WSD 3-MAY-71 9:09 6256

Initial Spec for NLS/User-File system Interface Routines

This is somewhat incomplete and inconsistant, but is offerred as a start

Primitives for stage0 FS interface 1

- open 1A
  - Parameters: Document number 1A1
    - Identification of requesting user 1A1A
  - Returns: 1A2
    - Success (document on immediate access storage and access permitted) 1A2A
      - JFN of physical file. 1A2A1
      - Type of structure document is within physical file 1A2A2
      - SID's delimiting docmuent structure within physical file. 1A2A3
    - Failure: 1A2B
      - Failure code: 1A2B1
        - Document does not exist 1A2B1A
        - User does not have access requested to this document 1A2B1B
        - Document is inaccessable at this time (e.g. locked, in transit between tape and disk or??) 1A2B1C
        - Code specifies which 1A2B1D
        - Document is not in immediate access storage 1A2B1E
        - unspecified error 1A2B1F
      - Failure message if unspecified error 1A2B2
      - Access information if document is not on immediate access storage. 1A2B3
        - Location of physical file. 1A2B3A
        - Identification of document within file (structure type, SID's) 1A2B3B

Estimated access time 1A2B3C

Functions: 1A3

Locate entry for document in master directory, error if not there. 1A3A

Check that user is allowed requested access to document. 1A3B

locate the physical document. 1A3C

Determine the state and location of document and return appropriate error if not immediately accessable. 1A3D

If immediately accessable, open the file and return JFN, etc. 1A3E

Locate (see also--Journal, 6928, 413c:gw) 1B

Parameters: STID or SID of master catalog entry 1B1

Returns: 1B2

STID or SID of master catalog entry for physical file containing document. 1B2A

SID's of document in physical file. 1B2B

mebbe information relaing to availability of physical file. 1B2C

Functions: 1B3

Given a master catalog entry, this routine follows the location links in the EIB until the master catalog entry for the physical file is found. 1B3A

In doing so, it makes a running computation of the location of the original document in the physical file. 1B3B

Any unforseen errors are handled as SIGNALS. 1B3C

Retrieve: 1C

Parameters: 1C1

Catalog number of document to be retrieved. 1C1A

Function code: 1C1B

Do not return until document is retrieved. 1C1B1

Retrieve document in background, and (interrupt/ don't interrupt) specified fork when done. (If this is possible in TENEX) 1C1B2

Returns: 1C2

Success: TRUE 1C2A

Failure 1C2B

FALSE 1C2B1

Error code: 1C2B2

Document does not exist. 1C2B2A

Document not accessible 1C2B2B

Function: 1C3

Locates document, and returns TRUE if it is accessable on immediate access storage. 1C3A

If document is in transit from another retrieval request, waits (or sets up background process to wait) on the completion of that retrieval. 1C3B

Otherwise, initiates retrieval of document from storage. 1C3C

If background retrieval is specified, then a TRUE return is made when the request is successfully made to the background system. 1C3D

If a real time retrieval is requested, then does not return until retrieval is successfully completed. 1C3E

(EIB) Entity Information Branch 2

Each entry in the master catalog has a sub-branch termed the entity information branch. 2A

This branch contains the information necessary to identify and retreive the physical document belonging to the catalog entry. 2B

The Information contained in this branch is: 2C

(1) Document locator 2C1

This is either a link to a a mother document (a document of which this document is a part), or a flag indicating that this document is a physical file. 2C1A

(2) Access control information 2C2

This is information indicating who may have read and/or write access to the document. 2C2A

There are a number of ways to restrict and enable access, such as passwords, bit tables, etc. 2C2B

One which seems as though it may be particularly appropriate here is to use identlists (see--Journal,6203, O:gw> or <Journal, 6215, O:gw>) for enabling or disabling access to persons or groups. 2C2C

(3) Use Statistics 2C3

Use statistics may be used in the evaluation of algorithms for moving documents between secondary and tertiary (and any other) storage levels and devices, as well providing information about system usage. 2C3A

Included in use statistics are: number or frequency of read/write accesses; Number of or frequency of access faults; information about nature of access (e.g. in bursts, who, etc). 2C3B

(4) If document is a physical file, then: 2C4

Access copy physical name. 2C4A

This is the TENEX file name of a copy of the file which is on readily accessable storage, e.g. disk. 2C4A1

If there is no such 'access copy', the field is empty. 2C4A2

Master copy information. 2C4B

TENEX File name. 2C4B1

Includes device name if other than disk. 2C4B1A version is always assumed to be the most recent one. 2C4B1B

Address of file. 2C4B2

Includes reel number if on tape, perhaps tape position, etc. 2C4B2A

May include a field indicating installation as part of address. 2C4B2B

Disposition of physical file, e.g. beng read/written from/to tape, etc. 2C4B3

It is assumed that when we begin using storage made available via the network, each 'storage facility' offered will be treated as a new device. 2C4B4

Estimated access time. 2C4B5

An estimate of the time required to access the master copy, including tape mounting, etc. 2C4B5A

Initial Spec for NLS/User-File system Interface Routines

<JOURNAL>6256.NLS;2, 3-MAY-71 9:10 WSD ; (Expedite) Title: Author(s): William S. Duvall/WSD; Distribution: Charles H. Irby/CHI; Keywords: File System Interface; Clerk: WSD;

William H. Paxton
Augmentation Research Center
Stanford Research Institute
Menlo Park, California 94025

To:
Access Copy

6259

Modular Programming System 1

Introduction (by DCE) 1a

A fully developed augmentation system of a few years hence will have a very large repertoire of commands, representing a rich vocabulary for eliciting help from the computer system. To experiment meaningfully with any one subset of commands, designed to support a special kind of intellectual task, the evaluation must rightfully be done within a working environment in which the subjects are doing all of their associated work in the way they would do it in the "complete workshop." 1a1

This means that to provide a progressive research environment in which rapid and significant evolution can take place, some sort of a "latest thing in complete workshops" must be maintained as a laboratory for each experimenter. To maintain this in separate installations is quite impractical. 1a2

The computer network offers an important hope here, in that it makes it possible for people at distributed locations to share a "latest thing in complete workshops" as an environment for their different, specific "tool-development experiments." 1a3

For several years ARC has been aiming toward an experimental future where this was the way in which our work on augmentation systems would be done--as part of a larger community in which many more people than we could marshal would be working on different fronts (and at different levels). 1a3a

For instance, much of our motivation toward the Dialog Support System has been to facilitate close collaboration between such distributed system-development participants. 1a3b

Besides being able to sustain collaborative dialog, the participants would be much helped if each could view a relatively stable system as the background in which he experimented with a new tool, and if he could very rapidly and independently create and modify new tool features. 1a4

Our launching of a Modular Programming System was done explicitly to serve this end. When NLS has been modularized, it will be possible for instance to permit a worker at Utah to be given "custodianship" of a private subset of modules pertaining to the manipulation of one kind of graphic-data packet in our file data nodes. 1a5

He would be given his private copies of the source-code files for these modules, and could add and/or modify them at will. His modules can be independently compiled by him at any time; and when he wishes to experiment with the resulting "new tool," his compiled modules can be linked into the rest of the NLS compiled-code module set at run time, perhaps in place of some modules that the standard version of NLS offers but that he is redoing. 1a5a

To experiment with his tool, he can use it in the midst of processes, methods, and information that are part of a busy (and evolving) working life in the whole workshop. 1a5b

Each person can do his private development with minimal burden on the support system, and with maximal protection to the other workshop users. 1a5c

The standard-NLS Module Set would be controlled and updated by a central community process, steadily integrating the improvements of the trial tools as they become thoroughly checked out. 1a5d

Objectives 1b

1: Development of state-of-the-art tools for system evolution 1b1

These tools make up a "Modular Programming System" which is intended to contribute to 1b1a

a: Improved design--modularity 1b1b

Control structures that encourage modular design 1b1b1

runtime entities, called processes, correspond to modules 1b1b1a

control is passed among the processes 1b1b1b

Control structures that encourage simple interfaces 1b1b2

processes communicate by messages sent over "ports" 1b1b2a

b: Improved debugging 1b1c

Source language debugging 1b1c1

Integrated with NLS 1b1c2

c: Ease of modifications 1b1d

Incremental compilation 1b1d1

Dynamic linking and symbolic references 1b1d2

Change single module rather than reloading entire system 1b1d3

d: Ease of development 1b1e

Make new configuration of existing modules rather than reprogram 1b1e1

Can develop a library of modules 1b1e2

e: Effective use of address space 1b1f

Modules relocatable in address space 1b1f1

Mapped in only as needed, not loaded otherwise 1b1f2

Certain segments may be swapped out of address space and reloaded when next accessed 1b1f3

