Inter-Office Memorandum

To	Cedar Interest	Date	October 12, 1982
From	Bill Paxton	Location	Palo Alto
Subject	The Tioga Editor	Organization	PARC/ISL

XEROX

Filed on: [Indigo]<Tioga>Documentation>TiogaDoc.Tioga and TiogaDoc.Press

Uses: [Indigo]<Tioga>Styles>TiogaDoc.Style and Cedar.Style

The Tioga Editor

Introduction

This is an overview of Tioga as available for Cedar 3.4 in October, 1982. Tioga is a system to help you prepare documents. Its two main components are an editor and a typesetter. The editor lets you create the text of a document. The typesetter composes the text into pages for printing. Tioga is already capable of dealing with simple technical papers and is also well suited to more mundane tasks such as writing programs and memos. In future versions, it will be suitable for complex technical documents and books and will support tables, math formulas, and figures containing synthetic graphics or scanned images.

If you are looking at this document on-line, you might want to use the level-clipping function to see the overall structure rather than simply plowing straight through. Hit the "Levels" button in the top menu, then hit "FirstLevelOnly" in the new menu that appears. That will show you the major section headings. Hit "MoreLevels" to see the subsections, or hit "AllLevels" to read the details.

Overview

Some editors represent a document as a list of paragraphs. In Tioga, a document is a tree structure rather than a list so that you can explicitly represent its hierarchical structure. In discussing the document tree we use Computer Science terminology in which a branch is recursively defined to be a node having zero or more children branches. The root node of the document tree is not displayed although it can be modified by a few special commands so the document basically appears to be a list of top-level branches.

Each node in the tree contains text. The characters of the text can have *looks* which control various aspects of their appearance such as font and size. Appearance is also influenced by the

format of the node which determines things such as vertical and horizontal spacing. The document contains names of looks and formats, but not the specific interpretation of them. The interpretations are instead collected in a *style* which can be shared by many documents. For example, in the style for this document there are definitions of formats for titles, headings, and standard paragraphs, and there are definitions of looks for emphasis and for small caps. Rather than copying the specific details for the formats and looks, the document refers to them by name so it is easy to change the definitions in the style and modify the appearance uniformly throughout the document. The details of the style language will be described in a forthcoming memo.

User Categories

The Tioga user-interface is "layered" so that beginning users can protect themselves from the confusion that results from mistakenly giving a command. In ways described below, you can tell the system that you are either a beginner, an intermediate, or an advanced user. If you do nothing, the default is beginner.

As far as the Tioga user-interface is concerned, the user category determines which keyboard commands are currently enabled. As a beginner, you get the commands that use the special keys at the left and right of the keyboard, plus CTRL-A and CTRL-W for backspace character and word, respectively. As an intermediate user, you add a large number of commands that use print keys in combination with the various shift keys. As an advanced user, you add the keyboard commands for manipulating the document tree structure. Any category of user can get at any of the commands by using the Edit Tool. The user category mechanism is meant to let you protect yourself, not to limit you. You are free to change your own category at any time you feel like it. For example, to declare yourself to be an intermediate user, edit your user profile to say "UserCategory: intermediate".

Input Devices

Mouse

The mouse has three buttons named LEFT, MIDDLE, and RIGHT corresponding to their physical layout.

Keyboard

Used for commands as well as text input. Here are the names for the special keys.

LOOK (looks shift).	top right blank key.
NEXT.	middle right blank key.
REPEAT.	key labelled ESC.
DELETE.	key labelled DEL.
LOAD FILE.	key labelled LF.

The bottom right blank key is an alternative CTRL key. You can use either CTRL key and either SHIFT key interchangeably.

Scrolling and Thumbing

To move a viewer to look at a different part of the document, move the cursor into the left margin until it becomes a double arrow pointing both up and down. The part of the margin that becomes gray is called the "scroll bar". The dark gray part shows the relative size and location of the currently visible portion of the document. Click LEFT to move the text adjacent to the arrow to the top of the viewer. Click RIGHT to move the text from the top of the viewer down to the arrow. These operations are called "scrolling" the document.

If you hold down MIDDLE in the scroll bar, the cursor becomes a right-pointing arrow. If you move the cursor to x% down from the top of the viewer and let up, the viewer will change to start about x% of the way through the document. This operation is called "thumbing".

Selections

The details of making selections will be covered later on. For now, you simply need to know that there is a single primary selection on the screen. The viewer containing the selection is referred to as the "selected viewer". Many of the following commands deal with the selection or the selected viewer.

Menus

Top level menu

Save writes a new version of the file

Requires confirmation before writing the file. Old version of the file is renamed to have a "$" at the end of the extension so that you can retrieve it if necessary.

If the version of the file on the disk has a more recent create date than the document you are about to save, the system will tell you. This is to warn you about situations in which you load a file and while it is in a viewer transfer a more recent version to your disk.

Get loads the file named by the selection into a viewer

If the confirming click is done with the left button, the file is loaded into the "clicked" viewer and replaces the previous contents. If it is done with the middle button, a new viewer is created below the clicked one. Finally, if it is done with the right button, the clicked viewer is closed and a new viewer appears in its place. A comment will appear in the message window to remind of this.

If you use a remote file name, such as /Cedar/Documentation/TiogaDoc.Tioga, the system will fetch the file and make the viewer "ReadOnly". This lets you browse remote files and copy information from them. In future releases, Tioga will support editing of remote files.

File Extensions

If the selected name does not include an explicit extension (i.e., there is no period in the selected text), Tioga will search for the file using a set of standard extensions. The default extensions are mesa, tioga, df, cm, config, and style. You can specify your own list of extensions with a user profile entry for "SourceFileExtensions".

If the selected name is of the form <alpha>.<beta> and such a file exists, it is opened. Otherwise, if <beta> is one of the standard set of extensions, you will be informed that the file doesn't exist. However, if <beta> is not a standard extension, the system tries to open the file as if you had simply selected <alpha>. If this succeeds, it searches in the file for a definition of <beta>. This convention is intended for use with programs that have many instances of <Interface>.<Item> in which <Interface>.mesa is a file containing a definition for <Item>.

GetImpl like *Get* but loads the file that implements the selected interface name

If the selected name does not include an explicit extension (i.e., there is no period in the selected text), Tioga will use a set of standard implementation extensions. Currently, mesa is the only default implementation extension. You can specify your own list of implementation extensions with a user profile entry for "ImplFileExtensions".

If the selected name is of the form <Interface>.<Item> and an implementation for the item is

currently loaded, the system will find the name of the file holding the implementation (our thanks to the Cedar runtime model for providing this information). Otherwise, the system tries to open the file as if you had simply selected <Interface>, and, if this succeeds, searches in the file for a definition of <Item>.

