Simple Runtime Reference

August 2009
Version 0.1.1

Functions

- Application - Various application related runtime functions
- Arrays - Various array related runtime functions
- Assertions - Checking the runtime state of an application
- Collection - Unordered set of items
- Conversions - Various conversion related runtime functions
- Dates - Various date and time related runtime functions
- Files - Various file related runtime functions Log - Logging related runtime functions
- Math - Various mathematical runtime functions
- Strings - Various string related runtime functions

Components

- AccelerometerSensor - Sensor to measure acceleration in 3 dimensions
- Button - Push-style button
- Canvas - Surface to draw on
- CheckBox - Two state button that can either be checked or un-checked
- EmailPicker - Editable text box using auto-completion to pick out an email address from contacts
- Form - Root component container
- Image - Component for displaying images
- Label - Text display
- LocationSensor - Sensor to provide information about the current location
- OrientationSensor - Sensor to measure absolute orientation in 3 dimensions
- Panel - Container for other components
- PasswordTextBox - Editable text box for entering passwords
- Phone - Component providing phone-related functionality
- RadioButton - Two state button that can either be checked or un-checked
- TextBox - Editable text box
- Timer - Timer component

Layouts

- FrameLayout - Layout for prominently showing a single component
- LinearLayout - Layout for placing components horizontally or vertically
- TableLayout - Layout for placing components in tabular form

Runtime Errors

- AssertionFailure - Indicates an assertion failure
- ConversionError - Indicates a failed attempt to convert a value of one type into another
- FileAlreadyExistsError - Indicates a failed attempt to create a file
- FileIOError - Indicates a problem accessing a file

- IllegalArgumentError - Indicates an illegal value for a function or procedure argument
- IndexOutOfBoundsError - Indicates an array or collection access with an index that is outside of bounds
- NoSuchFileError - Indicates that no file for the given name could be found
- PropertyAccessError - Indicates illegal access to a property
- UninitializedInstanceError - Indicates access to uninitialized object or array variable
- UnknownFileHandleError - Indicates usage of an unknown file handle
- UnknownIdentifierError - Indicates that an identifier could not be resolved at runtime

namespace com.google.devtools.simple.runtime

Application

Various application related runtime functions.

- AddMenuItem - Creates a new menu item with the given caption.
- Finish - Terminates this application.
- GetPreference - Retrieves the value of a previously stored preference (even from previous of the same program).
- StorePreference - Stores the given value under given name.
- SwitchForm - Display a different form.

AddMenuItem

```
Static Sub AddMenuItem(caption As String)
```

Creates a new menu item with the given caption.
The caption will also be used to identify the menu item in the menu event handler.

Parameters:
- `caption` - menu item caption

SwitchFormStatic

```
Sub SwitchForm(form As Form)
```

Display a different form.

Parameters:
- `form` - form to display

Finish

```
Static Sub Finish()
```

Terminates this application.

GetPreferenceStatic

```
Static Function GetPreference(name As String) As Variant
```

Retrieves the value of a previously stored preference (even from previous of the same program).

Parameters:
- `name` - name which was used to store the value under

Returns:
- value associated with name

StorePreference

```
Static Sub StorePreference(name As String, value As Variant)
```

Stores the given value under given name. The value can be retrieved using the given name any time (even on subsequent runs of the program).

Parameters:

- `name` - name to store value under
- `value` - value to store (must be a primitive value, objects not allowed)

namespace com.google.devtools.simple.runtime

Arrays

Various array related runtime functions.

- Filter - Filters the contents of an array.
- Join - Appends array elements to a become a single string.
- Split - Splits up the given string where a separator is being found.
- UBound - Return the size of an array dimension.

Filter

```
Static Function Filter(array As String(), str As String, include As Boolean) As String()
```

Filters the contents of an array.

Parameters:
- `array` - array to search in
- `str` - substring to search for in the array
- `include` - if true then include matching strings in the result, otherwise exclude them

Returns:
- array containing (non-)matching array entries

Join

```
Static Function Join(array As String(), separator As String) As String
```

Appends array elements to a become a single string.

Parameters:
- `array` - array containing strings to be appended
- `separator` - string append between array elements

Returns:

- string containing appended array elements

Split

```
Static Function Split(str As String, separator As String, count As Integer) As String()
```

Splits up the given string where a separator is being found.

Parameters:
- `str` - string to be split up
- `separator` - separator to look for
- `count` - number of times

Returns:
- array containing split string

UBound

```
Static Function UBound(array As Variant, dim As Integer) As Integer
```

Return the size of an array dimension.

Parameters:
- `array` - array whose size is requested
- `dim` - dimension (1 for the first dimension, and so on)

Returns:
- size of the array dimension

namespace com.google.devtools.simple.runtime

Assertions

Assertions allow test against assumptions about the runtime state of an application. A failing assertion will result in an AssertionFailure runtime error.

- AssertFalse - Tests whether an assertion is false.
- AssertTrue - Tests whether an assertion is true.

AssertTrue

```
Static Sub AssertTrue(expression As Variant)
```

Tests whether an assertion is true. Evaluates the given expression and causes an `AssertionFailure` runtime error if the expression does not evaluate to `True`.

Parameters:
- `expression` - expression to test

AssertFalse

```
Static Sub AssertFalse(expression As Variant)
```

Tests whether an assertion is false. Evaluates the given expression and causes an `AssertionFailure` runtime error if the expression does not evaluate to `False`.

