String Functions (Visual Basic)

Visual Studio 2015

Updated: July 20, 2015

For the latest documentation on Visual Studio 2017, see Visual Studio 2017 Documentation.

The following table lists the functions that Visual Basic provides to search and manipulate strings.

.NET Framework method	Description
Asc, AscW	Returns an `Integer` value representing the character code corresponding to a character.
Chr, ChrW	Returns the character associated with the specified character code.
Filter	Returns a zero-based array containing a subset of a `String` array based on specified filter criteria.
Format	Returns a string formatted according to instructions contained in a format `String` expression.
FormatCurrency	Returns an expression formatted as a currency value using the currency symbol defined in the system control panel.
FormatDateTime	Returns a string expression representing a date/time value.
FormatNumber	Returns an expression formatted as a number.
FormatPercent	Returns an expression formatted as a percentage (that is, multiplied by 100) with a trailing % character.
InStr	Returns an integer specifying the start position of the first occurrence of one string within another.
InStrRev	Returns the position of the first occurrence of one string within another, starting from the right side of the string.
Join	Returns a string created by joining a number of substrings contained in an array.
LCase	Returns a string or character converted to lowercase.
Left	Returns a string containing a specified number of characters from the left side of a string.
Len	Returns an integer that contains the number of characters in a string.

.NET Framework method	Description
LSet	Returns a left-aligned string containing the specified string adjusted to the specified length.
LTrim	Returns a string containing a copy of a specified string with no leading spaces.
Mid	Returns a string containing a specified number of characters from a string.
Replace	Returns a string in which a specified substring has been replaced with another substring a specified number of times.
Right	Returns a string containing a specified number of characters from the right side of a string.
RSet	Returns a right-aligned string containing the specified string adjusted to the specified length.
RTrim	Returns a string containing a copy of a specified string with no trailing spaces.
Space	Returns a string consisting of the specified number of spaces.
Split	Returns a zero-based, one-dimensional array containing a specified number of substrings.
StrComp	Returns -1, 0, or 1, based on the result of a string comparison.
StrConv	Returns a string converted as specified.
StrDup	Returns a string or object consisting of the specified character repeated the specified number of times.
StrReverse	Returns a string in which the character order of a specified string is reversed.
Trim	Returns a string containing a copy of a specified string with no leading or trailing spaces.
UCase	Returns a string or character containing the specified string converted to uppercase.

You can use the Option Compare statement to set whether strings are compared using a case-insensitive text sort order determined by your system's locale (`Text`) or by the internal binary representations of the characters (`Binary`). The default text comparison method is `Binary`.

Example

This example uses the `UCase` function to return an uppercase version of a string.

VB

```
        ' String to convert.
```

```
Dim LowerCase As String = "Hello World 1234"
' Returns "HELLO WORLD 1234".
Dim UpperCase As String = UCase(LowerCase)
```

Example

This example uses the `LTrim` function to strip leading spaces and the `RTrim` function to strip trailing spaces from a string variable. It uses the `Trim` function to strip both types of spaces.

VB

```
' Initializes string.
Dim TestString As String = "  <-Trim->  "
Dim TrimString As String
' Returns "<-Trim->  ".
TrimString = LTrim(TestString)
' Returns "  <-Trim->".
TrimString = RTrim(TestString)
' Returns "<-Trim->".
TrimString = LTrim(RTrim(TestString))
' Using the Trim function alone achieves the same result.
' Returns "<-Trim->".
TrimString = Trim(TestString)
```

Example

This example uses the `Mid` function to return a specified number of characters from a string.

VB

```
' Creates text string.
Dim TestString As String = "Mid Function Demo"
' Returns "Mid".
Dim FirstWord As String = Mid(TestString, 1, 3)
' Returns "Demo".
Dim LastWord As String = Mid(TestString, 14, 4)
' Returns "Function Demo".
Dim MidWords As String = Mid(TestString, 5)
```

Example

This example uses `Len` to return the number of characters in a string.

VB

```
' Initializes variable.
Dim TestString As String = "Hello World"
' Returns 11.
Dim TestLen As Integer = Len(TestString)
```

Example

This example uses the `InStr` function to return the position of the first occurrence of one string within another.

VB

```
' String to search in.
Dim SearchString As String = "XXpXXpXXPXXP"
' Search for "P".
Dim SearchChar As String = "P"

Dim TestPos As Integer
' A textual comparison starting at position 4. Returns 6.
TestPos = InStr(4, SearchString, SearchChar, CompareMethod.Text)

' A binary comparison starting at position 1. Returns 9.
TestPos = InStr(1, SearchString, SearchChar, CompareMethod.Binary)

' If Option Compare is not set, or set to Binary, return 9.
' If Option Compare is set to Text, returns 3.
TestPos = InStr(SearchString, SearchChar)

' Returns 0.
TestPos = InStr(1, SearchString, "W")
```

Example

This example shows various uses of the `Format` function to format values using both `String` formats and user-defined formats. For the date separator (`/`), time separator (`:`), and the AM/PM indicators (`t` and `tt`), the actual formatted output displayed by your system depends on the locale settings the code is using. When times and dates are displayed in the development environment, the short time format and short date format of the code locale are used.

Note

For locales that use a 24-hour clock, the AM/PM indicators (`t` and `tt`) display nothing.

VB

```
Dim TestDateTime As Date = #1/27/2001 5:04:23 PM#
Dim TestStr As String
' Returns current system time in the system-defined long time format.
TestStr = Format(Now(), "Long Time")
' Returns current system date in the system-defined long date format.
TestStr = Format(Now(), "Long Date")
' Also returns current system date in the system-defined long date
' format, using the single letter code for the format.
TestStr = Format(Now(), "D")

' Returns the value of TestDateTime in user-defined date/time formats.
' Returns "5:4:23".
TestStr = Format(TestDateTime, "h:m:s")
' Returns "05:04:23 PM".
TestStr = Format(TestDateTime, "hh:mm:ss tt")
' Returns "Saturday, Jan 27 2001".
TestStr = Format(TestDateTime, "dddd, MMM d yyyy")
' Returns "17:04:23".
TestStr = Format(TestDateTime, "HH:mm:ss")
' Returns "23".
TestStr = Format(23)

' User-defined numeric formats.
' Returns "5,459.40".
TestStr = Format(5459.4, "##,##0.00")
' Returns "334.90".
TestStr = Format(334.9, "###0.00")
' Returns "500.00%".
TestStr = Format(5, "0.00%")
```

See Also

Keywords
Visual Basic Runtime Library Members
String Manipulation Summary