PrevFile like *Get* but reloads the file that was previously in this viewer

Reset discards edits by reloading the filed version of the document

Store like *Save* but writes to the file named by the current selection

Clear creates an empty viewer

If the confirming click is done with the left button, the "clicked" viewer is cleared. If it is done with the middle button, a new empty viewer is created below the clicked one. Finally, if it is done with the right button, the clicked viewer is closed and a new empty viewer appears in its place.

The empty viewer will say "No Name" at the top in the place that would normally hold the file name. Naturally the most common thing to do with a "No Name" viewer is to load a file. If you type a file name into a "No Name" viewer and then hit LF, it is as if you selected the name and hit Get but no confirmation is required. CTRL-LF provides a similar function for GetImpl.

Time inserts the current time at the caret

Split creates a new viewer looking at the same document

The selection is highlighted in one viewer only, however edits will be reflected in all viewers for the document. Note that the split viewers can be independently closed, opened, or moved on the screen.

Places, Levels show/remove submenus

Places

The Places menu contains commands that cause the viewer to begin displaying at a new place in the document. The first three of the commands search for instances of the current selection. For these commands, the button used in clicking the menu item determines how the search is carried out click LEFT to search towards the end of the document, RIGHT to search towards the start of the document, or MIDDLE to search first towards the end and, if that fails, then towards the start. If the selection is visible in the viewer, the search starts there. Otherwise, it starts from the top of the viewer.

Find

Find another instance of the selected text.

In this command and the following two, capitalization matters in the search (e.g., hitting Find with "the" selected will not select "The").

Word

Find an instance of the selected text that is a "word" i.e., doesn't have adjacent letters or digits.

Def

Find a "definition" of the selected text i.e., an instance of the selected text that is immediately followed by a colon and doesn't have an immediately prior alphanumeric.

Position

This is useful with compiler error messages that give locations as character counts. The command scrolls to the selected character number and then selects it e.g., if "183" is selected, scroll to and select character number 183 in the document.

Normalize

If the document in this viewer contains the selection, scrolls to make the caret visible. Otherwise scrolls to the start of the document.

PrevPlace

Go back to the place that was previously visible in this viewer discounting manual scrolling that may have taken place since.

Reselect

Restore the most recent selection in this viewer and scroll to it.

Levels

These commands let you control how deep the display goes in the document tree structure.

FirstLevelOnly show only the top level nodes

MoreLevels show one more level than currently

FewerLevels show one fewer level than currently

AllLevels show all levels of the tree

Selections

Primary selections

The primary selection is the one that's around most of the time and is the usual site for edits. It is displayed with a solid underline or with video reverse. Make a primary selection with the mouse in ways described below. During certain editing operations such as Copy or Move there is a "secondary" selection which is displayed as a gray underline or background.

Insertion point

The insertion point goes with primary selection. It is shown by a blinking "caret" at one end or the other of the selection the end closer to the cursor when the selection was made or the one most recently extended.

Making selections

The selection hierarchy

The selection hierarchy consists of the following levels at which a selection can exist: point, character, word, node, branch, and document.

Point selection

The primary selection has a blinking caret at one end. Some operations, such as delete and type-in, reduce the selection to just the caret. This is called a point selection.

Character selection

Click LEFT with the cursor over the desired character. You can hold LEFT down and move to correct place before letting the button up.

You can cancel the new selection by hitting DEL while the mouse button is still down. The system will restore the previous selection.

Word selection

A "word" is defined as a sequence of letters and digits or a sequence of identical characters that are not letters or digits. Use the MIDDLE mouse button to make word selections. As with character selection, you can hold MIDDLE down and move to the correct word before letting up.

Node selection

Double click LEFT to select a node.

Branch selection

A branch is a node and any children branches it might have. Double click MIDDLE to select a branch.

Document selection

CTRL-D extends the selection to include the entire document. "D" stands for for "Document".

Selection extension

Extend an existing selection by pointing at a new endpoint and clicking RIGHT. The system will extend the end of the selection closer to the cursor when RIGHT goes down. You can hold RIGHT down and move the endpoint to a new position.

If you just click RIGHT, the selection is extended at the same level in the selection hierarchy. For example, a selection at the word level will be extended a word at a time. Double click RIGHT to extend at a lower level in selection hierarchy. Triple click to extend at a higher level in selection hierarchy. Thus, if you have a node-level selection and wish to extend it to a word position within a node, double click RIGHT to reduce the level to words and then do the extension. If necessary, you can then double click again to reduce to character level.

Editing by making selections

Delete selections

A *delete selection* is deleted as soon as the selection is completed. It is shown as video reverse. Hold down CTRL to make a delete selection. The selection is complete when you let up on both the mouse button and the CTRL key.

Pending-delete selections

"Pending-delete" selections are automatically deleted by subsequent insertions. They are shown with video reverse rather than solid underline. Whenever you extend a selection it is automatically made pending-delete. Notice that you don't have to actually change the selection when you "extend" it. For example, you can click MIDDLE to select a word and then immediately click RIGHT to make it pending-delete. Or you can combine these actions by "rolling" from the LEFT or MIDDLE mouse button to the RIGHT button to make a char or word selection pending-delete. For example, the sequence MIDDLE-down, RIGHT-down, MIDDLE-up, RIGHT-up will produce a pending-delete word selection.

Copy and Move

Source selections are made with the SHIFT key held down and are shown with a gray underline.

Copy source to primary

If you hold down the SHIFT key and select, the source selection will be copied. If the primary selection is currently pending-delete, it will be replaced by the copy of the source. Otherwise, the copy will be inserted at the caret.

Move source to primary

If you hold down the SHIFT key and the CTRL key, the source selection will be moved. As before, if the primary selection is pending-delete, it will be replaced by the source. Otherwise, the source will be moved to the caret.

Destination selections for Copy and Move

The operations described above work by copying or moving a source selection to the primary selection. However, often the primary selection is itself the thing you want to copy or move. The following two commands take care of these situations.

Copy primary

To copy the primary selection, hit CTRL-S, and with the control key still held down, select a destination. The copy takes place as soon as you let the keys up. The primary selection is made not-pending-delete as soon as you hit CTRL-S to indicate that it will be copied rather than moved.

Move primary

To move the primary selection, hit CTRL-Z, and with the control key still held down, select a destination. The move takes place as soon as you let the keys up. The primary selection is made pending-delete as soon as you hit CTRL-Z to indicate that it will be moved rather than copied.

Transpose selections

We now have covered commands to copy or move either the primary or the source selections. A final option is to transpose the primary and the source. To do this, hit CTRL-X, and with the control key still held down, select a source. The transpose takes place as soon as you let the keys and mouse buttons up.

