

Variables

FOSI has two kinds of variables: **counter variable** and **string variable**, which are described below.

NOTE: Arbortext Editor distinguishes between counter and string variables saved in the `pagedesc` and counter and string variables saved in the `styldesc` or `ftndesc`. You cannot change the value of a `styldesc` or `ftndesc` counter or string variable in the `pagedesc`. However, a string variable that is output in an `e-i-c` as a section number can also be output in a page header or footer.

Similarly, with one exception, you cannot change the value of a `pagedesc` counter or variable in an `e-i-c` in the `styldesc` or `ftndesc`. It is possible, however, to output the value of a `pagedesc` string variable in the `styldesc`. For example, a counter that is incremented and saved to a `textid` in a `pageres` and output in a page header or footer can also be output in a table of contents or index. The exception case is that a `pagedesc` counter can be reset to its initial declared value in an `e-i-c` that forces a page break.

Mirror-image title quadding on page 664 provides an example of using a variable in the `pagedesc` that is saved in the `styldesc`.

Counter variable

A **counter** variable is a numeric variable whose value may be zero or a positive integer. A declared counter has:

- a unique name
- an optional initial value (the default is the empty string)
- a numbering/lettering style
- a lettering sequence style for letters beyond “z” and “Z”: aa, ab, ac, ... (the default) or aa, bb, cc, ...
- an optional padding length for leading zeroes

The available numbering styles for counter variables are:

- Arabic number (the default): 0, 1, 2, ...
- Lowercase alphabet character: a, b, c, ...
- Lowercase Roman numeral: i, ii, iii, ...
- Uppercase alphabet character: A, B, C, ...
- Uppercase Roman numeral: I, II, III, ...
- Korean characters (all 14 consonants and 10 vowels available for sequence)
- Korean characters (14 consonants only)

Two lettering sequences are supported for letters beyond “z” and “Z”: ..., y, z, aa, ab, ac, ... (the default) and ..., y, z, aa, bb, cc, ...

Padding with leading zeroes can be specified for Arabic numbers.

The declared name of the counter can be referenced by:

- an `enumerat enumid`, which is used to change the value of the variable.
- a `reset resetlist`, which restores the initial value.
- a `savetext conrule`, which stores the current value of a counter
- a `usetext source`, which outputs the current value of a counter.
- a `specval attloc`, which tests the value of a counter using #FOSI

FOSI counter variables are time-dependent, which means the value of a counter at any particular point in the document is the last value saved to it. See **Enumerat** on page 462 and **Reset** on page 467 for details on changing the value of a counter.

The contents of a **counter** can be tested with `#FOSI` in an **attribute rule**, as described in **Att** on page 496.

Also see **Counter** on page 233 for more information.

COUNTER MODIFIERS

Counter modifiers can be used to specify the style in which the **counter** is displayed when it is saved to a **savetext conrule** or output from a **usetext source**. The modifier overrides the **style** characteristic specified in the **counter** category.

Supported modifiers are as shown in the following table.

Table 8 Supported Counter modifiers

Modifier	Description	Counter style	Examples
[AR]	Arabic numbers	arabic	1, 2, 3, 4
[RU]	Uppercase Roman numerals	romanuc	I, II, III, IV
[RL]	Lowercase Roman numerals	romanlc	i, ii, iii, iv
[AU]	Uppercase alphabetic letters	alphauc	A, B, C, D
[AL]	Lowercase alphabetic letters	alphalc	a, b, c, d
[KA]	Korean characters, all 14 consonants and 10 vowels available for sequence	kanada	
[JA]	Korean characters, 14 consonants only	jaeum	

COUNTER MODIFIER EXAMPLE

The FOSI fragment below shows a counter modifier in a **usetext source**. It would be the same in a **savetext conrule**.

Figure 33 Change Arabic numbers to Roman numerals**FOSI fragment**

```
<counter enumid="itemct style="arabic" ...>  
...  
<usetext source="itemct[RU],\.\ \"></usetext>
```

String variable

A FOSI **string** variable may contain the constructs described in **Table 74 Savetext conrule and usetext source syntax** on page 472.

