<OFFICE-15, TUTOR, TUTOR2.AUG;1,>, 18-Oct-84 15:21-EDT RVR ;;;;

Welcome back! We will begin with a very quick review of the subjects covered previously. If you have forgotten any of the material, you can type "jftutor<OK><OK>" (which will be echoed on your screen as "Jump (to) File (named) tutor!!") and you can review everything in detail. 1

The basic commands for moving around are the Jump commands, which you begin by typing "j". The simplest way of indicating the destination is by using the mouse to mark the statement you want at the top of the screen, and then pressing <OK> again. You can jump to the origin by typing "jo" (standing for "Jump (to) Origin") followed by two <OK>'s. 2

Again, remember that the way to view this file is with a sequence of Jump commands. The previous file was carefully arranged so that only complete statements appeared on the screen after each jump. In this file, that will not be the case; when you type "j" for "Jump" and mark the bottom statement on the screen, often a few lines of text that were on the bottom of the screen will be repeated at the top of the screen in the next view. 3

Notice that this is exactly what happens in the case of this statement. The first three lines of the statement will appear at the top of the next view when you type "j" and mark this statement, together with some more information which was part of the same paragraph, but was too long to fit entirely in the screenful you were just viewing. 4

At this point in your AUGMENT education, you should now know the following: 5

Almost all commands are two words long. The first word is generally a verb, which tells the computer what to do, and the second is a noun, indicating what to do it to. Typical verbs that we have encountered are: "Delete", "Insert", "Move", "Replace", and "Set". Typical nouns are: "Character", "Word", "Text", "Statement", and so on. 6

After specifying a command such as "Delete Word", you will need to indicate to AUGMENT which word should be deleted. This is generally done by "marking" the desired word using the mouse. Do this by positioning the cursor anywhere under the word to be deleted and pressing <OK>. Remember that the right button on the mouse is the same as the <OK> key on your terminal. The character you marked (hopefully in the word to be deleted) will be highlighted. A final <OK> (coming either from the keyboard or from the mouse) is required to complete the command. 7

Remember also that you can gain a lot of information by watching the prompts in the window at the top of your screen. They give you some idea of what the computer is expecting next. "C" stands for "command word", "M" stands for "mark", "T" stands for "type", and "OK" stands for the <OK> character on your keyboard or on the mouse. Remember that, in general, the backspace key will undo exactly one keystroke (whether a command word, a mark, or a character you type). The "CMD DELETE" key can be used to cancel an entire command, so that you can start over again. As a shortcut, the left mouse button is a backspace, and the middle button is the same as the command delete key. 8

There was one other thing covered previously, and that was the use of viewspec "y" to put blank lines between statements (so that it was easy to see where one statement ended and the next one began). The command to change viewspecs

was Set Viewspecs, but typing an "s" gave us "Sort" rather than "Set". Any

time there is a conflict of first letters like this, you need to type a space before the first letter and then start typing the letters of the command word until AUGMENT recognizes it. In the case of "Set Viewspecs", you need to type "<SP>se" to get "Set", and a single "v" for "Viewspecs". Remember that viewspec "y" turns on the blank lines, and that viewspec "z" turns off viewspec "y". 9

If you haven't used the system for a while, it might be a good idea now to practice a little with these commands on the following statement. Don't use the "Delete Statement" command until the very end, though. You may also want to try setting the viewspecs to "y" before you start, and back to "z" when you are done. 10

Fore skoree and 7 yrs years ago**our fatherz bought 4th upon the connit a nation new onceived in liberty, and dedicated to the proposition that all are equal. 11

