

# Using IF statements

05/02/2025 3:08 pm MDT

This article explains how to apply conditions to document auto-numbering, outgoing file name, and compilation pattern formats. The use of parameters is also discussed.

## Using IF statements for Document Auto-Numbering Format

If you have document numbers that differ depending on whether the document is a drawing versus if it is a document adding an IF statement teaches DocBoss how to build each number.

Example:

If Drawing: 4530KR-ISO-0001-**001**

If Document: 4530KR-GEN-0001

- Enable a custom doc code/card field (custfield1 in example below) for "Is Drawing" as text
- Create the following IF statement in the Doc No. Convention

```
IF("<IsDrawing>="Y","<PONumber>-<DocCode>-<Sequence>-<Sheet>","<PONumber>-<DocCode>-<Sequence>")
```

This tells DocBoss that if "Is Drawing" is "Yes" to add the sheet number, otherwise, leave the sheet number off the document number.

## Using IF statements for Outgoing File Name formatting

Like auto-numbering, a similar IF statement can be applied to Outgoing File Name formats.

If your documents need to have the Title added to the outgoing file name but drawings do not, the following IF statement can be used in the File Name Format

```
IF("<IsDrawing>="Y","<DocNoAcme>_Rev<RevAcme>","<DocNoAcme>_Rev<RevAcme>_<Title>")
```

Example:

If Drawing: 4530KR-ISO-0001-001\_Rev0.pdf

If Document: 4530KR-GEN-0001\_Rev0\_**General Document**.pdf

IF statements are also useful for other situations, such as if there is no supplier document number, include the customer's document number.

```
IF("<DocNoWolfe>="empty","<DocNoAcme>","<DocNoWolfe>")
```

Or, if there is no internal doc code, use customer doc code.

IF("<DocCodeInternal>""="empty", "<DocCode>", "<DocCodeInternal>")

## Using IF statements for Header/Footer Patterns

If statements can be used for header/footer patterns to draw different values for different conditions (e.g., cards with A01 and A02 doc code require a different heading).

For instructions on using if statements in header/footer patterns, see this article: [IF Statement for Header/Footer](#).

## Using IF Statements for Compilation Patterns

IF statements can be used in the following compilation patterns:

Bookmark pattern for cover page entry

Bookmark pattern for TOC entry

TOC/Bookmark pattern for cards

Example

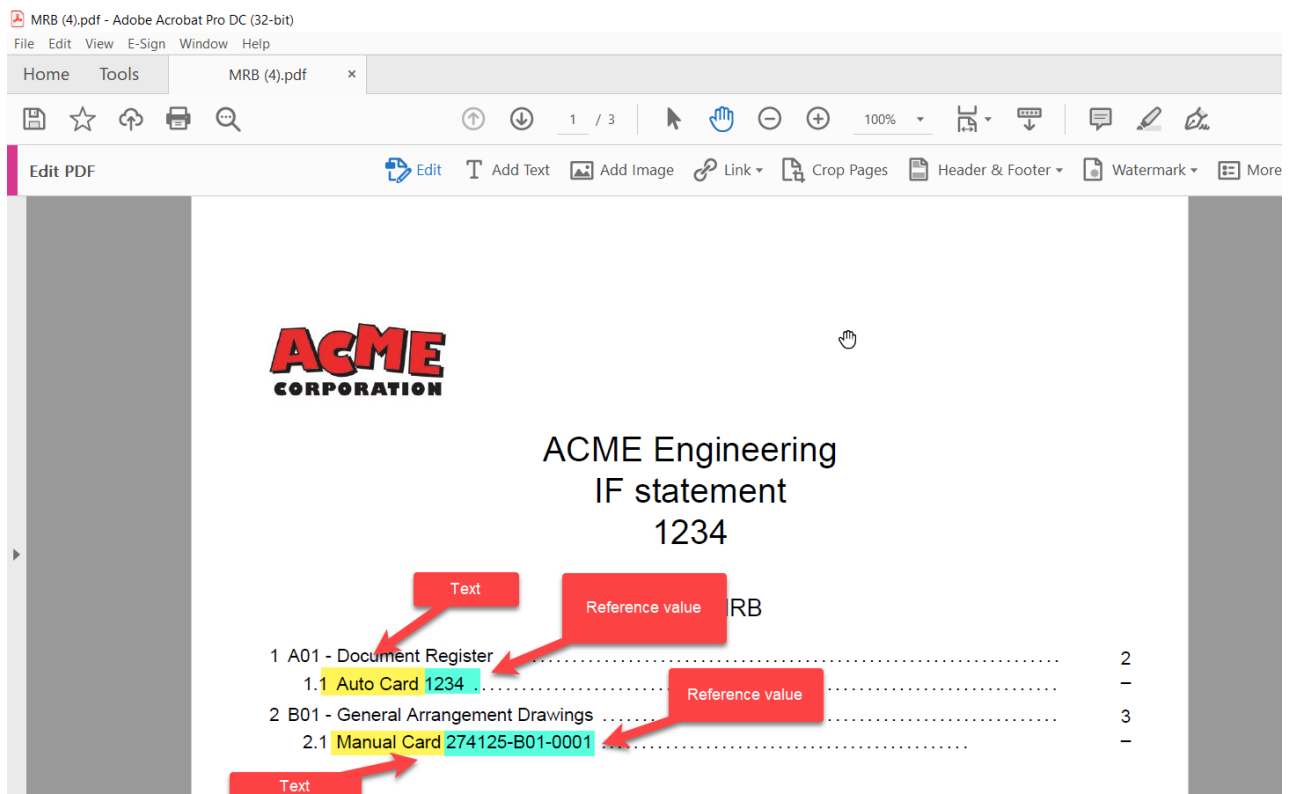
IF("<CardType>""="manual", "Manual Card card <Reference>", "Auto Card <Reference>")

If the card type is manual, display text Manual Card with reference variable, if not, display Auto Card with reference variable

Some examples are below:

| 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")"="" toc","auto="" 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 <="" td="" 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; secti"/>   |                          |                                |
| <input type="checkbox"/> Suppress cards in TOC   |  |                          |                                |

Result:



Another example: If card type is manual, display "Manual Card" as text and list tags, if auto-generated card, display "Auto Card" as text with reference value.

**Table of Contents and Bookmarks sections options**

Depth of TOC:\*  Depth of Bookmarking: 
  
Depth of TOC numbering:  Start of TOC numbering:\* 
  
Hide first level TOC numbers ☐

[How it works?](#)

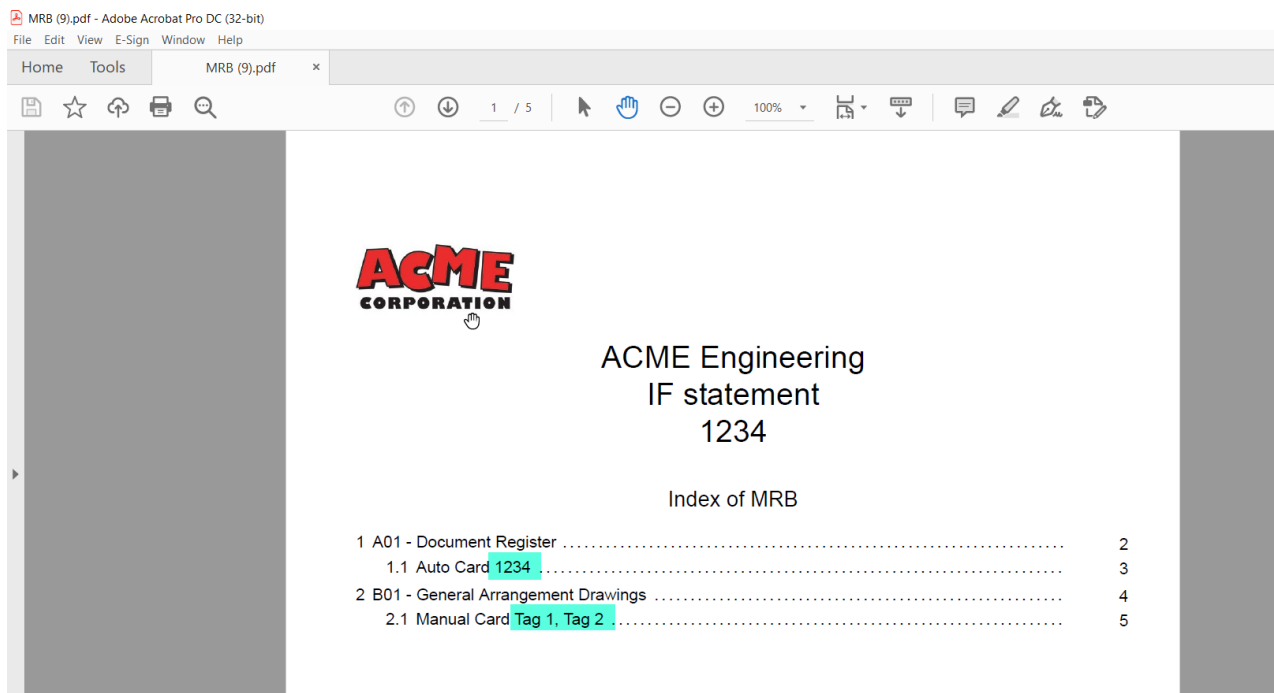
Bookmark pattern for cover page entry:\* 
  