Dynamically establish and modify configuration 1b1f4

f: Flexiblity--from "virtuality" of external references 1b1g

Modules communicate via ports 1b1g1

May connect any other module to the port 1b1g2

May replace any module by another that satisfies the interface requirements 1b1g3

Connection of ports binds the "virtual facilities" of the module to real facilities 1b1g4

The formation of these connections (and thus the binding of facilities for the module) is not only delayed until run time but may even be changed as the module runs 1b1g5

g: Reliability 1b1h

Simpler relationships and dependencies 1b1h1

Better definitions of interfaces 1b1h2

Ability to put a module into a test environment for verification 1b1h3

2: new version of NLS implemented with these tools 1b2

3: development of a library of modules 1b3

4: NLS usable as a base for application systems 1b4

5: NLS usable as an interface to other facilities 1b5

a: to other local processes (in the TSS sense) 1b5a

to give limited parallelism 1b5a1

b: to remote processes via the Network 1b5b

These programs may be 1b5b1

parts of NLS running on another TENEX, or 1b5b1a

special facilities available at specific sites. 1b5b1b

6: system evolution franchising 1b6

A main objective of CSES is to allow collaboration among distributed sites in the development of large systems. 1b6a

The preceeding objectives are oriented toward making it possible to give a "franchise" to some group to develop a portion of a system. 1b6b

Tasks 1c

1: Modular Programming System 1c1

This task includes the design, implementation, and documentation of a system providing the tools mentioned as the first objective for CSES. 1c1a

2: NLS in MPS 1c2

There will be a minimal redesign of NLS to be compatible with MPS and a rewriting of NLS in MPL. 1c2a

3: library of modules 1c3

This envolves the development of a library of modules which are used in NLS and related systems. This will require a more through study and reworking of the design of NLS in order to take better advantage of MPS. These modules should be as general as possible so that they can be used as components in the development of other systems. Complete documentation of each module is of course necessary. The development of retrival methods, standards, etc. for the module library is in the domain of SEAS and DSS. 1c3a

4: MPS using NLS modules 1c4

After NLS modules are available, MPS can be modified to make use of them in order to access NLS files directly for source language debugging and incremental compilation. 1c4a

5: applications based on NLS/MPS module library 1c5

In order to gain experience using the module library to construct systems, some application systems should be developed by ARC programmers. 1c5a

This will help to point out further generalizations of modules and shortcomings in the organization of NLS in MPS. 1c5b

It will also be valuable as a test of the facilites for accessing the module library. 1c5c

6: development of interface modules 1c6

a: modules which allow the creation of subprocesses (in the TSS sense) so that other programs can be executed in parallel. 1c6a

b: modules providing interface to remote processes via network 1c6b

7: subsets of NLS in parallel operation 1c7

Using the interface modules developed above, allow certain operations (such as Output Processor) to be executed in parallel with normal NLS operation. 1c7a

8: subsets of NLS at remote sites 1c8

Using the interface modules developed above, allow certain operations to be executed at remote sites.

These will probably be other TENEX installations at which MPS can be used. 1c8a

A "front end" to NLS, providing for interaction with the user, which can be used at other sites should be one goal of this task. 1c8b

9: experiments in franchising 1c9

(J6259) ; Title: Author(s): William H. Paxton/WHP; Distribution: Douglas C. Engelbart, Richard W. Watson, James C. Norton, Charles H. Irby/DCE RWW JCN CHI; Sub-Collections: SRI-ARC; Clerk: JCN; Origin: <NORTON>J7661.NLS;2, 18-OCT-71 8:41 JCN ; .RTJ=0; 18-OCT-71 9:25;

this is a test 1

(J6279) 15-APR-71 12:52; Title: Author(s): William S. Duvall/WSD; Clerk: WSD;

JCN 15-APR-71 15:04 6282

No good title

Let's try to have another NLS session soon, OK? 1

No good title

(J6282) 15-APR-71 15:04; (Expedite) Title: Author(s): James C. Norton/JCN; Distribution: Hardeman/BAH; Keywords: ; Clerk: JCN;

this is a test 1

(J6292) 16-APR-71 10:56; Title: Author(s): William S. Duvall/WSD;
Clerk: WSD;

test message 1

(J6293) 16-APR-71 11:02; Title: Author(s): William S. Duvall/WSD; Clerk: WSD;

Some Journal Changes

Minor Modifications to the Journal System 1

Some minor modifications have been made to the file AUTOJO and JNLDEL to permit a better formatting of messages. 1A

1. Major comments directed to all addressees will now appear on a page separate from the header. A modification in the procedure setjcomment was made. 1A1

2. Page numbering will now begin on the first page of the Journal document. This required some modifications in the output Processor because the PGN directive applied within a statement and not within a page. Walter now permits negative values of PGN. We start printing the number when it becomes greater than zero. Changes were made in setjcomment and setjwork. 1A2

3. The number of lines between the address and return address on the distribution copy has been increased by three in the file JNLDEL. 1A3

Some Journal Changes

<JOURNAL>6348.NLS;1, 16-APR-71 12:41 HGL ;Title: Author(s): Harvey G. Lehtman/HGL; Distribution: William S. Duvall, Richard W. Watson, James C. Norton/WSD RWW JCN; Keywords: Journal distribution ; Clerk: HGL;

Initial Baseline Record System

This file is a description of the initial Baseline Record System. Please note the word initial. This system has a great deal of evolving to do. I would appreciate a lot of dialogue about it. 1

The widespread use of the BRS (Baseline Record System) will require a fair sized change in the working methods of almost all of us -- which is a painful process. Unfortunately the BRS will be of little value unless it is widely used and worked on. Plans that are perhaps voluminous, but which are not up-to-date, are very nearly useless. 2

Thus the biggest problem with the BRS will probably be choosing a balance between the amount of information people want to have available and the amount which people are willing to write down and keep up to date. 3

There will be several things asked of people in the running of the BRS experiment: 4

The biggest thing is that everyone is asked to invest a fair amount of effort in entering and updating information in the BRS. 4A

Another is that people who use BRS information make the value the information has for them visible to others. This visibility could be a motivating force for people to make up-to-date information available. 4B

Another thing that will be asked (primarily of me I guess) is that there be a set of tools (NLS commands mostly) that make entering and updating information as painless as possible and a set of tools to aid getting various views and summaries of the information. 4C

There are several parts to the BRS: 5

The Baseline Record which is a collection of branches. Each branch contains information about one "task". 5A

The conventions for what information is to be kept about each task and the format of that information. 5B

The conventions about entering and updating task information. 5C

The tools that aid entering and updating information. 5D

The tools that aid in viewing and summarizing the information in the Baseline Record. 5E

Perhaps the Completed Tasks file(s) should be considered part of the BRS. 5F

Perhaps the Needs and Possibilities file(s) should be considered part of the BRS. 5G

The current state of each of the parts is: 6

The INITIAL Baseline Record exists in the file (MSR, BASEREC, :wh). Eventually (or maybe even soon) the Baseline Record will probably be dispersed over many files -- each person having custody of all the tasks for which he is the pusher. 6A

The conventions about the information to be kept and its format are described below for the initial BRS. These conventions are subject to much change. 6B

The conventions for updating information are (tentatively): 6C

The pusher of a task is responsible for keeping the information of that task up-to-date. 6C1

Each pusher should check his tasks at least once a week. 6C2

If schedules change, it would be nice if the Baseline Record reflected those changes soon after the schedule change, particularly if any other tasks would be affected by the change. 6C3

If schedules are changing very rapidly, maybe the convention would be that the Baseline Record were assumed to be up-to-date only on the 13th of each month or every Friday at midnight or something. I'd like to experiment with this. 6C3A

I would like everyone to make a pass over (MSR, BASEREC,) as soon as possible. Please add any tasks that you're planning on doing and check and update the information of the tasks in which you're involved. Feel free to add or update ANY information. Please do an Output File when finished updating. Feel free to ask me any questions. 6C4

