter via a "package handle", or PKH. The (entire) contents of a package are accessible to a remote process if and only if that particular process has successfully "opened" the package and thus obtained a PKH for it. 2b1

NOTE: The PSP itself is always considered open (with PKH=0) and need not, indeed cannot, be explicitly opened or closed. 2b1a

A Package Programmer's Guide (PPG), like the current document, is assumed to be available for each package implemented by a process. 2b2

PROCEDURES 3

Open packages 3a

OPNPKS (pknames => pkhs) 3a1

This procedure opens the local process' packages PKNAMES, and makes them known via the handles PKHS, and thus their contents accessible, to the invoking process. 3a2

Argument/result types: 3a3

pknames= LIST (%pkname% CHARSTR, ...) 3a3a
pkhs = LIST (%pkh% INTEGER, ...) 3a3b

Close packages 3b

CLSPKS (pkhs => costs) 3b1

This procedure closes those of the local process' packages known to the invoking process via the handles PKHS, and thus makes their contents inaccessible to the invoking process. The cost COSTS in cents of each package's use is returned. 3b2

Argument/result types: 3b3

pkhs = LIST (%pkh% INTEGER, ...) 3b3a
costs= LIST (%cents% INTEGER, ...) 3b3b

Read data store 3c

RDDATA (datastore => value) 3c1

This procedure returns the value VALUE of a specified element DATASTORE of a data store in one of the local process' packages known to the invoking process. 3c2

Argument/result types: 3c3

datastore= DSELECTOR* 3c3a
value = any 3c3b

Write data store 3d

WRDATA (datastore, value) 3d1

This procedure assigns a new value VALUE to a specified element DATASTORE of a data store in one of the local process' packages known to the invoking process. 3d2

Argument/result types: 3d3

datastore- DSELECTOR* 3d3a
value - any 3d3b

Create temporary data store 3e

CRTTMP (tmpname, value) 3e1

This procedure creates a temporary data store TMPNAME with initial value VALUE, in the local process' PSP. Once created, the temporary data store can be used and manipulated like any other data store in the local process. Unlike other data stores, however, it is known only to the process that created it. 3e2

Argument/result types: 3e3

tmpname- CHARSTR 3e3a
value - any 3e3b

Delete temporary data store 3f

DELTMP (tmpname) 3f1

This procedure deletes from the local process' PSP, the temporary data store TMPNAME previously created by the invoking process via CRTTMP. 3f2

Argument/result types: 3f3

tmpname- CHARSTR 3f3a

No operation 3g

NOP (argument => argument) 3g1

This procedure is a NOP, simply echoing its argument as its result. It can be called remotely to verify the channel to, and proper functioning of the local process. 3g2

Argument/result types: 3g3

argument= any 3g3a

DATA STORES 4

PKS List of offered packages 4a

This read-only, reader-dependent data store contains a list of the packages PKNAMEs implemented by the local process and available to the invoking process. 4a1

Data structure type: 4a2

<pks> LIST (<pkname> EMPTY, ...) 4a2a

The Process Management Package
PMP Version 2

22-NOV-74

James E. White
Augmentation Research Center

Stanford Research Institute
Menlo Park, California 94025

PMP is an organizational tool that operates within the setting provided by the Procedure Call Protocol (PCP -- 24459,), with which the reader of the present document is assumed familiar.

JEW 22 NOV 74 7:46PM The Process Management Package

(J24462) 22-NOV-74 15:47;;;; Title: Author(s): James E. (Jim) White/JEW; Sub-Collections: SRI-ARC; Clerk: JEW; Origin: < WHITE, PCP-PMP,NLS;30, >, 22-NOV-74 13:17 JEW ;;;; ####;

PREFACE 1

The Procedure Call Protocol (PCP) is an inter-process and/or inter-host protocol that permits a collection of processes within one or more ARPANET hosts to communicate at the procedure call level. In effect, it makes the component procedures of remote software systems as accessible to the programmer as those within his own system. PCP specifies both a virtual programming environment (VPE) in which remote procedures may be assumed to operate, as well as the inter-process exchanges that implement it. 1a

The Multi-Process Software System (MPSS) whose construction PCP makes practical and of which the NSW is an example, consists of collections of "procedures" and "data stores" called "packages", in one or more "processes", interconnected in a tree structure by "physical channels". Procedures within a process have free access to the procedures (and data stores) of each process adjacent to it in the tree structure, and may call upon them as if they were local subroutines. Superimposed upon the tree structure is a more general set of interconnections which give non-adjacent processes in the tree the same kind of access to one another. 1b

The MPSS is implemented by: 1c

1) low-level protocols which provide the basic, inter-process communicaton (IPC) facilities by which channels are implemented: an inter-host IPC protocol (PCPHST), an inter-Tenex-fork IPC protocol (PCPFRK), and data structure format specifications for both connection types (PCPFMT). 1c1

2) PCP proper, which largely defines the VPE (especially, the procedure call and return mechanism) and specifies the inter-process control exchanges required to implement it. 1c2

3) a set of system packages, implemented within each process, which augment PCP proper by providing mechanisms by which user procedures can: call remote procedures (implemented by the Procedure Interface Package, PIP), manipulate remote data stores (implemented by the PCP Support Package, PSP), and interconnect processes (implemented by the Process Management Package, PMP). 1c3

4) user packages in each process. 1c4

INTRODUCTION 2

The Multi-Process Software System 2a

PCP makes practical the construction of large, Network-based, distributed software systems, each module implemented on a different machine, with a different operating system, in a different programming language. The Process Management Package (package name = PMP) provides the necessary tools for interconnecting two or more processes to form a coherent, Multi-Process Software System (MPSS). PMP contains those facilities required to construct, manipulate, and dismantle the MPSS: procedures for creating, deleting, logically and physically interconnecting processes, and allocating and releasing processes. 2a1

The Process Tree 2b

Let MPSS be, at any point in time, a tree structure of processes. Every MPSS begins with a single, pre-existent "root" process to which all other processes are ultimately subordinate. During the course of its execution, the root process creates one or more subordinate processes, one or more of which may create subordinate processes of its own, and so forth. 2b1

A process is said to be the "direct inferior" of the process that created it, and the "indirect inferior" of each process further up in the process tree. A process is said to be the "direct superior" of each process it creates, and the "indirect superior" of each process further down in the tree. A process may have, at any point in time, an arbitrary number of directly inferior processes, but (of course) only one directly superior process. 2b2

Process Addresses 2c

An existing process creates a direct inferior by offering its "process address" to the CRTPRC procedure. Process addresses which begin with a dollar sign ('$) are reserved for MPSS-wide assignment; all others are available for local assignment. The following universal process addresses are currently defined: 2c1

1) "$F" <SP> host <SP> filename 2c1a

This process address designates a process created from the executable file FILENAME in the file system of ARPANET host HOST, and run at that host. FILENAME is assumed to be an executable file, and the process which results from it one that supports PCP. HOST is a standard host name or decimal host address, and defaults to that of the local process, if "<SP> host" is absent. 2c1a1

2) "$N" <SP> host <SP> socket 2c1b

This process address designates a process created by and run at ARPANET host HOST, in response to an ICP to contact socket SOCKET (specified in decimal). The process so created is assumed to support PCP. 2c1b1

Process Names 2d

Every process has, in addition to an address, a generic "process name", which is returned by the CRTPRC procedure and which implies the set of packages (see the PCP Support Package) which the process supports. The combination of a process' name and address is called its "ident", and is designated throughout this document by the following shorthand: 2d1

PRCIDENT* ==> LIST (%prcname% CHARSTR, %prcaddr% CHARSTR) 2d1a

Known Processes 2e

Once it has been created, a process is referenced within the MPSS by means of a "process handle", or PH. A PH is a local handle to a process, and each process that wishes to reference a process must first obtain its own PH for it. A process B is said to be "known" to another process A if and only if A has a handle to B. A process may call procedures in any process known to it. PMP prevents A from obtaining a handle to B without, at the same time, obtaining for B a handle to A. 2e1

A process' direct superior is always known to it via a special PH whose value is SUPER=-1. A PH is assigned to each direct inferior at its creation. A process is always known to itself via the special PH whose value is SELF=0. 2e1a

Inter-Connecting Processes 2f

Logical Channels 2f1

The tree structure that naturally results from the process of creating inferiors is not sufficient, in general, to describe the inter-process relationships that must exist in many real MPSSs. It may be required, for example, that two processes I1 and I2 with the same direct superior S be able to call each other's procedures, i.e. that I1 and I2 be known to each other as well as to S. Therefore, PMP permits any two processes in an MPSS to be made known or "introduced" to each other. 2f1a

Any two processes (I1 and I2, in the example above) can be introduced to each other by any process that knows them both (S). The introduction involves the assignment of process handles by which the two processes can refer to one another, and the establishment between them of a "logical channel" for PCP messages. The logical channel requires the continued support of the introducing process (S), which relays PCP messages between I1 and I2. 2f1b

Introductions can be "cascaded". That is, once I1 and I2 have been introduced, I1 (for example) can in turn introduce I2 to any other process it knows (e.g. one of ITS direct inferiors). The logical channel that results from cascaded introductions involves two or more intermediate processes, all of which are involved in relaying PCP messages between the introduced processes. 2f1c

A logical channel consists, conceptually, of two "ends" connected by one or more "segments", each end being in one of the introduced processes, and each segment in one of the intermediate processes which comprise the logical channel. Each logical channel is known to the process which creates it via an "introduction handle" (IH), and each end and segment (in priviledged PMP procedures) is known via a "segment handle" (SH) to the process that contains it. The segment handle for an end doubles as the process handle by which the process at the other end is known to the local process. 2f1d