Parameters:
- `expression` - expression to test

namespace com.google.devtools.simple.runtime

Collection

A collection is an ordered set of items. Unlike arrays where all members must have the same data type, collections do not have that restriction.

- Add - Adds a new item to the collection.
- Clear - Removes all items from the collection.
- Contains - Checks whether an item is already part of the collection.
- Count - Returns the number of items in the collection.
- Item - Returns the item at the specified position.
- Remove - Removes an item from the collection.

Clear

```
Sub Clear()
```

Removes all items from the collection.

Add

```
Sub Add(item As Variant)
```

Adds a new item to the collection.

Parameters:

- `item` - item to be added

Item

```
Function Item(index As Integer) As Variant
```

Returns the item at the specified position.

Parameters:

- `index` - item position

Returns:
- item

Count

```
Property Count As Integer
```

Returns the number of items in the collection. This is a read-only property.

Contains

```
Function Contains(item As Variant) As Boolean
```

Checks whether an item is already part of the collection.

Parameters:
- `item` - item to look for

Returns:
- `True` if the item is already in the collection

Remove

```
Sub Remove(item As Variant)
```

Removes an item from the collection.

Parameters:
- `item` - item to remove

namespace com.google.devtools.simple.runtime

Conversions

Various conversion related runtime functions.

- Asc - Returns the unicode value of the first character of the given string.
- Chr - Returns a string for the given unicode value.
- Hex - Returns a string containing the hexadecimal value for the given value.

Asc

```
Static Function Asc(str As String) As Integer
```

Returns the unicode value of the first character of the given string.

Parameters:
- `str` - string to convert first character of

Returns:
- unicode value of first character of `str`

Chr

```
Static Function Chr(value As Integer) As String
```

Returns a string for the given unicode value.

Parameters:
- `value` - unicode value to convert into a string

Returns:
- string consisting of given unicode value

Hex

```
Static Function Hex(v As Variant) As String
```

Returns a string containing the hexadecimal value for the given value. If the given value is not w whole number then its integer part will be used.

Parameters:

- `v` - value

Returns:

- string with hexadecimal value of `v`

namespace com.google.devtools.simple.runtime

Dates

Various date and time related runtime functions.

- DATE_YEAR DATE_MONTH DATE_DAY DATE_WEEK DATE_HOUR DATE_MINUTE DATE_SECOND - Date/time interval kind constants.
- DATE_JANUARY DATE_FEBRUARY DATE_MARCH DATE_APRIL DATE_MAY DATE_JUNE DATE_JULY DATE_AUGUST DATE_SEPTEMBERDATE_OCTOBER DATE_NOVEMBER DATE_DECEMBER - Month constants.
- DATE_MONDAY DATE_TUESDAY DATE_WEDNESDAY DATE_THURSDAY DATE_FRIDAY DATE_SATURDAY DATE_SUNDAY - Weekday constant.

- DateAdd - Adds a time interval to the given date.
- DateValue - Creates a date from the given string.
- Day - Returns the day of the month for the given date.
- FormatDate - Converts and formats the given date into a string.
- Hour - Returns the hours for the given date.
- Minute - Returns the minutes for the given date.
- Month - Returns the month of the given date.
- MonthName - Returns the name of the month for the given date.
- Now - Returns the current date and time.
- Second - Returns the seconds for the given date.
- Timer - Returns the current system time in milliseconds.
- Weekday - Returns the weekday for the given date.
- WeekdayName - Returns the name of the weekday for the given date.
- Year - Returns the year of the given date.

DATE_YEAR, DATE_MONTH, DATE_DAY, DATE_WEEK, DATE_HOUR, DATE_MINUTE, DATE_SECOND

```
Const DATE_YEAR As Integer
Const DATE_MONTH As Integer
Const DATE_DAY As Integer
Const DATE_WEEK As Integer
Const DATE_HOUR As Integer
Const DATE_MINUTE As Integer
Const DATE_SECOND As Integer
```

Date/time interval kind constants.

DATE_JANUARY, DATE_FEBRUARY, DATE_MARCH, DATE_APRIL, DATE_MAY, DATE_JUNE, DATE_JULY, DATE_AUGUST, DATE_SEPTEMBER, DATE_OCTOBER, DATE_NOVEMBER, DATE_DECEMBER

```
Const DATE_JANUARY As Integer
Const DATE_FEBRUARY As Integer
Const DATE_MARCH As Integer
Const DATE_APRIL As Integer
Const DATE_MAY As Integer
Const DATE_JUNE As Integer
Const DATE_JULY As Integer
Const DATE_AUGUST As Integer
Const DATE_SEPTEMBER As Integer
Const DATE_OCTOBER As Integer
Const DATE_NOVEMBER As Integer
Const DATE_DECEMBER As Integer
```

Month constants.

DATE_MONDAY, DATE_TUESDAY, DATE_WEDNESDAY, DATE_THURSDAY, DATE_FRIDAY, DATE_SATURDAY, DATE_SUNDAY

```
Const DATE_MONDAY As Integer
Const DATE_TUESDAY As Integer
Const DATE_WEDNESDAY As Integer
Const DATE_THURSDAY As Integer
Const DATE_FRIDAY As Integer
Const DATE_SATURDAY As Integer
Const DATE_SUNDAY As Integer
```

Weekday constant.

