Matches Operator

Description

Compares a string value to a pre-defined pattern. Evaluates to 1 (true) or 0 (false).

Syntax

string **Match | Matches** *pattern*

Remarks

Match is a comparison operator. It compares the string value of string to the defined pattern specified by pattern. If the patterns match, the evaluation is 1 (true); if they do not match, the evaluation will be 0 (false). The values that are compared are the number and types of characters.

For example, if the string value of InputNumber in the expression:

```
InputNumber Matches "5N"
```

consists of five numbers, it will evaluate to 1 (true).

Parameters

The Matches operator has the following parameters.

Parameter	Description
Pattern	Must be enclosed in quotation marks, or it must be a variable identifier. Pattern specifications are evaluated from left to right. The maximum length permitted for the string is 254 characters; the pattern specification expression may not exceed 254 characters.

Pattern	Meaning
NA	An integer followed by A tests for that number of alphabetic characters.
NN	An integer followed by N tests for that number of numeric characters.
NX	An integer followed by X tests for that number of any characters.
NZ	An integer followed by Z tests for up to that number of characters.
"string"	A literal string enclosed in quotes tests for that exact text.

If n is zero, the match is interpreted to mean any number of that pattern. For example, the pattern 0A indicates any number of alphabetic characters.

Example

```
phone = "(203)555-1212"
flag = phone Matches "'('3N')'3N'-'4N"
/* Expression will evaluate to 0 (false) because the phone number has no spaces around the dash or the
parentheses. */
CN = "B1234WA"
flag = CN Matches "1A4N2A"
* Expression will evaluate to 1 (true).
```