OK. The review is over. It is now time for some new material. 12

Probably the most important thing to learn is a more efficient way to get around in your file. So far, we know of only two ways -- Jump (to) MARK, and Jump (to) Origin. It turns out that each statement in your file has a statement number associated with it, and you can see those numbers or not, depending on what you want. Like the blank lines between statements, the numbers are turned on and off with a viewspec. Viewspec "m" turns on the numbers, and viewspec "n" turns them off. Go ahead and use the "Set Viewspecs" command to turn on the numbers. (Remember to type the space before the "se", or you will get "Sort", which we have not yet discussed.) The whole command will be "<SP>sevm<OK>". As soon as you press the final <OK>, the screen will be redrawn with the statement numbers on, and each statement will be preceded by its number. 13

I can't tell you what the number of this statement is because I don't know whether you deleted the exercise statement or not. This statement's number is probably either 13 or 14. 14

Try deleting this statement with the numbers on, and watch the numbers change. 15

The numbers of the statements preceding the deleted one do not change -- statement number 12 is still the twelfth statement after the origin -- but the numbers of statements following the deleted statement will change. 16

Statement numbers will give us a good way to get around in a file. The command we use will be another Jump command, but this time instead of typing "o" for "Origin" or marking the desired statement, we will give the name of another object (noun) to jump to. 17

OK. Let's try it. Turn your statement numbers on if they aren't already, and note the number of the current statement (so that you can come back). Then type the command "Jump (to) Link" (by typing "jl"), followed by the number of the statement you want to jump to (say 2, in this case). You will get back the same way -- Jump (to) Link, followed by the number of the statement you are reading now. The command is completed by typing the final <OK>. Do this now, and then return here. 18

•

Go ahead and try jumping around a little, but come back here. Try some illegal jumps (for example, to statement 1000000) and see what happens. 19

(Some advanced people may already be worried about the fact that statement numbers may change. Later, we will discover that each statement actually has

two numbers, one changing dynamically and indicating the actual position in the file, and one permanent number that can always be used. More about this later.) 20

The word "Link" probably seems a little strange to you. You may wonder "Why not call it Number, or Address?" It turns out that a link is much more than this, and as you proceed, you will find more and more things that it can be. For now, simply consider it to be the number of the statement in a file. 21

The Jump command is perhaps the most useful and powerful command in AUGMENT. We will now look at some of its other features. Most of the Jump commands display the prompt "V:" at the end (Jump (to) Link is one of the few exceptions). Up to now, we have simply typed <OK>, and the command has completed normally. In fact, the "V:" stands for "viewspecs", and whenever it appears, you can type viewspecs that will take effect when the jump is completed. When you jump to a new place in a file, it is likely that you will want to see it with a different "view". 22

Before you try experimenting with this feature, read the next few paragraphs to learn how to get out of trouble in case you happen to get into it. 23

Most letters are valid viewspecs, but now we only know four -- "y", "z", "m", and "n". Some of the other viewspecs can cause rather bizarre changes to your view of the file, but there is a new command, "Reset Viewspecs", which will put them all back to what they were when you started. Typing "r" gives the command "Replace", which you don't want, so as with "Set", you must begin that command by typing a space. You will then need to type "res" for the computer to recognize the verb as "Reset". Then type "v", followed by <OK>, and you will be back to your original set of viewspecs. 24

Let's try out some Jump commands making use of the "V:" prompt. Again, note the statement number where you are, so that you can get back here (set viewspec "m" to turn on the numbers if they are off). Then perform a few experiments of the following type: Do either Jump (to) MARK or Jump (to) Origin, but instead of immediately giving the final <OK> when the "V:" prompt appears, type one or more viewspecs, and note that your view will change when the jump is completed. So far the only viewspecs we know about are "y" (blank lines on), "z" (blank lines off), "m" (numbers on), and "n" (numbers off). There are many other viewspecs, and you are welcome to fool around with them, but remember the Reset Viewspecs command, and the Jump (to) Link command to get back here. (For the truly adventuresome, there is a way to specify viewspecs in a Jump (to) Link command. After you type the number of the statement to which you wish to jump, follow it with a colon and then the desired viewspecs before typing <OK>. For example, to get to statement 3 with numbers on and blank lines between statements, type "jl3:my<OK>".) 25

