



This PDF is generated from authoritative online content, and is provided for convenience only. This PDF cannot be used for legal purposes. For authoritative understanding of what is and is not supported, always use the online content. To copy code samples, always use the online content.

eServices Field Codes Reference Manual

String Functions

5/4/2025

String Functions

The eServices Field Codes include the following string functions:

[+] Find

Find

Description

Finds a substring within a string. Returns the 0- based character position of the found substring. Returns -1 if the substring is not found.

Syntax

Find(*SearchIn*, *SearchFor*)

Find String

Argument	Description
SearchIn	The string to search in
SearchFor	The string to search for

Remarks

Examples of Find String

Example	Result
<\$Find("Hello, World!", "H")\$>	0
<\$Find("Hello, World!", "lo")\$>	3
<\$Find("Hello, World!", "Qbert")\$>	-1

[+] Left

Left

Description

Returns a string containing a specified number of characters from the left side of a specified string.

Syntax

Left(*String*, *Number*)

Left String

Argument	Description
String	The string from which the leftmost characters are returned.
Number	The number of characters to return. If 0, an empty string ("") is returned. If greater than the length of String, then the entire string is returned.

Remarks

Examples of Left String

Example	Result
<\$Left("Hello, World!", 5)\$>	"Hello"
<\$Left("Hello, World!", 0)\$>	""
<\$Left("Hello, World!", 25)\$>	"Hello, World!"

[+] Length

Length

Description

Returns the length of a string.

Syntax

Length(*String*)

Remarks

Example of Length String

Example	Result
<\$Length("Hello")\$>	5

[+] Mid

Mid

Description

Returns a specified substring of a string.

Syntax

Mid(*String*, *Start*, *Length*)

Mid String

Argument	Description
String	The string from which the substring is returned.
Start	The 0 - based character position at which the substring begins. If <i>Start</i> is greater than the length of <i>String</i> , then an empty string ("") is returned.
Length	The number of characters to return. If <i>Length</i> is 0, then an empty string ("") is returned. If <i>Length</i> is greater than the portion of <i>String</i> after <i>Start</i> , then all the characters after <i>Start</i> are returned.

Remarks

Examples of Mid String

Example	Result
<\$Mid("Hello, World!", 2, 3)\$>	"llo"
<\$Mid("Hello, World!", 25, 5)\$>	""
<\$Mid("Hello, World!", 7, 25)\$>	"World!"

[+] Replace

Replace

Description

Returns a string in which all instances of a specified substring have been replaced with another string.

Syntax

Replace(*String*, *Find*, *ReplaceWith*)

Replace String

Argument	Description
String	The string containing the substring to replace
Find	The substring to search for
ReplaceWith	The replacement string

Remarks

Examples of Replace String

Example	Result
<\$Replace("Hello", "l", "*")\$>	"He**o"
<\$Mid("Hello", "j", "*")\$>	"Hello"
<\$Mid("Hello", "Hello", "")\$>	" "

[+] Right

Right

Description

Returns a string containing a specified number of characters from the right side of a specified string.

Syntax

Right(*String*, *Number*)

Right String

Argument	Description
String	The string from which the rightmost characters are returned.

Argument	Description
Number	The number of characters to return. If 0, an empty string ("") is returned. If greater than the length of <i>String</i> , then the entire string is returned.

Remarks

Examples of Right String

Example	Result
<\$Right("Hello, World!", 5)\$>	"orld!"
<\$Right("Hello, World!", 0)\$>	""
<\$Right("Hello, World!", 25)\$>	"Hello, World!"

[+] To Lower

ToLower

Description

Returns a string that has been converted to lowercase.

Syntax

ToLower(*String*)

Remarks

Example of ToLower String

Example	Result
<\$ToLower("Hello, World!")\$>	"hello, world!"

[+] ToUpper

ToUpper

Description

Returns a string that has been converted to uppercase.

Syntax

ToUpper(*String*)

Remarks

Example of ToUpper String

Example	Result
<\$ToUpper("Hello, World!")\$>	"HELLO, WORLD! "

[+] Trim

Trim

Description

Returns a copy of a specified string without specified leading or trailing characters.

SyntaxTrim(*String*, [*CharSet*])

Trim String

Argument	Description
String	The string from which to trim
CharSet	Optional. The characters to trim. If omitted, then white space (" \t\r\n") is trimmed.

Remarks

Examples of Trim String

Example	Result
<\$Trim(" Howdie ")\$>	"Howdie"
<\$Trim("Howdie", "Howd")\$>	"ie"
<\$Trim("Howdy", "y")\$>	"Howd"

[+] TrimLeft

TrimLeft

Description

The same as Trim, except it trims only leading characters.

Syntax

TrimLeft(*String*, [*CharSet*])

[+] TrimRight

TrimRight

Description

The same as Trim, except it trims only trailing characters.

Syntax

TrimRight(*String*, [*CharSet*])

[+] FullName

Wrap

Description

Returns a string that has been word-wrapped to a specified line length.

Syntax

Trim(*String*, *LineLength*, [*LinePrefix*, [*Eol*]])

Wrap String

Argument	Description
String	The string to wrap.
LineLength	The maximum length, in characters, of any line, including LinePrefix (if specified), but not Eol.
LinePrefix	Optional. A string to prefix to each line. Often used to "quote" e-mails being replied to. If omitted, lines are not prefixed.
Eol	Optional. A string to use as a line terminator. If omitted, lines are terminated with "\r\n" as usual.

Remarks

Example:

```
<$Wrap(> "Once upon a midnight dreary",<
11,<
">",<
"*\r\n")$>
```


Result:

```
>Once upon*<  
>a midnight*<  
>dreary*
```