

MY Proposal for MY-0015 / Response 30

Option 1: Optional third parameter for legacy support

Malaysia's proposal for the CEILING function considers three aspects which improves on Ecma-376, Ecma-January Dispositions and Ecma-BRM Proposal:

- 1) Mathematical accuracy: Positive values rounds up away from zero, and Negative values, by default, rounds up towards zero.
- 2) Ease of use: The optional (default = 1) second significance parameter will be evaluated on its absolute value, removing the limitation of requiring the signs to be the same. Previously differing signs will cause an unnecessary #NUM error.
- 3) Legacy support: The optional (default = false) third parameter for legacy support toggles the behaviour of which direction negative values are evaluated.

The following changes are proposed:

3.17.7.33 CEILING

Syntax:

CEILING (*x* [*significance*] [*legacy-support*])

Description: Computes a value that is *x* rounded up.

Arguments:

Name	Type	Description
<i>x</i>	number	The value to be rounded
<i>significance</i>	number	<p>The optional multiple to which <i>x</i> is to be rounded.</p> <p>If not specified, the default value for significance is 1</p> <p>[Note: the absolute value of the multiple is used, so the CEILING function will return a CEILING irrespective of the signs of <i>x</i> and the significance.]</p> <p>If <i>x</i> is negative, and significance is negative, then the value is rounded down (away from zero). If <i>x</i> is negative, and significance is positive, then the value is rounded up, towards zero.</p> <p>If <i>x</i> is positive, and significance is negative, then the value is #NUM!</p>

		If x and/or significance is zero, then the result is zero.
legacy-support	number	The optional flag to indicate support for legacy CEILING calculations. If legacy-support is 0 (default), then for negative values of x, x is rounded up, towards zero. If legacy-support is 1, then for negative values of x, x is rounded down, away from zero to reproduce legacy behaviour of this function.

Return Type and Value: number The rounded-up value of x.

~~However, if x and significance have different signs, #NUM! is returned.~~

[Example:

CEILING(2.5) rounds 2.5 up to the nearest default multiple of 1; that is, to 3

CEILING(-2.5) rounds -2.5 up to the nearest default multiple of 1; that is, to -2

CEILING(2.5, 1) rounds 2.5 up to [the](#) nearest multiple of 1; that is, to 3

[CEILING\(-2.5, 2 \) rounds -2.5 up to the nearest multiple of 2; that is, to -2](#)

CEILING(-2.5, -2) rounds -2.5 ~~up-down~~[up](#) to [the](#) nearest multiple of -2; that is, to -2

CEILING(1.5, 0.1) rounds 1.5 up to the nearest multiple of 0.1; that is, to 1.5

CEILING(0.234, 0.01) rounds 0.234 up to the nearest multiple of 0.01; that is, to 0.24

[CEILING\(-0.234, 0.01 \) rounds -0.234 up to the nearest multiple of 0.01; that is, to -0.23](#)

[CEILING\(-0.234, 0.01, 1 \) rounds -0.234 down to the nearest multiple of 0.01; that is, to -0.24](#)

[CEILING\(-2.5, 2, 1 \) rounds -2.5 down to the nearest multiple of 2; that is, to -4](#)

end example]

[\[Migration Note: Applications importing legacy or transition files utilizing CEILING functions with negative significance numbers, must append the legacy support flag with value 1\]](#)

Option 2: Using Reserved Prefixes with '.'

Another way to prevent legacy issues is to use a reserved prefixes which allow multiple algorithms to implement a function of the same name.

- Current functions remain unchanged
- All current functions are also available in the ECMA prefix
- All current functions are also available with the ISO prefix but with mathematically correct behaviour (i.e ISO.CEILING() is correct)
- Provides a basis for future migration of OpenFormula namespaces or the use of OpenFormula functions

Note: ISO.CEILING() is the preferred function for new formulas

With respect to the use of `.` for the delimiter, S3.27.5.1 Limits example indicates that function names override other kinds of names, in particular cell references. However, a simple implementation will have a 1 token lookahead, and which makes the use of `.` ambiguous here: consequently `.` is used. Furthermore, it seems that Open Formula is syntactically incompatible with the Office formula language.

Instruction 1: Add new section

Syntax:

ISO.CEILING(x [, sig])

Arguments:

Name	Type	Description
x	number	The value to be rounded
significance	number	The optional multiple to which x is to be rounded. If not specified, the default value for significance is 1 [Note: the absolute value of the multiple is used, so the CEILING function will return the mathematical ceiling irrespective of the signs of x and the significance.] If x and/or significance is zero, then the result is zero.

Return Type and Value: number The rounded-up value of x.

[Example:

ISO.CEILING(4.3) = 5
 ISO.CEILING(-4.3) = -4
 ISO.CEILING(4.3, 2) = 6
 ISO.CEILING(4.3, -2) = 6
 ISO.CEILING(-4.3, 2) = -4
 ISO.CEILING(-4.3, -2) = -4
 end example.]

