

TIP 

If you code page layout debugging capability in a FOSI, be sure to include complete information about it in the FOSI documentation.

How to output page layout debugging information

It is good practice to include page layout debugging information in your FOSI because it often comes in handy. For instance, when there are several page models in a `pageset`, it may not be clear which is being used for each output page. Also, when optional content does not appear in page headers/footers, there is no way to tell if the header/footer is correctly formatted.

With page layout debugging code in the FOSI, each page model can output a string that identifies it, and hard-coded string variables can be output in headers/footers in order to verify their formatting.

An ACL variable is used to turn the debugging information on and off. It can be entered at the command line, or a menu item or keymapping could be used instead. To avoid accidentally publishing pages with debugging information, a message can be displayed in the Edit window to alert the user that debugging mode is enabled.

In this example, FOSI code is shown for one page model. In practice, the code needs to be included for all page models. In addition, the `savetexts` that provide values for the debugging strings must be coded for every element that could be the top tag in a document.

Figure 356 Conditional debugging strings

ACL command

```
set $debug=yes
```

FOSI fragment

```
<stringdecl textid="debug-doc-security.txt" literal="">
<stringdecl textid="debug-recto0.txt" literal="">
<stringdecl textid="debug-recto1.txt" literal="">
<stringdecl textid="debug-rectobb.txt" literal="">
<stringdecl textid="debug-versobf.txt" literal="">
<stringdecl textid="debug-verso1.txt" literal="">
<stringdecl textid="debug-wp-security.txt" literal="">
<stringdecl textid="doc-security.txt" literal="">
<stringdecl textid="wp-security.txt" literal="">
...
<!-- Include the following code for each page model -->
<!-- $debug=yes shows debugging text-->
<rectopg>
...
<pagespec pgid="wp-recto0">
```

```

<header>
<usetext source="doc-security.txt[BO],debug-doc-security.txt[FI]">
<subchars charssubsetref="doc-security-header">
...
<usetext source="wp-security.txt[BO],debug-wp-security.txt[FI]">
<subchars charssubsetref="wp-security-header">
...
<region xoff="36pt" yoff="36pt" width="1in" height="36pt">
<usetext source="debug-wp-recto0.txt[FI]"></usetext>
</region>
...
<!-- Include the attspecs below in the e-i-c for any element -->
<!-- that can be the top tag in a document -->
<e-i-c gi="toptag">
<charlist inherit="1"...>
...
<!-- debug strings appear in preview/print/PDF when $debug=yes -->
<att>
<specval attname="debug" attloc="SYSTEM-VAR" attval="yes">
<charsubset>
<savetext textid="debug-doc-security.txt" conrule="\doc-security\">>
<savetext textid="debug-wp-security.txt" conrule="\wp-security\">g
<savetext textid="debug-recto0.txt" conrule="\recto0\">
<savetext textid="debug-recto1.txt" conrule="\recto1\">
<savetext textid="debug-rectobb.txt" conrule="\rectobb\">
<savetext textid="debug-versobf.txt" conrule="\versobf\">
<savetext textid="debug-versol.txt" conrule="\versol\">
</charsubset>
</att>
<!-- message appears in Edit window when $debug=yes -->
<att logic="and">
<specval attname="debug" attloc="SYSTEM-VAR" attval="yes">
<specval attname="editor-only" attloc="SYSTEM-VAR" attval="yes">
<charsubset>
<usetext source="\NOTE: Debugging strings appear in page headers/footers\">
<subchars charssubsetref="onscreen-message">
...

```