Now that you know how to move around in a file, you will learn how to move to other files. The only other file that we know about so far is the first one in this series, called TUTOR. Read the next statement entirely (so that you'll know how to get back), and then try the exercises. 26

•

CE, 10-Sep-96 16:32-PDT < USER:TUTOR, TUTOR2.AUG.2, > 4

Jumping to different files is accomplished by using the Jump (to) File (named) command. You can also use the Jump (to) Link command. Instead of typing a statement number, simply type the file name followed by a comma and then the final <OK>. You may, if you wish, follow the comma with a statement number, and you will be taken to that statement number in that file. To get to statement number 5 in the TUTOR file, for example, type "jl" (for Jump (to) Link), followed by "tutor,5<OK>". To get to statement number 17 in this file, type "jltutor2,17<OK>". Note the number of the statement you are reading now, jump to somewhere in the TUTOR file, and then come back here. (Again, for

experts, you may change the viewspecs as you go. Jump (to) Link tutor,3:yn<OK> would put you at statement number 3 in the TUTOR file with blank lines on and statement numbers off.) 27

STRUCTURE 28

We will now start on another topic which will be extremely useful and important. It is already illustrated by this statement. Note that it is indented, and if you have the statements numbers on, you will notice that it is not a number, but rather a number followed by the letter "A" (probably 27A or 28A). 28A

The idea of arranging material in the form of a structured file is similar to the outlines you used to write in elementary school. The indentation indicates subordination. Statements underneath another generally have something to do with the statement above. Books, for example, may have many chapters, and each chapter will have its own sections. Each section may have subsections, and so on. Sections in chapter 2 generally are all related to the title of chapter 2 more than to chapter 3, and so on. The following example may help to illustrate this. Use the Jump command, and mark "NORTH AMERICA" below (use viewspec "z"): 28B

NORTH AMERICA 28B1

CANADA 28B1A

QUEBEC (PROVINCE) 28B1A1

QUEBEC (CITY) 28B1A1A

UNITED STATES 28B1B

CALIFORNIA 28B1B1

CUPERTINO 28B1B1A

LOS ANGELES 28B1B1B

SAN FRANCISCO 28B1B1C

MISSISSIPPI 28B1B2

OHIO 28B1B3

CLEVELAND 28B1B3A

NORTH AMERICA is a main heading, and has subtopics which are all countries.

CE, 10-Sep-96 16:32-PDT < USER:TUTOR, TUTOR2.AUG.2, > 5

Under each country are main subdivisions -- provinces in the case of Canada, and states in the case of the United States. The pattern continues in the obvious way. 28C

Study the numbering of the statements as well. (If the numbering is off, turn it on with viewspec "m".) It is a little more cumbersome than the numbering you used to use in outlines in elementary school, but each number tells you exactly where you are in the file. These numbers can be used in the Jump (to) Link command in exactly the same way as the original numbers you have been using. If a file has a statement numbered 1A (this one doesn't), then Jump (to) Link 1a<OK> will take you there. 28D

A typical use for structuring might be the following. Suppose that you

wish to compose a document from scratch. You might first type in the main topics, then go through and fill in subtopics under the main topics, and finally fill in the paragraphs containing the actual material. In this way, the document you produced would be guaranteed to be well organized. On the other hand, if you are trying to read a structured document produced by someone else, it will be possible to view an outline consisting of only the highest level statements (using a new viewspec). When you read through this outline and find the topic you are most interested in, you can jump to it and display everything down to two levels. In this way, you can see all the subtopics. This scheme can be continued until you arrive at the text you are really interested in reading. 28E

Let us try this on the next portion of the file, which is simply a copy of the statements naming places in North America that you saw earlier. 28F