There are two kinds of string variables in a FOSI: time-dependent (tdv) and time-independent (tiv), as detailed below.

TIME-DEPENDENT VERSUS TIME-INDEPENDENT VARIABLES

When a string variable is declared in a `stringdecl` category, setting `status="0"` makes it a time-dependent variable (tdv). The contents of a tdv are assumed to be available whenever the string variable is output from a `usetext souce`. Setting `status="1"` declares a time-independent variable (tiv), which means at least one formatting pass is required for the tiv to have a value.

A variable that is declared as time-independent signals that a value is not available when the variable is encountered during the first formatting pass through the document. Instead, at least one more formatting pass is necessary in order to save the desired content to the variable. This is illustrated in **Figure 112 Time-independent variable for caption** on page 238 .

STRING VARIABLE EXAMPLES

A time-independent variable is used for creating a table of contents from `<title>` elements in the document. An example of this is shown in **Figure 225 Appended tiv generates a table of contents** on page 475. A table of contents for each chapter in a document also uses a time-independent variable, as shown in the following example.

Figure 34 Scoped tiv for chapter TOCs

CHAPTER 1	
LOREM IPSUM DOLOR SIT AMETER	
CONTENTS	
Sed ultrices condimentum	Pellentesque quis posuere metus
aliquam 4	pretium consectetur..... 8
Curabitur feugiat imperdiet velit	Quisque pretium eros ut sem
sodales laoreet..... 6	dignissim scelerisque..... 10

STRING MODIFIERS

A `textid` can be saved to more than once in a document or a page, as well as in one or more `pageress` in the `pagedesc`. **String modifiers** are used with `textids` to specify which value to use.

STRING MODIFIERS FOR PAGEDESC OUTPUT

The value of a time-dependent string variable (`tdv`) can change on one page. For example, if every `<formal-para>` has a required `<title>`, which is saved to a `tdv` such as `title.txt`, then the value of `title.txt` at the top of the page may be different than the value of `title.txt` at the bottom of the page. If `title.txt` is output in the page header, which one of the different titles on that page should it be? FOSI string modifiers provide a way to specify what is desired.

FOSI has five modifiers that specify the:

- first value saved to the string variable on that page: [FI]
- the existing value of the string variable at the top of the page, before any other value has been saved to it: [TO]
- the last value of the string variable on the page: [BO]
- every value of the string variable, starting with the first value saved to it on that page: [FB]
- every value of the string variable, starting with the value at the top of the page before any other value has been saved to it: [TB]

“Dictionary headers” are probably the best way to explain the purpose of string modifiers. The page header in a dictionary generally has the first word defined on that page and the last word on the page, such as “jawbone — jeopardy.”

In the dictionary I’m looking at, the definition for `jawbone` starts on the first line on the page. The definition of `jeopardy` starts near the bottom of the page and continues on the top of the next page. The header on the next page is “jetty — jitney.” The definition for `jetty` starts on the third line on the page.

Many documents share this need to output the first “something” (often a title) on a page and/or the last “something” on a page in page headers and/or footers. A further refinement is distinguishing between the first something that continues from the previous page from the first something on the current page.

...

TIP 

Software users dislike and may even fear error messages. To minimize questions and complaints, your best bet is to eliminate error messages whenever possible so users don't see them.

When a document is formatted, elements in the document are matched with `e-i-cs` in the FOSI, and any `savetext` categories in an `e-i-c` are processed to assign a value to each string variable. The value of a string variable is output from a `usetext` source in an `e-i-c`. For example: `usetext source="title.txt[FI]"`.

String modifiers are required to get the output you want. If a string modifier is not included when required, an error message will be displayed that `[BO]` is assumed.

