OH, NO! MY DATES IMPORTED AS NUMBERS!

You just imported a 4,000 page PDF only to see that you didn't set up the template correctly.

In this example, the dates in the original PDF were formatted in the form of MMDDYYYY, but had no punctuation marks such as slashes or dashes to separate the month, the day, and the year.

DATE	ITEM
01052013	TRUMPETS
04232013	DRUMS
09142013	VIOLINS
10262013	TUBAS
12072013	CLARINETS

When imported into IDEA, the date field was imported as a numeric field.

DATE	ITEM
1052013	TRUMPETS
4232013	DRUMS
9142013	VIOLINS
10262013	TUBAS
12072013	CLARINETS

The problem with these date fields morphing into number fields is that the leading zero was lost.   This set the dates into two different formats:

- MDDYYYY for dates between January 1 and September 30
- MMDDYYYY for dates between October 1 and December 31

Because the dates are in two different formats, simply appending a virtual date field and using the @NTOD function to convert the numbers to dates or using Field Manipulation to change the data type to a date field would not work.

⚠ CAUTION: Please note that this technique works for dates in the following formats:

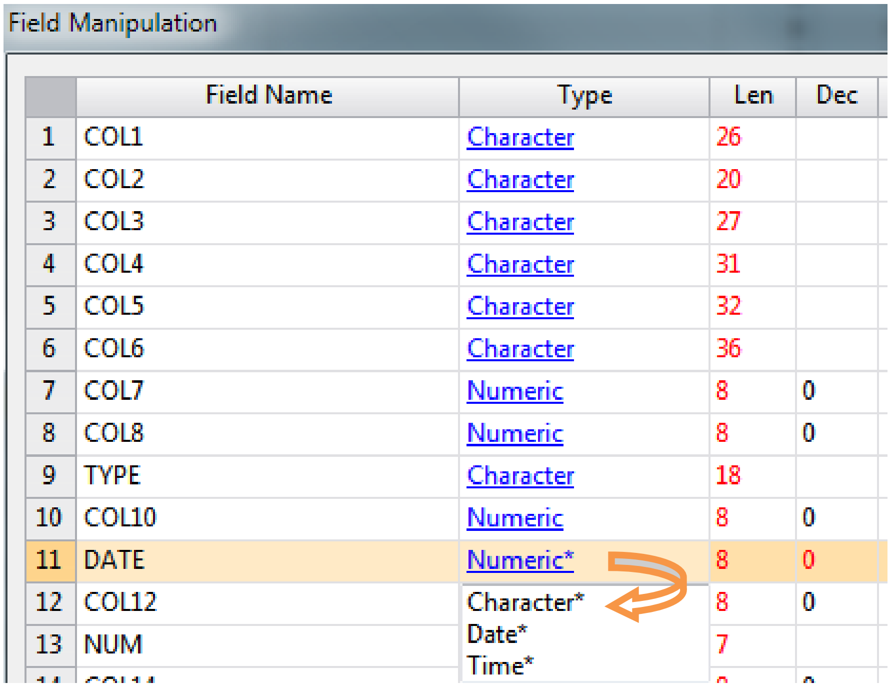
- MMDDYYYY
- DDMMYYYY
- MMDDYY
- DDMMYY

If your dates begin with a 4-character year, you will not need to use this technique. More importantly, if the dates in your original data are formatted some other way than what is listed above, this technique will require modifications.

Here's how to correct this kind of problem if you don't have time to re-import and correct the template.

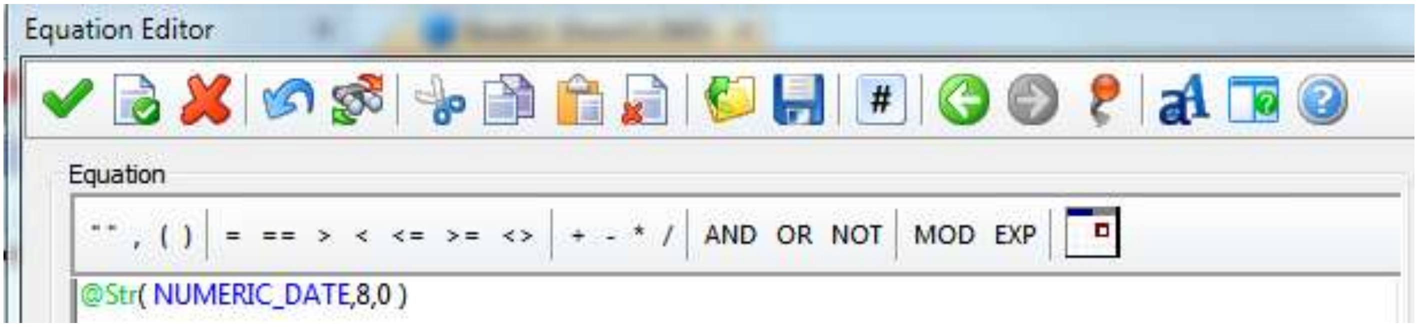
STEP 1

Either use Field Manipulation to change the numeric field into a character field; or append a virtual character field and use it to reformat the numbers into characters using @Str.