Miscellaneous

Cancelling a selection

Hit DEL before finishing the selection and the previous selection will be restored. This works during either primary, source, destination, or transpose selections.

Placeholders

A "placeholder" is all the text between a matching pair of placeholder brackets, . Hit NEXT to find and select the next placeholder beyond the current selection. Hit SHIFT-NEXT to find the previous one. Recall that NEXT is the blank key to the right of RETURN.

If there isn't another placeholder, NEXT will move the selection to the end of the document. If the selection is already at the end, NEXT will try to find the next nested text viewer. Similarly, SHIFT-NEXT will move the selection to the start of the document if it doesn't find a previous placeholder and will try to find the previous nested text viewer if it is already at the start. This allows you to use NEXT both when filling in placeholders within a document and when filling in text viewers in tools.

Select visible expand selection to blanks

CTRL-V expands the selection to include the "visible" characters on the left and right ends.

Select matching brackets

CTRL-] extends the selection to the left and right to find a matching pair of [..]'s. Similarly for CTRL-}, CTRL-), and CTRL->. Note that you hit CTRL with the right bracket to extend the selection. In addition, you can hit CTRL with the left bracket to insert matching brackets around the selection.

Fine point: Unfortunately, our keyboards don't have both left and right quote keys. However most of the fonts, including TimesRoman and Helvetica, do provide a left and right single quote. The keyboard key inserts a right single quote (code 047); the left single quote (code 140) can be inserted using the MakeOctalCharacter command in the Edit Tool or with the Insert Matching Single Quotes command (CTRL-'). The Edit Tool also has a command which extends the selection to find a matching pair of left and right single quotes. The situation for double quotes is even less uniform. Some fonts, such as Classic, have a left double quote (code 264) in addition to a right double quote (code 042). However, most fonts have only the 042 double quote, so the Tioga commands for inserting and matching double quotes use that code exclusively.

Editing

Text input

Typed-in characters are inserted at the caret.

To insert the current time use CTRL-T "T" for *Time*.

When you insert a carriage return by hitting RETURN, the system will automatically copy the blank characters (tabs and spaces) from the start of the previous line. To suppress this, type SHIFT-RETURN and only the carriage return will be inserted.

To insert control characters or characters with a specific octal code, use CTRL-K or CTRL-O. The former will change the character before the caret to a control character, while the latter will convert the three digits before the caret to the corresponding octal character. The inverse operations are also available as CTRL-SHIFT-K and CTRL-SHIFT-O.

Abbreviation Expansion

CTRL-E "E" for *Expand abbreviation*

When you hit CTRL-E, the caret is moved to the right of the selection if necessary and the keyname to the left of the caret is then replaced by the expansion text (according to the definition which is linked to the style in a manner described below). If the keyname had looks, they are added to the expansion. If the keyname was all caps, the expansion is made all caps too. If the keyname had an initial cap, the first character of the expansion is made uppercase (useful at the start of sentences, for example). If the definition node has a non-null format, the format of the caret node is changed to be the same as the definition node. If the expansion contains a placeholder, the first placeholder is selected. Otherwise the entire expansion is selected.

Definitions for abbreviations come from Tioga documents which are automatically read by the system when needed. The name of the appropriate abbreviations file is determined by the style that is in effect at the caret when the expansion takes place. For example, if the style is "Report", the abbreviations will come from the file "Report.Abbreviations". You can override this by explicitly naming the abbreviations file along with the keyname. For example, "Mesa.proc" will expand the abbreviation for "proc" from the file "Mesa.Abbreviations" independent of the style in effect at the caret. (Fine point: Since the system interprets a period before the keyname to mean that you're specifying a particular abbreviations file, you cannot type a vanilla abbreviation after a period.)

Each definition in an abbreviations file consists of a separate branch. The top node of the branch holds the keyname followed by an equals sign and then the text expansion. The rest of the

branch, if any, is copied after the caret node as part of expanding the abbreviation. Any text following the keyname is moved to the end of the last child node in the branch. The definition may also include a list of operations to be performed after the expansion has been inserted. These operations have the same format as those in EditTool and are placed in parentheses after the keyname and before the equals sign.

Delete Character or Word

BackSpace: CTRL-A, CTRL-H, or BS. Deletes the character to the left of the caret.

Fine point: If the caret is at the start of a node, this does a Join command (q.v.).

BackWord: CTRL-W, or CTRL-BS. Deletes the word to the left of the caret.

DeleteNextChar: CTRL-SHIFT-A, CTRL-SHIFT-H, or SHIFT-BS. Deletes the character to the right of the caret.

DeleteNextWord: CTRL-SHIFT-W, or CTRL-SHIFT-BS. Deletes the word to the right of the caret.

Delete

As mentioned above, you can delete something by selecting it with CTRL held down. In addition, you can delete the current selection by hitting DEL.

Paste

Paste: CTRL-P. The most recently deleted text is copied to the caret. You can also use the Edit Tool to save the current selection to be pasted later.

Copy, Move, Replace, and Transpose

These operations are all carried out by making selections. They are described in detail in the previous section.

Insert matching brackets

CTRL-[adds a matching pair of [..]'s to the ends of the selection. Similar CTRL commands exist for {, (, <, -, ', and ". CTRL-B inserts matching placeholder brackets .

Note: CTRL-' inserts a left single quote (code 140) and a right single quote (code 047); CTRL-" inserts the same character (code 042) at each end of the selection.

Case

All lower: CTRL-C. Makes the selection all lower case.

All caps: CTRL-SHIFT-C. Makes the selection all upper case.

Initial caps: CTRL-double C. Capitalizes each word in the selection.

First cap: CTRL-SHIFT-double C. Capitalizes the first word of the selection

Repeat

Hitting ESC will repeat the most recent non-empty edit sequence starting with the current selection. Edit sequences are separated by user-made selections. For example, if you select a word, delete it, type a new one, and then select something else, the edit sequence is delete followed by text entry. If you hit Repeat, the system will do a delete and retype the new word.

Auto-repeat is done by ESC-select: if you hold down the ESC key while making a selection, a Repeat will automatically be done as soon as the selection is completed. This is useful when

you're doing a large number of repeats.

Undo

Hitting SHIFT-ESC undoes the most recent edit sequence and restores the selection to its prior state. If you want to undo more than just the most recent sequence, use the Edit History tool which is described later.

Tree structure editing

As explained in the introduction, Tioga documents consist of a tree of nodes. The following commands let you break, join, and nest nodes in the tree.

Break: CTRL-RETURN break node at insertion point to create a new node.