The viewspecs that you will be interested in are the following: "a" means show one level less, "b" means show one level more, "c" means show all levels, and "d" means show only one level. Jot these down for future reference. When you begin, you see all levels. In the upper right corner of your screen is a set of letters that indicate a few of the viewspecs in force. Viewspecs like "y" and "m" are not mentioned, because you can see for yourself whether blank lines or statement numbers are on or off. The two ALL's you see indicate numbers of lines and levels visible. The one on the left is the number of levels, and you are currently viewing all of them. When you finish reading this paragraph, jump to NORTH AMERICA, below, and set viewspec "d". Notice that the viewspec window in the upper right corner of the screen indicates that only the first-level statements are visible. Then use the Set Viewspecs command a few times, and set "b" (for one more level) each time. Notice that your view changes, and the number of visible levels indicated in the viewspec window changes to match it. When you are done, set viewspec "c" (or Reset Viewspecs). Go ahead and do this now. 29

NORTH AMERICA 30

CANADA 30A

QUEBEC (PROVINCE) 30A1

QUEBEC (CITY) 30A1A

UNITED STATES 30B

CALIFORNIA 30B1

•

CUPERTINO 30B1A

LOS ANGELES 30B1B

SAN FRANCISCO 30B1C

MISSISSIPPI 30B2

OHIO 30B3

CLEVELAND 30B3A

You will notice that viewspec "d" shows only NORTH AMERICA. Adding one level shows all the countries in the list. Adding another level shows the major subdivisions of the countries, and so on. In the case of this example, it is not particularly useful, but in the case of a large document it can be

experts, you may change the viewspecs as you go. Jump (to) Link tutor,3:yn<OK> would put you at statement number 3 in the TUTOR file with blank lines on and statement numbers off.) 27

STRUCTURE 28

We will now start on another topic which will be extremely useful and important. It is already illustrated by this statement. Note that it is indented, and if you have the statements numbers on, you will notice that it is not a number, but rather a number followed by the letter "A" (probably 27A or 28A). 28A

The idea of arranging material in the form of a structured file is similar to the outlines you used to write in elementary school. The indentation indicates subordination. Statements underneath another generally have something to do with the statement above. Books, for example, may have many chapters, and each chapter will have its own sections. Each section may have subsections, and so on. Sections in chapter 2 generally are all related to the title of chapter 2 more than to chapter 3, and so on. The following example may help to illustrate this. Use the Jump command, and mark "NORTH AMERICA" below (use viewspec "z"): 28B

NORTH AMERICA 28B1

CANADA 28B1A

QUEBEC (PROVINCE) 28B1A1

QUEBEC (CITY) 28B1A1A

UNITED STATES 28B1B

CALIFORNIA 28B1B1

CUPERTINO 28B1B1A

LOS ANGELES 28B1B1B

SAN FRANCISCO 28B1B1C

MISSISSIPPI 28B1B2

OHIO 28B1B3

CLEVELAND 28B1B3A

NORTH AMERICA is a main heading, and has subtopics which are all countries.

CE, 10-Sep-96 16:32-PDT < USER:TUTOR, TUTOR2.AUG.2, > 5

Under each country are main subdivisions -- provinces in the case of Canada, and states in the case of the United States. The pattern continues in the obvious way. 28C

Study the numbering of the statements as well. (If the numbering is off, turn it on with viewspec "m".) It is a little more cumbersome than the numbering you used to use in outlines in elementary school, but each number tells you exactly where you are in the file. These numbers can be used in the Jump (to) Link command in exactly the same way as the original numbers you have been using. If a file has a statement numbered 1A (this one doesn't), then Jump (to) Link 1a<OK> will take you there. 28D

A typical use for structuring might be the following. Suppose that you

wish to compose a document from scratch. You might first type in the main topics, then go through and fill in subtopics under the main topics, and finally fill in the paragraphs containing the actual material. In this way, the document you produced would be guaranteed to be well organized. On the other hand, if you are trying to read a structured document produced by someone else, it will be possible to view an outline consisting of only the highest level statements (using a new viewspec). When you read through this outline and find the topic you are most interested in, you can jump to it and display everything down to two levels. In this way, you can see all the subtopics. This scheme can be continued until you arrive at the text you are really interested in reading. 28E