	Field Name	Type	Len	Dec
1	COL1	Character	26	
2	COL2	Character	20	
3	COL3	Character	27	
4	COL4	Character	31	
5	COL5	Character	32	
6	COL6	Character	36	
7	COL7	Numeric	8	0
8	COL8	Numeric	8	0
9	TYPE	Character	18	
10	COL10	Numeric	8	0
11	DATE	Numeric*	8	0
12	COL12		8	0
13	NUM		7	

OR

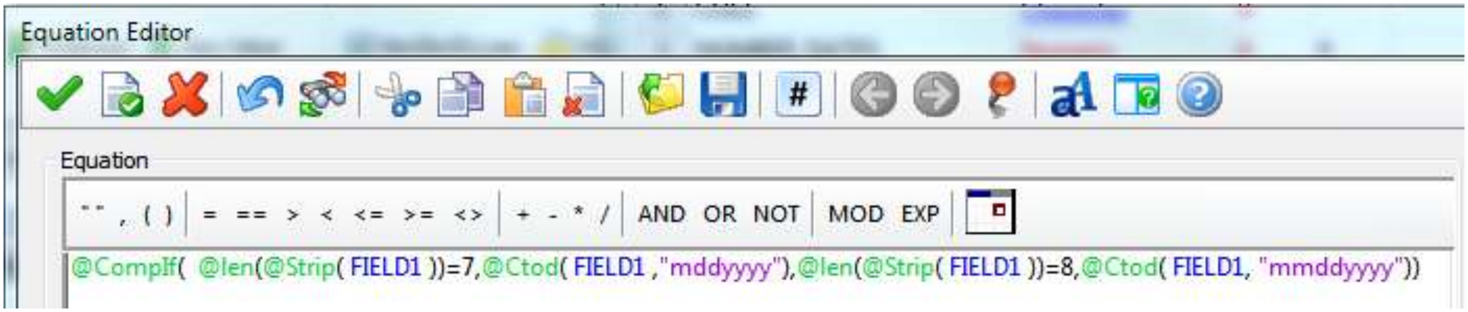


@Str(NUMERIC_DATE,8,0)

1. @Str is the function
   a. NUMERIC_DATE is the name of the field
   b. 8 is the maximum length you want the data to be. We used "8" because that is the same length as an IDEA date field.
   c. 0 is the number of decimals. We used 0 because we don't generally use decimals in dates.

STEP 2

Append a field as a virtual date field and use a nested equation to format the character data into date data.



```
@CompIf( @len(@Strip( FIELD1 ))=7,@Ctod( FIELD1 ,"mddyyyy"),@len(@Strip( FIELD1 ))=8,@Ctod( FIELD1, "mmddyyyy"))
```

The @CompIf function has two condition/results sets:

1. If :
   a. CONDITION: the length of Field 1, after stripping out the spaces, equals 7
      @Len(@Strip(FIELD1))=7,(@Ctod(FIELD1 ,"mddyyyy"))

   b. RESULT: then convert the data to a date using the format of “MDDYYYY”.

      @Ctod(FIELD1 ,"mddyyyy"))

      This takes care of the dates between January 1 and September 30.
      NOTE: if you are confident your data is very clean, you do not need to include @Strip.

2. If:
   a. CONDITION: the length of Field 1, after stripping out the spaces, equals 8

   b. RESULT: then convert the data to a date using the format of “MMDDYYYY”

      @Ctod(FIELD1 ,"mmddyyyy"))

      This takes care of the dates between October 1 and December 31.

   3. If needed, you can add on to the @CompIf statement with a third condition/result set to take into account any dates formatted as MDYYYY: @len(@Strip(FIELD1))=6,@Ctod(FIELD1, "mdyyyy")

Here’s the secret to using @CompIf with date fields – do *not* use the normal practice of ending the @CompIf equation with “1,0)”.

@CompIf will expect a default date-related condition and a specific default date for a result. In the case of dates, it would be better to see a cell with “**Error**” in it rather than a specific default date. The default date would not stand out against the other dates in the field and could easily be misinterpreted as legitimate data.