STRING MODIFIERS FOR STYLDESC OUTPUT

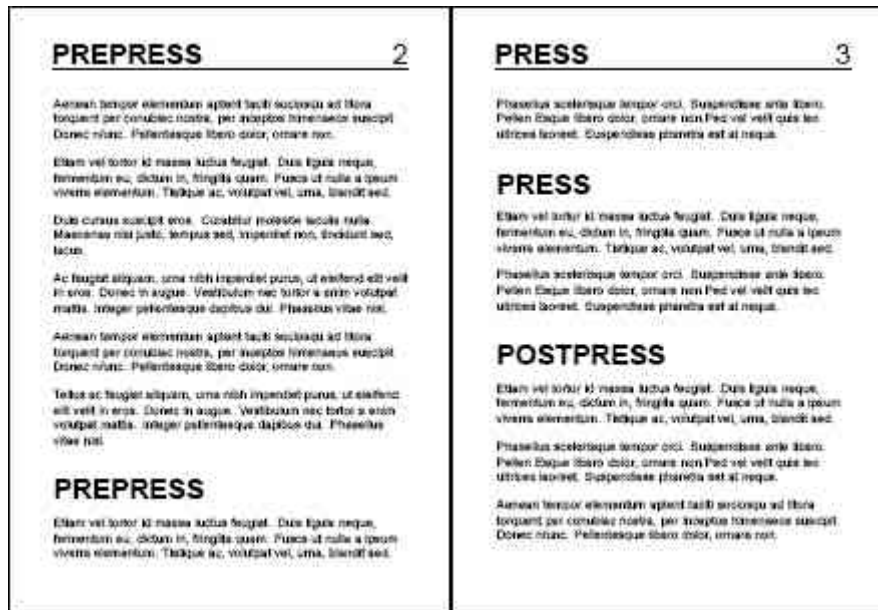
that may be appended to the name of a string variable saved in the `Styldesc` when it is output from in a page header or footer or to a string variable saved in the `Pagedesc` with a page number value.

`pagedesc` string variables that are output in the `styldesc` usually use the `[BO]` modifier, which returns the value of the string variable after the `pageres` is processed. `[FI]` and `[TO]` return the string variable value before the `pageres` is processed, which is rarely if ever needed.

STRING MODIFIER EXAMPLES

The first five examples illustrate modifiers for output in the `pagedesc`. The `[FI]` modifier specifies the first value assigned to the string variable on that page, as shown in the following example:

Figure 35 First title.txt on page

**XML fragment**

```
<section><title>Print Processes</title>...</section>
<section><title>Prepress</title>...</section>
<section><title>Press</title>...</section>
<section><title>Postpress</title>...</section>
```

FOSI fragment

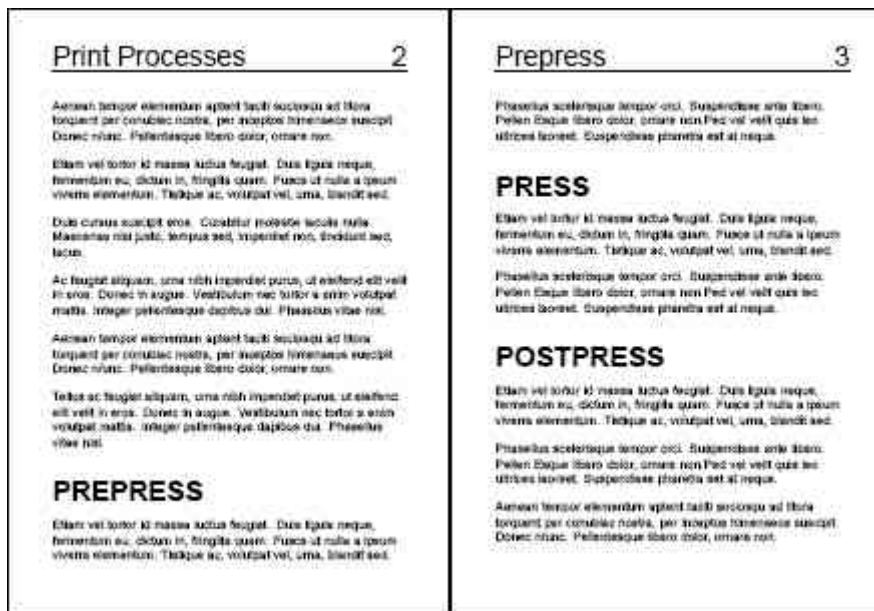
```
<stringdecl textid="title.txt" literal="">
...
<header nomdepth="23pt" spaflow="14pt">
<usetext source="title.txt[FI],spacefill,folioct">
<subchars charsubsetref="block">
<font size="18pt">
<leading lead="18pt"></subchars>
</usetext>
<ruling thick="1pt" lentype="rel" relen="col" type="single">
<leading inherit="1" lead="4pt">
<textbrk startln="1" endln="1">
</ruling>
</header>
...
<e-i-c gi="title" context="section">
<charlist inherit="1" charsubsetref="title allcaps">
<font inherit="1" size="18pt">
<leading inherit="1" lead="18pt">
```

```
<savetext textid="title.txt" conrule="#CONTENT">
```

```
...
```