There are no tools currently planned to aid entering and updating information with the exception of a branch named Template in (MSR, BASEREC,). Please let me know of any ideas you have (by means of the Journal?). 6D

There is one tool currently planned to aid in viewing and summarizing the information in the Baseline Record. The tool will be an Execute Evaluate Set command. This command will be the subject of a future (soon) memo. 6E

The Completed Tasks file(s) exists as the file (MSR, DONETASK,). The format of the entries in that file will be the subject of a future memo. For now just move the task branch from BASEREC to DONETASK. 6F

The Needs and Possibilities file(s) exists as the file (MSR, NP,). The format of the entries in that file will be the subject of a future memo. For now use the same format as in BASEREC. 6G

7

The following two branches describe the format and kind of information for a task branch in the Baseline Record: 8

(Task'Name) m/w m/w [day] <who 9

Info: This branch should contain at least a description of the task. It may contain anything else anyone wants to put here that does not fit any place else. This branch should always be present. 9A

Buyers: This branch would contain the names of all the "buyers" of this task. As a start I suggest that the name of a buyer is either a goal of ARC or a link to a task branch. See memo of Doug handed out at meeting of 4/14 for things that can be considered ARC goals. 9B

Sub-Cons: This branch should list all tasks (as links) which are "Sub-Contractors" for this task. (All Sub-Contractor task branches should list this task as a Buyer.) A Sub-Contractor task is probably any task upon which this one is dependent. 9C

Require: This branch would contain the "Requirements" for this task, usually as a link. The Requirements are the (functional) specs for this task. The Requirements are written by the Buyer. 9D

Design: This branch would contain links to any other files pertinent to this task. The (proposed) Design for satisfying the Requirements should be linked to here (or written here if its short). 9E

Dates: A list of any dates when portions of the task are to be finished that may be of interest to people. 9F

Cost: An estimate of the cost of the task: peoples time in terms of man-weeks and any new hardware in terms of $. Later we may wish to have more detail here such as CPU time, file space, supplies, etc. 9G

Subtasks: Each branch of the substructure of this statement would have the same format as a task branch except that none of its sub-branches need be present. 9H

People: The sublist of this statement would consist of one statement for each person listed under who for the task. Each statement might be of the form: 9I

m/w m/w [%-time] <who 9I1

Conventions: 10

The name of the task should be a visible (use single quotes instead of spaces) of less than 28 characters inclosed in parens. 10A

m/w represents a date in the form of the wth week of the mth month, e.g., 4/1 means the 2nd week of the 4th month. 10B

The two m/w's after the task name are the begin-week (date work on the task is scheduled to begin) and the end-week (date the task is estimated to be completed). 10C

??? for a begin- or end-week means the start or completion of the task has not yet been scheduled 10D

--- for a begin- or end-week means the task started before this file was begun of the task goes on forever 10E

[day] means the task is scheduled for completion on that day of the week -- usually will only be present during the week the task is to be completed 10F

<who means a list of those people who are to work on the task -- the first name is the "task pusher" 10G

<??? means nobody is yet slated to work on that task 10H

See (MSR, BASEREC, :hdbnsj) for examples. 10I

It is not necessary that all the information be filled in, particularly for small tasks. The conventions for now are: 10J

the task name, begin-date, end-date, and who must always be present even if the last three are question marks 10J1

the [day] thing need never be present 10J2 the Info: branch must always be present since task names are usually not descriptive enough 10J3

when we become more organized and formal and so on, every task must have Buyers:, Require:, and Design:, branches; since it would require a lot of time to complete all of those right away, why don't we let them be filled out as people have time for it; it would be nice to get everything filled out soon 10J4

the Sub-Cons: branch would only be present if there are Sub-Contracts 10J5

the Dates: branch would probably only be present if the task were fairly long and somebody wanted to know when some stage of it would be completed (like if they were dependent on that part of it) 10J6

the Cost: branch should always be present; it is a great aid in figuring out people schedules and eventually money things 10J7

the Subtasks: and People: branches may be present anytime anyone wants them 10J8

Unanswered Questions (on which I would appreciate dialogue): 11

What information described above isn't necessary or is too much of a drag to keep up to date. 11A

What other information ought to be kept in the Baseline Record. 11B

What is a Task, a Sub-Task, a Sub-Contracted task, etc. 11C

Where is the dividing line between a task being in the Needs and Possibilities file and in the Baseline Record. 11D

Who or what are possible Buyers? If the idea of ARC goals being the ultimate Buyers is accepted, what is on the list of out gaols. 11E

What tools are viewing and updating the BRS. 11F

<JOURNAL>6349.NLS;1, 16-APR-71 13:00 BLP ; (Expedite) Title: Author(s): Bruce L. Parsley/BLP; Distribution: Hardeman, Marilyn F. Auerbach, Walter L. Bass, Roger D. Bates, Mimi S. Church, William S. Duvall, Douglas C. Engelbart, Martin E. Hardy, J. D. Hopper, Charles H. Irby, Mil Jernigan, Harvey G. Lehtman, John T. Melvin, Jeanne B. North, James C. Norton, William H. Paxton, Ed K. Van De Riet, Dirk H. Van Nouhuys, Kenneth E. Victor, Don I. Andrews, James A. Fadiman, Richard W. Watson/BAH MFA WLB RDB MSC WSD DCE MEH JDH CHI MEJ HGL JTM JBN JCN WHP EKV DVN KEV DIA JAF RWW; Keywords: baseline record; Clerk: BLP; Origin: <MSR>PLANP.NLS;13, 16-APR-71 10:25 BLP ;

Some Thoughts on System Measurement

I have had Mil copy and send to you some papers which I thought might help you in planning your attack on system measurement and tuning. Some of the techniques also yield useful information for finding bugs. The materials being sent includes: 1

3 page from my book summarizing some basic ideas 1A

Campbell, D.J., and Heffner: Measurement and Analysis of Large Operating Systems During System Development AFIPS, FJCC, 1968 1B

Cantrell, H.N., and A.L. Ellison: Multiprogramming System Performance Measurements and Analysis, AFIPS, SJCC, 1968 1C

Bussell, B., R.A. Koster: Instrumenting Computer Systems and Their Programs, AFIPS, FJCC, 1970 1D

Saltzer J H, and J W Gintell, The instrumentation of MULTICS, Comm. ACM Aug. 1970. 1E

There are two or three other papers which may be of interest which I will have copied for you. 1F

The key point is that in the past 3 years a lot of people have thought about hardware and software techniques for tuning and measuring and we should be plugged into this literature. My own feeling is that besides adjusting various parameters and certainly before we rewrite much code or change our hardware configuration, we should have a thought out plan of hardware/software measurement The experience of most groups I know who have done measurement is that the bottlenecks were not where they thought they were. For what it's worth, I wouldrecommend a hierarchical approach. First find percent time in system, user, idle. Second, find various causes of idle-waiting for drum, disc, no process ready to run, etc. Then percent time in key system areas scheduler, memory allocation, file sys etc. Then study queues etc inside these areas. I am sure you have already thought of all this, but maybe it was worth restating. Although I do not know TENEX yet, if I can help in any way, I would like to--if nothing else than as an ear to clarify any of your thoughts. In any case, because this is an area of interest to me, I would appreciate being included in the distribution on any dialog you have on this subject. Thanks. 2

Some Thoughts on System Measurement

<JOURNAL>6350.NLS;1, 16-APR-71 14:13 WSD ; (Expedite) Title: Author(s): Richard W. Watson/RWW; Distribution: Don I. Andrews, Kenneth E. Victor, Roger D. Bates, Ed K. Van De Riet, John T. Melvin, Douglas C. Engelbart/DIA KEV RDB EKV JTM DCE (I didnn't send you the papers); Clerk: WSD;

Visit by Al Sassus

I have invited Al Sassus of Shell Development to come down Friday April 23 to talk to us about some of his experiences with system measurement and tuning. 1

Al has spent the last 2 years thinking about accounting and system measurement and he spent last year using hardware and software techniques to tune Shell's 360/65 running OS/MVT. Hopefully some of his experience may be relevant to us. 2

