[Self-help Document] Parametrization and Expression Language

Last updated by | Jackie Huang | Jan 4, 2022 at 12:24 AM PST

Path: Data Factory/Authoring or Development Issues/Parameterization and Expression Language
ID: 32629484
Description: Default ways in which expressions are being passed in parameters is not working, need more ways to pass parameters

Expressions and Functions

Recommended Steps

- Recently, ADF has added some validations capabilities in ADF portal experience to detect common expression errors

- Please note that even simple typographical errors in expressions may cause errors at run time

- A syntax or validation error reported during runtime can be caused by an expression error in dataset setting and/or activity definition referenced in the pipeline. So it's important to check expressions carefully in *all* relevant entities

- Please be extra careful when writing JSON within expressions, particularly when using arrays types. For example, if you want to use a JSON array, like [1, 2, 3], as the value of an expression in a ForEach activity, you can write expressions "@json('[1,2,3]')" or "@createArray(1,2,3)", but **not** "@[1,2,3]"

- *Only* single quotes can be used within expressions, while double quotes are reserved to mark the ends of the entire expression. Single quotes can be escaped with an additional single quote, however it's may be hard to see depending on font and easy to get wrong. Please use extra caution when leveraging these capabilities. For example, *one* in following expression is within pairs of two single quotes (not a pair of double quotes): "@json('{''one'':1}')".

- Consider testing expressions in small pipelines during development, so you can iterate quickly to find and fix expression-level issues before using the expression in a more complex context. Please consider utilize the "Debug" mode in ADF portal experience. See [Iterative Development and Debugging](#) ⧉

Recommended Documents

Supported by ADF:

- Build dynamic contents with [functions and parameters](#) ⧉

- [Expression Language](#) ⧉

- [System Variables](#) ⧉

- [Functions](#) ⧉ including:

- [String](#) ↗ functions only apply to strings
- [Collection](#) ↗ functions operate over collections, such as arrays, strings, and in some cases dictionaries
- [Logical](#) ↗ functions are useful inside conditions
- [Conversion](#) ↗ functions convert between native types
- [Math](#) ↗ functions are used for either *integer* or *float*
- [Date](#) ↗ functions operate over timestamps

How good have you found this content?

😊 🙁