Let us try this on the next portion of the file, which is simply a copy of the statements naming places in North America that you saw earlier. 28F

The viewspecs that you will be interested in are the following: "a" means show one level less, "b" means show one level more, "c" means show all levels, and "d" means show only one level. Jot these down for future reference. When you begin, you see all levels. In the upper right corner of your screen is a set of letters that indicate a few of the viewspecs in force. Viewspecs like "y" and "m" are not mentioned, because you can see for yourself whether blank lines or statement numbers are on or off. The two ALL's you see indicate numbers of lines and levels visible. The one on the left is the number of levels, and you are currently viewing all of them. When you finish reading this paragraph, jump to NORTH AMERICA, below, and set viewspec "d". Notice that the viewspec window in the upper right corner of the screen indicates that only the first-level statements are visible. Then use the Set Viewspecs command a few times, and set "b" (for one more level) each time. Notice that your view changes, and the number of visible levels indicated in the viewspec window changes to match it. When you are done, set viewspec "c" (or Reset Viewspecs). Go ahead and do this now. 29

NORTH AMERICA 30

CANADA 30A

QUEBEC (PROVINCE) 30A1

QUEBEC (CITY) 30A1A

UNITED STATES 30B

CALIFORNIA 30B1

•

CE, 10-Sep-96 16:32-PDT < USER:TUTOR, TUTOR2.AUG.2, > 6

CUPERTINO 30B1A

LOS ANGELES 30B1B

SAN FRANCISCO 30B1C

MISSISSIPPI 30B2

OHIO 30B3

CLEVELAND 30B3A

You will notice that viewspec "d" shows only NORTH AMERICA. Adding one level shows all the countries in the list. Adding another level shows the major subdivisions of the countries, and so on. In the case of this example, it is not particularly useful, but in the case of a large document it can be

extremely useful. If you go to a large document with viewspec "d", you will see only the chapter titles -- sort of like looking at a table of contents. When you spot the chapter you are interested in, you can jump to that (by using the Jump (to) MARK command), and set viewspec "b" as you go. This will put the chapter heading at the top of the screen, and underneath it you will see one more level down -- all of the section titles. You can continue this process until you find exactly the material you are looking for. 31

Since this and the previous file were introductory, it was impossible to make full use of the structuredness of AUGMENT files. In the next file, TUTOR3, you will make heavy use of the structure of the file. 32

The left "ALL" or number at the upper right corner of your screen represents the number of levels shown, or how deeply into the structure you are viewing. As was stated before, the right "ALL" or number represents the number of lines of each statement you are viewing. Clipping this down to one or two lines also helps you to get a quick overview of the contents of a file; generally, in a well written paragraph, the first line or two should give a good idea of the rest of the contents. As with level clipping, line clipping is controlled by viewspecs, and the pattern is exactly the same: viewspecs a, b, c, and d represented one less, one more, all, and one level, respectively; viewspecs q, r, s, and t represent one less, one more, all, and one line of each statement being shown. 33

You may want to jot these down as well, before you begin experimenting with them (again, if you get into viewspec trouble, use the Reset Viewspecs command to bail out). Following this statement is another group of statements that you can view using both the level clipping and line clipping viewspecs (any combination of them is valid -- to see exactly what is in effect, look at the numbers or ALL's at the top of your screen, and remember that the left one is levels and the right one is lines). Jump to the next statement (with statement numbers on), and try the following viewspec combinations using the Set Viewspecs command (you can set more than one at a time): dt, ds, ct, dbt, dbtr. (Jot these down, too. They will disappear when you start fiddling with the viewspecs.) Note both the view of the file and the numbers in the viewspec window. Try other viewspec cobinations. 34

This is the first line of a first-level statement,
this is the second line,
•
CE, 10-Sep-96 16:32-PDT < USER:TUTOR, TUTOR2.AUG.2, > 7

and this is the third. 35

Here is a second-level statement which also has three lines in it. Do not worry yet about how the strange line breaks were forced in the statement above -- it will be covered later. 35A