He will arrive at noon to have lunch with us before his talk. 3

Visit by Al Sassus

<JOURNAL>6353.NLS;1, 16-APR-71 16:17 RWW ; (Expedite) Title: Author(s): Richard W. Watson/RWW; Distribution: Don I. Andrews, Kenneth E. Victor, Ed K. Van De Riet, John T. Melvin, Charles H. Irby, Roger D. Bates, Douglas C. Engelbart, James C. Norton/DIA KEV EKV JTM CHI RDB DCE JCN; Clerk: RWW;

WSD 22-APR-71 20:44 6356

Comments on 6349--Baseline Record System

Bruce: 1

With regard to the initial baseline record system (Journal, 6349,) I have these immediate comments: 1A

I strongly advocate that 'identifications' be used to identify people within BASEREC. 1A1

I notice that this is not initially the case. 1A1A

There are a lot of advantages of using a standard identification system which is un-ambiguous, and an initial one is in use by the Journal and catalog systems. 1A1B

If you object to using 'Initials' (as I believe you once said you did) provision is made for an arbitrary identification string, however I think that it would require some discussion before abandoning initials. 1A1C

I refer you to (journal, 6203,) and (Journal, 6215,) for further discussion of the identification system. 1A1D

As an additional incentive, convenient handles may be provided to the NLS user level L10 programmer for manipulating idents at a nominal cost. 1A1E

re (Journal, 6349, 9b:gw)--Buyers and ARC goals 1A2

The document distributed by DCE on 4/14 did not contain a list of ARC goals. 1A2A

Rather, it contained a few of the goals which were currently of highest priority to ARC. 1A2B

(Journal, 6349, 9c:gw) Subcontractors 1A3

This seems a prime candidate for back-links, which would allow us to automatically generate a sub-contractor list from the buyer list. 1A3A

(Journal, 6349, 9d:gw) Requirements. 1A4

As a semantic point, I have always considered functional specs to relate more to the design rather than the requirements. 1A4A

(Journal, 6349, 9g:gw) Costs 1A5

The estimating of costs is a difficult task in itself, and I think that we could profit greatly by attempting to develop a technique in this area. 1A5A

In this direction, it seems desirable to avoid 'seat of the pants' estimates, and rather devise specific estimating techniques, which may be tested and the subsequent results evaluated as to accuracy. 1A5B

I think that if we systematically attack this problem, a satisfactory technique may be developed with relatively few evolutional generations. 1A5C

I have a number of ideas to throw in to the initial pot if we decide to work on this. 1A5D

(Journal, 6349, 9h:gw) Sub tasks 1A6

Is it possible to eliminate this category by treating subtasks as sub-contractors?? 1A6A

Probably not, but it seems worth considering. 1A6B

(Journal, 6349, 10a:gw) Task Names 1A7

If a task were given a group identification (see--journal,6215,), that could be used for naming the corresponding branch in the baseline record, andthen the user would not have to muck around with single quotes instead of spaces. 1A7A

We could get some other benifits from this too, in the nature of automatically directing Journal stuff towards that task, which included in its list the idents of concerned and affected people/tasks. 1A7B

And so on. 1A7C

Besides, a ' is not a legal character in a name field (allowed, yes, but not legal) 1A7D

And then we would not have to limit names to 28 characters. 1A7E

(Journal, 6349, 10b:gw) Dates 1A8

Lets keep a standard date format throughout all of our stuff. 1A8A

It will make things much easier when we write processors to do our work for us. 1A8B

If we need an abbreviated way of typing in a date, lets create a special NLS command to help us. 1A8C

Perhaps this can be related to NLS text macros (see -- Journal, 5274,) 1A8D

Or mebbe to deferred execution/executable text 1A8E

(Journal, 6349, 10g:gw) <who 1A9

I don't understand why we need the '< there??? 1A9A

Comments on 6349--Baseline Record System

<JOURNAL>6356.NLS;2, 22-APR-71 20:44 WSD ; (Expedite) Title: Author(s): William S. Duvall/WSD; Distribution: Bruce L. Parsley, Douglas C. Engelbart, James C. Norton/BLP DCE JCN; Keywords: baseline Record Comment; Clerk: WSD;

File System Baseline Plan 1

Overview 2

The file system as is currently envisioned is a component of the master catalog system. 2A

Its functions are: 2A1

Provide an interface through which documents listed in the master catalog may be accessed. 2A1A

We use access in the sense of 'being made available and present' for reading and or writing. 2A1A1

Thus, if stated simply, the interface accepts from a program or user a document name, and instructions to locate it, replace it, etc. 2A1A2

Provide a library for manipulating documents and master catalog entries with regard to access, physical location and disposition, and physical maintenance of documents. 2A1B

This includes the controlling of privilidged access, allocation to secondary, tertiary, etc. storage, archiving, and so on. 2A1B1

For the sake of convience, we will consider the library as having a part for the handling of physical files (e.g. tape routines, copy routines, etc.); a section which is devoted to manipulating the master catalog entries; and a locator section, which is capable of determining the physical configuration of a document from a catalog entry. 2A1B2

Eventually, all files within the NLS environment will be represented in the master catalog system, and the services of the file system will apply to all of these files. 2B

Many of the documents will be elements of one or more sub-collections. 2B1

In order to facillitate occaisional special handling of files within the scope of the subcollection, extensions of the file system may be built. 2B2

For this reason, the central file system will be built so as to serve as a library not only to NLS or user functions, but also to extensions of the file system itself. 2B3

Extensions should be planned and incorporated in a manner such that they may be in turn used by other extensions in a bootstrap manner. 2B4

Stage 0 File System 3

As a starting point, a stage 0 file system is proposed which is oriented towards providing the basic file system library, and extending it sufficiently to allow the automatic loading of cataloged documents when accessed by catalog number within the environs of NLS. 3A

The most significant restriction of the stage 0 system is that it only allows read access of the documents. 3A1

Significant features are: 3A2

Automatic archiving and retrieval of documents 3A2A

Allowing documents to be branches, plexes, groups, and statements in addition to files, and doing this in a manner such that a user/program need not worry about the structure of the document. 3A2B

Automatically allocating secondary and tertiary storage, with an effort to optimise usage. 3A2C

'Background' access requests for documents whose retrieval time is long. 3A2D

Features which may or may not be included: 3A3

Sets as documents (evaluated or dynamic). 3A3A

This depends somewhat on the set system (see -- msr, baserec, sets); 3A3A1

Features which are by-products. 3A4

Revised NLS file machinery 3A4A

This will hopefully simplify things for the user. 3A4A1

Entities in links. 3A4B

The user will be able to specify and entity in a link down to the statement level. 3A4B1

Not promised yet, but possibly some support for a private archive system if deemed desirable (which is questionable). 3A4C

Brief Baseline and Stage 0 File System Description

<JOURNAL>6357.NLS;2, 23-APR-71 9:42 WSD ; (Expedite) Title: Author(s): William S. Duvall/WSD; Distribution: Bruce L. Parsley, Douglas C. Engelbart, Charles H. Irby/BLP DCE CHI; Keywords: File System Baseline; Clerk: WSD;

Proposal for Including Entity Identification in Link Syntax. 1

In order to implement the Stage 0 File System as is envisioned, it is necessary to allow a link to identify a structural entity as the document containing the statement being addressed. 2

The syntax proposed here does not effect the problem of addressing textual entities by links. 3

Syntax: 4

```
   <link> ::= ('( / '< / "--")  [<collection name> ',] [<entity
   name> ',] [ '[ <entity type> [<address> [', <address>]] '] ] [
   <address> ] [': <viewspecs> ] ')                                    4A
```

Examples: 4A1

(Journal, 6202, 0:gw) 4A1A

(Journal, jrnl1, [Branch j1234]:gw) 4A1B

(collection, filname, [Group 1a, 1b] 2:w) 4A1C

Discussion 4A2

<collection name> ::= <user name> / <sub-collection name> 4A2A

The collection name identifies the collection to which the entity named by te <entity name> belongs. 4A2A1

When a link is parsed, a check is first made to see if the collection name is on of those recognised by the Master catalog system. 4A2A2

If it is, then the entity name is assumed to be the name of an entity belonging to that subcollection, and the link is decoded using the File System machinery for handling the particular sub-collection (see--Journal, 6357, 2b2:gw). 4A2A3

If the collection name is not recognised by the File System, it is assumed to be a TENEX user name, and is processed according to the existing link machinery. 4A2A4