PMP uses the PH-field of PCP messages to support its relaying activity. Whenever it is sent a message whose PH-field is other than SELF, a process looks up PH in an internal table containing the handles of directly superior and inferior processes and local ends and segments, extracts from the table the handles of the next two processes along the logical channel, stores the handle of the second in the PH-field of the messsage, and sends the message to the first. 2f1e

Physical Channels 2f2

Because logical channels can become long, PMP permits a physical channel (like that which connects a process and its direct superior) to be established when the two processes are introduced, and "associated" with the logical one. 2f2a

> Physical channels may also be used for purposes other than carrying PCP messages. The File package for example, described in another document, uses them to transmit files between processes. 2f2a1

The association of a physical channel with a logical one simply reduces the overhead of PCP message transmission; it gains for the connected processes no additional capabilities. 2f2b

Each process that supports PMP provides to remote processes by means of its PRCLOC and CHNTYPMNU data stores, information about its physical location and the kinds of physical channels it supports. The pseudo data types CHNTYPSEL*, PORT*, and PRCLOC* will be used throughout the current document to designate, respectively, a type of physical channel, a physical port, and a process' physical

location. The formats of these parameters and the concepts they represent are discussed in the PCP document. 2f2c

Configuring the MPSS 2g

An MPSS is in one sense, simply a collection of procedures and data stores partitioned among some arbitrary number of processes. Since PMP (with the aid of the PCP Support and Procedure Interface Packages, described in other documents) provides mechanisms by which procedures and data stores can be accessed no matter what their location, the system's components could, in principle, be arbitrarily partitioned among processes. 2g1

In practice, however, the programmer must assume that calling a remote procedure or manipulating a remote data store is considerably more expensive, in terms of both the real and processing time required, than calling or manipulating a local one, an operation which may be as inexpensive as a single machine instruction. The components of the system must therefore be partitioned among processes with intelligence and care. 2g2

PROCEDURES 3

Process Manipulation 3a

Create process 3a1

CRTPRC (prcaddr => ph, prcname) 3a1a

This procedure attaches as a direct inferior of the local process via a physical channel, an instance of the process whose address is PRCADDR, which it makes known to the local process via PH. The newly-created process, whose generic name PRCNAME is returned and which is assumed to support PCP, is initialized and assigned a handle to the local process, its direct superior. 3a1b

Argument/result types: 3a1c

prcaddr- CHARSTR 3a1c1
ph - INTEGER 3a1c2
prcname- CHARSTR 3a1c3

Delete process 3a2

DELPRC (Ph => cost) 3a2a

This procedure terminates, detaches from the local process (deleting the physical channel), and discards the previously-created process known to the local process via PH. PH and, of course, the handle via which the local process was known to the deleted process, are invalidated. COST is the cost in cents of the process' use. 3a2b

Argument/result types: 3a2c

ph - INTEGER 3a2c1
cost- INTEGER 3a2c2

Introduce processes 3a3

ITDPRCS (ph1, ph2, direct => ph12, ph21, ih) 3a3a

This procedure introduces to each other, the two processes known to the local process via PH1 and PH2, making the second known to the first via PH12, and the first to the second via PH21. If DIRECT is TRUE, the procedure establishes a physical channel between the two processes; otherwise, PCP messages will be relayed via the logical channel which connects them. The procedure returns a local "introduction handle" IH by which the introduction becomes known to the local process. 3a3b

Since PH12 and PH21 are known, initially, only to the invoking process, it must, in subsequent exchanges with them, notify the newly-connected processes that they have been introduced, and pass each its PH for the other. 3a3c

Implementation: 3a3d

```
%allocate and initialize segment table entry%                3a3d1
   allocate seg sh = SEG (empty, PH1, PH2, empty)            3a3d1a
%identify the two processes -- in parallel for speed%        3a3d2
   @ PH1: RDDATA (PRCIDENT => prcident1)                     3a3d2a
   @ PH2: RDDATA (PRCIDENT => prcident2)                     3a3d2b
%create a logical-channel end in process 1%                  3a3d3
   if find end PH1 = END (locphA, remshA, prcidentA)         3a3d3a
      then @ locphA:                                         3a3d3a1
         DUPLOGCHNHLF! (prcident2, remshA, sh =>             3a3d3a1a
            sh(1), PH12)                                     3a3d3a1a1
      else @ PH1:                                            3a3d3a2
         CRTLOGCHNEND! (prcident2, sh =>                     3a3d3a2a
            sh(1)) and PH12 _ sh(1)                          3a3d3a2a1
%create a logical-channel end in process 2
 -- in parallel with same for process 1 above, for speed%   3a3d4
   if find end PH2 = END (locphB, remshB, prcidentB)         3a3d4a
      then @ locphB:                                         3a3d4a1
         DUPLOGCHNHLF! (prcident1, remshB, sh =>             3a3d4a1a
            sh(4), PH21)                                     3a3d4a1a1
      else @ PH2:                                            3a3d4a2
         CRTLOGCHNEND! (prcident1, sh =>                     3a3d4a2a
            sh(4)) and PH21 _ sh(4)                          3a3d4a2a1
%allocate introduction table entry%                          3a3d5
   allocate intro IH = INTRO (PH12, PH21, EMPTY, sh)         3a3d5a
%create and associate physical with logical channel%         3a3d6
```

```
        if DIRECT = TRUE then                                          3a3d6a
           CRTPHYCHN (PH1, PH2 => poh1, poh2, IH(3))                   3a3d6a1
           %in parallel for speed%                                     3a3d6a2
              @ PH1: ASOPHYCHN! (poh1, PH12)                           3a3d6a2a
              @ PH2: ASOPHYCHN! (poh2, PH21)                           3a3d6a2b

     Argument/result types:                                            3a3e

        ph1    = INTEGER                                               3a3e1
        ph2    = INTEGER                                               3a3e2
        direct= BOOLEAN                                                3a3e3
        ph12   = INTEGER                                               3a3e4
        ph21   = INTEGER                                               3a3e5
        ih     = INTEGER                                               3a3e6

  Separate processes                                                   3a4

     SEPPRCS (ih)                                                      3a4a

     This procedure separates two processes whose previous
     introduction is known to the local process via IH, by
     invalidating the process handles assigned by ITDPRCS and
     deleting the physical channel (if any) between them.              3a4b

     Implementation:                                                   3a4c

        %locate introduction table entry%                              3a4c1
           find intro IH = INTRO (ph12, ph21, pch, sh)                 3a4c1a
        %delete physical channel (if any)%                             3a4c2
           if pch not = EMPTY then DELPHYCHN! (pch)                    3a4c2a
        %locate segment table entry%                                   3a4c3
           find seg sh = SEG (remsh1, locph1, locph2, remsh2)          3a4c3a
        %delete logical-channel end in each process
         -- in parallel for speed%                                     3a4c4
           @ locph1: DELLOGCHNHLF! (remsh1)                            3a4c4a
           @ locph2: DELLOGCHNHLF! (remsh2)                            3a4c4b
        %release segment and introduction table entries%               3a4c5
           release seg sh                                              3a4c5a
           release intro IH                                            3a4c5b

     Argument/result types:                                            3a4d

        ih= INTEGER                                                    3a4d1
```

Allocate processors 3a5

ALOPCRS (count) 3a5a

This procedure allocates COUNT (more) local processors to the invoking process, increasing by that number the number of procedures the invoking process is guaranteed to be able to set in simultaneous execution within the local process. 3a5b

Argument/result types: 3a5c

count- INTEGER 3a5c1

Release processors 3a6

RELPCRS (count) 3a6a

This procedure releases COUNT of the local processors currently allocated to the invoking process, decreasing by that number the number of procedures the invoking process is guaranteed to be able to set in simultaneous execution within the local process. 3a6b

Argument/result types: 3a6c

count- INTEGER 3a6c1

Physical Channels 3b

Create physical channel 3b1

CRTPHYCHN (ph1, ph2 => poh1, poh2, pch) 3b1a

This procedure creates a physical channel, which it makes known to the local process via PCH, between the processes known to the local process via PH1 and PH2, using the ports known to the two processes via POH1 and POH2, respectively. 3b1b

CRTPHYCHN is provided simply as a convenience to the programmer; it can be implemented using other, lower-level PMP procedures. 3b1c

Implementation: 3b1d

```
%allocate and initialize physical-channel table entry%      3b1d1
   allocate pchan PCH = PCHAN (PH1, EMPTY, PH2, EMPTY)      3b1d1a
%fetch locations of two processes -- in parallel for
speed%                                                      3b1d2
   @ PH1: RDDATA (PRCLOC => prcloc1)                        3b1d2a
   @ PH2: RDDATA (PRCLOC => prcloc2)                        3b1d2b
%negotiate channel type with process 2 and allocate a
port%                                                       3b1d3
   @ PH1: RDDATA (CHNTYPMNU => chntypmnu1)                  3b1d3a
   @ PH2:                                                   3b1d3b
      ALOPHYPOR (chntypmnu1, prcloc1 =>                     3b1d3b1
         chntypsel2, port2, POH2)                           3b1d3b1a
   PCH(4) _ POH2                                            3b1d3c
%allocate a corresponding port in process 1%                3b1d4
   @ PH1:                                                   3b1d4a
      ALOPHYPOR (chntypsel2, prcloc2 =>                     3b1d4a1
         chntypsel1, port1, POH1)                           3b1d4a1a
   PCH(2) _ POH1                                            3b1d4b
%burrow a channel from both ends
 -- in parallel of necessity%                               3b1d5
   @ PH1: CRTPHYCHNEND (poh1, port2)                        3b1d5a
   @ PH2: CRTPHYCHNEND (poh2, port1)                        3b1d5b
```

Argument/result types: 3b1e

```
ph1 = INTEGER                                               3b1e1
ph2 = INTEGER                                               3b1e2
poh1= INTEGER                                               3b1e3
poh2= INTEGER                                               3b1e4
```

```
pch = INTEGER                                                    3b1e5
```