DateAdd

```
Static Sub DateAdd(date As Date, intervalKind As Integer, interval
As Integer)
```

Adds a time interval to the given date.

Parameters:

- `date` - date to add to
- `intervalKind` - kind of interval (one of `DATE_YEAR`, `DATE_MONTH`, `DATE_DAY`, `DATE_WEEK`, `DATE_HOUR`, `DATE_MINUTE` or `DATE_SECOND`)
- `interval` - units to add

DateValue

```
Static Function DateValue(value As String) As Date
```

Creates a date from the given string.
Dates must be formatted as follows: MM/DD/YYYY hh:mm:ss or MM/DD/YYYY where MM is the month (01-12), DD the day (01-31), YYYY the year (0000-9999), hh the hours (00-23), mm the minutes (00-59) and ss the seconds (00-59).

Parameters:

- `value` - string to convert

Returns:

- date

Day

```
Static Function Day(date As Date) As Integer
```

Returns the day of the month for the given date.

Parameters:

- `date` - date to get day of

Returns:

- day (range 1 - 31)

FormatDate

```
Static Function FormatDate(date As Date) As String
```

Converts and formats the given date into a string.

Parameters:
- `date` - date to format

Returns:
- formatted date

Hour

```
Static Function Hour(date As Date) As Integer
```

Returns the hours for the given date.

Parameters:
- `date` - date to use hours of

Returns:
- hours (range 0 - 23)

Minute

```
Static Function Minute(date As Date) As Integer
```

Returns the minutes for the given date.

Parameters:
- `date` - date to use minutes of

Returns:
- minutes (range 0 - 59)

Month

```
Static Function Month(date As Date) As Integer
```

Returns the month of the given date.

Parameters:
- `date` - date to use month of

Returns:
- month (one of `DATE_JANUARY`, `DATE_FEBRUARY`, `DATE_MARCH`, `DATE_APRIL`, `DATE_MAY`, `DATE_JUNE`, `DATE_JULY`, `DATE_AUGUST`, `DATE_SEPTEMBER`, `DATE_OCTOBER`, `DATE_NOVEMBER` or `DATE_DECEMBER`)

MonthName

```
Static Function MonthName(date As Date) As String
```

Returns the name of the month for the given date.

Parameters:
- `date` - date to use month of

Returns:
- name of month

Now

```
Static Function Now() As Date
```

Returns the current date and time.

Returns:

- current date and time

Second

```
Static Function Second(date As Date) As Integer
```

Returns the seconds for the given date.

Parameters:
- `date` - date to use seconds of

Returns:
- seconds (range 0 - 59)

Timer

```
Static Timer() As Long
```

Returns the current system time in milliseconds.

Returns:
- current system time in milliseconds

Weekday

```
Static Function Weekday(date As Date) As Integer
```

Returns the weekday for the given date.

Parameters:

- `date` - date to use weekday of

Returns:

- weekday (one of `DATE_SUNDAY`, `DATE_MONDAY`, `DATE_TUESDAY`, `DATE_WEDNESDAY`, `DATE_THURSDAY`, `DATE_FRIDAY` or `DATE_SATURDAY`)

WeekdayName

```
Static Function WeekdayName(date As Date) As String
```

Returns the name of the weekday for the given date.

Parameters:

- `date` - date to use weekday of

Returns:

- name of weekday

Year

```
Static Function Year(date As Date) As Integer
```

Returns the year of the given date.

Parameters:

- `date` - date to use year of

Returns:

- year

namespace com.google.devtools.simple.runtime

Files

Various file related runtime functions.

- Close - Closes a file previously opened.
- Delete - Deletes a file.
- Eof - Checks whether the current file position is at the end of the file.
- Exists - Checks whether a file or directory exists.
- IsDirectory - Checks whether the given name is the name of an existing directory.
- Mkdir - Creates a new directory.
- Open - Opens an existing file or creates a new file for reading or writing.
- ReadBoolean - Reads a Boolean value from a file.
- ReadByte - Reads a Byte value from a file.
- ReadDouble - Reads a Double value from a file.
- ReadInteger - Reads an Integer value from a file.
- ReadLong - Reads a Long value from a file.
- ReadShort - Reads a Short value from a file.
- ReadSingle - Reads a Single value from a file.
- ReadString - Reads a String value from a file.
- Rename - Renames a file.
- Rmdir - Deletes a directory.
- Seek - Positions the file pointer to an absolute position.
- Size - Returns the size of a file.
- WriteBoolean - Writes a Boolean value to a file.
- WriteByte - Writes a Byte value to a file.
- WriteDouble - Writes a Double value to a file.
- WriteInteger - Writes an Integer boolean value to a file.
- WriteLong - Writes a Long value to a file.
- WriteShort - Writes a Short value to a file.
- WriteSingle - Writes a Single value to a file.
- WriteString - Writes a String to a file.

Rename

```
Static Sub Rename(oldname As String, newname As String)
```

Renames a file. Causes a runtime error if the file doesn't exist.

Parameters:
- `oldname` - file name before renaming
- `newname` - file name after renaming

Delete

```
Static Sub Delete(name As String)
```

Deletes a file.

Parameters:
- `name` - name of file to delete

Mkdir

```
Static Sub Mkdir(name As String)
```

Creates a new directory.

Parameters:
- `name` - name of new directory

Rmdir

```
Static Sub Rmdir(name As String)
```

Deletes a directory.