<entity name> ::= $LD 4A2B

The entity name identifies the entity (file or document) containing the entity being referenced by the link. 4A2B1

For TENEX files, the entity name will be aa physical file name. 4A2B2

For documents within the Journal collection, the entity name will normally be a catalog accession number. 4A2B3

We have to handle the exception cases created by the links to messages in the 'JRNL1' etc. files. 4A2B3A

Mebbe we should convert those existing links by hand, if there are not to many of them?? 4A2B3B

<entity type> ::= 'B/ 'G/ 'P/ 'S/ 'F/ "ES"/ "DS"/ "Branch"/ "Group"/ "Plex"/ "Statement"/ "File"/ "Evaluated Set"/ "Dynamic Set"/ EMPTY 4A2C

The entity types have the obvious meanings. 4A2C1

If no entity type is specified, "File" is assumed. 4A2C2

The addresses immediately following the entity type inside of the square brackets identify the entity with respect to the containing entity (that identified by collection name and entity name). 4A2C3

If there are no such addresses, then the origin of the containing entity is assumed for all necessary addresses. 4A2C4

If there is one such address, and the entity requires more that one address, the value of that address is assumed for all absent fields. 4A2C5

Current statement address 4A2D

The address immediately preceding the viewspec field identifies the statement which is addressed by the link, i.e. that which will be set at the top of the display in DNLS, or the value of the CSP in TNLS. 4A2D1

This address is interpreted with respect to the entity identified by the link. 4A2D2

If the entity is a branch or statement, then the origin is the top statement in that branch, 1 is the first down, etc. 4A2D2A

IF the entity is aa group or plex, the head statement is assumed to be statement 1, and all other statements are relative accordingly. 4A2D2B

In this case, a dummy 'origin' statement for the document is created. 4A2D2B1

If the entity is a set, then the origin is assumed to be the highest statement in the evaluated set (which contains the evaluation information, see--journal, 6207, 3b1a:gw) 4A2D2C

If the address is omitted, the relative origin is assumed. 4A2D3

This syntax is compatible with the current link syntax, so no conversion is necessary. 5

The syntax proposed here distorts the aesthetic qualities of the link parsing machinery. 6

Currently, the link is parsed from right to left, and is context dependent. 6A

This introduces, in essence, a left to right context free element. 6B

It is difficult to avoid this if we wish to make the link syntax seem natural to the user, while being upward compatible with the existing syntax. 6C

One of the possibilities explicitly allowed by the proposed syntax is that the entity described by a link need not be a file, but rather need only be an identifiable entity. 7

This feature is used by the file system to assure that there is only one catalog entry for a physical file. 7A

When links can point to other links and be executed with indiriction, this will allow an arbitrary level of relativeness to a structure. 7B

Link Syntax for Identifying Structural Entities

<JOURNAL>6358.NLS;2, 23-APR-71 11:06 WSD ; (Expedite) Title: Author(s): William S. Duvall/WSD; Distribution: Bruce L. Parsley, Charles H. Irby, Douglas C. Engelbart, Mimi S. Church/BLP CHI DCE MSC; Keywords: Link Structural Entities; Clerk: WSD;

RWW 23-APR-71 12:45 6637

Summary of TNLS Features and Commands

Marilyn Auerbach has custody of this file. Please report any errors to her . If the system does not behave as described here please report this to Charles.

Syntax Equations 1

Syntactic entities are indicated by strings of upper case letters. These entitites are either defined in the body or are standard NLS terminology assumed known. 1A

The following characters have the indicated meaning in syntactic definitions:

Symbol	Meaning
/	OR
[ ]	quantity enclosed is optional
( )	quantity enclosed grouped for convenience in the expression
$	any number of entity following
;	end of syntax equation

1B

When any of the above symbols or letters are to be used literally, they are preceded by an apostrophe '. 1C

Any place we have used lowercase in the syntax equations, uppercase could also be used. Where we have shown uppercase as in Viewspecs, uppercase must be used. 1D

Address 2

An address in TNLS is a statement identifier, character position pair. The syntax of address specification is: 2A

```
ADDRESS = $ (ADDRESSELEMENT [SP] / NOTHING);                    2A1

ADDRESSELEMENT=':(SNAME[SP'f]/SNUM)/
                  LINK /
                 [INTEGER]STRUCREL /
                 ALTMODE/
                 CNTRLMARKER /
                 INDIRECT/
                [INTEGER] '[LIT']['f] /
                [INTEGER]'<LIT'>['f] /
                 LEFTRIGHT /
                MARKER /
                RETURN /
                AHEAD;                                          2A2
```

The first type of address element above must be followed by a SP if other address elements are to follow. 2A3

SNAME = NLS statement name; 2A4

SNUM = NLS statement number; 2A5

INTEGER = integer; 2A6

CHARPOS = character position in a statement; 2A7

STRUCREL = 'u / 'd / 'p / 's / 'h / 't / 'e / 'n / 'b; 2A8

ALTMODE altmode character; 2A9

CNTRLMARKER '.; 2A10

INDIRECT = '↑ [INTEGER]; 2A11

LIT = literal string of characters; 2A12

LEFTRIGHT = ['+/'-][INTEGER]('c/'w/'i/'v); 2A13

MARKER = 'L3LD; 2A14

LINK =
'([USERNAME,FILENAME,/FILENAME,][(SNAME/SNUM)][':VIEWSPECS]
'); 2A15

LD = LETTER/DIGIT; 2A16

RETURN = 'r; 2A17

AHEAD = 'a; 2A18

NOTHING = no character required; 2A19

T1 = CHAR / WORD / VISIBLE / INVISIBLE / LINK / NUMBER; 2A20

T2 = TEXT; 2A21

S1 = STATEMENT/ BRANCH / PLEX; 2A22

S2 = GROUP; 2A23

LEVADJ = 'u/'d; 2A24

Control of the Control Marker 3

There is a pointer in TNLS which at any instance in time is pointing to a statement and a character within the statement. This pointer is called the Control Marker (CM). 3A

As a command is being evaluated several Temporary Control Markers (TCMs) may be created. When the final CA is given terminating a command, one of these TCM's is assigned to the CM. A TCM is initialized to the current value of the CM. 3B

The various kinds of ADDRESSELEMENTS affect the value of a TCM as follows: 3C

SNAME and SNUM cause the TCM to be set to the first character of the indicated statement. 3C1

CNTRLMARKER causes the TCM to be set to the current value of CM. 3C2

STRUCREL causes the TCM to be reset to the first character of a statement relative to the current statement pointed to by the TCM. 3C3

INDIRECT causes the TCM to be reset to a position indicated by the nth LINK from the position in the statement pointed to by the TCM at the point where the INDIRECT is encountered, where n is the integer after the '↑. All further ADDRESSELEMENTS after the INDIRECT reset the TCM to addresses in the file pointed to by the link, except CNTRLMARKER which resets TCM to the current contents of CM. 3C4

'[LIT'] causes search for the string LIT, bounded by any characters or beginning or end of statement from the current value of TCM. When found, TCM points to the last character of the instance of LIT found. 3C5

'<LIT'> causes search for the string LIT, bounded by characters not (letters or digits) or beginning or end of statement from the current value of TCM. When found, TCM points to the last character of the instance of LIT found. 3C6

If '[LIT'] or '<LIT'> are preceded by a number n, then the nth instance is searched for. 3C7

When searching reaches the end of the file, it stops and fails. 3C8

'[LIT'] or '<LIT'> followed by 'f causes search to start at the first statement in the file regardless of what address had been accumulated by TCM up to this point. 3C9

Content searches will not find occurrences of LIT which cross statement boundaries. 3C9A

ALTMODE causes search for the next occurrence of LIT. 3C10

MARKER causes TCM to be set to the ADDRESS stored in the named marker. 3C11

LEFTRIGHT essentially causes the position pointed to by the TCM to be moved left or right in the current statement. Movement is relative by the number of entities indicated by INTEGER. + is assumed if neither + or - is indicated. A number not followed by an entity type is defaulted to mean CHARACTER. The algorithm for movement to entities is: 3C12

If the CM is on the entity, then go left or right to the nth occurrence of the entity as indicated by INTEGER. If the CM is not on the entity, then go left or right to the n+1th occurrence of the entity as indicated by INTEGER. (To be changed) 3C12A