Delete physical channel 3b2

DELPHYCHN (pch) 3b2a

This procedure deletes the previously-created physical channel known to the local process via PCH, and is priviledged if the physical channel is associated with a logical one. 3b2b

DELPHYCHN is provided simply as a convenience to the programmer; it can be implemented using other, lower-level PMP procedures. 3b2c

Implementation: 3b2d

```
%locate physical-channel table entry%                    3b2d1
   find pchan PCH = PCHAN (ph1, poh1, ph2, poh2)         3b2d1a
%dissolve channel from both ends%                        3b2d2
   @ PH1: DELPHYCHNEND (poh1)                            3b2d2a
   @ PH2: DELPHYCHNEND (poh2)                            3b2d2b
%release physical-channel table entry%                   3b2d3
   release pchan PCH                                     3b2d3a
```

Priviledged calls to DELPHYCHNEND are made if and only if the call to DELPHYCHN is priviledged. 3b2d4

Argument/result types: 3b2e

```
pch- INTEGER                                             3b2e1
```

Physical Channel Support 3c

Create end of physical channel 3c1

CRTPHYCHNEND (poh, remport) 3c1a

This procedure creates one end of (i.e. helps to create) a physical channel of the type specified when the local port was allocated, between the port known to the local process via PCH, and remote port REMPORT. 3c1b

Argument/result types: 3c1c

```
poh    - INTEGER                                         3c1c1
remport- PORT*                                           3c1c2
```

Delete end of physical channel 3c2

DELPHYCHNEND (poh) 3c2a

This procedure deletes one end of (i.e. helps to delete) the previously-created physical channel attached to the port known to the local process via POH, and is priviledged if the physical channel is associated with a logical one. 3c2b

Argument/result 3c2c

poh- INTEGER 3c2c1

Allocate physical port 3c3

ALOPHYPOR (chntypmnu, remloc => chntypsel, port, poh) 3c3a

This procedure allocates a port PORT within the local process and known to it via POH, for subsequent use in establishing a physical channel of one of the types CHNTYPSEL permitted by CHNTYPMNU, between the local process, and the remote process at location REMLOC. 3c3b

Argument/result types: 3c3c

chntypmnu- LIST (CHNTYPSEL*, ...) 3c3c1
remloc - PRCLOC* 3c3c2
chntypsel- CHNTYPSEL* 3c3c3
port - PORT* 3c3c4
poh - INTEGER 3c3c5

Release physical port 3c4

RELPHYPOR (poh) 3c4a

This procedure releases the previously-allocated port known to the local process via POH. 3c4b

Argument/result types: 3c4c

poh- INTEGER 3c4c1

Logical Channel Support 3d

Associate physical with logical channel 3d1

ASOPHYCHN! (poh, ph) 3d1a

This priviledged procedure associates with the logical channel between the local process and that known to the local process via PH, the physical channel attached to the port known to the local process via POH. 3d1b

Argument/result types: 3d1c

poh- INTEGER 3d1c1
ph - INTEGER 3d1c2

Disassociate physical from logical channel 3d2

DSOPHYCHN! (ph) 3d2a

This priviledged procedure dissociates the previously-associated physical channel from the logical channel between the local process and that known to the local process via PH. 3d2b

Argument/result types: 3d2c

ph - INTEGER 3d2c1

Create end of logical channel 3d3

CRTLOGCHNEND! (prcident, chnsh => sh) 3d3a

This priviledged procedure creates a logical channel end with handle SH in the local process, by virtue of which process PRCIDENT will know the local process, and chains it to the logical channel segment with handle CHNSH in the invoking process. 3d3b

Implementation: 3d3c

%allocate and initialize end table entry% 3d3c1
allocate end SH = END (invoker, CHNSH, PRCIDENT) 3d3c1a

Argument/result types: 3d3d

prcident= PRCIDENT* 3d3d1
chnsh = INTEGER 3d3d2
sh = INTEGER 3d3d3

Delete half of logical channel 3d4

DELLOGCHNHLF! (sh) 3d4a

This priviledged procedure deletes the logical channel segment/end with handle SH in the local process. If SH designates a segment, the delete request is propagated to the process to which the segment is chained, effectively deleting the rest of the logical channel in one direction. 3d4b

Implementation: 3d4c

%locate segment/end table entry% 3d4c1
find seg/end 3d4c1a
SH = 3d4c1a1
SEG (remshA, locphA, invoker, remshB) or 3d4c1a1a
END (invoker, remshA, prcidentA) 3d4c1a1b
%propagate delete request% 3d4c2
if SH is a SEG then @ locphA: DELLOGCHNHLF! (remshA) 3d4c2a
%delete segment/end table entry% 3d4c3
release seg/end SH 3d4c3a

Argument/result types: 3d4d

sh= INTEGER 3d4d1

Duplicate half of logical channel 3d5

DUPLOGCHNHLF! (prcident, oldsh, chnsh => newsh, ph) 3d5a

This priviledged procedure creates a logical channel segment/end with handle NEWSH in the local process, like the existing segment/end with handle OLDSH in the local process. The copy is chained to the same process(es) as the original, one of which is the invoking process, but to different segments in those processes. The copy is chained on one side to the segment/end with handle CHNSH in the invoking process, and on the other to a segment/end whose handle is to be obtained by calling DUPLOGCHNHLF in the segment's process. The process of obtaining this latter handle, of course, effectively duplicates the rest of the logical channel in one direction, and thus gives process PRCIDENT access to the process at the end. 3d5b

Implementation: 3d5c

```
%lookup original segment/end table entry%                    3d5c1
   find seg/end                                              3d5c1a
      OLDSH =                                                3d5c1a1
         SEG (remshA, locphA, invoker, remshB) or            3d5c1a1a
         END (invoker, remshA, prcidentA)                    3d5c1a1b
%duplicate segment/end%                                      3d5c2
   if OLDSH is an END                                        3d5c2a
      then                                                   3d5c2a1
         allocate end                                        3d5c2a1a
            NEWSH = END (invoker, CHNSH, PRCIDENT)           3d5c2a1a1
         PH _ NEWSH                                          3d5c2a1b
      else                                                   3d5c2a2
         allocate seg                                        3d5c2a2a
            NEWSH = SEG (empty, locphA, invoker, CHNSH)
                                                             3d5c2a2a1
         @ locphA:                                           3d5c2a2b
            DUPLOGCHNHLF! (PRCIDENT, remshA, NEWSH =>        3d5c2a2b1
               NEWSH(1), PH)                                 3d5c2a2b1a
```

Argument/result types: 3d5d

```
prcident- PRCIDENT*                                          3d5d1
oldsh     - INTEGER                                          3d5d2
chnsh     - INTEGER                                          3d5d3
newsh     - INTEGER                                          3d5d4
ph        - INTEGER                                          3d5d5
```

DATA STORES 4

PRCIDENT Ident of local process 4a

This read-only data store contains the ident of the local process. 4a1

Data structure type: 4a2

<prcident> LIST (<prcname> CHARSTR, <prcaddr> CHARSTR) 4a2a

PRCLOC Local process's physical location 4b

This read-only data store specifies the local process' physical location. 4b1

Data structure type: 4b2

<prcloc> PRCLOC* 4b2a

CHNTYPMNU Physical channel types supported by local process 4c

This read-only data store specifies the types of physical channels which the local process supports. 4c1

Data structure type: 4c2

<chntypmnu> LIST (CHNTYPSEL*, ...) 4c2a

FREEPCRS Count of unallocated local processors 4d

This read-only data store contains the number of local processors which are currently unallocated. 4d1

Data structure type: 4d2

<freepcrs> INTEGER 4d2a

ALOCPCRS Local processor allocations 4e

This read-only data store lists the number COUNT of processors currently allocated to each process known to the local process (via PH). 4e1

Data structure type: 4e2

<alocpcrs> LIST (<%ph% INTEGER> %count% INTEGER, ...) 4e2a

KEV 11-NOV-74 11:10 24463

source debugging

we (dave maynard probably) will put a psid or zero into 400002, and the address of an astring in 400001. i suggest the astring be of the following format:

<dirname>filename.extname;version

i suggest the compiler append to this string either the compile time, or the write time of the second rel file.

source debugging

(J24463) 11-NOV-74 11:10;;;; Title: Author(s): Kenneth E. (Ken) Victor/KEV; Distribution: /DIA([ACTION]) DSM([ACTION]) CHI([INFO-ONLY]) ; Sub-Collections: SRI-ARC; Clerk: KEV;

DCE 11-NOV-74 14:31 24464

Digital-speech I/O demo available at SRI

See JML, she'll make appointment for group of us. Dick and Jim:
notify any you think might be interested.

DCE 11-NOV-74 14:31 24464

Digital-speech I/O demo available at SRI

During an extended discussion in Washington with Carl Herold of SRI (Who is in Phil Whalen's Dept.), I learned of some work going on at SRI with Vocoders -- being tested for operational characteristics. Seems that some equipment is temporarily at SRI that can demonstrate speech throughput that has been squeezed optionally through either 2400 or 4800-baud digital. 1

Had a call this morning from John Van Geen, ext 3234; Carl told him of my interest, and he offers to demonstrate the equipment. Says it may not be here much beyond this week. I'll have Jeanne Leavitt schedule a half hour; waiting until Tuesday to make the appointment in case any of you would like to be included and have special timing requests. 2

DCE 11-NOV-74 14:31 24464

Digital-speech I/O demo available at SRI

(J24464) 11-NOV-74 14:31;;;; Title: Author(s): Douglas C. Engelbart/DCE; Distribution: /RWW([ACTION]) CHI([ACTION]) JML([ACTION]) JCN([INFO-ONLY]) ; Sub-Collections: SRI-ARC; Clerk: DCE;

JAKE 11-NOV-74 15:00 24465

Journal delivery problems

I can't seem to get the right combination of useroptions or what have you to get my journal mail delivered without a header for each delivery. Also, messages and journal items that I have sent recently are not showing up in my author file. CHI looked at this a couple of days ago and said it was OK but it doesn't appear to be. Can someone either tell me what I am doing wrong or let me know if the system has a bug. Thanks. Jake 1

JAKE 11-NOV-74 15:00 24465

Journal delivery problems

(J24465) 11-NOV-74 15:00;;;; Title: Author(s): Elizabeth J. (Jake) Feinler/JAKE; Distribution: /FEEDBACK([ACTION]) JDH([INFO-ONLY]) ; Sub-Collections: SRI-ARC FEEDBACK; Clerk: JAKE;

EKM 11-NOV-74 15:42 24466

Tape Format

We will call Elizabeth and speck with her about the file.
Documentation of the specially modified insert sequential will be
sent to relevant ARC programmers in the Journal.

