

# Using IF statement for Compilation TOC and Bookmark patterns

09/16/2024 1:41 pm MDT

IF statements can be used to apply patterns based on if a card falls into a certain requirement. For example, If a card type is manual, display as X, if a card type is auto-generated, display as Y.

An example of when this would be used is if you would like to display the "Reference" for auto-generated cards, but you'd like to use the "Document Number" for manually generated cards.

IF statements can be used in the following fields:

Bookmark pattern cover page entry

Bookmark pattern for TOC entry

TOC/Bookmark pattern for cards

Table of Contents and Bookmarks sections options

Depth of TOC:*	<input type="text" value="3"/>	Depth of Bookmarking:	<input type="text"/>
Depth of TOC numbering:	<input type="text"/>	Start of TOC numbering:*	<input type="text" value="1"/>
Hide first level TOC numbers	<input type="checkbox"/>		
Bookmark pattern for cover page entry:*	<input type="text" value="&lt;Title&gt;"/>		
Bookmark pattern for TOC entry:*	<input "manual="" &lt;cardtype&gt;="manual" ,="" card="" toc";"auto="" toc)"="" type="text" value="IF("/>		
TOC/Bookmark pattern for cards:*	<input "manual="" &lt;cardtype&gt;="manual" &lt;reference&gt;")"="" &lt;reference&gt;";"auto="" ,="" card="" type="text" value="IF("/>		
Suffix when files cannot be embedded in databook PDF:	<input type="text" value="* [cannot be embedded in PDF]"/>		
<input type="checkbox"/> Append secondary codes (if they exist) to each card with suffix:	<input type="text" value="(Includes &lt;SecondaryCodeList&gt;"/>		
<input type="checkbox"/> If card has secondary codes, replicate the card TOC entry under EVERY secondary code with suffix:	<input type="text" value="(Included in &lt;PrimaryDocCode&gt; sec"/>		
<input type="checkbox"/> Suppress cards in TOC			

**Note:** "CardType" with values "Manual" / "Auto" is not presented as a button in the pattern variable list and must be typed in.

**Parameters are available to be used in Bookmark Patterns as follows:**

**|split("-",X,"Y")** where - is delimiter, X is segment (integer), Y is optional parameter, text to replace value in segment in the case if delimiter is not found inside value, or there is no X segment inside value.

Example: Customer Doc Number value is 123-456-789, and you want to get "456" value into own cell - you would set <Documents.Customer\_Doc\_Number|split("-",2)> for this cell.

**|count=X** where X is integer (or |count(X) or |count(X,"Y")

This allows you to limit the number of values in the list to X

**|length(X,"Y")** where X is integer, Y is optional parameter, ellipsis

This allows you to limit the number of displayed characters for text variables. XXX should be replaced by the number of characters to display. YYY should be replaced by text which the system automatically will add at the end

of truncated value i.e. if you want to limit the display to 40 characters in the project name, you would use  
<Project\_Name|length(40, "...")>

**|uppercase**

This allows you to set the output in all caps

**|separator="X"**, where X is text/symbol

This allows you to change separator between values in list variables, for example it you want to change "," to "-"

---