The [TO] modifier specifies the value of the string variable at the top of the page. For example:

Figure 36 title.txt value at page top

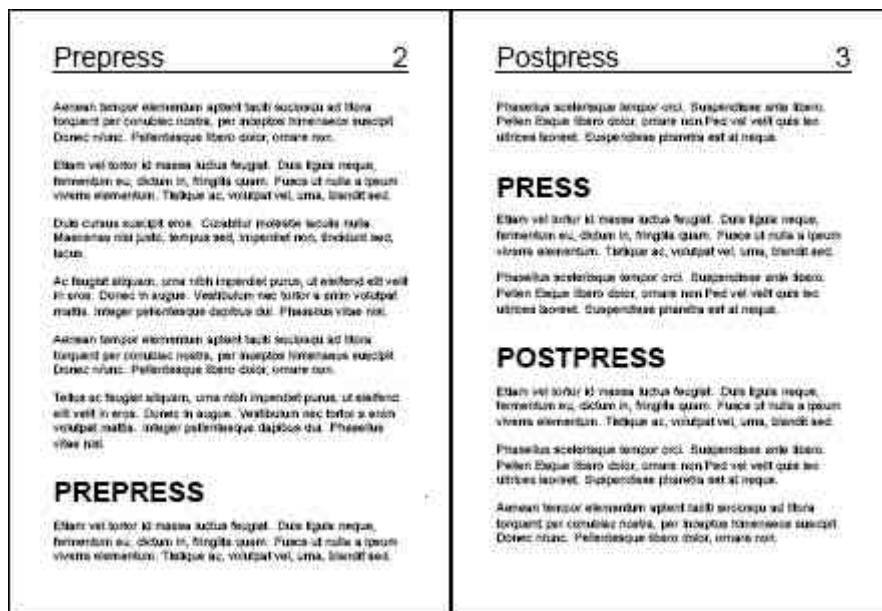


FOSI fragment

```
<header nomdepth="23pt" spaflow="14pt">
<usetext source="title.txt[TO],spacefill,folioct">
...
```

In the following example, the [BO] modifier specifies the value of the string variable at the bottom of the page.

