FormAssembly Functions Update - What To Expect

Download the PDF of this article.

In this Article

Related Articles

Overview

On March 16, 2023, FormAssembly will be updating the functions used within our product in areas such as connectors, emails, thank you pages, and redirects, to more closely match Excel functions. This change will allow our team to work off of a supported library and will provide an updated range of functions.

Most of our FormAssembly functions will remain the same and will not require any changes to be made.

However, a few functions will be updated during this release and the improvements to error handling may affect your forms. We will cover both aspects of this update in this document.

Error Handling Improvements

With this update, our functions will perform more consistently and reliably. Previously, it was possible to use incorrect formatting in some functions and formulas without a visible error. This update will more strictly enforce correct formatting. If you see an error that did not occur prior to this update, you may have incorrect formatting within your function or formula.

For example, the @AND() function previously allowed formulas to reference field aliases within @AND() without assigning a value. However, **this was not the correct format for the @AND() function**. After this update, the correct format is being enforced and a value must be assigned to the field alias to make sure it evaluates properly.

Previous Behavior:

```
@AND(%%tfa_1%%,%%tfa_2%%)
```

Updated Behavior:

```
@AND(%%tfa_1%%="value",%%tfa_2%%="value")
```

If you have any questions, you can reach out to our Support Team for assistance or check out this help doc to view the correct formatting for our functions and formulas.

This update also has the added benefit of improving our connector error handling. **Because of this, you may see more errors appearing in your connector logs.** Previously, issues may not have triggered an error in every circumstance but now they will be noted in the logs along with any successful step logs. If you need assistance reading your logs, you can reach out to our Support Team.

Updates to Functions

The affected functions are:

- The "ERROR" condition in @IF()
- Logical comparison between strings and numbers
- Timezones in @HOUR()
- Strings in @ISODD() and @ISEVEN()

These functions are currently not operating as they should in **specific circumstances**. After the update, they will perform as intended.

Let's go over each of these in more detail.

The ERROR condition in @IF()

Currently, if an error occurs in the condition portion of @IF() then the formula will output the FALSE value. This means that you will never see an error in your formula, and instead will see the FALSE value which can lead to confusion. Now you will see if an error occurs!

Current Behavior:

```
@IF(ERROR_CONDITION, "yes", "no") -> no
```

Updated Behavior:

```
@IF(ERROR_CONDITION, "yes", "no") -> ERROR_CONDITION
```

This change will only affect you if you're **expecting** an error to occur and want the FALSE value to appear when one does. If this is your intention, you can alter your formulas to use @IFERROR() instead.

Example:

```
@IF(IFERROR(ERROR_CONDITION, FALSE()), "yes", "no")
```

Logical comparison between strings and numbers

Currently, an incorrect behavior occurs if you try to compare a number and a string of text. Comparing the two leads to a TRUE value being output, instead of a FALSE value. With this update, this behavior will be fixed.

Current Behavior:

```
@IF( 123 > 'my string', 'true_value', 'false_value') -> true_value
```

Updated Behavior:

```
@IF( 123 > 'my string', 'true_value', 'false_value') ->  false_value
```

This change will only affect you if you're **expecting** a TRUE value when comparing a number to a string of text. If you are not expecting this, no changes will be needed.

Timezones in @HOUR()

Currently, the @HOUR() function returns the value at a different timezone. After this update, it will return the correct value.

Current Behavior:

```
@HOUR('6:30 AM') -> 11
```

Updated Behavior:

```
@HOUR('6:30 AM') -> 6
```

This change will only affect you if you're **expecting** a different timezone to be output. If you are not expecting this, no changes will be needed.

Strings in @ISODD() and @ISEVEN()

Currently, text strings are evaluated as even values in @ISODD() and @ISEVEN(). This behavior will be removed with this update.

Current Behavior:

```
@ISEVEN("string") -> TRUE
```

Updated Behavior:

```
@ISEVEN("string") -> #VALUE!
```

This change will only affect you if you're **expecting** a text string to evaluate as TRUE. If you are not expecting this, no changes will be needed.