Here is a third-level statement with only two lines in it. It will all be visible when the line viewspec number is 2 or more. 35A1

Here is a one-line first-level statement. 36

Here is a one-line second-level statement. 36A

Here is a one-line third-level statement. 36A1

Here is a multiple-line fourth-level statement. When you are seeing at least 4 levels, try setting the viewspecs so that they show only one line, and then keep adding one more line at a time, until you can see the whole thing. If you are wondering what the limits are, try setting all lines (or levels), and subtracting one (using "a" or "q"). 36A1A

Well, hopefully you now have a good idea about how structure can be used to your advantage when viewing text. Now we will learn how to build structure into your own files. 37

It is not much harder than inserting a statement. Recall that to enter a new statement into your file, you used the command "Insert Statement". After the noise words "(to follow statement at)", you marked the statement in the file that you wanted the new statement to follow. What you did before was simply to begin typing the text of the new statement. If you try it again (type "is" now, and mark this statement), you will note that the prompt includes "L", which stands for "level adjustment". If you want the new statement at the same level as the one you marked, simply type <OK>, or begin typing the new statement. If you want the new statement to be under the marked one, type "d<OK>" ("d" stands for "down"). Similarly, "u<OK>" would make the next statement at a higher level than the statement marked ("u" stands for "up"). 38

Important: The level adjustment is relative to the statement you mark. If you mark a fifth-level statement and give no level adjustment, the new statement will be at the fifth level. If you indicate "d" for down, the new statement will be below the marked statement (at the sixth level). If you say "u", it will be at the fourth level, and so on. (To put a statement up two levels, type "uu". Why do you suppose "dd" makes no sense?) 39

Go ahead and create an outline of your own (perhaps a family tree of a person, with the children underneath, and the grandchildren under that). Insert it under this statement. 40

Well, that's plenty of material for now. The main subjects covered here have been improved methods for jumping around and an introduction to structure. We have covered a lot of new viewspecs, and have learned to insert statements at different levels. A short review section follows this, and the next installment appears in a file named TUTOR3 (Jump (to) File (named) tutor3<OK> V:<OK> remember?) 41

•

CE, 10-Sep-96 16:32-PDT < USER:TUTOR, TUTOR2.AUG.2, > 8

REVIEW 42

New viewspecs 42A

a -- show one level less 42A1

b -- show one level more 42A2

c -- show all levels 42A3

d -- show one level only 42A4

m -- statement numbers on 42A5

n -- statement numbers off 42A6

q -- show one line less 42A7

r -- show one line more 42A8

s -- show all lines 42A9

t -- show one line only 42A10

Old viewspecs 42B

y -- blank lines on between statements 42B1

z -- no blank lines between statements 42B2

Jump commands 42C

Jump (to) MARK viewspecs <OK> 42C1

Jump (to) Origin viewspecs <OK> 42C2

Jump (to) File (named) filename <OK> viewspecs <OK> 42C3

Jump (to) Link link <OK> 42C4

Links (and examples) 42D

Statement numbers: 15 2 12a 12a2b 42D1

File names (followed by a comma): tutor, tutor2, 42D2

Combinations: tutor,3 tutor2,12a 42D3

Links with viewspecs: 15:dm tutor,3:army tutor2,:ds 42D4

Other commands 42E

Verb Noun, where Verb = Insert, Delete, Move, or Replace, and Noun = Character, Text, Word, or Statement. 42E1

Reset Viewspecs 42E2

Set Viewspecs 42E3

Delete Modifications 42E4

It might be a good idea to refresh your knowledge of the editing commands learned previously as well. Try inserting some new statements (with structure), and editing them. You may want to try changing the structure of the material you create. Try to figure out how to move statements so that they are under different statements, or so that different levels are moved to the same level. The next file to look at is named TUTOR3. 42F