Parameters:
- `name` - name of directory to delete

IsDirectory

```
Static Function IsDirectory(name As String) As Boolean
```

Checks whether the given name is the name of an existing directory. Causes a runtime error if the directory doesn't exist.

Parameters:
- `name` - name to check

Returns:
- `True` if the name belongs to an existing directory, `False` otherwise

---

Exists

```
Static Function Exists(name As String) As Boolean
```

Checks whether a file or directory exists.

Parameters:
- `name` - file to check

Returns:
- `True` if the file or directory exists, `False` otherwise

Open

```
Static Function Open(name As String) As Integer
```

Opens an existing file or creates a new file for reading or writing.

Parameters:
- `name` - name of file to open or create

Returns:
- file handle

Close

```
Static Sub Close(handle As Integer)
```

Closes a file previously opened.

Parameters:
- `handle` - handle of file to close

Eof

```
Static Function Eof(handle As Integer) As Boolean
```

Checks whether the current file position is at the end of the file.

Parameters:
- `handle` - handle of file to check

Returns:
- `True` if the end of the file was reaches, `False` otherwise

Seek

```
Static Function Seek(handle As Integer, offset As Long) As Long
```

Positions the file pointer to an absolute position.

Parameters:
- `handle` - handle of file
- `offset` - absolute position within file

Returns:
- new position within file

Size

```
Static Function Size(handle As Integer) As Long
```

Returns the size of a file.

Parameters:
- `handle` - handle of file

Returns:
- file size

WriteString

```
Static Sub WriteString(handle As Integer, value As String)
```

Writes a String to a file.

Parameters:
- `handle` - handle of file
- `value` - value to write

ReadString

```
Static Function ReadString(handle As Integer) As String
```

Reads a String value from a file.

Parameters:
- `handle` - handle of file

Returns:
- value read

WriteBoolean

```
Static Sub WriteBoolean(handle As Integer, value As Boolean)
```

Writes a Boolean value to a file.

Parameters:
- `handle` - handle of file
- `value` - value to write

ReadBoolean

```
Static Function ReadBoolean(handle As Integer) As Boolean
```

Reads a Boolean value from a file.

Parameters:
- `handle` - handle of file

Returns:
- value read

WriteByte

```
Static Sub WriteByte(handle As Integer, value As Byte)
```

Writes a Byte value to a file.

Parameters:
- `handle` - handle of file
- `value` - value to write

ReadByte

```
Static Function ReadByte(handle As Integer) As Byte
```

Reads a Byte value from a file.

Parameters:
- `handle` - handle of file

Returns:
- value read

WriteShort

```
Static Sub WriteShort(handle As Integer, value As Short)
```

Writes a Short value to a file.

Parameters:
- `handle` - handle of file

- `value` - value to write

ReadShort

```
Static Function ReadShort(handle As Integer) As Short
```

Reads a Short value from a file.

Parameters:
- `handle` - handle of file

Returns:
- value read

WriteInteger

```
Static Sub WriteInteger(handle As Integer, value As Integer)
```

Writes an Integer boolean value to a file.

Parameters:
- `handle` - handle of file
- `value` - value to write

ReadInteger

```
Static Function ReadInteger(handle As Integer) As Integer
```

Reads an Integer value from a file.

Parameters:
- `handle` - handle of file

Returns:
- value read

WriteLong

```
Static Sub WriteLong(handle As Integer, value As Long)
```

Writes a Long value to a file.

Parameters:
- `handle` - handle of file
- `value` - value to write

ReadLong

```
Static Function ReadLong(handle As Integer) As Long
```

Reads a Long value from a file.

Parameters:
- `handle` - handle of file

Returns:
- value read

WriteSingle

```
Static Sub WriteSingle(handle As Integer, value As Single)
```

Writes a Single value to a file.

Parameters:
- `handle` - handle of file
- `value` - value to write

ReadSingle

```
Static Function ReadSingle(handle As Integer) As Single
```

Reads a Single value from a file.

Parameters:
- `handle` - handle of file

Returns:
- value read

WriteDouble

```
Static Sub WriteDouble(handle As Integer, value As Double)
```

Writes a Double value to a file.

Parameters:
- `handle` - handle of file
- `value` - value to write

ReadDouble

```
Static Function ReadDouble(handle As Integer) As Double
```

Reads a Double value from a file.

Parameters:

- `handle` - handle of file

Returns:

- value read

namespace com.google.devtools.simple.runtime

Log

Logging related runtime functions.

- Error - Logs an error message.
- Info - Logs an info message.
- Warning - Logs an warning message.

Error

```
Static Sub Error(moduleName As String, message As String)
```

Logs an error message.

Parameters:
- `moduleName` - name of the module reporting the message (e.g. "Simple Runtime Library")
- `message` - text to log

Warning

```
Static Sub Warning(moduleName As String, message As String)
```

Logs an warning message.

Parameters:
- `moduleName` - name of the module reporting the message (e.g. "Simple Runtime Library")
- `message` - text to log

Info

```
Static Sub Info(moduleName As String, message As String)
```

Logs an info message.

Parameters:

- `moduleName` - name of the module reporting the message (e.g. "Simple Runtime Library")
- `message` - text to log

namespace com.google.devtools.simple.runtime

Math

Various mathematical runtime functions.

- E -Euler's constant.
- PI - Pi.