Join: CTRL-J join node at insertion point with previous node.

Nest: CTRL-N move selected nodes to deeper nesting level in tree.

UnNest: CTRL-SHIFT-N move selected nodes to shallower nesting level in tree.

Break & Nest: CTRL-I simultaneously insert a new node and nest it.

Break & UnNest: CTRL-SHIFT-I simultaneously insert and unnest.

Looks

Characters have looks which are named by the lower case letters "a" to "z". Looks are interpreted by the style to change the appearance of the text. For example, look "e" might stand for *"emphasis"* and might result in italic face in one style and bold face in another. Each character has a set of looks thus it may have several looks simultaneously, but each look occurs only once. You can use the Edit Tool to read or change the set of looks for selected characters. The following keyboard commands are also available for dealing with looks.

Selection Looks

You can change the selection looks with the following commands. (Recall that the LOOK shift is the top blank key to the right of BS.)

LOOK-char to add to selection looks.

LOOK-SHIFT-char to remove from selection looks.

LOOK-space to remove all selection looks.

Caret Looks

Caret looks determine the looks of typed-in text. The caret picks up the looks of the adjacent selected text whenever a selection is made. Changing the selection looks also changes the caret looks. However, if you wish to change the looks of the caret without changing the selection looks, click the character key twice in quick succession.

LOOK-char-char to add to caret looks.

LOOK-SHIFT-char-char to remove from caret looks.

LOOK-space-space to remove all caret looks.

Using Selections To Copy Looks

To copy the looks of some existing text to the primary selection, hit CTRL-Q, and then with the

CTRL key still held down, select the text with the looks you want to copy. The source looks replace any looks the selection previously had.

Automatic Mesa formatting

The CTRL-M command scans the selection for Mesa keywords, comments, and procedure names and gives them looks k, c, and n respectively. (As a convenience during typein, the entire caret node is reformatted if the selection is a caret only.) With the standard Cedar style, the keywords will then be displayed in small caps, the comments will be italic, and the procedure names will be boldface. We expect to provide more extensive reformatting capabilities in the future.

Formats

Just as characters have looks, nodes have formats. The "format" is the name of a rule in the style that tells how to modify various parameters when displaying the node. For example, a style for documents might contain formats for titles, headings, quotations, standard paragraphs, etc. The Edit Tool has facilities for reading and changing the format of a node, or you can use the commands described below. (By convention, the null format name is equivalent to "default".)

Setting and inserting caret node format

CTRL-_ will delete the word to the left of the caret and make it the format of the caret node.

Fine point: In most cases you will give this command immediately after typing the format name, so the caret will naturally be in the correct place. However, to handle cases in which you select the name before hitting CTRL-_, the caret will automatically be forced to the right of the selection at the start of this command.

CTRL-SHIFT-_ inserts the format name.

This gives you a simple way to find out the format of a selected node.

Using selections to copy formats

To copy the format of some exisiting node to the selection nodes, hit CTRL-F, and then with the CTRL key still held down, select the node with the format you want to copy.

The Edit Tool

The Edit Tool provides a variety of operations on Tioga documents.

The Edit Tool menu contains some special commands in addition to the usual viewer operations. The "Bigger" command enlarges the Edit Tool, while the "Smaller" menu item does the opposite. The "Stop" button lets you interrupt lengthy substitutions and sorts. The second line of the menu contains buttons for search and substitute which are discussed below.

The text fields in the Edit Tool follow the convention that clicking the field name with LEFT causes the contents of the field to be selected pending-delete while clicking with RIGHT causes the field to be cleared and selected.

Search and Substitute

Search

To do a search, enter the text you're looking for in the "Target" field, select where you want the search to start, and click "Search" with the left mouse button to search forward, with the right button to search backwards, or with the middle button to search first forward then

backwards. The system searches from the current selection and updates the selected viewer if the search succeeds. (Fine point: in both searches and substitutes, the match is limited to a single node we do not yet have mechanisms for doing matches across node boundaries.)

The multiple choices below the "Replacement" field control what is matched in searches and replaced in substitutes. You can select or deselect an item by clicking it with the mouse. White text on black background means that the item is selected; black text on white means it is not selected.

Text if this option is selected, match characters of target text when searching.

Looks if selected, match looks of target text when searching.

Format match format of target node.

Style match style of target node.

Comment match comment property of target node.

For example, if you pick the Looks option and deselect the Text option, you can search for any text that has a particular set of looks. If you pick only Text, the matching will ignore the looks of the target. If you pick Text and Looks, the matching text must match both the characters and the looks of the target text.

The other options deal with node properties. If you pick Format, the matching will be limited to nodes with the same format as the target node. Similarly, if you pick Style, the matching will only look at nodes whose style is the same as the target node's. Finally, if you pick Comment, the match will consider only nodes with the same value of the Comment property (TRUE or FALSE) as the target.

The first two rows of boxes below the multiple choices give you control over how matching is performed. Click LEFT with the cursor over a box to change the choice next to it. The various choices are as follows:

1. Case of matching text
   - Match Case matching text must have same case as target text.
   - Ignore Case matching text does not have to have same case as target text.
2. Interpretation of target text
   - Match Literally don't treat target text as a pattern.
   - Match as Pattern do treat target as pattern. (Patterns are described below.)
3. Context of target text
   - Match Anywhere ignore context of matching text.
   - Match Words Only matching text must not have adjacent letters or digits.
   - Match Entire Nodes Only matching text must span entire node.
4. Matching target looks
   - Subset as Looks Test looks of matching text must include target looks.
   - Equal as Looks Test looks of matching text must be identical to target looks.

Substitute

To do a substitution, enter the new text in the "Replacement" field, enter the text to be replaced in the "Target" field, and hit "Substitute" in the menu at the top of the Edit Tool. The Text/Looks/Format/... options guide the search in the usual manner and also control what is replaced.

If you pick Text and Looks, the matching text will be replaced just as if you had selected it with pending delete and made a source secondary selection of the replacement text.

If you pick only the Looks option, the matching text will have the target looks removed and the replacement looks added.

If you pick only Text, the replacement text will have the looks of the replaced text added to it. (Fine point: if the looks of the replaced text are not uniform, the looks of the first character will be used throughout.)

If you pick Format, the matching node will get the format of the replacement node. Similarly, picking Style causes the matching node to get the style of the replacement node, and picking Comment causes the matching node to get the same value of the Comment property as the replacement node.

The final three boxes in the Search&Substitute section give you further control over this operation.