Syntax:

ECMA.CEILING(x, sig)

Arguments:

Name	Type	Description
<i>x</i>	number	The value to be rounded
<i>significance</i>	number	<p>The multiple to which <i>x</i> is to be rounded.</p> <p><u>If <i>x</i> is negative, and <i>significance</i> is negative, then the value is rounded down (away from zero). If <i>x</i> is negative, and <i>significance</i> is positive, then the value is rounded up, towards zero.</u></p> <p><u>If <i>x</i> is positive, and <i>significance</i> is negative, then the value is #NUM!</u></p> <p><u>If <i>x</i> and/or <i>significance</i> is zero, then the result is zero.</u></p>

Return Type and Value: number The rounded-up value of *x*.

[Example:

ECMA.CEILING(4.3, 2) = 6
 ECMA.CEILING(4.3, -2) = #NUM!
 ECMA.CEILING(-4.3, 2) = -4
 ECMA.CEILING(-4.3, -2) = -6
 end example]

This will change the syntax on how ISO 29500 saves its formulas with the necessary prefixes. For legacy and transitional files, the default namespace is ECMA, but new files will default to the ISO namespace for formulas.

Part 4, §3.17.2.4, page 2,517, lines 2 27:

function-name=

prefixed-function-name | predefined-function-name | user-defined-function-name ;

prefixed-function-name=

"ISO.", predefined-function-name |

"ECMA.", predefined-function-name

Instruction 3

Add paragraph to Part 4, §3.17.7, at end.

All predefined functions may be used with their simple name, or with the prefix `ECMA.` or with the prefix `ISO.` with the following exception: the predefined function named `CEILING()` in this standard may only be used with the prefix `ECMA.` only. The predefined function named `ISO.CEILING()` is specified in this standard.

Option 3: Using Reserved Prefixes with ':'

Another way to prevent legacy issues is to use a reserved prefixes which allow multiple algorithms to implement a function of the same name.

- Current functions remain unchanged
- All current functions are also available in the ECMA prefix
- All current functions are also available with the ISO prefix but with mathematically correct behaviour (i.e ISO:CEILING() is correct)
- Provides a basis for future migration of OpenFormula namespaces or the use of OpenFormula functions

Note: ISO:CEILING() is the preferred function for new formulas

With respect to the use of `:` for the delimiter, S3.27.5.1 Limits example indicates that function names override other kinds of names, in particular cell references.

Instruction 1: Add new section

Syntax:

ISO:CEILING(x [, sig])

Arguments:

Name	Type	Description
x	number	The value to be rounded
significance	number	The optional multiple to which x is to be rounded. If not specified, the default value for significance is 1 [Note: the absolute value of the multiple is used, so the CEILING function will return the mathematical ceiling irrespective of the signs of x and the significance.] If x and/or significance is zero, then the result is zero.

Return Type and Value: number The rounded-up value of x.

[Example:

ISO:CEILING(4.3) = 5

ISO:CEILING(-4.3) = -4

ISO:CEILING(4.3, 2) = 6
 ISO:CEILING(4.3, -2) = 6
 ISO:CEILING(-4.3, 2) = -4
 ISO:CEILING(-4.3, -2) = -4
 end example.]

Syntax:

ECMA:CEILING(x, sig)

Arguments:

Name	Type	Description
<i>x</i>	number	The value to be rounded
<i>significance</i>	number	<p>The multiple to which <i>x</i> is to be rounded.</p> <p><u>If <i>x</i> is negative, and <i>significance</i> is negative, then the value is rounded down (away from zero). If <i>x</i> is negative, and <i>significance</i> is positive, then the value is rounded up, towards zero.</u></p> <p><u>If <i>x</i> is positive, and <i>significance</i> is negative, then the value is #NUM!</u></p> <p><u>If <i>x</i> and/or <i>significance</i> is zero, then the result is zero.</u></p>

Return Type and Value: number The rounded-up value of *x*.

[*Example:*

ECMA:CEILING(4.3, 2) = 6
 ECMA:CEILING(4.3, -2) = #NUM!
 ECMA:CEILING(-4.3, 2) = -4
 ECMA:CEILING(-4.3, -2) = -6
 end example]

This will change the syntax on how ISO 29500 saves its formulas with the necessary prefixes. For legacy and transitional files, the default namespace is ECMA, but new files will default to the ISO namespace for formulas.

Part 4, §3.17.2.4, page 2,517, lines 2 27:

function-name=

prefixed-function-name | predefined-function-name | user-defined-function-name ;

prefixed-function-name=

"ISO:", predefined-function-name |

"ECMA:", predefined-function-name

Instruction 3

Add paragraph to Part 4, §3.17.7, at end.

All predefined functions may be used with their simple name, or with the prefix `ECMA:` or with the prefix `ISO:` with the following exception: the predefined function named `CEILING()` in this standard may only be used with the prefix `ECMA:` only. The predefined function named `ISO:CEILING()` is specified in this standard.

Ballot Decisions

Adopt Option 1: Usage of CEILING with optional third parameter for legacy support

Adopt Option 2: Usage of ISO.CEILING() and ECMA.CEILING() prefix with '.' token

Adopt Option 3: Usage of ISO:CEILING() and ECMA:CEILING() prefix with ':' token