- Abs - Returns the absolute value of the given value.
- Atn - Returns the arctangent for the given value.
- Atn2 - Returns the angle theta from the conversion of rectangular coordinates (x, y) to polar coordinates (r, theta).
- Cos - Returns the cosine for the given value.
- DegreesToRadians - Converts an angle measured in degrees to an approximation in radians.
- Exp - Returns e (euler's constant) raised to the power of the given value.
- Int - Returns the integer part of the given number.
- Log - Returns the natural logarithm for the given number.
- Max - Returns the greater of two values.
- Min - Returns the smaller of two values.
- RadiansToDegrees - Converts an angle measured in radians to an approximation in degrees.
- Rnd - Returns a random number in the range between 0.0 (inclusive) and 1.0 (exclusive).
- Sgn - Indicates the sign for the given value.
- Sin - Returns the sine for the given value.
- Sqr - Returns the square root for the given value.
- Tan - Returns the tangent for the given value.

E

```
Const E As Double
```

Euler's constant.

PI

```
Const PI As Double
```

Pi.

Abs

```
Static Function Abs(v As Variant) As Variant
```

Returns the absolute value of the given value.

Parameters:
- `v` - value

Returns:
- absolute value

Atn

```
Static Function Atn(v As Double) As Double
```

Returns the arctangent for the given value.

Parameters:
- `v` - value

Returns:
- arctangent of `v`

Atn2

```
Static Function Atn2(y As Double, x As Double) As Double
```

Returns the angle theta from the conversion of rectangular coordinates (x, y) to polar coordinates (r, theta>).

Parameters:
- `y` - the ordinate coordinate
- `x` - the abscissa coordinate

Returns:
- the theta component of the point (r, theta) in polar coordinates that corresponds to the point (x, y) in Cartesian coordinates

---

Cos

```
Static Function Cos(v As Double) As Double
```

Returns the cosine for the given value.

Parameters:
- `v` - value

Returns:
- cosine of `v`

Exp

```
Static Function Exp(v As Double) As Double
```

Returns e (euler's constant) raised to the power of the given value.

Parameters:
- `v` - value

Returns:
- e to the power of `v`

Int

```
Static Function Int(v As Variant) As Long
```

Returns the integer part of the given number.

Parameters:
- `v` - value

Returns:
- integer part of `v`

Log

```
Static Function Log(v As Double) As Double
```

Returns the natural logarithm for the given number.

Parameters:
- `v` - value

Returns:
- natural logarithm for `v`

Max

```
Static Function Max(v1 As Variant, v2 As Variant) As Variant
```

Returns the greater of two values.

Parameters:
- `v1` - first value
- `v2` - second value

Returns:
- greater value of `v1` and `v2`

Min

```
Static Function Min(v1 As Variant, v2 As Variant) As Variant
```

Returns the smaller of two values.

Parameters:
- `v1` - first value
- `v2` - second value

Returns:
- smaller value of `v1` and `v2`

Rnd

```
Static Function Rnd() As Double
```

Returns a random number in the range between 0.0 (inclusive) and 1.0 (exclusive).

Returns:
- random number (between 0.0 and 1.0)

Sin

```
Static Function Sin(v As Double) As Double
```

Returns the sine for the given value.

Parameters:
- `v` - value

Returns:
- sine of `v`

Sgn

```
Static Function Sgn(v As Double) As Integer
```

Indicates the sign for the given value.

Parameters:
- `v` - value

Returns:
- for positive values, 0 for zero, and -1 for negative values

---

Sqr

```
Static Function Sqr(v As Double) As Double
```

Returns the square root for the given value.

Parameters:
- `v` - value

Returns:
- square root of `v`

Tan

```
Static Function Tan(v As Double) As Double
```

Returns the tangent for the given value.

Parameters:
- `v` - value

Returns:
- tangent of `v`

DegreesToRadians

```
Static Function DegreesToRadians(d As Double) As Double
```

Converts an angle measured in degrees to an approximation in radians.

Parameters:
- `d` - value in degrees

Returns:
- radian approximation to d degrees

RadiansToDegrees

```
Static Function RadiansToDegrees(r As Double) As Double
```

Converts an angle measured in radians to an approximation in degrees.

Parameters:
- `r` - value in radians

Returns:

- degree approximation to `r` radians

namespace com.google.devtools.simple.runtime

Strings

Various string related runtime functions.

- InStr - Searches for a string in another string.
- InStrRev - Searches for a string in another string starting at the end of that string.
- LCase - Converts the given string to all lowercase.
- Left - Returns the specified number of characters from the start of the given string.
- Len - Returns the number of characters in the given string.
- LTrim - Removes leading space characters from the given string.
- Mid - Returns the specified number of characters from the given string starting from the given index.
- Replace - Replaces occurrences of one string with another string in the given string.
- Right - Returns the specified number of characters from the end of the given string.
- RTrim - Removes trailing space characters from the given string.
- StrComp - Compares the two given strings lexicographically.
- StrReverse - Reverses the given string.
- Trim - Removes leading and trailing space characters from the given string.
- UCase - Converts the given string to all uppercase.

InStr

```
Static Function InStr(str1 As String, str2 As String, start As Integer) As Integer
```

Searches for a string in another string.