NOTHING causes the current value of the CM to be used as the address. 3C13

TNLS maintains a list organized as a ring of the last 5 intrafile CM values. The elements RETURN and AHEAD move the CM backward or forward in the ring. 3C14

Command Delete: 3C15

One can exit from any command back to the TNLS command level by hitting CD. 3C15A

Explicit Setting of the Control Marker 3D

Syntax: SP (ADDRESS / & / &a / r/a) CA; 3D1

To find out where the CM is pointing is accomplished with two commands. 3D2

Location is printed out in the form SNUM '(CHARPOS') 3D2A

Syntax: '.; 3D2A1

Location by printing the statement and use of '<LF>' just before the character pointed to. 3D2B

Syntax: '/; 3D2B1

Statements Affecting Markers 3D3

Execute Marker List 3D3A

Syntax: 'e 'm 'l CA; 3D3A1

Execute Marker Release 3D3B

Syntax: 'e 'm 'r CA; 3D3B1

Execute Marker Fix 3D3C

Syntax: ('f LIT CA ADDRESS CA /
'e'm('f LIT CA ADDRESS CA; 3D3C1

Links 3D4

Syntax: ('(/'</'='-)[USERNAME,FILENAME,/ FILENAME,]
[(SNAME/SNUM)][':VIEWSPECS](')/'>) 3D4A

Go back one in interfile link with: 3D4B

Syntax: SP '& CA; 3D4B1

Go ahead one in the interfile link ring with: 3D4C

Syntax: SP '&'a CA; 3D4C1

Five links movements are stored. 3D4D

Input/Output Commands 4

Execute Reset 4A

Syntax: 'e 'r CA CA CA; 4A1

Execute Unlock File 4B

Syntax: 'e 'u CA; 4B1

Load File 4C

Syntax: 'l 'f FILENAME (CR/SP/ALTMODE) CR; 4C1

Load 940 File 4D

Syntax: 'l '9 NAME".kdf;"# CR; 4D1

Load Checkpoint 4E

Syntax: 'l/'l'n/'l'o CA; 4E1

Update File 4F

Syntax: 'u CA; 4F1

Execute Insert Sequential 4G

Syntax: 'e'i ('t/'9) ADDRESS LEVADJ FILENAME (CR/ALTMODE) CR 4G1

Output File 4H

Syntax: 'o 'f FILENAME (CR/ALTMODE) CR; 4H1

Output Device printer or teletype 4I

Syntax: 'o 'd ('p FILENAME (CR/ALTMODE) CR/'t) 4I1

Output Processor 4J

Output Sequential 4K

Syntax: 'o's('y/SP/CA'n) FILENAME (CR/ALTMODE) CR 4K1

Output Checkpoint 4L

Syntax: 'o ['c] CA 4L1

Execute File Verify 4M

Syntax: 'e 'f CA 4M1

Output Quickprint 4N

Syntax: 'o 'q FILENAME (CR/ALTMODE) CR 4N1

↑r can be used within a command (to cause retyping of a FILENAME or LIT up to the CM) which has had many characters or words edited with ↑a or ↑w and is not clean on the page as a result. 4O

'↑ prints statement back from the one where CM points 4P

LF prints the Next statement below where CM points. 4Q

Print 4R

```
Syntax:  p S1 ADDRESS CA [VIEWSPECS] CA
         p (S2) ADDRESS CA ADDRESS CA [VIEWSPECS] CA          4R1
```

Output Processor Directives 4S

Output Processor Directives can be placed anywhere in the text of the file. 4S1

Execute Assimilate 4T

Syntax: e a ADDRESS CA LEVADJ CA FILENAME CR CR (S1 ADDRESS CA/S2 ADDRESS CA ADDRESS CA) CA 4T1

Structural Editing 5

All commands operating on statements, branches, plexes, groups can be terminated by C. instead of CA which leaves you in the current command mode at the point after the first ADDRESS selection. We have shown it in the syntax for those commands in which it is probably useful. (as of 4/16/71 not fully implemented) 5A

Append: to from 5B

Syntax: 'a ADDRESS CA ADDRESS CA LIT $ (C. ADDRESS CA LIT) CA; 5B1

Breakstatement: 5C

Syntax: 'b ADDRESS CA LEVADJ CA; 5C1

breaks statement at first following invisible 5C2

Copy/Move: 5D

```
Syntax:  ('c/'m) (T1 ADDRESS CA ADDRESS CA /
                  T2 ADDRESS CA ADDRESS CA ADDRESS CA /
                  S1 ADDRESS CA ADDRESS CA LEVADJ $(C.
                     ADDRESS CA LEVADJ) CA /
                  S2 ADDRESS CA ADDRESS CA ADDRESS CA
                     LEVADJ $(C. ADDRESS CA ADDRESS CA
                     LEVADJ CA);                                5D1
```

Copies or moves TO FROM 5D2

Delete: 5E

'd ((T1/S1) ADDRESS CA /
(T2/S2) ADDRESS CA ADDRESS CA); 5E1

Execute Content Analyzer: 5F

Syntax 'e 'c ADDRESS CA CA; 5F1

Execute Edit 5G

Syntax: 'e 'e ADDRESS CA; 5G1

↑h = backspace character 5G2

↑d = copy the rest of the old text into the new and terminate 5G3

↑e = insert between angle brackets without changing your place in the old text (same as before) 5G4

↑f = copy one character 5G5

↑g = skip up through the character typed following ↑g and type a %. 5G6

↑n = backspace one in old and new text (the same as before) 5G7

↑o = copy up to following character (same as before) 5G8

↑p = skip up to following typed character and type % (same as before) 5G9

↑q = backspace statement in old and new 5G10

↑r = retype line up to this point (same as before) 5G11

↑s = skip one character in old statement, type % 5G12

↑z = copy up through following character 5G13

↑u = copy through end of old statement 5G14

↑bw = backspace word in new statement 5G15

Execute Status: 5H

Syntax: 'e 's CA;; 5H1

Execute Viewchange: 5I

Syntax: 'e 'v CA 5I1

Because the normal type of syntax equation for this command is not easy to read, we adopt the following convention. Identation indicates what can be or must be typed next. 5J

Once one is in Viewchange, one can enter one of four submodes: 5K

'c Character defined: 5L

'f Feedback: 5M

's Shift character defined; 5N

't Text area defined: 5O

One can remain in the above submodes and perform as many definitions as required. 5P

Character defined: 5P1

d CHARACTER 5P1A

```
SP
    'C'A ['n/CHARACTER]
    'C'D ['n/CHARACTER]
    'C'. ['n/CHARACTER]
    'B'C ['n/CHARACTER]
    'B'S ['n/CHARACTER]
    'B'W ['n/CHARACTER]
       SP ['n/CHARACTER]
    'l literal escape
    'n null
    't tab
    SP CHARACTER
    CHARACTER
CHARACTER
CA (required for termination of command)
CA                                                  5P1A1
```

Feedback defined: 5P2

```
'c character
    (INTEGER/NOTHING)
'i indenting -- command
           (INTEGER/NOTHING)
'l level switch
CA                                                  5P2A
```

```
      Shift character defined:                                        5P3

         'l lower case
             CHARACTER
         'u upper case
             CHARACTER
         'c control case
             CHARACTER
              'c character
                   CHARACTER
              'w word
                   CHARACTER
              'p permanent
                   CHARACTER
         'o off
         CA                                                           5P3A

      Text area defined:                                              5P4

         't tabs
             (INTEGER/NOTHING)
         'i indenting
             (INTEGER/NOTHING)
         'l lines per page
             (INTEGER/NOTHING)
         'r rows per total page
             (INTEGER/NOTHING)
         CA                                                           5P4A

   Execute Quit:                                                      5Q

      Syntax:  'e 'q CA;                                              5Q1

Goto                                                                  6

   Syntax:  'g 't CA;                                                 6A

   tries to compile program starting at statement of CM; arms
   content analyser with a program                                    6B

Insert                                                                7

   Syntax:  'i (T1/T2) ADDRESS CA LIT CA /
               (S1/S2) ADDRESS CA LEVADJ CA LIT
               $ (C. LEVADJ LIT) CA;                                  7A

Move:  see Copy                                                       8
```

