

2.1.11 Operations

[Return to DIDO CLI Background](#)

CamelCase

[Return to Top](#)

CamelCase strips leading and trailing blanks, converts multiple whitespace characters to a single space, converts the text to LowerCase and then Uppercases the first letter of each word.

```
CamelCase ::= ( IN TextValue ) RETURN TextValue;
```

Example

```
myTextString = "  This is  a  Test";  
echo CamelCase(myTextString);
```

Results In

```
"ThisIsATest"
```

camelCase

[Return to Top](#)

camelCase performs the **CamelCase** operation and then lowercases the first letter.

```
camelCase ::= ( IN TextValue ) RETURN TextValue;
```

Example

```
myTextString = "  This is  a  Test";  
echo camelCase(myTextString);
```

Results In

```
"thisIsATest"
```

CompressWhiteSpace

[Return to Top](#)

CompressWhiteSpace converts multiple spaces within the text to a single space. It does not remove leading or trailing spaces.

```
CompressWhiteSpace ::= ( IN TextValue ) RETURN TextValue;
```

Example

```
myTextString = "  This is   a  Test";  
echo CompressWhiteSpace(myTextString);
```

Results In

```
"This is a Test"
```

CurrentDate

[Return to Top](#)

CurrentDate returns the current date from the system clock using [ISO 8601-1:2019 Date and time -- Representations for information interchange -- Part 1: Basic rules](#). See [DateTimeValue](#)

```
CurrentDate ::= ( EMPTY_PARAM ) RETURN DateValue;
```

Example

```
myTextString = "  This is   a  Test";  
echo CurrentDate();
```

Results In

This would be the result for 3:30 pm on September 7, 2019 in NY Eastern Daylight Savings Time (EDT)

```
"2019-09-07T15:50-04:00"
```

CurrentTime

[Return to Top](#)

CurrentTime returns the current time from the system clock. See [TimestampValue](#).

```
CurrentTime ::= ( EMPTY_PARAM ) RETURN TimestampValue;
```

This would be the result for the current time in milliseconds¹⁾

```
"1617929032476"
```

IsNull

[Return to Top](#)

IsNull tests if the parameter passed is has no value (i.e., 'NULL').

```
IsNull ::= ( DidoObject ) RETURN BooleanValue;
```

Example

```
myTextString = NULL;  
echo myTextString is Null IsNull(myTextString);  
myTextString = " This is a Test";  
echo myTextString is Null IsNull(myTextString);  
myTextString = "";  
echo myTextString is Null IsNull(myTextString);
```

Results In

This would be the result for 3:30 pm on September 7, 2019 in NY Eastern Daylight Savings Time (EDT)

```
myTextString is Null TRUE
```

```
myTextString is Null FALSE  
myTextString is Null FALSE
```

Length

[Return to Top](#)

Length returns the number of characters in the TextValue. **Note:** if the TextValue is NULL, **Lenght** returns a minus 1 (i.e., -1).

```
Length ::= ( IN Text ) RETURN IntegerValue;
```

Example

```
myTextString = NULL;  
echo length(myTextString);  
myTextString = "";  
echo length(myTextString);  
myTextString = "This is a Test";  
echo length(myTextString);
```

Results In

```
-1  
0  
14
```

LowerCase

[Return to Top](#)

LowerCase converts the text to LowerCase, it does not modify the whitespace.

```
LowerCase ::= ( IN Text ) RETURN TextValue;
```

Example

```
myTextString = "  This is    a  Test";
```

```
echo LowerCase(myTextString);
```

Results In

```
"  this is  a  test"
```

PadC

[Return to Top](#)

PadC centers the TextValue within the given padding length. If the length of the Text string is longer than the padding length, the left and right characters are trimmed equally until it fits.

```
PadC ::= ( IN TextValue, IN IntegerValue ) RETURN TextValue;
```

Example

```
myTextString = "Test";  
echo      |...:.....|...: RULER  
echo PadC(myTextString,15);
```

Results In

```
|...:.....|...: RULER  
"      Test      "
```

NVL

[Return to Top](#)

The **NVL** operations evaluates the two parameters passed to it. If the first value is NOT NULL, then the first value is returned. If the first value is NULL, then the second value is returned.