Figure 37 Title.txt at page bottom

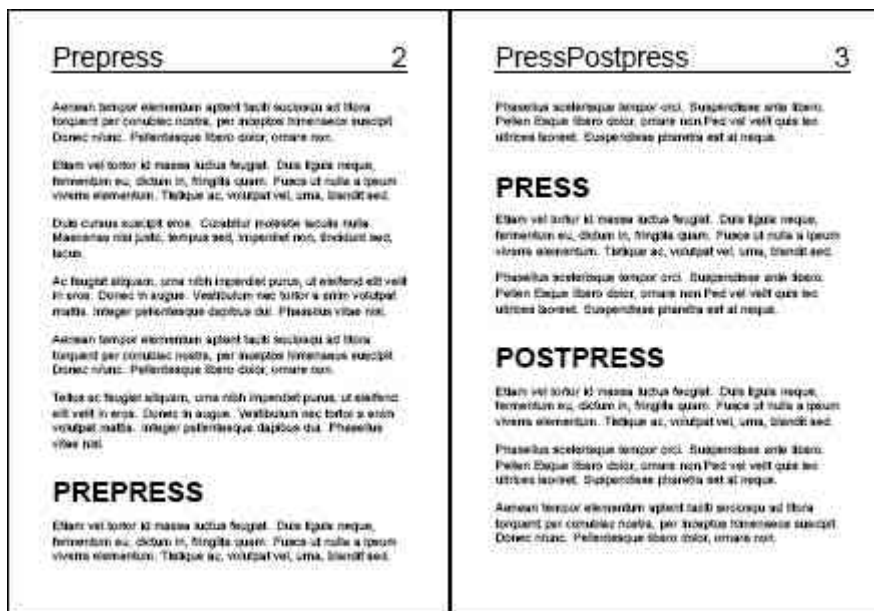


FOSI fragment

```
<header nomdepth="23pt" spaflow="14pt">
<usetext source="title.txt[B0],spacefill,folioct">
...
```

[FB] outputs all the titles on the page, starting with the first title on the page.

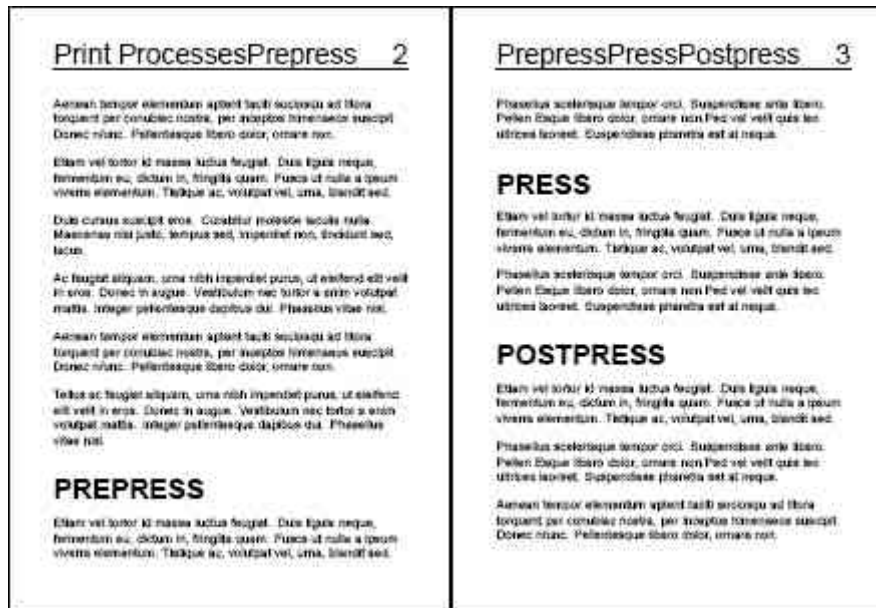
Figure 38 Page top through page bottom title.txt

**FOSI fragment**

```
<header nomdepth="23pt" spaflow="14pt">
<usetext source="title.txt[FB],spacefill,folioct">
...
```

[TB} outputs all the titles on the page, starting with the title in effect at the top of the page.