Replace: 9

```
Syntax: 'r  T1 ADDRESS CA (('y/CA/SP)  LIT /'n ADDRESS CA) CA/
            T2 ADDRESS CA ADDRESS CA (('y/CA/SP) LIT /'n
ADDRESS CA ADDRESS CA CA) /
           S1 ADDRESS CA (('y/CA/SP) LIT/'n ADDRESS CA)
               $ (C. LEVADJ LIT) CA
            S2 ADDRESS CA ADDRESS CA
              (('y/CA/SP) LIT /'n ADDRESS CA ADDRESS CA)
               $ (C. LEVADJ LIT) CA;                          9A
```

Substitute 10

```
Syntax:  's S1 ADDRESS CA (new) LIT CA (old) LIT $(C. LIT CA
LIT) CA;         S2 ADDRESS CA ADDRESS CA LIT CA LIT $(C. LIT
CA LIT) CA;                                                   10A
```

Transpose: 11

```
Syntax:  't (T1/S1) ADDRESS CA ADDRESS CA CA /
           (T2/S2) ADDRESS CA ADDRESS CA ADDRESS CA ADDRESS CA
CA;                                                           11A
```

Viewspecs: 12

```
Syntax:  'VIEWSPECS v CA;                                     12A

   VIEWSPECS = $VIEWSPECELEMENT
   VIEWSPECELEMENT = LEVEL/LINE/TOGGLE
   LEVEL = 'a/'b/'c/'d/'e/'x/'w
   LINE = 'q/'r/'s/'t/'x/'w
   TOGGLE = 'g/'h/'l/'i/'j/'k/'m/'y/'z/'A/'B/'C/'D/'K/        12A1

   SEMANTICS                                                  12A2

   Level:                                                     12A3

      a L←L-1                                                 12A3A

      b L←L+1                                                 12A3B

      c L←All                                                 12A3C

      d L←1                                                   12A3D

      e relative                                              12A3E

      x L and T←1                                             12A3F
```

w L and T←ALL 12A3G

Lines: 12A4

q T←T-1 12A4A

r T←T+1 12A4B

s T←ALL 12A4C

t T←1 12A4D

Toggles: Display mode 12A5

g branch only 12A5A

*h normal 12A5B

l plex only 12A5C

Content Analyzer 12A5D

i on, display only statements which pass content analyzer program filter when Content Analyzer is given. 12A5E

*j off 12A5F

k reverse, display statements which fail to pass content analyzer program filter. 12A5G

Statement numbers 12A6

m on 12A6A

* n off 12A6B

blank lines 12A7

y on 12A7A

* z off 12A7B

identation 12A8

* A on 12A8A

B off 12A8B

statement names 12A9

* C on 12A9A

D off 12A9B

statement signatures 12A10

K on 12A10A

*L off 12A10B

* is default setting 12A11

X Set 12B

```
Syntax:  x (m (c/l)CA)/
               (T1/STATEMENT) ADDRESS /
                T2 ADDRESS CA ADDRESS CA          12B1
```

Other parts of NLS accessible from TNLS which have their own syntax 13

Journal System 13A

Content Analyzer 13B

Calculator 13C

Output Processor 13D

Collector-Sorter 13E

RWW 23-APR-71 12:45 6637

Summary of TNLS Features and Commands

<JOURNAL>6637.NLS;2, 23-APR-71 12:46 RWW ; Title: Author(s): Richard W. Watson/RWW; Distribution: Walter L. Bass, Roger D. Bates, Mimi S. Church, William S. Duvall, Douglas C. Engelbart, Martin E. Hardy, Fred P. Hocker, J. D. Hopper, Charles H. Irby, Mil Jernigan, Harvey G. Lehtman, John T. Melvin, Jeanne B. North, James C. Norton, Bruce L. Parsley, Ed K. Van De Riet, Richard W. Watson/WLB RDB MSC WSD DCE MEH FPH JDH CHI MEJ HGL JTM JBN JCN BLP EKV RWW; Clerk: RWW;

QUARTERLY MANAGEMENT REPORT 4

(covering the period 9 October 1970 through 8 April 1971)

Other QMR's will be entered retroactively soon

1

1A

ARPA Order Number: 967, Program: 1A1

Title: Network Information Center and Computer-Augmented Team Interaction 1A2

Contractor: Augmentation Research Center, Stanford Research Institute 1A3

Date of Contract: 9 February 1970 1A4

Amount of Contract: $2,410,480 1A5

Contract Number: F30602-70-C-0219 1A6

Principal Investigator: Dr. Douglas C. Engelbart, phone (415) 326-6200, ext. 2220 1A7

Contract Expiration Date: 9 May 1972 1A8

1A9

I RESEARCH PROGRAM AND PLAN 1B

As per our proposal and contract, work is progressing in the following areas: 1B1

A. Network Participation 1B2

Further development of the Network Operating System 1B2A

Development and operational administration of the Network Information Center (NIC). 1B2B

Use by ARC of the Network facilities as they become available and as appropriate 1B2C

B. Team Augmentation Research 1B3

Development of a user- and service-system design discipline 1B3A

Management techniques to coordinate augmented design teams 1B3B

Special user subsystems to support team collaboration 1B3C

C. Computer Facility evolution and maintenance 1B4

II MAJOR ACCOMPLISHMENTS 1C

The transfer of NLS, the development of special compilers with which it is implemented, and all of its related features is completed to the point where we are now using the PDP-10 NLS system operationally in our day-to-day work. 1C1

The initial Journal system was used operationally throughout the Fall of 1970 and up to the start of final transfer activities to the PDP-10 in February 1971. This initial system proved very useful in operational applications. The Journal system is now transferred to the PDP-10 and has been further improved to provide for more automatic entry and distribution of documents. 1C2

The Network Information Center has continued to be active in supporting the collection and dissemination of relevant information in the Network experiment. 1C3

III PROBLEMS ENCOUNTERED 1D

No major problems 1D1

IV FISCAL STATUS 1E

Estimated expenditures and commitments to date are: $ 1,315,000, excluding computer lease commitments. 1E1

Estimated funds required to complete the work are: $ 1,095,480. 1E2

Estimated date of completion of work: February 9, 1972. 1E3

V ACTION REQUIRED BY THE GOVERNMENT 1F

None 1F1

VI FUTURE PLANS 1G

Concentration of effort in the following areas is planned for the coming quarter. 1G1

A. Network Participation,and development of the Network Information Center. 1G2

B. Journal and Dialogue Support System (DSS) 1G3

C. Further computer facility evolution, including improvement of operational efficiency of the PDP-10 and further organization of the Center to provide reliable service to the ARPA Network through the NIC. 1G4

Submitted by:

D. C. Engelbart, Principal Investigator

1H

Approved:

D. R. Brown, Director Information Science Laboratory 1I

QUARTERLY MANAGEMENT REPORT 4
(covering the period 9 October 1970 through 8 April 1971)

<JOURNAL>6638.NLS;2, 23-APR-71 12:56 JCN ; (Expedite) Title: Author(s): Augmentation Research Center/ARC; Distribution: Augmentation Research Center, Augmentation Research Center, James C. Norton, Douglas C. Engelbart/ARC ARC JCN DCE; Keywords: ; Clerk: JCN;
Origin: <NORTON>QMR4.NLS;1, 23-APR-71 12:45 JCN ;

Further notes on DEX-- a proposal

Further notes on DEX-- a proposal

This is a modification of an earlier note concerning the implementation of DEX. The primary changes are in the section on the implementation of the first phase of the system. This section is now relativly complete and is submitted as an official proposal. Also included is a discussion of some basic design principles and needs. Later design proposals for later phases will appear in this file. It will be entered intothe Journal when the Journal is again operational. 1

Note that all of this is subject to modifcation and discussion. Comments will be appreciated. 2

Thoughts about the Deferred Execution System (DEX) arising from a conversation with DCE on 12 March 1971: 3

The needs and general idea of DEX-- part of a full spectrum of augmentation aids for the user. 4

As we make the resources of our site available to users on the network, and as the load on the system increases here, it becomes more clear that an additional option for doing low priority, non-interactive text creation and editing would be desirable. 4A

Input could be through paper tape here or through some form of queueing mechanism here and out on the network. 4A1

Compatability with existing systems-- DNLS and TNLS would be nice. 4B

Commands should be as similar as possible. With half-duplex terminals, it may be advantgeous to modify the TNLS command specification and feedback. 4B1