EKM 11-NOV-74 15:42 24466

Tape Format

We have created a new file in directory Riddle called ARPATAPE.NLS at OFFICE-1. This we created quite easily when we discovered what was in the tape you sent us. It seems that every line ends with a carriage return followed by a variable number of spaces followed by a carriage return followed by a line feed. What looked on the print out like double carriage returns at the end of paragraphs were actually line lenths full of spaces followed by carriage return, spaces, carriage return, line feed. 1

The file we created looks as though it conforms pretty well to the structural specifications Bill Carlson sent us except for words which were hyphenated on the original tape and for statements that had an incorrect number of blanks at the beginning of a line. 2

We did not do anything about hyphenation or incorrect levels because we felt you would like to see the first results; we can help you fix these things while linked to you if you wish. 2a

We noticed that the original text file from the tape occupied 82 Tenex pages where the NLS file only took 62 pages. 3

The hyphenation problem can be simply cured by doing a substitute over the entire file of a NULL character (Control-N) for "- ". Don't forget the space after the hyphen. 4

There are a couple of places where the structure isn't right because there were the wrong number of spaces at the beginning of the statement. These can easily be moved down or up as necessary. We used the following structuring algorithm: 5

The level of a statement was determined by the number of leading spaces in its first line-- no spaces for highest level, one space for second level, etc. (Semantic examination of text could also be used, but this was more efficient and the "errors" are easily fixed by hand online in NLS.) 5a

We would like to know if you plan to send more tapes in the same format. If you do we can formalize and write up the procedure for creating a structured NLS file. The whole thing only takes a few seconds to execute when the procedure is known. 6

Sorry the job was not done to your wishes originally, but there are so many possibilities for both input and output that without detailed specification of the input file format (which was not at all obvious to us without a character by character examination since printouts did not show the CR followed by spaces followed by CRLF) and without detailed specification of the desired output. Thus we sent you the test file you received and expected a critique much sooner so we could more closely match your needs. 7

Tape Format

(J24466) 11-NOV-74 15:42;;;; Title: Author(s): Elizabeth K. Michael/EKM; Distribution: /EAR([ACTION]) WEC([ACTION]) RWW([INFO-ONLY]) EKM([INFO-ONLY]) HGL([INFO-ONLY]) DSM([INFO-ONLY]) ; Sub-Collections: SRI-ARC; Clerk: EKM; Origin: < MICHAEL, TAPEMSG.NLS;4, >, 11-NOV-74 15:34 EKM ;;;;####;

CHI 11-NOV-74 15:59 24467

speech demo

I would like to go to speech demo. 1

CHI 11-NOV-74 15:59 24467

speech demo

(J24467) 11-NOV-74 15:59;;;; Title: Author(s): Charles H. Irby/CHI; Distribution: /JML([INFO-ONLY]) ; Sub-Collections: SRI-ARC; Clerk: CHI;

JMB 11-NOV-74 16:32 24468

TENEX documentation; REF--24444,>

Dear Pooh, I would like a copy each of: TENEX EXECUTIVE Manual, TENEX Users Guide, and TENEX TECO for my office. Thanks, your faraway next door neighbor (Jeanne) 1

JMB 11-NOV-74 16:32 24468

TENEX documentation; REF--24444,>

(J24468) 11-NOV-74 16:32;;;; Title: Author(s): Jeanne M. Beck/JMB;
Distribution: /POOH([ACTION]) ; Sub-Collections: SRI-ARC; Clerk:
JMB;

DVN 11-NOV-74 19:24 24469

Response To 24454

Right on.

1

DVN 11-NOV-74 19:24 24469

Response To 24454

(J24469) 11-NOV-74 19:24;;;; Title: Author(s): Dirk H. Van Nouhuys/DVN; Distribution: /JOAN([ACTION] please add 24454 to the drit note book) SRI-ARC([INFO-ONLY]) DLS([INFO-ONLY] I thought this would ammuse you too) ; Sub-Collections: SRI-ARC; Clerk: DVN;

DVN 11-NOV-74 21:50 24470

Responses to Coments on User's Gudie

I wrote this thursday, hadn't gotten around to sending it.

DVN 11-NOV-74 21:50 24470

Responses to Coments on User's Gudie

(vannouhuys,novguide,1a4) The Quick Reference Gudie is what is called around ARC the cue card...I thought I should call it by the name printed on it. (vannouhuys,novguide,4d1b3) A brace is that funny wiggling thing like two tildes on end. I think it is clear on the terminal. 1

DVN 11-NOV-74 21:50 24470

Responses to Coments on User's Gudie

(J24470) 11-NOV-74 21:50;;;; Title: Author(s): Dirk H. Van Nouhuys/DVN; Distribution: /RWW([INFO-ONLY]) ; Sub-Collections: SRI-ARC; Clerk: DVN;

DVN 11-NOV-74 22:09 24471

Visit Log: Richard Smith of SRI Economics with Reference to a Text Editor for R.R. Donnelley

LINK RACK ;;SINCE(5-NOV-74 00:00) 1

(,#a:ctr) (directory,) (fiche,) 1a

(,#j:ctr) (newprim,:;SINCE(28-JUL-74 00:00);kK) 1b

((,#jr:w) (newmess,,t) 1c

(,lit) 1d

(odp,#x) 1e

:(userguides,locator,5:xebn) 1f

TOP(userguides,locator,1:xn) 1g

(,play) (mylin,) 1h

Real Alphabetic sort: (programs,sortalphabetic,) 1i

(documentation,help, systems:xeb) (documentation,help, :;["DHvN"] AND ["..."];k) 1j

(documentation,manual,do:xebb) 1k

(documentation,final,:xeb) 1l

(twocc,new) 1m

sendmail crapname:< (vannouhuys)[Send-mail],PC;1 > 1n

Journal documents (most recent first) 2

SLJ 10-NOV-74 00:01 24454
Let's Call A Spade A Spade ...
Message: Just consider: We might call a common garden spade: a personalized earth-moving equipment module; a mineralogical mini-transport; a personalized strategic tellurian command and control module; an air-to-ground interface contour adjustment probe; a leveraged tactile-feedback geomass delivery system; a man-machine energy-to-structure converter; a one-to-one individualized geophysical restructurizer; a portable unitized earthwork synthesis system; an entrenching tool (Firesign Theater); a zero-sum dirt level adjuster; a feedback-oriented contour management probe and digging system; a gradient disequilibrator; a mass distribution negentroprizer; a dig-it-all system; and extra-terrestrial transport mechanism.....Spades, not words, should be used for

DVN 11-NOV-74 22:09 24471
Visit Log: Richard Smith of SRI Economics with Reference to a Text Editor for R,R, Donnelley

shovelling, But words should help us unearth the truth,
--excerpt from Dream Machines/Computer Lib
*****Note: [INFO-ONLY] *****

2a

JEW 8-NOV-74 15:01 24450
ARC Dialog Support: ROUGH DRAFT for a Report Chapter
Location: (MJOURNAL, 24450, 1:w)
*****Note: [INFO-ONLY] *****

2b

Comments: This was journalized because we seemed to need formatted copies from time to time,

2b1

JEW 8-NOV-74 13:49 24448
ARC Journal: ROUGH DRAFT of Report Chapter
Location: (MJOURNAL, 24448, 1:w)
*****Note: [INFO-ONLY] *****

2c

LAC 7-NOV-74 12:07 31279
Info from DVN on COM prices and procedures
Location: (MJOURNAL, 31279, 1:w)
*****Note: [INFO-ONLY] *****

2d

Comments: This is a pointer at information about DDSI COM and hardcopy procedures and prices, Bill asked me to cend him a copy as he couldnt find his orriginal copy of the information,

2d1

KIRK 6-NOV-74 18:45 24433
Publish user-subsystem has Index and TOC, Not Format,
Message: Check out the Publish user-subsystem and let me know what you think, I erred by calling "Format" in a previous message,
*****Note: [INFO-ONLY] *****

2e

MEH 6-NOV-74 17:39 24431
My Thoughts about Recording Written Dialogue, and a Suggestion,
Ref: 24393, 24404,
Location: (MJOURNAL, 24431, 1:w)
*****Note: [INFO-ONLY] *****

2f

MEH 6-NOV-74 17:28 24430
My thoughts about recording dialogue, and a suggestion: ref: 24393, 24404,
Location: (MJOURNAL, 24430, 1:w)

DVN 11-NOV-74 22:09 24471

Visit Log: Richard Smith of SRI Economics with Reference to a Text Editor for R.R. Donnelley

*****Note: [INFO-ONLY] *****

2g

DCE 25-OCT-74 09:05 24320
More care about spelling in our written communications
Location: (MJOURNAL, 24320, 1:w)
*****Note: [ACTION] *****