Parameters:
- `str1` - string to search in
- `str2` - string to search for
- `start` - search start index within `str1`

Returns:
- index at which `str2` was found within `str1` or a negative value if `str2` was not found within `str1`

InStrRev

```
Static Function InStrRev(str1 As String, str2 As String, start As Integer) As Integer
```

Searches for a string in another string starting at the end of that string.

Parameters:
- `str1` - string to search in
- `str2` - string to search for
- `start` - search start index within `str1`

Returns:
- index at which `str2` was found within `str1` or a negative value if `str2` was not found within `str1`

LCase

```
Static Sub LCase(ByRef str As String)
```

Converts the given string to all lowercase.

Parameters:
- `str` - string to convert to lowercase

UCase

```
Static Sub UCase(ByRef str As String)
```

Converts the given string to all uppercase.

Parameters:
- `str` - string to convert to uppercase

Left

```
Static Function Left(str As String, len As Integer) As String
```

Returns the specified number of characters from the start of the given string.

Parameters:
- `str` - string to return characters from
- `len` - number of characters to return

Returns:
- substring of the given string

Right

```
Static Function Right(str As String, len As Integer) As String
```

Returns the specified number of characters from the end of the given string.

Parameters:
- `str` - string to return characters from
- `len` - number of characters to return

Returns:
- substring of the given string

Mid

```
Static Function Mid(str As String, start As Integer, len As
Integer) As String
```

Returns the specified number of characters from the given string starting from the given index.

Parameters:
- str - string to return characters from
- start - start index within str
- len - number of characters to return

Returns:
- substring of the given string

Len

```
Static Function Len(str As String) As Integer
```

Returns the number of characters in the given string.

Parameters:
- `str` - string to get length of

Returns:
- number of characters (length) of the given string

Trim

```
Static Sub Trim(ByRef str As String)
```

Removes leading and trailing space characters from the given string.

Parameters:
- `str` - string to trim

LTrim

```
Static Sub LTrim(ByRef str As String)
```

Removes leading space characters from the given string.

Parameters:
- `str` - string to trim

RTrim

```
Static Sub RTrim(ByRef str As String)
```

Removes trailing space characters from the given string.

Parameters:
- `str` - string to trim

Replace

```
Static Sub Replace(ByRef str As String, find As String, replace As String, start As Integer, count As Integer)
```

Replaces occurrences of one string with another string in the given string.

Parameters:
- `str` - string to modify
- `find` - string to find
- `replace` - string to replace found string with
- `start` - start index within str
- `count` - number of times to perform the replacement (-1 to replace all occurrences)

StrComp

```
Static Function StrComp(str1 As String, str2 As String) As Integer
```

Compares the two given strings lexicographically.

Parameters:
- `str1` - first string of comparison
- `str2` - second string of comparison

Returns:
- 0 if the strings are equal, a negative number if `str2` follows `str1` and a positive number if `str1` follows `str2`

StrReverse

```
Static Sub StrReverse(ByRef str As String)
```

Reverses the given string.

Parameters:
- `str` - string to reverse

namespace com.google.devtools.simple.runtime.errors

AssertionFailure

Runtime error indicating an assertion failure.

namespace com.google.devtools.simple.runtime.errors

ConversionError

Runtime error indicating a failed attempt of converting a value of a type into a value of another type, e.g. the String "foo" into an Integer, but also converting from a base type to a derived type where there is no relationship.

namespace com.google.devtools.simple.runtime.errors

FileAlreadyExistsError

Runtime error indicating that the attempt to create a file failed because there is a file already existing with the same name.

namespace com.google.devtools.simple.runtime.errors

FileIOError

Runtime error indicating a problem accessing a file.

namespace com.google.devtools.simple.runtime.errors

IllegalArgumentError

Runtime error indicating an illegal value for a function or procedure argument.

namespace com.google.devtools.simple.runtime.errors

IndexOutOfBoundsError

Runtime error indicating an array or collection access with an index that is outside of bounds.

namespace com.google.devtools.simple.runtime.errors

NoSuchFileError

Runtime error indicating that no file for the given name could be found.

namespace com.google.devtools.simple.runtime.errors

PropertyAccessError

Runtime error indicating write access to a read-only property or read access to a write-only property.

namespace com.google.devtools.simple.runtime.errors

UnknownFileHandleError

Runtime error indicating an unknown file handle.

namespace com.google.devtools.simple.runtime.errors

UnknownIdentifierError

Runtime error indicating that an identifier could not be resolved at runtime.

namespace com.google.devtools.simple.runtime.errors

UninitializedInstanceError

Runtime error indicating an access to an instance or array variable that is not properly initialized.

namespace com.google.devtools.simple.runtime.components

AccelerometerSensor

Sensor to measure acceleration in 3 dimensions, and also detect shaking.

Events

- Initialize - Initialization event.
- AccelerationChanged - Acceleration change event.
- Shaking - Shaking event.

Properties

- Available - Available property (read-only property).
- Enabled - Enabled property.
- XAccel - X acceleration property (read-only property).
- YAccel - Y acceleration property (read-only property).
- ZAccel - Z acceleration property (read-only property).

Initialize

```
Event Initialize()
```

Event raised upon component initialization.

AccelerationChanged

```
Event AccelerationChanged(xAccel As Single, yAccel As Single,
zAccel As Single)
```

Event raised when the acceleration in any of the 3 dimensions changes.

Parameters:

- `xAccel` - acceleration minus Gx on the x-axis
- `yAccel` - acceleration minus Gy on the y-axis
- `zAccel` - acceleration minus Gz on the z-axis

Shaking

```
Event Shaking()
```

Event raised when the device is being shaken.