The differences between the parts of the spectrum should appear to be as minor as possible. Going from a fully interactive display system to the less interactive teltype system to the non-interactive deferred execution system should be painless and transparent to the user. 4B2

The Old FLTS-- Discussion of Features 5

FLTS provided the user the capabilities of creating and editing files off-line. In creating a completely new file one could insert statements. The method differed in a crucial manner from that used in the current NLS. Whereas in NLS one specifies the location after which the statement is to be entered, in FLTS the user specified a number which was to be the number of the newly entered node. 5A

Additionally, in FLTS one did not have to specify an instruction mnemonic for insertions-- simply the specification of a node number followed by text cuased the text to be entered as a node in the file structure. The end of the text was specified by two CRLFs. 5B

The user had a bit of control over errors in punching the input tape. Immediate deletions of characters, words or lines could be specified using the characters <, > and / respectively. One could specify a deferred deletion using a simple place code similar to the location specifications in NLS. Deferred deletions could serve the same purposes as immediate deletions; they could also be inserted in previous statements (using a construction similar to the Insert Character or Text commands in NLS) to permit later changes in text. 5C

Any Command could be cancelled by Un-commands. Commands could be immediately aborted by typing a dollar sign ($). 5D

Other available commands included Append, Delete, Replace, Move, and Copy. 5E

Specification of locations again referred to the number a node (or group of nodes) was to get. 5E1

Directives coresponding to thee current Output Processor directives permitted formatting of Hard-Copy output. (The term Pass 4 came from the fact that this print out was the fourth pass of the FLTS compiler. 5F

Some proposed features of DEX 6

As many of the current repetoire of NLS commands as possible should eventually be put into DEX. (There are, of course, several which would be meaniingless in an off-line system.) 6A

It also is evident that an off-line system not necessarily using paper tape as input would be useful particularly for people on the network. 6B

The basic principles of design involved in this project should be the following: 6C

1. As much of the core NLS procedures as possible should

be used in the implementation of the system. This is fairly evident-- proliferation of similar routines is expensive in time and wasteful of the resources of the computer. 6C1

2. The system should be as easy to use as possible. 6C2

3. As part of the array of augmentation tools, DEX should be compatible with TNLS and DNLS. 6C3

Some needs arise as immediate consequences of these design principles; others come after some thought; others still are open to discussion: 6D

1. Specification of items should be with relation to what the user has in front of him at the time of creation of the DEX tape or file. 6D1

The user editing a pre-existant document will refer to items with repect to the numbers they had on their last printout. References to new items will be with repect to the numbers given on entry. Moves, insertions and other commands should not affect this. 6D1A

As a result an interpolative specificatio scheme is necessary. To insert a node between nodes 1 and 2 on the same level the user should be able to specify a node 1.5. 6D1B

2. It shoulld be possible to enter nodes in any order. 6D2

The creation of dummy nodes which will be cleaned up at final output is useful. 6D2A

3. It should be possible to cancel any command. 6D3

A history of each special node could be contained in a word in a list containing in one half a PSID, in the other 18 flags--dummy flag, move flag, interpolative flags, etc. 6D3A

4. If the user repeats an already used number the information should not be thrown away. The successive occurances become successors of the first statement. They may be accessed using parenthesized occurance numbers, e.g., 1(1), 1(2), 2a(3)1, etc. 6D4

5. An implicit command at the end of a program causes the

creation or updating of an NLS file which may be used on the other systems. All dummy statements without substructure are deleted; those with substructure remain with asterisks as their text strings. All numbering is corrected. (Interpolations and repeats are renumbered along with all other statements.) 6D5

6. As much of the tape as is possible should be processed without generating errors. 6D6

7. The user should be able to see all he has done on the printout of the file being modified and the text being entered. Thus control characters can not be used in punching tape as they do not print on the Teletype. 6D7

Proposed implementation schedule. 7

Phase 1-- Paper tape -- fresh material input. 7A

The first phase in the implementation of a deferred execution facility will be a system permitting off line insertion of files into the system. These files may be entered onto paper tape or into some sort of queue at other network sites to be inserted and processed at some later time. 7A1

A queueing mechanism is not to be implemented at present. Rather the user is to load the paper tape reader, go into the EXEC, assign the reader to his job using the ASSIGN PTR: command, copy the tape to a TENEX file, deassign the reader, and then enter NLS specifying a new device, OFF-LINE, his initials as usual and then the name of the file to which the paper tape had been copied. 7A1A

The tape will be processed off-line at low priority. Eventually we hope to permit the user to enter his job into something like a printer queue to process when the system has time. 7A1B

Because of the demand for a decrease in the load on the system it is desirable to get something up as soon as possible. 7A2

There will be commands capable of doing several basic file creation tasks: 7A3

Insert a statement at a particular level in a particular location. 7A3A

This is more difficult than is immediately apparent. In order to satisfy Principle 1 above and also Need 1 we must write a new parsing algorithm for NLS node specification. We must be clever-- the core routines expect the PSID after which the new item is to go along with level adjust specifications. We must get this information out of the node specification. 7A3A1

The syntax of the command will be identical to that in FLTS with the following exception. The end of the command will be specified by an exclamation point. (See Need 7 above.) This will serve as the command accept character in all other cases. The use of CRLFs as acceptance characters is to be avoided since it is possible that the user may wish to use them in a node. It is easier to put a literal escape (') before a real printing character.. 7A3A2

Other special characters are described in FLTS descriptions and in (J5280,:). 7A3A3

Insert statements in any order. 7A3B

This is not difficult if we start the creation of the history list to facilitate the presence of dummy nodes. If a node is specified with an existant number and that number refers to a dummy node, it becomes the node. If it refers to a real node we handle the problem as discussed in Need 4 above. 7A3B1

Interpolate a statement into a location between (or before) earlier specified statements. 7A3C

This is perhaps not essential in the first pass given the desirability of imminent implementation of the first phase. It would add additional complexity to the parsing algoritnI 7A3C1

Create file from the input material. Enter it into the system. 7A3D

The user could specify the name of a file to be output to. Alternatively, a default would be an NLS extension of the file named at the time the tape program was entered. 7A3D1

Print out a file; specify the format and structure of the print out. 7A3E

We could permit an Output Device Printer command. 7A3E1

Modify commands made earlier in the input-- delete commands to insert statements, change the location specified by an insert command, modify a create file command-- change the name of the file created. Modify printout specification commands. 7A3F

At first very basic modifications-- we do not plan to permit appends or deferred deletions at first. 7A3F1

Note that because of the way we plan the syntax of the commands and the semantics we plan to attach to them there is no need for level adjust characters. The user explicitly specifies the number a node is to get. Included in this specification is the level. Also note that this precludes the use of a center-dot mode. 7A3G

This is a good idea. With a centerdot, it would be difficult for an off-line user too recall the number of a node entered in the middle of a center-dot sequence. 7A3G1

In the future we plan to permit something of a macro definition facility which would permit an unused symbol to be declared to represent a node. If, for example, 2a12b were defined to be #, 2a12b3 could be written #3. This should take away some of the objection to the lack of the center-dot facility. 7A3G2

The first phase system could be implemented early the week of 26 April. 7A4

Phase 2-- Access and modify existing files. 7B

The input commands and text can be on paper tape or can be in a subsystem of the Exec or can be queued at remote sites. The background job will process items in the queue at appropriate times. 7B1

In addition to the commands of Phase 1, the following additional commands will be added: 7B2

Insertion of new statements, words, characters. 7B2A

Deletion of items. 7B2B

Modify the sructure of the existing file. 7B2C

Copy and merge items from other files; copy items from the same file. 7B2D

Phase 3-- Make DEX commands compatable with the on-line system. Permit essentially executable text. 7C

Maybe modify commands in TNLS and DNLS to make all systems as similar as possible. 7C1

Further notes on DEX-- a proposal

<JOURNAL>6738.NLS;2, 23-APR-71 15:44 HGL ; (Expedite) Title: Author(s): Harvey G. Lehtman/HGL; Distribution: Douglas C. Engelbart, Richard W. Watson, Charles H. Irby, Bruce L. Parsley, James C. Norton, William S. Duvall/DCE RWW CHI BLP JCN WSD; Keywords: DEX; Clerk: HGL;
Origin: <LEHTMAN>DEXT.NLS;1, 23-APR-71 15:36 HGL ;