2h

(Info) Journal documents for information only (most recent first) 2i

KIRK 11-OCT-74 00:38 24196
Line lengths in NLS
Location: (JJOURNAL, 24196, 1:w)
*****Note: [INFO-ONLY] *****

2j

JAKE 10-OCT-74 23:51 24195
Trip report - future plans for the Arpanet
Location: (JJOURNAL, 24195, 1:w)
*****Note: [INFO-ONLY] *****

2k

Comments: This is the proper text for a journal item you received a few days ago which inadvertently contained ACMs initial file, Sorry for the mix-up, Jake, 2k1

NDM 10-OCT-74 18:01 24189
Viewspec Cards: COM formatted
Location: (JJOURNAL, 24189, 1:w)
*****Note: [INFO-ONLY] *****

2l

Comments: Since the directives that the journal put in will screw up this file (formatted to 1/1000"), you must create your own work file from what was originally my file, I.e, copy from my origin on into a new file, DO NOT PRINT THE JOURNAL FILE DIRECTLY. 2l1

KIRK 10-OCT-74 17:30 24188
More last-minute mods to NLS-8
Message: Since "Jump (to) File Named TYPEIN" is essentially the same thing as "Jump (to) File <SPACE> TYPEIN", and "Jump (to) File <SPACE> BUG" is more easily done with "Jump (to) File BUG" and <SPACE> is a non-intuitive, inconsistant "Commandword", the following things should have been done when the Jump to Name command was added: ALT should have been armed when specifying the FILE entity by TYPEIN, The "Jump (to) File <SP>" command should

DVN 11-NOV-74 22:09 24471
Visit Log: Richard Smith of SRI Economics with Reference to a Text Editor for R.R. Donnelley

be deleted. Charles agrees these changes should be made, so I updated the help information accordingly when I added the New command.
*****Note: [INFO-ONLY] *****

2m

JMB 6-OCT-74 19:18 24170
Trouble documenting a command when I can't find out the story - the Protect command
Location: (JJOURNAL, 24170, 1:w)
*****Note: [INFO-ONLY] my present means of receiving news on changes aren't working*****

2n

EJK 4-OCT-74 09:26 31121
Comments on New NLS
Location: (JJOURNAL, 31121, 1:w)
*****Note: [ACTION] *****

2o

FDBK 30-SEP-74 16:17 24104
User Feedback Decisions leading to NLS-8,2
Location: (JJOURNAL,24104,1:w)
*****Note: [ACTION]
(Secondary Distribution Copy from KIRK)*****

2p

Comments: This document contains the status of user feedback decisions for NLS-8,2. It is over 50 pages long, we advise you NOT to print it. Read it online. For the new features and bug fixes, see the Documented branch. For those suggestions that have been rejected, see the Rejected branch. The items scheduled to be done in the next version are in <NLS,MODS,>. Those items which remain as Needs & Possibilities are in <FEEDBACK,FDBK,FUTURE>.
Secondary Distribution Copy

2p1

FDBK 3-OCT-74 23:47 24161
User Feedback Decisions leading to NLS-8,3
Location: (JJOURNAL, 24161, 1:w)
*****Note: [INFO-ONLY] *****

2q

Comments: This document contains the status of user feedback decisions for NLS-8,3. It is over 50 pages long, we advise you NOT to print it. Read it online. For the new features and bug fixes, see the Documented branch. For those suggestions that have been rejected, see the Rejected branch. The items

DVN 11-NOV-74 22:09 24471

Visit Log: Richard Smith of SRI Economics with Reference to a Text Editor for R,R, Donnelley

scheduled to be done in the next version are in <NLS,MODS,>, Those items which remain as Needs & Possibilities are in <FEEDBACK,FDBK,FUTURE>. 2q1

RLB2 2-OCT-74 12:02 24120
WHAT IS A SIMPLE DRAWING?
Location: (JJOURNAL, 24120, 1:w)
*****Note: [INFO-ONLY] *****
2r

EJK 30-SEP-74 10:33 31104
Feedback on NNLS
Location: (JJOURNAL, 31104, 1:w)
*****Note: [ACTION] *****
2s

RLB2 30-SEP-74 12:10 24096
DISPLAYS FOR NLS GRAPHICS CAPABILITY
Location: (JJOURNAL, 24096, 1:w)
*****Note: [INFO-ONLY] *****
2t

JAKE 30-SEP-74 09:52 24093
Trip Report - Future Management and Programs of the Arpanet
Location: (JJOURNAL, 24093, 1:w)
*****Note: [INFO-ONLY] *****
2u

KIRK 28-SEP-74 01:28 24087
Syntax could be better
Message: I think we made a big mistake when we decided to describe the syntax in terms of how you specify something (CONTENT, SOURCE, DESTINATION) instead of what you need to specify (STATEMENT, CHRACTER, etc.) when we decided we didn't have room to put both. I'm journalizing this for the record in hopes that it can be rectified someday.
*****Note: [INFO-ONLY] *****
2v

ACM 26-SEP-74 14:07 24066
Trip Report - Arpanet Book Discussion
Location: (HJOURNAL, 24066, 1:w)
*****Note: [ACTION] *****
2w

DIA 26-SEP-74 09:23 24060
New (Experimental) version of L10 Compiler
Location: (HJOURNAL, 24060, 1:w)

DVN 11-NOV-74 22:09 24471

Visit Log: Richard Smith of SRI Economics with Reference to a Text Editor for R.R. Donnelley

*****Note: [INFO-ONLY] *****

2x

FDBK 25-SEP-74 14:16 24054
User Feedback Decisions leading to NLS-8,1
Message: <HJOURNAL, 24051,> contains the status of user feedback decisions for NLS-8,1. It is over 100 pages long, we advise you NOT to print it. Read it online. For the new features and bug fixes, see the Documented branch. For those suggestions that have been rejected, see the Rejected branch. The items scheduled to be done in the next version are in <NLS,MODS>. Those items which remain as Needs & Possibilities are in <feedback,fdbk,future>.
*****Note: [INFO-ONLY] *****

2y

EKM HGL CHI RWW 25-SEP-74 16:57 24056
NLS Task Shopping List for NSW
Location: (HJOURNAL, 24056, 1:w)
*****Note: [INFO-ONLY] *****

2z

JAKE 25-SEP-74 16:16 24055
A Plea and a Proposal
Location: (HJOURNAL, 24055, 1:w)
*****Note: [ACTION] *****

2a0

JAKE 25-SEP-74 12:06 24053
ARPA Book Chapter Outline
Location: (HJOURNAL, 24053, 1:w)
*****Note: [INFO-ONLY] *****

2aa

FDBK 24-SEP-74 23:37 24051
User Feedback Decisions leading to NLS-8,1
Location: (HJOURNAL, 24051, 1:w)
*****Note: [INFO-ONLY] *****

2ab

JAKE 24-SEP-74 20:59 24049
Contact Report: NIC Discussion with Craig Fields, ARPA IPTO
Location: (HJOURNAL, 24049, 1:w)
*****Note: [INFO-ONLY] *****

2ac

EJK 24-SEP-74 09:47 31090
Comments on NNLS 24 Sep 74
Location: (HJOURNAL, 31090, 1:w)

DVN 11-NOV-74 22:09 24471

Visit Log: Richard Smith of SRI Economics with Reference to a Text Editor for R.R. Donnelley

*****Note: [ACTION] *****

2ad

(dosomething)

3

RWW 11-SEP-74 08:42 23938
ARC Participation in the Design of an SRI Text Handling System
Location: (HJOURNAL, 23938, 1:W)
*****Note: [INFO-ONLY] *****

3a

NDM 7-AUG-74 09:30 23742
Visit with Bill Carlson
Location: (GJOURNAL, 23742, 1:W)
*****Note: [INFO-ONLY] *****

3b

DSM 29-JUL-74 16:15 23692
Modifications Planned to NLS for OFFICE-1 before October 1st.
Location: (GJOURNAL, 23692, 1:W)
*****Note: [INFO-ONLY] *****

3c

JEW 29-JUL-74 19:23 23694
Preview of Inter-Host/Inter-Fork Procedure Call Protocol
Location: (GJOURNAL, 23694, 1:W)

3d

Comments: For those interested in contributing to the design of the protocol to be used in the NLS split. This document is incomplete and unpolished, but should indicate the direction in which I'm headed. Now is the time to offer suggestions. 3d1

CHI 28-JUL-74 16:35 23689
NSW software plan for 29-July to 1-October-74
Location: (GJOURNAL, 23689, 1:W)
*****Note: [INFO-ONLY] *****

3e

RWW 23-JUL-74 13:11 23667
Suggested Changes to Help System
Location: (GJOURNAL, 23667, 1:W)
*****Note: [INFO-ONLY] *****

3f

JEW 19-JUL-74 14:23 23649
FTPFRK (2,0) Programmer's Guide
Location: (GJOURNAL, 23649, 1:W)

DVN 11-NOV-74 22:09 24471

Visit Log: Richard Smith of SRI Economics with Reference to a Text Editor for R.R. Donnelley

*****Note: [INFO-ONLY] *****

3g

Comments: Feel free to pass this along to anyone you think might benefit from it.

3g1

EAR 11-JUL-74 09:20 23596
NSW Microfiche Format
Location: (GJOURNAL, 23596, 1:W)
*****Note: [INFO-ONLY] *****

3h

Comments: These are sendmessages preserved for future reference.

3h1

RWW 9-JUL-74 15:06 23555
Notes on Talk With Tom Humphrey on SRI Text System
Location: (GJOURNAL, 23555, 1:W)
*****Note: [INFO-ONLY] *****

3i

DIA 30-MAY-74 09:44 23165
New Line Processor program description for Users
Location: (GJOURNAL, 23165, 1:W)
*****Note: [INFO-ONLY] *****

3j

Comments: Contains switch and light meanings and error reporting procedures.