Figure 39 Title.txt at page top through page bottom

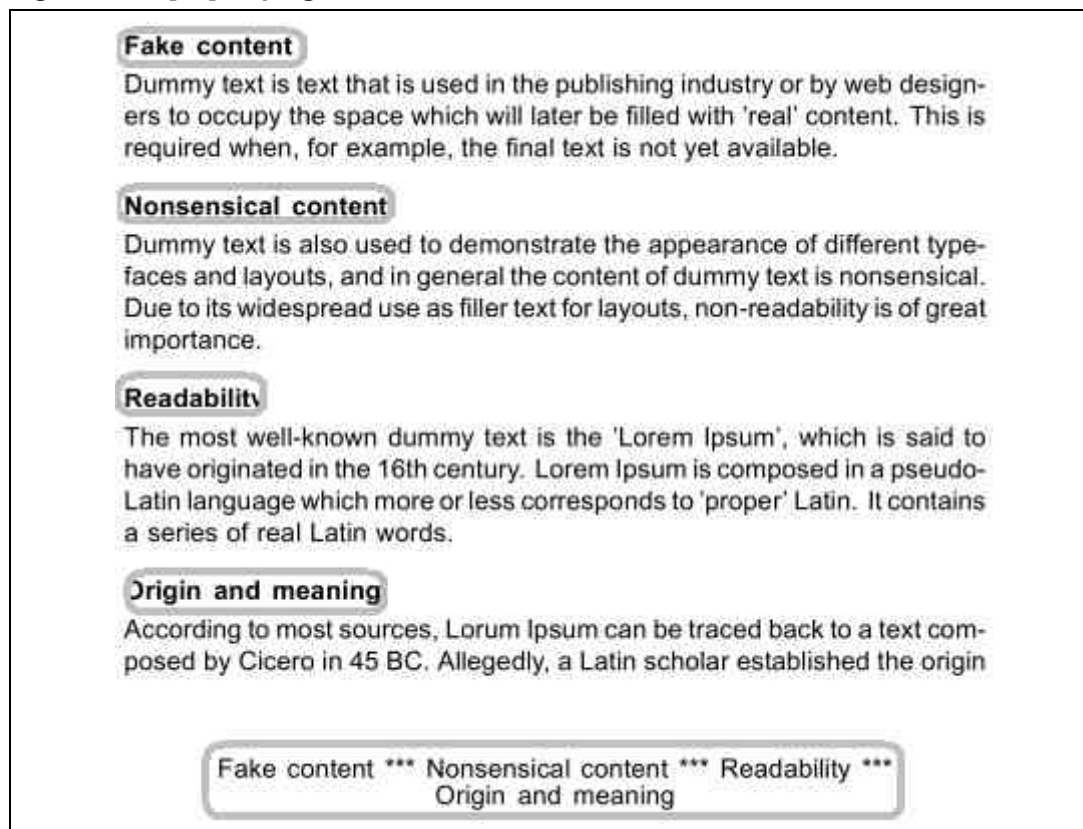


FOSI fragment

```
<header nomdepth="23pt" spaflow="14pt">
<usetext source="title.txt[TB],spacefill,folioct">
...
```

The next example illustrates how to format the contents of the string variable returned by [FB] and [TB].

Figure 40 [FB] in page footer

**FOSI fragment**

```
<stringdecl textid="section-title.txt">
<footer nomdepth="30pt" spaflow="12pt">
<usetext source="section-title.txt[FB]">
<subchars charsubsetref="footer"></subchars>
</usetext>
</footer>

<e-i-c gi="title" context="section">
<charlist inherit="1" charsubsetref="title">
<savetext textid="section-title.txt"
conrule="<FB.fmt>,#CONTENT,</FB.fmt>" placemnt="before">
</charlist>
</e-i-c>
```

```

<e-i-c gi="FB.fmt" occur="notlast">
<charlist inherit="1">
<usetext source="\ *** \" placemnt="after"></usetext>
</charlist>
</e-i-c>

```

```

<e-i-c gi="FB.fmt" occur="last">
<charlist inherit="1"></charlist>
</e-i-c>

```

[BO] is used in the last example to save the current page number to a Table of Contents.

Figure 41 Page numbers in a TOC

FOSI fragment

```

<counter initial="0" style="romanlc" enumid="frontfoliocr">
<stringdecl textid="frontfoliocr.txt" literal="">
...
<pageset id="front.page" blankpg="0" orient="portrait">
<pageres>
<enumerat increm="1" enumid="frontfoliocr">
<savetext textid="frontfoliocr.txt" conrule="frontfoliocr">
</pageres>
<pagespec pgid="front.recto">
...
i-c gi="title" context="* front">
<charlist inherit="1" charsubsetref="title">
<savetext textid="toc.app" append="1"
conrule="<toc.fmt>,#CONTENT,spacefill,frontfoliocr.txt[BO],</toc.fmt>">
...

```