Example

```
value_1 : IntegerValue := NULL;  
value_2 : IntegerValue := "1234";  
value_3 : IntegerValue := NULL;
```

```
echo "1:2 -> " + NVL ( value_1, value_2);  
echo "2:3 -> " + NVL ( value_2, value_3);
```

Results In

```
1:2 -> 1234  
2:3 -> 1234
```

PadL

[Return to Top](#)

PadL pads the TextValue to the left until the padding length is met. If the padding length is smaller than the length of the TextValue, characters are trimmed on the left until it fits.

```
PadL ::= ( IN TextValue, IN IntegerValue ) RETURN TextValue;
```

Example

```
myTextString = "Test";  
echo      |...:.....|...: RULER  
echo PadL(myTextString,15);
```

Results In

```
|...:.....|...: RULER  
"          Test"
```

PadR

[Return to Top](#)

PadR pads the TextValue to the right until the padding length is met. If the padding length is smaller than the length of the TextValue, characters are trimmed on the right until it fits.

```
PadR ::= ( IN TextValue, IN IntegerValue ) RETURN TextValue;
```

Example

```
myTextString = "Test";  
echo      |...:.....|...: RULER  
echo PadL(myTextString,15);
```

Results In

```
|...:.....|...: RULER  
"Test      "
```

TitleCase

[Return to Top](#)

TitleCase converts the text to LowerCase, it does not modify the whitespace.

```
TitleCase ::= ( IN TextValue ) RETURN TextValue;
```

Example

```
myTextString = "  This is  a  TEST";  
echo TitleCase(myTextString);
```

Results In

```
"  This Is  A  Test"
```

Trim

[Return to Top](#)

Trim trims all leading and trailing [whitespace](#) from the line.

```
Trim ::= Trim( IN TextValue ) RETURN TextValue;
```

Example

```
myTextString = "  This is    a  TEST  ";  
echo Trim(myTextString);
```

Results In

```
"This is    a  TEST"
```

TrimL

[Return to Top](#)

Trims all leading (i.e., left side) [whitespace](#) from the line.

[Return to Top](#)

TrimL trims all leading [whitespace](#) from the line.

```
TrimL ::= TrimL( IN TextValue ) RETURN TextValue;
```

Example

```
myTextString = "  This is    a  TEST  ";  
echo Trim(myTextString);
```

Results In

```
"This is    a  TEST  "
```

TrimR

[Return to Top](#)

TrimR trims all trailing [whitespace](#) from the line.

```
TrimR ::= TrimR( IN TextValue ) RETURN TextValue;
```

UpperCase

[Return to Top](#)

UpperCase converts all the ASCII characters to their Uppercase equivalents.

```
UpperCase ::= ( IN TextValue ) RETURN TextValue;
```

Example

```
myTextString = "This is a Test";  
echo LowerCase(myTextString);
```

Results In

```
"THIS IS A TEST"
```

WhiteSpaceToUnderscore

[Return to Top](#)

WhiteSpaceToUnderscore converts all whitespace characters to underscores ('_').

```
WhiteSpaceToUnderscore ::= ( IN TextValue ) RETURN TextValue;
```

Example

```
myTextString = " This is  a TEST ";  
echo WhiteSpaceToUnderscore(myTextString);
```

Results In

```
"_This_is__a_TEST_"
```

¹⁾

CurrentMiillis, Accessed 8 April 2021, <https://currentmillis.com/>

From:
<https://www.omgwiki.org/dido/> - **DIDO Wiki**

Permanent link:
https://www.omgwiki.org/dido/doku.php?id=dido:public:s_cli:05_contents:01_prt:03_langconst:04_operations:start&rev=1619917358

Last update: **2021/05/01 21:02**