Available

```
Property Available As Boolean
```

This property indicates whether the sensor is available on the device running the application. This property is read-only.

Enabled

```
Property Enabled As Boolean
```

Reading from the Enabled property indicates whether the sensor is generating data. Writing to the Enabled property will turn sensor data generation on or off. Data generation is enabled by default.

XAccel

```
Property XAccel As Single
```

Reading the value of this property returns the most recent x acceleration value. In order for this property to supply meaningful values, the sensor needs to be available and enabled. Writing to this property will accelerate the device at the given rate. Use this only in a controlled environment as sudden acceleration may cause severe injury... No, just kidding - this property is read-only.

YAccel

```
Property YAccel As Single
```

Reading the value of this property returns the most recent y acceleration value. In order for this property to supply meaningful values, the sensor needs to be available and enabled. This property is read-only.

ZAccel

```
Property ZAccel As Single
```

Reading the value of this property returns the most recent z acceleration value. In order for this property to supply meaningful values, the sensor needs to be available and enabled. This property is read-only.

namespace com.google.devtools.simple.runtime.components

Button

Simple Button component.

Events

- Initialize - Initialization event.
- Click - Click event.
- GotFocus - Focus received event.
- LostFocus - Focus lost event.

Properties

- BackgroundColor - Property for controlling the component's background color.
- Column - Property for controlling the column position when using a table layout.
- Height - Property for controlling the component's height.
- Row - Property for controlling the row position when using a table layout.
- Width - Property for controlling the component's width.
- FontBold - Property for controlling the component's font weight.
- FontItalic - Property for controlling the component's font style.
- FontSize - Property for controlling the component's font size.
- FontTypeface - Property for controlling the component's font typeface.
- Justification - Property for controlling the component's text justification.
- Text - Property for controlling the component's text.
- TextColor - Property for controlling the component's text color.
- Enabled - Property for controlling whether the component is enabled.
- Image - Property for controlling an image shown on the component.

Initialize

```
Event Initialize()
```

Event raised upon component initialization.

Click

```
Event Click()
```

Event raised after the button is clicked or touched.

GotFocus

```
Event GotFocus()
```

Event raised after the component gains focus.

LostFocus

```
Event LostFocus()
```

Event raised after the component lost focus.

BackgroundColor

```
Property BackgroundColor As Integer
```

Reading from the BackgroundColor property returns the current background color of the component. The color value is encoded as &Haarrggbb where aa represents the alpha value (&H00 - transparent to &HFF - opaque), rr represents the red, gg the green and bb the blue component of the color. Writing to the BackgroundColor property will set the background color of the component.
There are a number of predefined color constants: `Component.COLOR_NONE`, `Component.COLOR_BLACK`, `Component.COLOR_BLUE`,`Component.COLOR_CYAN`, `Component.COLOR_DKGRAY`, `Component.COLOR_GRAY`, `Component.COLOR_GREEN`, `Component.COLOR_LTGRAY`,`Component.COLOR_MAGENTA`, `Component.COLOR_RED`, `Component.COLOR_WHITE` and `Component.COLOR_YELLOW`.

Column

```
Property Column As Integer
```

Reading from the Column property returns the current column position within a table layout. Writing to the Column property will set the column position of the component within a table layout. This property has no meaning for any other layout.

Height

```
Property Height As Integer
```

Reading from the Height property returns the current height of the component in pixels. Writing to the Height property changes the height of the component to the given value in pixels. There are two special values. `Component.LENGTH_PREFERRED` sets the preferred height of the component which depends on the contents of the component. `Component.LENGTH_FILL_PARENT` sets the height of the component to its maximum to fill the height of its parent container.

Row

```
Property Row As Integer
```

Reading from the Row property returns the current row position within a table layout. Writing to the Row property will set the row position of the component within a table layout. This property has no meaning for any other layout.

Width

```
Property Width As Integer
```

Reading from the Width property returns the current width of the component in pixels. Writing to the Width property changes the width of the component to the given value in pixels. There are two special values. `Component.LENGTH_PREFERRED` sets the preferred width of the component which depends on the contents of the component. `Component.LENGTH_FILL_PARENT` sets the width of the component to its maximum to fill the width of its parent container.

FontBold

```
Property FontBold As Boolean
```

Reading from this property indicates the font weight. A value of `True` means that the component font is bold, `False` means normal. Writing to this property changes the changes the font weight.
The default value of this property is `False`.

FontItalic

```
Property FontItalic As Boolean
```

Reading from this property indicates the font style. A value of `True` means that the component font is italic, `False` means normal. Writing to this property changes the changes the font style.
The default value of this property is `False`.

FontSize

```
Property FontSize As Single
```

Reading from this property returns the font height in points. Writing to this property changes the changes the font height.
The default value of this property is 14 points.

FontTypeface

```
Property FontTypeface As Integer
```

Reading from this property returns the font typeface. The value must be one of `Component.TYPEFACE_DEFAULT`, `Component.TYPEFACE_SERIF`,`Component.TYPEFACE_SANSSERIF` or `Component.TYPEFACE_MONOSPACE`. Writing to this property changes the changes the font

typeface.
The default value of this property is `Component.TYPEFACE_DEFAULT`.