1. First character capitalization of the replacement text
   - First cap like replaced — if the replaced text starts with a capital letter, force the first letter of the replacement to be a capital too.
   - Don't change caps — leave the replacement capitalization alone.
2. What is done to the matching text
   - Do Replace — do the usual substitute or replace.
   - Do Operations — instead of doing a replace, select the matching text and then do the operations currently in the "Operations" field of the Edit Tool.
3. Where the substitutions will take place
   - Within Selection Only — substitute is limited to current selection.
   - After Selection Only — substitute after selection to end of document.
   - In Entire Document — substitute in the entire selected document.

Case-by-Case Substitutes

In addition to doing global substitutes, you can decide on a case-by-case basis whether or not to replace the matching text by the new text. Use the search commands to find the first matching text. Then if you hit "Yes", the system will do a "Replace" followed by a search forward. If you hit "No", it will skip the "Replace" and simply do another search. Thus to selectively substitute, start with a search, then do "Yes" for the cases you want to change and "No" for the others. The "Replace" command simply does for the current selection what a substitute would do for a match. Finally, the "Count" command is available to tell you how many substitutions would take place without actually changing the document.

In order to do a search or substitute, you will typically need to fill in the Target or Replacement fields in the Edit Tool. Naturally, this changes the selection, and before you can do the operation, you must restore the selection to the place where you actually want it to take place. The system helps you with this by saving the primary selection if it is not in the Edit Tool when you click either the Target or the Replacement button. The commands along the top of the Edit Tool Seach, Substitute, Yes, No, Replace, and Count restore the saved selection if the primary selection is in the Edit Tool when they are clicked. The net effect is that if you start out with the selection in the right place, you can click the Target or Replacement buttons, fill in the needed information, and then directly click one of the commands without needing to reselect since the system will do it for you.

Patterns for Search and Substitute

When you specify that the target be matched as a pattern rather than literally, the following symbols in the target text are interpreted specially. A short summary of these symbols appears at the bottom of the Search&Substitute section of the Edit Tool.

'	Match the next character in the pattern exactly.
~	Match any character except the next one in the pattern.
#	Match any single character.

*	Match any sequence of characters.
@	Match any single alphanumeric character (letter or digit).
&	Match any sequence of alphanumeric characters.
~@	Match any single non-alphanumeric character.
~&	Match any sequence of non-alphanumeric characters.
%	Match any single blank character.
$	Match any sequence of blank characters.
~%	Match any single non-blank character.
~$	Match any sequence of non-blank characters.
|	Match start or end of node.
{	Mark start of resulting selection.
}	Mark end of resulting selection.
<	Mark start of named subpattern.
>	Mark end of named subpattern.

The named subpatterns are of use in substitutes that reorder or duplicate parts of the matching text. The full syntax for a named subpattern is <name:subpattern>. The special case of "match any sequence of characters" is provided as a default i.e., <name> is equivalent to <name:*>. As far as the matching is concerned, the occurrence of <name:subpattern> is the same as if the subpattern had appeared without a name, but it has the side-effect of remembering the subsection it matched. The "Replacement" field can contain <name>'s corresponding to named subpatterns in the target. The replacement text is constructed by replacing the <name>'s with the section of replaced text that matched the subpattern. The replacement is automatically considered to be a pattern whenever the target is you don't need to do anything special to get the <name>'s in the replacement interpreted as subpatterns.

For example, if the target is

Target: WHILE <pred>[<arg>] DO

and the replacement is

Replacement: WHILE <arg> IS <pred> DO

then "WHILE blue[moon] DO" will be converted to "WHILE moon IS blue DO".

Looks, Formats, Styles, and Properties

Looks

The Looks commands let you read and modify the looks of the caret or the selection. The looks are shown as a series of letters in the "Looks characters" field. The box lets you pick whether you want the looks for the selection or the looks for the caret.

Get	fills the "Looks characters" field with the letters for the caret/selection looks.
Set	reads the "Looks characters" field and sets the looks of the caret/selection.
Clear	removes all looks from the caret/selection.
Add	adds the specified looks to the caret/selection.
Sub	removes the specified looks from the caret/selection.

Formats

The Format commands let you read and modify the format for the root node or the selected nodes. The box at the right lets you pick the case you want.

Get fills the "Format name" field.

Set reads the "Format name" field and sets the node's format.

Clear removes the node's format name. This is the same as specifying "default" format.

Styles

A style is a collection of interpretations for looks and formats. The Style commands let you read and modify the name of the style for the root node or the selected nodes. The new style applies to the specified node and all the nodes within its sub-branches that do not themselves have explicit styles.

Fine point: If a document does not have an explicit style specified for it, Tioga will use the default style. You can specify a default style by means of a user profile entry of the form DefaultStyle: <style name>. If you don't have such an entry, "Cedar" style will be used as the default.

Get fills the "Style name" field.

Set reads the "Style name" field and sets the node's style.

Clear removes any style specification from the node.

LoadStyleDefinition reads the "Style name" field and reloads the style definition from that file with extension "Style". (Don't put the extension in the field just put the style name and let the system add the extension.) Do this operation after you have edited (and saved) the style definition and want to load the new version.

LoadAbbreviations reads the "Style name" field and reloads the abbreviation definitions from that file with extension "Abbreviations". Do this after you have edited the abbreviations and want to load the new version.

Properties

A node can have an arbitrary set of "properties". Each property consists of a name and a value. Certain properties are used by the system, but you (and your programs) are free to add others.

The "Property name" field specifies the name of a property. (Incidentally, case distinctions do matter in property names "foo" is a different property than "Foo".) The "Property value" field specifies a value. The box lets you pick whether you are talking about properties of the root node or the selected nodes.

Get fills the "Property value" field with the current value of the named property.

Set reads the "Property value" field and sets the property value.

Remove removes the named property from the node.

List fills the "Property name" field with the names of the node's properties.

In addition to setting and reading properties, there is a mechanism that lets you find nodes with a certain property value. For example, if you have annotated a document with comments under the property name "MyOpinion", and you want to find nodes in which your comment included the word "good", enter the name in the "Property name" field and the text in the "Value pattern" field. Then use the "Find" command to search forward or backwards from the caret node for a node with a value of the property that matches the pattern. (Click with LEFT to search forward, RIGHT to search backwards.) Value patterns can use the standard set of special characters for searches. If a match is found, the node is selected and the property value is displayed.

Comment Property

All nodes have a "Comment" property which is either "TRUE" or "FALSE". If its value is TRUE, the text of the node is not seen by programs such as the compiler that are only interested in

the basic text contents. This makes it possible to intermix documentation and program text without requiring special escape characters to mark the start and end of comments. You can set the Comment property by using the Edit Tool or with the following commands.