3j1

NDM 20-AUG-74 09:50 31011
Comments on Output Processor Section in Final Report
Location: (GJOURNAL, 31011, 1:W)

3K

DCE 21-AUG-74 07:52 23831
NLS Version Numbers
Location: (GJOURNAL, 23831, 1:W)
*****Note: [INFO-ONLY] *****

3l

(fortherecord)x(userguides,commands,)

4

DCE 9-AUG-74 19:01 23756
Possibility of Providing Report Development Support for DoD Inter-Netting Study Group
Location: (GJOURNAL, 23756, 1:W)

DVN 11-NOV-74 22:09 24471

Visit Log: Richard Smith of SRI Economics with Reference to a Text Editor for R.R. Donnelley

*****Note: [INFO-ONLY] ***** 4a

Comments: Summary background discussion; tentative possibilities 4a1

dir 5

Delete Plex (vannouhuys,directory,1) 5a

Copy Directory vannouhuys(vannouhuys,directory,)d All N Time Read N Size N Date Read N Number Accesses N Size N Sort Read N Group Deletion 5b

Substitute Text Plex (vannouhuys,directory,1)
n pgsSize in Pages n RdLast ReadnAccessesNo. of Accesses (reads + writes)yn 5c

Jump Link (directory,:x) 5d

Update File Old 5e

Show Disk 5f

docyoudear 6

Connect Directory documentationkwcs 6a

Delete Plex (documentation,directory,1) 6b

Copy Directory documentation(documentation,directory,)d All N Time Read N Size N Last N Date Read N Number Accesses N Size N Sort Read NGroup Lastn 6c

Substitute Text Plex (documentation,directory,1)
n pgsSize in Pages n RdLast ReadnAccessesNo. of Accesses (reads + writes)yn 6d

Jump Link (directory,:x) 6e

Update File Old 6f

Show Disk 6g

(documentation,xhelp,how :xes) (documentation,howto,),#x:w)
(septline, #x) 7

comdir 8

Visit Log: Richard Smith of SRI Economics with Reference to a Text Editor for R.R. Donnelley

Connect Directory com 8a

Delete Plex (com,directory,1) 8b

Copy Directory com(com,directory,)d All N Time Read N Length N Last N Date Read N Number Accesses N Size N Sort Read N Group Lastn 8c

Substitute Text Plex (documentation,directory,1)
n pgsSize in Pages n RdLast ReadnAccessesNo. of Accesses (reads + writes)y 8d

Jump Link (directory,:x) 8e

Update File Old 8f

Show Disk 8g

mess 9

Mov Ple (newmess,1) (oldmess,) 9a

Jum Lin (oldmess,) 9b

Upd Fil Old 9c

Got Pro Loa Pro message 9d

Jum Lin (vannouhuys,newmess,) 9e

Got Mes 9f

Mov Mes vannouhuys,message.txt; 9g

Sor Mes 9h

Qui To Bas 9i

Update Fil Old 9j

(startupx) 10

Execute Programs 10a

Set Buffer 8 10b

Execute Programs 10c

DVN 11-NOV-74 22:09 24471

Visit Log: Richard Smith of SRI Economics with Reference to a Text Editor for R.R. Donnelley

Exe Pro Loa Pro mouse 10d

(archive) 11

author Journal documents written 12

DVN 11-NOV-74 19:24 24469
Response To 24454
Message: Right on,
*****Note: Author Copy***** 12a

DVN 8-NOV-74 15:20 24451
The Dictionary Has Two p's,
Message: See (vannouhuys,novguide,"equipped")
*****Note: Author Copy***** 12b

DVN 7-NOV-74 22:51 24440
Conversations about COM with George Lithograph
Location: (MJOURNAL, 24440, 1:W)
*****Note: Author Copy***** 12c

DVN 7-NOV-74 17:02 24438
Request to Journalize Draft on Journal System
Message: Carlson for Lukasic (sp?) has asked for some information on the Journal. Among other things I would like to send him your draft (documentation,final,6b) as support. Why don't we journalize it, clearly marked DRAFT, so we get a nice, familiar format and can get at it easilly again, since I note it has been used this way a couple ot times before.
*****Note: Author Copy***** 12d

DVN 7-NOV-74 16:46 24437
One More thought about Journal Deliveryy
Message: As a step toward (documentation, final,,6b6c4) and the rest of that plex, what the journal should do is enter in everyone's initial file an author,keword, and arrival data catalog of journal items sent to her or that she sent.
*****Note: Author Copy***** 12e

DVN 6-NOV-74 10:39 24425
400 Cockrels
Message: A lady just called frm Tabor farms who said that Mrs Jeanne Beck at a different xtension had ordered 400 cockrels and

DVN 11-NOV-74 22:09 2447

Visit Log: Richard Smith of SRI Economics with Reference to a Text Editor for R.R. Donnelley

since SRI owed them $268 from 1973 from previous orders of cockrels she wanted them to pay up befoe she shipped more. I thought there was a mistake and shunted her to the extension number she had. But, Jeanne, if you have been ordering cockrels, it's time to pay up.
*****Note: Author Copy*****

12

DVN 6-NOV-74 10:22 24424
Locator Has Lost the Journal Indices
Message: They may be on line, but a sampling of links in locator takes me to files that say they are not online.
*****Note: Author Copy*****

12g

DVN 4-NOV-74 22:12 24406
References and Thoughts about Output to COM from Office-1
Location: (MJOURNAL, 24406, 1:w)
*****Note: Author Copy*****

12h

DVN 4-NOV-74 09:04 24394
Response to Outline for Output Processor Primer
Message: Jeanne, your outline looks neat to me. I would be sure to include a scenario of making a format via the format system.
*****Note: Author Copy*****

121

Comments: This comments on (hjournal,24389,) 1211

DVN 1-NOV-74 21:41 24386
Edition II of DCA Paper
Message: I spent some time Friday afternoon talking on the phone with Susan about the DCA paper. Mostly we got some ODP directives straightened out. She also mentioned that Lyons had sought local printing for Edition II see--vannouhuys,oldmess,#dcasc) and gotten a minimum of 15 days for 100 copies; couldn't we do as well? I think we might. It would take some forwarning of DDSI. The paper would be ready early next week (they are still editing). It might be necessary for me (or Dean) to go to DDSI and watch what came out and make corrections via a terminal there and bring the file down again if necessary, but that might be worth while for that paper.
*****Note: Author Copy*****

12j

DVN 31-OCT-74 09:42 24368
Your Help with <vannouhuys,septline,>

DVN 11-NOV-74 22:09 24471
Visit Log: Richard Smith of SRI Economics with Reference to a Text Editor for R.R. Donnelley

Message: It looks neat tom me, I am going to circulate this version to Don and Martin.....While we're at it, I would appreciate your thoughts on a draft of a brief document on TNLS addressing, in <hamilton,tnlsaddressing,>.
*****Note: Author Copy*****

12k

DVN 31-OCT-74 09:11 24367
Question about Conformation of the Lineprocessor User's Guide to Specificatons
Message: Particularlly in organization, the Lineprocessor User's Guide does not conform to the specifications set out in (journal,24335,ib. Don, since 24335 was the first I learned of those specifications but I have spoken to you several times in the intervening weeks, I wonder if you think the guide fulfilles users' needs except as we noted in our last conversation?
*****Note: Author Copy*****

121

DVN 30-OCT-74 08:54 24361
Watching
Message: How nice to know you are still watching us from afar....I have deleted my citation to your message that gave the DCA publication scheudle and the online journal catalogs are a shambles, could you send it to me again?...There is some chance I will be in Washington next week for the Demo, but I think the odds are against it.
*****Note: Author Copy*****

12m

DVN 28-OCT-74 21:51 24343
The Need for a Way to Create Formatted, Sequential Files Suitable for Printing at Terminals at Other Sites
Message: It sometimes happens that we want to pass a file through the formatting steps of the Output Processor for transmisson to some one who will print it out at a terminal as a sequential file, e.g. as part of a sendmessage. A file created by the command Output Printer contains some control characters intended for our line printer that make it unsuitable for printing at a terminal. A procedure exists for passing this file through sendprint to scrub out the control characters, but it is awkward to use and creates a file that may contain long lines which TENEX then wraps around with a double star. It appears that if the output teletype command could aternatively output to a file, that file would be suitable for this use.
*****Note: Author Copy*****

12n

DVN 11-NOV-74 22:09 24471
Visit Log: Richard Smith of SRI Economics with Reference to a Text Editor for R.R. Donnelley

DVN 28-OCT-74 09:21 24334
Please Send Sample XGP Fonts
Message: Glad to hear XGP can change type sizes. Please do send samples of all available fonts.
*****Note: Author Copy*****

12o

DVN 25-OCT-74 13:39 24326
Conversation with Connie McLindon about ARPA and ISI-XGP
Location: (MJOURNAL, 24326, 1:w)
*****Note: Author Copy*****

12p

DVN 25-OCT-74 13:22 24325
Functional Documents and Journal Numbers
Location: (MJOURNAL, 24325, 1:w)
*****Note: Author Copy*****

12q

DVN 24-OCT-74 22:17 24318
More On Journal Citations
Location: (MJOURNAL, 24318, 1:w)
*****Note: Author Copy*****

12r

DVN 24-OCT-74 20:13 24317
The Salesman from George Lithograph Will be Here Tmorrow
Message: When Walter Bass was still here, ARC spoke to George Lithograph, a local firm with a good reputation in the printing field, about doing our COM work. Recently they aquiredd a new COM device, a Singer 6000, and remembered us enough to have a salesman call. I have an appointment with him tomorrow at 2:30, and I'm sure he would be glad to talk with anyone who want to join us. He has asked for and I intend to supply a sample tape of our output(jjournal,12214,) and our specifications(journal,14093,).
*****Note: Author Copy*****

12s

DVN 24-OCT-74 09:28 24293
Is Aything Happening on COMing the DCA Paper?
Message: Is anything happening? Since I don't presently have anything else to go to COM, I'm not planning to send the file you made tonight unless I hear from you otherwise.
*****Note: Author Copy*****

12t

DVN 24-OCT-74 09:04 24292
Is Documentation Holding Up NLS-8?

