To: MTB Distribution

From: E. J. Wallman

Date: 1978 August 07

Subject: Embedded Controls Feature for compose.

This MTB proposes a generalized mechanism for compose that allows the user to insert formatting parameter changes anywhere in the text and to place any arbitrary bit string in the output data stream to the output device.

As the capabilities of Multics compose reach further and further toward the standards of traditional typesetting, more and more sophisticated features are demanded. Certain documentation standards require that a text formatter be able to change formatting parameters at lowest possible level, that is, in the middle of a word. The reader has undoubtedly seen documents in which changes of this kind are made; most notably are those documents containing words with emboldened, italicized, or underlined syllables and the many forms of subscript and superscript notation.

Some of these typographic features may be accomplished fairly easily with the existing compose controls, but many lead to great difficulty. For example, when using the DTC300/s terminal (or any of the many incremental motion output devices now on the market), a simple half-line shift is accomplished by transmitting an escape sequence to the device. This implies that the user must place ASCII ESC characters in his data and suffer the pangs of editing with the terminal in ^edit mode. (Even with compose's greatly extended formatting capabilities, there are still many formatting features that are beyond it and must be done manually by the user with a text editor.) Moreover, in order to transmit the escape sequence to the device, tty_ must be put into rawo mode. It is apparent that the skills and knowledge to do this in some general form, while available to any competent system programmer, are far beyond the casual documentation user.

An even greater difficulty is the fact that, if the above approach to sophistication is taken, the document becomes inextricably tied to a particular terminal device and cannot be printed on any other device; including the line printer! It was for this last reason that the conventional artwork constructs were "invented" for compose.

Now, however, the support of phototypesetting devices opens an entire new dimension of device capabilities. The syntax of the existing artwork constructs is not rich enough to include the wide range of font changes, point size changes, leading, and other device controls necessary to produce a document on a phototypesetter.

compose already has the ability to perform several actions on input file lines that are subject to substitution of variables; namely, simple symbolic variable value substitution, evaluation of expressions, and insertion of characters returned by any active function or obtained from the user_input I/O switch. The feature to be added is the ability to recognize and process any compose control appearing in the input data, regardless of its position with respect to line boundaries. (The arbitrary bit strings mentioned above will be retrieved from an external device support table by a new compose control. The device support table and the new control are the subjects of two other MTBs.)

The mechanism for the feature merely encloses the desired control with symbol delimiters in a line that is subject to substitution of variables as follows:

```
     .ur <arbitrary text>%.ctl <ctl_param> ...%<more text>
```

The embedded control may not contain symbol delimiters when it is passed to the control processor since the first symbol delimiter following the opening "%." is taken to be the closing delimiter for the control. This means that, for complex constructs, the embedded control must be at the outermost nesting level so that it is the last item substituted in the line. For example, the user desires a cross-index list of the usage of certain word roots and writes a special "hit" macro to contruct the list. The usage of the macro could be (for the word root "limit"):

```
.ur The un%.ifi hit limit%ed use of such de%.ifi hit limit%ers
.ur eliminates all of the aforesaid %.ifi hit limit%tations.
```

The user documentation for this feature was inadvertently published before the feature was completed. See "Embedded Controls" on page 4-10 of AZ98, Rev. 0.