CTRL-\ set comment property of all selected nodes to TRUE

CTRL-SHIFT-\ set comment property of all selected nodes to FALSE

Miscellaneous

Sort and Reverse

These commands let you sort or reverse lists of things. The "Sort" command sorts things in alphabetical order, ignoring case. "Sort-and-remove-duplicates" is useful when you are merging sets of things. The box below the Sort button lets you pick whether you want increasing or decreasing order in the result. The right box lets you pick what will be sorted: text delimited by blanks, lines delimited by carriage returns, or branches of the document tree. Leading blanks are ignored when sorting lines or branches.

Operations

These commands let you construct simple edit macros. The "Operations" field holds a text description of a command sequence. The "GetLast" command fills in the Operations with the description of the most recent sequence, i.e., the one a Cancel would cancel or a Repeat would repeat. "Do" executes the description in the operations field. "Begin" marks the start of a command sequence. "End" fills the Operations with the description of the commands since the most recent "Begin". "SetCom" stores the operations under the CTRL-number key for the number in the "Command [0..9]" field. Conversely, "GetCom" fills the "Operations" field with the current CTRL-number definition.

To see how this works, select the following word: "hello". Hit DEL and type in "howdy". Now hit the "GetLast" button. The Operations field should now contain: Delete "howdy". Edit the Operations field contents to replace "howdy" by "goodbye", then hit the "SetCom" button (first put 1 in the Command field if it's not already there). Now select the "howdy" you typed earlier and hit CTRL-1. If all went well, the "howdy" will have been deleted and "goodbye" inserted in its place. More examples of edit macros will be given later.

Searches and Operations on Files

You can use the EditTool to look through a list of files for one in which the current search specifications are satisfied. This can be accomplished by clicking the "SearchEachFile" button after filling in the "Files" field with the file names (or "@" followed by the name of a file that holds the list of file names). When you click SearchEachFile, a new viewer is created, and one-by-one the files will be loaded and searched until a match is found. The list of files will be updated when a match is found, so that when you are finished with one file you can click SearchEachFile again to look for the next one. You can hit the "Stop" button at the top of the EditTool to interrupt the search, but you should not try doing other operations while the search is in progress since it resets the selection each time it loads a file.

Occasionally you will want to apply certain operations to an entire set of files. You can do this by filling in the "Operations" field and the "Files" field, and then clicking "DoForEachFile". A new viewer will be created, and one-by-one the files will be loaded, selected, edited, and saved. No confirmation is required for the saves, so the entire process can go on without you. In fact, you should not try to do anything else while this is going on since the operations use the primary selection. The one exeception is the "Stop" button which you can hit to terminate the process.

Operations via the User Exec

You can invoke a set of operations from the User Exec as well as directly from the Edit Tool.

When you run TiogaExecCommands, one of the commands it registers is DoTiogaOps which expects a command line containing operations in the same format as in the operations field. One possible application of this is to create a command file that initializes various Edit Tool switches to your favorite settings. For example, you could use the following to set up some of the search parameters: DoTiogaOps IgnoreCase MatchWords MatchPattern.

Edit Commands

This section of the Edit Tool contains buttons for all the basic edit commands. These are useful if you can't remember what keys to hit for an infrequent command or if you've set your user category to beginner or intermediate to filter out certain commands. Many of the buttons correspond to commands that have been documented above. However, a few of them are primarily of use in constructing edit macros and have not been mentioned before. For completeness, all the buttons will be given a brief description.

Modifying selections

CaretBefore move caret to front of selection

CaretAfter move caret to end of selection

CaretOnly reduce selection to a caret

Grow selection grows in hierarchy of point, character, word, node, branch

Document select entire document

PendingDelete make primary selection pending delete

NotPendingDelete make primary selection not pending delete

MakeSecondary make the primary selection become the secondary selection

MakePrimary make the secondary selection become the primary selection

CancelSecondary remove the secondary selection

CancelPrimary remove the primary selection

Copy, Move, and Translate

ToPrimary copy/move secondary to primary

ToSecondary copy/move primary to secondary

Transpose transpose the primary and the secondary

Moving the caret

GoToNextChar make primary caret only and move one character toward end of document or one toward start if you click with RIGHT instead of LEFT

GoToNextWord like GoToNextChar, but move toward end by "words" or one toward start if you click with RIGHT instead of LEFT

GoToNextNode move caret one node toward end or one toward start if you click with RIGHT instead of LEFT

Saving and Restoring the selection

SaveSel-A save primary selection to restore later

RestoreSel-A restore the selection previously saved by SaveSel-A

SaveSel-B save primary selection to restore later

RestoreSel-B restore the selection previously saved by SaveSel-B

Extend the selection to matching brackets

(...) extend primary selection to matching parens

<...> extend to matching angle brackets

{...} extend to matching curly brackets

[...] extend to matching square brackets

"..." extend to matching double quotes
‘...’ extend to matching single quotes
-...- extend to matching dashes
... extend to matching placeholder brackets

Find placeholders
Next ... move primary selection to next placeholder
Prev ... move to previous placeholder

Delete and Paste
Delete delete the primary selection
Paste insert saved text at caret
SaveForPaste save text for later pasting; overwrites text saved by Delete

Repeat and Undo
Repeat repeat the last command sequence
Undo undo the last command sequence

Deleting character/word next to caret
BackSpace delete the character to the left of the caret
BackWord delete the word to the left of the caret
DeleteNextChar delete the character to the right of the caret
DeleteNextWord delete the word to the right of the caret

Inserting matching brackets around the selection
Add( ) insert parens around the selection
Add< > insert angle brackets
Add{ } insert curly brackets
Add[ ] insert square brackets
Add" " insert double quotes
Add‘ ’ insert single quotes
Add- - insert dashes
Add insert placeholder brackets

Capitalization
AllCaps make the selection upper case
AllLower make the selection lower case
InitialCaps make the words in the selection start with caps
FirstCap make the selection start with a cap

Special characters
MakeOctalCharacter convert the 3 digits before the caret to the corresponding character
MakeControlCharacter convert the character before the caret to a control character
UnMakeOctalCharacter convert the character before the caret to 3 digit representing its character code
UnMakeControlCharacter convert the control character before the caret to a normal character

Tree structure commands
Break break node at caret
Join join caret node to one before it

Nest move selection deeper in tree

UnNest move selection higher in tree

Miscellaneous commands

CommentNode set Comment property of all selected nodes to TRUE

NotCommentNode set Comment property of all selected nodes to FALSE

ExpandAbbreviation expand the abbreviation to the left of the caret

MesaFormatting add Mesa looks, formats, and style to selection

Command 0 1 2 3 4 5 6 7 8 9 do saved command macro

Writing Edit Macros

A few examples should get you started writing your own edit macros. First, assume you find yourself doing a lot of edits of the form <stuff> becomes <pred>[<stuff>] for various values of <stuff> and <pred>. You might like a single command to insert the brackets and move the caret to the place where you will insert the <pred>. To construct such a command, first select a particular <stuff>, hit the Add[ ] button in the Edit Tool, the CaretBefore button, and the GoToNextChar button using the right mouse button. Then hit GetLast to fill the Operations field, enter "1" in the "Command [0..9]" field, and hit SetCom to save the macro definition. Now you can select <stuff>, hit CTRL-1, and be ready to insert before the left bracket.