DVN 11-NOV-74 22:09 24471

Visit Log: Richard Smith of SRI Economics with Reference to a Text Editor for R.R. Donnelley

Message: I feel I have lost touch with why NLS-8 has not come up as the running system at Offie-1. If it is waiting for documentation I would like t know about it. The state of documentation is essentially as described in (mjournal,24247,2) except that a draft of the two-page document (mjournal,24247,2h) now exists and a draft of the revised viewspec cards (JRNL22, J24266:gw) and (journal,jrnl22,j24262) is in review.
*****Note: Author Copy*****

12u

DVN 24-OCT-74 08:41 24290
Faileur to Properly Journalize the NSW Proposal
Message: It's my fault that the NSW proposal is not online under its correct number (23352). It was originally printed with the number preassigned to Mil and a simulated journal header for publicatioln purposes, Sometime in September Dick asked Joan to journalize it corrctly. She tried to do so with my help, but the Journal system was suffering from a bug at the time so that when it failedto act on our request for Mil's preassigned number and instead gave us a new number, it did not give an error message. Some time later we discovered we had failed. I did not get arround to trying again unil, as a matter of fact, yesterday. The journal will normally grant access to a preassigned number either if the number is assigned to an author or if th sender is connected to the assignee's directory. Of course Mil is not an author and I discovered yesterday that her directory no longer exists. I expect this journal item will reach Dave Hopper and he will advise m how to proceed. When I hear from him I will journalize the proposal under the right number forwith.
*****Note: Author Copy*****

12v

DVN 23-OCT-74 11:22 24269
For A user otiôn to Turn off Journal notification
Location: (JOURNAL, JRNL22, J24269:gw)
*****Note: Author Copy*****

12w

Message: 12w1

I think the feature of the journal interrupting your work to tell you when it delivers is a pain in the ass; there should be a useruption to defend users against it. Nor do I like delivery itno the classes Information and action. It is bad enough trying to force items into those blunt catagories when you send them. 12w1a

DVN 23-OCT-74 09:10 24266

DVN 11-NOV-74 22:09 24471
Visit Log: Richard Smith of SRI Economics with Reference to a Text Editor for R.R. Donnelley

Revised Viewspec Cards
Location: (JOURNAL, JRNL22, J24266:gw)
*****Note: Author Copy*****

12x

Comments: If you have suggestions, please let me know. 12x1

Message: 12x2

Jeanne Beck hasbrought the little viewspec/Keyset cards up to date and made some changes in the fomat, improvements as I see it. I amseneding the revision thrugh the review process, but with luck everyone will OK it and we can send the file to DDSI thursday night. The draft is in <userguides,viewspeccard,> and some further explanation is in (journal,jrn122,j24262) 12x2a

DVN 21-OCT-74 19:39 24263
Alba Amicorum
Message: Could yo do me the favor of asking Caroline what "alba amicorum" might mean in the context of Christian religious books?
*****Note: Author Copy*****

12y

DVN 18-OCT-74 13:49 24247
MINUTES OF DOCUMENTATION MEETING OF 10-14-74: Status of Documentation, Plans for Introductory Hardcopy for Help, Plans for Something for Learners to Read.
Location: (MJOURNAL, 24247, 1:w)
*****Note: Author Copy*****

12z

DVN 17-OCT-74 21:26 24242
Missing Indeces: All the Links in the Attached Group Yield the Message File Not Online
Location: (MJOURNAL, 24242, 1:w)
*****Note: Author Copy*****

12a@

DVN 17-OCT-74 12:53 24237
Proposal Posibility: Output Processorr Direct to XGP [To add this item to DPCS subcollection]
Message: See <mjournal,24134,>
*****Note: Author Copy*****

12aa

SRI-ARC 16-OCT-74 16:22 23912
NLS-8 Command Summary [as of 6-OCT-74]

DVN 11-NOV-74 22:09 24471
Visit Log: Richard Smith of SRI Economics with Reference to a Text Editor for R.R. Donnelley

Location: (MJOURNAL, 23912, 1:W)
*****Note: Author Copy*****

12ab

DVN 16-OCT-74 16:09 24234
Minutes of Documentation Meeting of October 7; Command Summary, Userguides, Help and Syntax, Proofing
Location: (MJOURNAL, 24234, 1:W)
*****Note: Author Copy*****

12ac

DVN 16-OCT-74 14:54 24233
Primer, DCA Internetting Study Drafts, Font Test Tape to DDSI
Location: (MJOURNAL, 24233, 1:W)
*****Note: Author Copy*****

12ad

DVN 16-OCT-74 09:10 24228

Message: mes watson,message.txt;
*****Note: Author Copy*****

12ae

DVN 15-OCT-74 13:42 24220
The Next Move in DPCS for Montgomery
Message: Naturally I am interested in the possibilitiesof NLS publications services to the people in Montgomery, What is the next move?
*****Note: Author Copy*****

12af

DVN 10-OCT-74 21:47 24192
Role of Nucleator
Message: Doug, and I and Nielsen have substantially agreed that I will be a nucleator. There are some budget considerations incompletely resolved but the general plan now is that my time committed to such work will graudally rise from its present 10-15% to about 50% in January and probably more later. We will have to think carefully how we can most effectively use the remainder of my time. Neilsen is anxious that I not do anything that makes me appear to ARC as an outsider. I have not taken any action on replacement until things clarify a bit more.
*****Note: Author Copy*****

12ag

DVN 10-OCT-74 20:03 24190
Anthropomorphism Can Aid Clarity
Location: (JJOURNAL, 24190, 1:W)

DVN 11-NOV-74 22:09 24471
Visit Log: Richard Smith of SRI Economics with Reference to a Text Editor for R,R, Donnelley

*****Note: Author Copy*****

12ah

DVN 10-OCT-74 08:16 24180
Dean's Priorities
Message: I cast my vote for a revised OP Users' Guide first, We have about30 of these guys left and give them out almost daly, (We have a good supply of a slightly deffective printing of the same version,) Second the bibliographic subsystem,
*****Note: Author Copy*****

12ai

DVN 9-OCT-74 09:29 24172
Journal Confounds Bugs with Dreams
Message: I am a member of a group exploring possibilities of controling dreams, One of the techniques is to tell anyone who appears in your dreams about the dream, Recently Elizabeth Michael appeared in some of my dreams, Since she was travelling and I had to send her some information about demonstration files anyway, I reported my dream to her in a journal item, In one of its rare moments of humour the journal gave the same name (24170) to two items, the dream and an item by Jeanne Beck reporting a bug, The dream has the higher version number so people loading the item get the dream, Try (jjournal,24170,nls;1,1:w) if you want to learn about the bug,
*****Note: Author Copy*****

12aj

DVN 8-OCT-74 20:37 24170

Location: (JJOURNAL, 24170, 1:w)
*****Note: Author Copy*****

12ak

DVN 30-SEP-74 22:21 24105
Information on Printing Through COM
Location: (JJOURNAL, 24105, 1:w)
*****Note: Author Copy*****

12al

numbers 13

22128 13a

Help Command language specs 13b

22130 Ken' CML Paper 13c

DVN 11-NOV-74 22:09 24471

Visit Log: Richard Smith of SRI Economics with Reference to a Text Editor for R.R. Donnelley

22131 Line processor User guide 13d

22132 NLS-8 User's Glossary 13e

22133 Final Report 13f

Userguides: 14

(haiku)5 7 5 15

September Meetings,
Anger glances off order,
Days fade into beige, 15a

doromadary:srl bah 15b

Trying the System
After so many Years
Is Rather Trying.....Tom Humphry 15c

Preassigned numbers: 16

LINK RACK 17

(lit) 17a

(agentann,20:cmt) 17a1

(documentation,mickey,) 17a2

(documentation,jane,) 17a3

(blind-light,:xyK) 17a4

(cutprog,ncut:x) 17a5

(beautest,) 17a6

(cutprog,:x) 17a7

(gugcut,) 17a8

(my,name;,:) 17a9

(essayassim,) 17a10

(documentation) 17b

DVN 11-NOV-74 22:09 24471

Visit Log: Richard Smith of SRI Economics with Reference to a Text Editor for R.R. Donnelley

Privacy:(ijournal,10341,) 17b1

(vannouhuys,qmr,:xb) 17b2

(documentation,help,help *control-q>kwcs 17b3

(documentation,help,)jjsyzygyg 17b4

(documentation,snort, mmb)jjsyzygyg 17b5

(bbn-tenex,walace,dolt) 17b6

(imlac,imlac-users-guide,:xb) 17b7

(documentation,dimex,)ergsdnb 17b8

(hjournal,10088,) 17b9

(hjournal,10035,) 17b10

OPUG 17b11

(userguides,op-dirlist,1) 17b11a

(userguides,op-direxp,1) 17b11b

(userguides,op-dirindex,1) 17b11c

(userguides,op-intro,1) 17b11d

(userguides,op-valop,1) 17b11e

AKW paper: (Ijournal,14724,1) 17b12

Co-ordinate Services paper: (Mjournal,12445,1) 17b13

(journalref) 17c

(journal,jrn121,j23320) 17c1

(journal,jcat,j13001:;["): WA"];kxr) 17c2

(nic,locator,2 :xmbb) 17c3

Recent Entries:(journal,tjcat,1:xbbrrD) (journal,tjcat,:;[" Bair "];kxbbrr) 17c4

(tejournal,jnjn40620,nlp;1,1) 17c5

DVN 11-NOV-74 22:09 24471

Visit Log: Richard Smith of SRI Economics with Reference to a Text Editor for R.R. Donnelley