Bookmark pattern for TOC entry:\* 
  
TOC/Bookmark pattern for cards:\*

Suffix when files cannot be embedded in databook PDF:

☐ Append secondary codes (if they exist) to each card with suffix: ☐ If card has secondary codes, replicate the card TOC entry under EVERY secondary code with suffix: 
  
☐ Suppress cards in TOC
  
☒ Show page numbers on the TOC

TOC/Bookmark pattern for doc code:\* 
  
☐ Append Included In Codes to Primary Code (if applicable) using suffix: When Included In Codes are added to the compilation layout, add suffix to TOC/Code: When Not Applicable Codes are added to the compilation layout, add suffix to TOC/Code:

Result:



**Note:** The <CardType> with values "manual" / "auto" is not presented as an option in the pattern variable list and must be typed in manually.

## Additional options: parameters

Parameters can be used to sort/modify outputs in specific ways. The options available are as follows:

### 1. |split("-",X,"Y")

- Where "-" is the delimiter (text/symbol), "X" is the segment (e.g., 1, 2, etc.), and "Y" (*optional*) is the value to use if no delimiter exists OR the defined X segment is not found.
- Example (compilation): The customer doc number is 123-456-789, and the middle value ("456" - i.e., segment 2) needs to be separated. To do this, the variable and parameter would be set as follows:  
<Documents.Customer\_Doc\_Number|split("-",2)>.

### 2. |count(X)

- Where "X" is the number of values (e.g., 1, 2, etc.)
- Limits the number of list items by the number defined.
- Example (compilation): If there are 5 list items, only the first X (e.g., 2) are displayed.

### 3. |length(X,"Y")

- Where "X" is the number of characters (e.g., 1, 2, etc.), and "Y" (*optional*) is the text that is displayed at the end of the truncated value.
- Limits the number of displayed characters for text variables.
- Example (compilation): The project name must be a maximum of 40 characters. If there are more than 40 characters in the project name, the text "..." should be displayed at the end. To do this, the variable and parameter would be set as follows: <Project\_Name|length(40, "...")>

### 4. |uppercase

- Modifies text output to display in capital letters
- Example: "Demo value" → "DEMO VALUE"

5. |separator="X",

- Where "X" is the value (text/symbol) that separates list items.
- Updates the pre-existing separator between list items
- Example: Delimiter is a comma (,) but is changed to a semi-colon (;)



Parameters are available for use in the outgoing file naming pattern(s) and/or the compilation TOC/Bookmark patterns, but are unavailable for use in the document auto-numbering pattern.

---