A macro to delete matching parentheses will show the use of the SaveSel and RestoreSel commands. Make a selection anywhere inside a pair of matching parens. Give the following sequence of commands: (...) to extend the selection, CaretAfter to position the caret at the right, SaveSel-A so we can restore the selection later, BackSpace to delete the right paren, RestoreSel-A to get the selection back, CaretBefore to move the caret to the front, and finally DeleteNextChar to delete the left paren. Now you can hit GetLast and SetCom to save this macro.

Other operations on the Edit Tool can also be programmed using edit macros. For example, assume you want a macro to substitute "which" for "that". The following set of commands will load the target and replacement fields and do the substitute: click RIGHT to select and clear the target field, type "that" as the target text, click RIGHT to select and clear the replacement field, type "which" as the replacement text, and finally hit the Substitute button. This macro will only change the target and replacement fields. The other parameters of the substitute are not changed and thus behave like "free variables" of the macro. If you want to bind more of the choices, such as Match Case or Ignore Case, you simply include those commands as part of the macro. For example, you could select Ignore Case just before restoring the selection, and the macro would always set the ignore case flag when it was executed. Note that you must hit the "Target" and "Replacement" buttons to enter the target and replacement text. Directly selecting in those fields would not produce a correct macro because it would terminate the edit sequence and would also fail to identify which field was being filled in.

When you save an edit macro under a CTRL-number key, it only stays around for the rest of the session. If you want to save one for tomorrow, you can of course copy the operations to a file and redefine it another time. But if you really want to make a macro a permanent part of your user interface, you can add it to your "TIP table" which describes the translation from keyboard and mouse actions into executable tokens such as those in edit macros. The details of how to do this are given in a later section.

The Edit History Tool

The Edit History tool lets you undo edits in the same way that Undo does, but it lets you go back farther in history than just the most recent sequence. To create an Edit History tool, type "EditHistory" to the Exec. At the top of the tool are two fields and four commands. The bottom part of the tool is a text field to hold descriptions of previous edit events.

The system keeps a history of a certain number of the most recent edit events. You can find out how large this history is by the "Get" command which will enter a number in the "history size" field. The "Set" command will change the history size to whatever value you've entered in the size field. The default size is 20; you can change this by making an entry in your User.Profile of the form "EditHistory: <history-size>".

The "Show" command will display the events starting with the number in the "since event number" field. If that field is empty it will show as many as are still remembered. The format of the entries is <event-number>, TAB, and then a list of operations. The "Undo" command undoes the edits since the specified event number.

Printing and the TypeSetter Tool

In the glorious future, Tioga will have an interactive typesetter that will let you make incremental revisions to a typeset document to adjust pagination, page layout, and other details before actually printing it. Michael Plass and I have started work on this, but until it's available, the current typesetter will do a more than adequate job of letting you print your files. The following documentation was written by Michael; see him with questions or bug reports.

The TSetter tool provides a convenient way of driving the Tioga typesetter. Start it up by typing "Run TSetter" to the executive. The typesetter icon will come up showing the name of the print server it is "connected" to; this name can be specified in your profile under the heading "Hardcopy.PressPrinter". When you open the icon, you will see the menu

Stop Pause Get Print All New

plus a fill-in field labeled "Document:". The normal way to use the typesetter is to select the viewer or file name, and click "Get" to add the document name to the queue. Then click "Print" to start the typesetter up; if all goes well, the only other thing you have to do is walk over to the printer and pick up your output.

You can put several things in the queue before starting up the typesetter, and go do something else while it is working away. The queue is shown after the "Document:" button and is in fact editable; it is a good idea to click "Pause" before editing, though, to keep the typesetter from taking the queue out from under you. Clicking "All" instead of "Print" funnels the whole queue into one big press file, which avoids a lot of header pages if you want to print many small files.

Hitting "Stop" will bring the typesetter to a grinding halt and will also abort any transmission to a print server.

Destroying a tool will not stop it; it will just finish running and then go away.

If you try to typeset a press file, it will just get sent to the server without modification. After it is successfully transmitted, it will be deleted if its name ends in a "$".

To make a TSetter tool that sends its output to a different place, type the server name somewhere, select it, and click "New". If the selection is less than two characters long, the resulting tool will be called "TSetter", and will just write a press file without trying to send it anywhere. It is OK to run multiple typesetters at the same time.

If you want to print multiple copies, "Grow" the tool and you will discover a "Copies" field to fill in. This number is reset to 1 after each file is transmitted, to keep you from wasting a lot of paper by forgetting to reset it.

When you make the tool bigger, you will also discover a button labeled "Temporary Press Files", followed by a Boolean. If this is TRUE, the typesetter will create press files with names that end in "$", so that they will be deleted after they are successfully transmitted. Clicking the button toggles the Boolean.

To drive the typesetter from a program (or the executive), there is an interface called "TSViewer" that lets you create a tool and simulate clicks on its buttons.

TIP Tables for Tioga

The acronym "TIP" stands for "terminal interface package". TIP tables describe the translation from keyboard and mouse actions into executable tokens. The standard TIP table for Tioga is found in the file "Tioga.TIP" and contains all the gory details about things such as multi-clicks of mouse buttons with various combinations of shift keys. It's unlikely that you want to try changing anything in Tioga.TIP, but you can consider adding commands to your own TIP table for things such as your favorite set of edit macros. Your TIP table can be "layered" on top of the Tioga table so that the system will try to interpret actions according to your table before looking at the standard one. The User.Profile contains an entry named "TiogaTIP" in which you can enter the file name of your TIP table to be layered over the standard Tioga table. The entry should look like something like this

TiogaTIP: MyOwn.tip Default

where "MyOwn.tip" is the file name for your table. Notice that your table goes first in the list; definitions in tables occurring early in the list take precedence over those that appear later. The special name "Default" refers to Tioga's default TIP table and will typically be the last entry in the list.

The file "MyOwn.tip" contains a sample table that you can edit to produce your own. It also contains documentation about the syntax of TIP tables and the macro package which is used with them.

Use the Exec command TiogaReadProfile to reload a TIP table after you've changed it, or, if your user category is advanced, hit CTRL-! to reload the Tioga profile information.