(initials) 17d

(norton,jcn,:mx) 17d1

(stone,dls,author:ctr) 17d2

(bbn-net,njn,author:ctr) 17d3

(bair,jhb,author:ctr) 17d4

(engelbart,dce,author:ctr) 17d5

(identfile,idents.master,kwac)
(identfile,idents.master,3:;["Ge"]10s150CH "BBN";kg)
(identfile,idents.master,3:;["Ge"] AND["BBN"];kg) 17d6

(identfile,idents.master,"Kruz")
(identfile,idents.master,3:;["Ge"]10s150CH "BBN";kg)
(identfile,idents.master,3:;["Ge"] AND["BBN"];kg) 17d7

jprogram) 17e

(hopper,kpgms,:x) 17e1

(cutprog,:x) (cutprog,tryone) 17e2

(nic,keysorts,:xb) 17e3

(norton,jcn,programs :xb) 17e4

(edprogs,:x) 17e5

(user-progs,makeref,) 17e6

(user-progs,append,) 17e7

(user-progs,letter,) 17e8

(user-progs,-contents,) 17e9

(play) 17f

(zymurgy,:o) 17f1

(taxon,:x) 17f2

(mylin,:x) 17f3

DVN 11-NOV-74 22:09 24471

Visit Log: Richard Smith of SRI Economics with Reference to a Text Editor for R.R. Donnelley

(agentann,16) 17f4

(documentation,mickey,) 17f5

(documentation,jane,) 17f6

17f7

(trainingaids) 17g

(nic,locator,2i;xbb) 17g1

(auerbach,syllabus,;x) 17g2

(train,:;SINCE(29-Mar-73 9:50);k) 17g3

(structures,) 17g4

(taxon,;x) 17g5

(apple,) 17g6

(where,) 17g7

(headfoot, ;xrm) 17g8

(jernigan,a-nictnls,) 17g9

17h

NLSControlCharacters 18

CA="D,|;ECHO=!; 18a

Give editor Lee Work a copy of the humphry report 19

DVN 11-NOV-74 22:09 24471

Visit Log: Richard Smith of SRI Economics with Reference to a Text Editor for R.R. Donnelley

(J24471) 11-NOV-74 22:09;;;; Title: Author(s): Dirk H. Van Nouhuys/DVN; Distribution: /DPCS([INFO-ONLY]) MAP2([INFO-ONLY]) ; Sub-Collections: DPCS SRI-ARC; Clerk: DVN; Origin: < VANNOUHUYS, DVN.NLS;534, >, 6-NOV-74 17:30 DVN ;;;;####;

KIRK 11-NOV-74 22:11 24472

criteria for user programs

Jim Bair tells me that certain user programs are not to be available at office-1. In particular, the "mouse" user subsystem. The reasoning went something like "it competes with things you can do other ways in NLS and it uses buttons for functions different from their use in the rest of NLS" though I must admit the more I listened to the reasons, the more my temper may have kept me from understanding them. 1

As the one picked to translate and maintain userprograms, I would like to understand, accept, agree and abide by criteria used to decide what NLS tools and user programs are to be made available. I would like to point out that all of the userprograms do things that can be done other ways in nls and that in fact, all of the things NLS does can be done without using a computer. Furthermore, all programs and NLS itself use the same buttons for different functions. The current criteria are meaningless to me. 2

I was under the impression that if a program was 1) guaranteed to work as advertised, and 2) did not allow access to protected files or information, (e.g. the identification subsystem) it could be made available to the general user. 3

I certainly was not prepared to have certain programs censored for what appear to be quite arbitrary reasons. If this is going to be the criteria for user programs, I want nothing to do with their conversion or maintenance. 4

Just what kind of a market place is NLS going to be? 5

criteria for user programs

(J24472) 11-NOV-74 22:11;;;; Title: Author(s): Kirk E. Kelley/KIRK;
Distribution: /JCN([ACTION]) RWW([ACTION]) DCE([ACTION])
JHB([INFO-ONLY]) DVN([INFO-ONLY]) RLL([INFO-ONLY]) NPG([
INFO-ONLY]) RLB2([INFO-ONLY]) ; Sub-Collections: SRI-ARC NPG;
Clerk: KIRK;

DVN 11-NOV-74 22:14 24473

Visit Log: Richard Smith of SRI Economics with Reference to a Text Editor for R.R. Donnelley

Friday I spent an hour showing the system and chatting about publications with Richard H. Smith (x4879) of SRI's Economics department. 1

The R.R. Donnelley corporation is a printing compnay. It is large, it does abut $400 million/year business including things like the Sears catalog. They have hired SRI Economics to survey ther field of text editors to choose one that they may acquire and offer as a service to their customers. He is going on a tour of the country this week to view various editors. He mentioned Hendrix and Bowne, I mentioned Comarco (journal,15124,) and Autotext to him. 2

We had all the usual problems of trying to give some meaningful notion of NLS to someone with very little relevant background. I gave hm copies of The Augmented Knowledge Workshop (journal,14724,), Coordinated Information Services (journal,12445,), Online Team Environment (journal,13041,) and the Output Processor User's Guide (journal,12209,) et. sequens. 3

Of course the idea came up of suggesting NLS to 1)the Economics department an 2) Donnelley. 4

In the first case I explained about SRI's slot, that I believed the idea had already been suggested, and that he or whoever talk with Mike Placko. 4a

In the second I tried to emphasize that NLS was too expensive and unspecialized and developmental for many practical applicatons and that we were only interested in utility customers that wanted to participate in a development comunnity, etc. Howeve, Donnelley is large enough that a useful relationship is conceivable. 4b

I emphasized to him I was interested in any trip report that might emerge from his survey and that he he should keep in touch. 5

DVN 11-NOV-74 22:14 24473

Visit Log: Richard Smith of SRI Economics with Reference to a Text Editor for R.R. Donnelley

(J24473) 11-NOV-74 22:14;;;; Title: Author(s): Dirk H. Van Nouhuys/DVN; Distribution: /DPCS([INFO-ONLY]) MAP2([INFO-ONLY]) ; Sub-Collections: DPCS SRI-ARC; Clerk: DVN;

JMB 11-NOV-74 22:19 24474

Re the OP Primer--MJOURNAL, 24389,>

Dick, have you thought more about what the Output Processor Primer should be like; should this work integrate at all with Dean Meyer's work on the Introductory section of the OP Users guide; did you see the outline I made? (You asked me to wait for you to read my outline and talk to Dean before I talked or worked with Dean at all on our ideas.)

JMB 11-NOV-74 22:19 24474

Re the OP Primer--MJOURNAL, 24389,>

(J24474) 11-NOV-74 22:19;;;; Title: Author(s): Jeanne M. Beck/JMB; Distribution: /RWW([ACTION]) NDM([INFO-ONLY] you are probably pretty far along on the OP users guide by now, huh? I haven't done much) ; Sub-Collections: SRI-ARC; Clerk: JMB;

JMB 11-NOV-74 22:24 24475

Output Processor illustration page

Please send me any further drafts of that Illustration page for Output Processor directives that dean showed me. I'm going to use it in the Primer, OK? 1

Output Processor illustration page

(J24475) 11-NOV-74 22:24;;;; Title: Author(s): Jeanne M. Beck/JMB;
Distribution: /DVN([ACTION]) ; Sub-Collections: SRI-ARC; Clerk:
JMB;

KIRK 11-NOV-74 23:50 24476

RLB2 should be a member of NPG group.

I couldn't find RLB2 in the membership list for NPG. Seems he should be there. 1

KIRK 11-NOV-74 23:50 24476

RLB2 should be a member of NPG group.

(J24476) 11-NOV-74 23:50;;;; Title: Author(s): Kirk E. Kelley/KIRK;
Distribution: /MLK([ACTION]) CHI([INFO-ONLY]) FEED([INFO-ONLY
]) ; Sub-Collections: SRI-ARC; Clerk: KIRK;

KIRK 12-NOV-74 00:03 24477

<Userguides,locator,> update for NLS-8

Someday this should be available via help.

KIRK 12-NOV-74 00:03 24477

<Userguides,locator,> update for NLS-8

I have updated userguides locator somewhat for NLS-8, I didn't know what to do about the users' guides, which ones stay and which ones go? Anne, you should be able to decide and do that. Then the document should be printed and re-released. 1

KIRK 12-NOV-74 00:03 24477

<Userguides,locator,> update for NLS-8

(J24477) 12-NOV-74 00:03;;;; Title: Author(s): Kirk E. Kelley/KIRK;
Distribution: /POOH([ACTION]) DIRT([INFO-ONLY]) ;
Sub-Collections: SRI-ARC DIRT; Clerk: KIRK;

KIRK 12-NOV-74 00:14 24478

userguides, summary, (NLS-7) but where is userguides, commands, (NLS-8)?

I retrieved userguides, commands, last night from archive.

KIRK 12-NOV-74 00:14 24478

userguides, summary, (NLS-7) but where is userguides, commands, (NLS-8)?

I notice NLS-8 documentation is getting archived and only NLS-7 documentation is left in directory userguides. This is unfortunate, NLS-8 documentation should be marked not to be archived. Jeff has a run file that will do it right (files that have once been archived can't be marked not to be archived without his program). Pooh, it seems you would be the appropriate person for this. 1

KIRK 12-NOV-74 00:14 24478

userguides, summary, (NLS-7) but where is userguides, commands, (NLS-8)?

(J24478) 12-NOV-74 00:14;;;; Title: Author(s): Kirk E. Kelley/KIRK; Distribution: /POOH([ACTION]) DVN([INFO-ONLY]) JMB([INFO-ONLY]) ; Sub-Collections: SRI-ARC; Clerk: KIRK;