Styles

A style is a collection of interpretations for looks and formats. The interpretations are represented as programs written in a simple language built on top of the JaM interpreter.

If a document doesn't specify a style, the system will use a default style. You can say what the default will be by adding a user profile entry of the form DefaultStyle: <style name>.

.... more to come ....

ScreenRules, PrintRules, and StyleRules.

Attached styles

The current set of formatting parameters are ...

Tioga User Exec Commands

Note: If the following commands are not known to the UserExecutive, type "Run TiogaExecCommands".

ReadTiogaTipTables

Causes Tioga to read its TIP tables again. If your user category is advanced, you can also invoke this operation by selecting in any Tioga document and hitting CTRL-!.

WritePlain

Make Tioga files unformatted by eliminating everything except the plain text. Inserts leading tabs before each node according to its nesting in the tree and terminates each node with a carriage

return.

ReadIndent

Build Tioga files with one node per line of source with indenting based on white space at the start of lines.

TiogaMesa

Convert Mesa files to Tioga format by combining a ReadIndent with a CTRL-M over the entire file.

Tioga User Profile Entries

User category

As described above, this entry lets you control your user category. The alternatives are Beginner, Intermediate, and Advanced. The entry is of the form

UserCategory: Intermediate

TIP Table

This entry specifies the TIP tables to use with Tioga documents. The entry is of the form

TiogaTIP: MyOwnTip.tip Default

See the section on TIP tables for more information.

Default Style

This entry specifies the default style to be used with documents that do not explicitly name a style. The entry is of the form

DefaultStyle: Cedar

Default submenus

This entry specifies which menus, if any, should automatically be displayed when you create a new Tioga viewer by clicking one of the buttons in the upper right corner or by giving an Exec command. The entry is of the form

DefaultTiogaMenus: places levels

or

DefaultTiogaMenus: none

You may delete or reorder the menu names to suit your tastes.

Default file extensions

This determines what extensions Tioga should look for in opening files. The entry is of the form

SourceFileExtensions: mesa tioga df cm config style

You may also have an entry for implementation extensions to be used with the "load impl" commands.

ImplFileExtensions: cedar mesa

Scroll top offset

When you do a Find command you may want to see a few lines in front of the match to give you more context. This parameter tells Tioga how many extra lines to want in such situations. The entry is of the form

ScrollTopOffset: 1

Scroll bottom offset

When you are typing and the caret goes to a new line just off the bottom of the viewer, Tioga will automatically scroll the viewer up a little to make the caret visible again. This parameter controls how far up to scroll; a big number causes larger but less frequent glitches.

ScrollBottomOffset: 3

Selection Caret

The default behaviour for Tioga is to place the caret at the end nearer the cursor when the selection is made. Some people have requested to have the caret always placed at one end or the other, hence this profile entry. The choices are before, after, and balance. The entry is of the form

SelectionCaret: before

Selection Displacement

This lets you specify a vertical displacement for making selections. Tioga behaves as if you had pointed this number of points higher up the screen so that you can point at things from slightly below them. The entry is of the form

YSelectFudge: 2

Command Summary

This is a short summary of the Tioga commands available using the keyboard and mouse. Other commands are available through the Edit Tool, the Edit History Tool, and various User Exec operations.

The extra keys on the keyboard are used for common editing actions. Recall that the bottom right blank key can be used as another CTRL key, and you can use either CTRL key and either SHIFT key interchangeably.

ESC	Repeat last action
SHIFT-ESC	Undo last action
ESC-select	Automatic repeat of last action when finish selection
DEL	Delete
LF	Load file in "No Name" viewer
CTRL-LF	Load Impl file in "No Name" viewer
BS	Backspace character
SHIFT-BS	Delete next character
CTRL-BS	Backspace word
CTRL-SHIFT-BS	Delete next word
NEXT	Find next placeholder (middle blank key)
SHIFT-NEXT	Find previous placeholder
RETURN	Insert carriage return with leading spaces copied from previous line
SHIFT-RETURN	Insert carriage return
CTRL-RETURN	Break node

Mouse button clicks are used for making selections.

CLICKS	LEFT	MIDDLE	RIGHT
SINGLE	Select letter	Select word	Extend selection at current level
DOUBLE	Select node	Select branch	Reduce selection level and extend
TRIPLE			Increase selection level and extend

Selections are used for delete, copy, and move.

CTRL-select	Delete when finish selection
SHIFT-select	Copy to primary
CTRL-SHIFT-select	Move to primary

LOOK commands are used for editing looks. (The LOOK shift is the top blank key.)

LOOK-char	Add look to selection
LOOK-SHIFT-char	Remove from selection
LOOK-space	Remove all from selection

Except for CTRL-A, CTRL-H, and CTRL-W, the following commands are not enabled for beginning users. We provide both CTRL-A and CTRL-H as a convenience to users with strong habits from previous systems. A "*" indicates a command enabled for advanced users only.

CTRL-A	Backspace character
CTRL-SHIFT-A	Delete next character
CTRL-B	Insert matching placeholder brackets
CTRL-C	Lower case
CTRL-SHIFT-C	Upper case
CTRL-CLICK-C	Initial caps
CTRL-SHIFT-CLICK-C	First cap
CTRL-D	Select document
CTRL-E	Expand abbreviation

CTRL-F-select	Copy format to primary *
CTRL-H	Backspace character
CTRL-SHIFT-H	Delete next character
CTRL-I	Indent (does Break and Nest) *
CTRL-SHIFT-I	Unindent (does Break and UnNest) *
CTRL-J	Join nodes *
CTRL-K	Make control character
CTRL-SHIFT-K	Unmake control character
CTRL-M	Automatic MESA formatting
CTRL-N	Nest *
CTRL-SHIFT-N	UnNest *
CTRL-O	Make octal character
CTRL-SHIFT-O	Unmake octal character
CTRL-P	Paste
CTRL-Q-select	Copy looks to primary
CTRL-S-select	Copy primary to selected destination
CTRL-T	Time
CTRL-V	Select visible (expand to selection to blanks)
CTRL-W	Backspace word
CTRL-SHIFT-W	Delete next word
CTRL-X-select	Select for transpose with primary
CTRL-Z-select	Move primary to selected destination
CTRL-]	Select matching [..]'s (same for },), and >)
CTRL-[	Add matching [..]'s (same for ", ', -, {, (, and <)
CTRL-!	Have Tioga read its TIP tables again *
CTRL-_	Set format to word before caret *
CTRL-SHIFT-_	Insert format name *
CTRL-\	Set comment property TRUE *
CTRL-SHIFT-\	Set comment property FALSE *