Justification

```
Property Justification As Integer
```

Reading from this property returns the text justification. The value must be one of `Component.JUSTIFY_LEFT`, `Component.JUSTIFY_CENTER` or `Component.JUSTIFY_RIGHT`.
Writing to this property changes the changes the text justification.
The default value of this property is `Component.JUSTIFY_LEFT`.

Text

```
Property Text As String
```

Reading from this property returns the text displayed by the component. Writing to this property changes the changes the text displayed.

TextColor

```
Property TextColor As Integer
```

Reading from the TextColor property returns the current color of the text displayed by this component. The color value is encoded as &Haarrggbbwhere aa represents the alpha value (&H00 - transparent to &HFF - opaque), rr represents the red, gg the green and bb the blue component of the color. Writing to the TextColor property will set the color for the text of the component.
There are a number of predefined color constants: `Component.COLOR_NONE`, `Component.COLOR_BLACK`, `Component.COLOR_BLUE`,`Component.COLOR_CYAN`, `Component.COLOR_DKGRAY`, `Component.COLOR_GRAY`, `Component.COLOR_GREEN`, `Component.COLOR_LTGRAY`,`Component.COLOR_MAGENTA`, `Component.COLOR_RED`, `Component.COLOR_WHITE` and `Component.COLOR_YELLOW`.

Enabled

```
Property Enabled As Boolean
```

Reading from the Enabled property indicates whether the button is enabled. Writing to the Enabled property will enabled or disable the button.
Buttons are enabled by default.

Image

```
Property Image As String
```

Reading from this property returns the path of the image currently shown on the component. If there is no image shown an empty string will be returned. Writing to this property changes the image shown on the component.

namespace com.google.devtools.simple.runtime.components

Canvas

The Canvas component supplies a surface to draw on.

Events

- Initialize - Initialization event.
- Touched - Touch event.

Properties

- BackgroundColor - Property for controlling the component's background color.
- Column - Property for controlling the column position when using a table layout.
- Height - Property for controlling the component's height.
- Row - Property for controlling the row position when using a table layout.
- Width - Property for controlling the component's width.
- BackgroundImage - Property for controlling the background image of the canvas.
- PaintColor - Property for controlling the paint color.

Functions

- Clear - Clears the canvas.
- DrawCircle - Draws a circle at the given coordinates on the canvas, with the given radius.
- DrawLine - Draws a line between the given coordinates on the canvas.
- DrawPoint - Draws a point at the given coordinates on the canvas.

Initialize

```
Event Initialize()
```

Event raised upon component initialization.

Touched

Event Touched(x As Integer, y As Integer)

Event raised when the device screen is touched.

BackgroundColor

```
Property BackgroundColor As Integer
```

Reading from the BackgroundColor property returns the current background color of the component. The color value is encoded as &Haarrggbb where aa represents the alpha value (&H00 - transparent to &HFF - opaque), rr represents the red, gg the green and bb the blue component of the color. Writing to the BackgroundColor property will set the background color of the component. The default background color of the Canvas component is `Component.COLOR_WHITE`.
There are a number of predefined color constants: `Component.COLOR_NONE`, `Component.COLOR_BLACK`, `Component.COLOR_BLUE`,`Component.COLOR_CYAN`, `Component.COLOR_DKGRAY`, `Component.COLOR_GRAY`, `Component.COLOR_GREEN`, `Component.COLOR_LTGRAY`,`Component.COLOR_MAGENTA`, `Component.COLOR_RED`, `Component.COLOR_WHITE` and `Component.COLOR_YELLOW`.

Column

```
Property Column As Integer
```

Reading from the Column property returns the current column position within a table layout. Writing to the Column property will set the column position of the component within a table layout. This property has no meaning for any other layout.

Height

```
Property Height As Integer
```

Reading from the Height property returns the current height of the component in pixels. Writing to the Height property changes the height of the component to the given value in pixels. There are two special values. `Component.LENGTH_PREFERRED` sets the preferred height of the component which depends on the contents of the component. `Component.LENGTH_FILL_PARENT` sets the height of the component to its maximum to fill the height of its parent container.

Row

```
Property Row As Integer
```

Reading from the Row property returns the current row position within a table layout. Writing to the Row property will set the row position of the component within a table layout. This property has no meaning for any other layout.

Width

```
Property Width As Integer
```

Reading from the Width property returns the current width of the component in pixels. Writing to the Width property changes the width of the component to the given value in pixels. There are two special values. `Component.LENGTH_PREFERRED` sets the preferred width of the component which depends on the contents of the component. `Component.LENGTH_FILL_PARENT` sets the width of the component to its maximum to fill the width of its parent container.

BackgroundImage

```
Property BackgroundImage As String
```

Writing to this property changes the background image shown on the component. The value assigned to this property should be the name of a file in the project's assets directory. This property is write-only.

Clear

```
Sub Clear()
```

Clears the canvas (fills it with the background color).

DrawPoint

```
Sub DrawPoint(x As Integer, y As Integer)
```

Draws a point at the given coordinates on the canvas.

Parameters:
- `x` - x coordinate
- `y` - y coordinate

DrawCircle

```
Sub DrawCircle(x As Integer, y As Integer, r As Single)
```

Draws a circle at the given coordinates on the canvas, with the given radius

Parameters:
- `x` - x coordinate
- `y` - y coordinate
- `r` - radius

DrawLine

```
Sub DrawLine(x1 As Integer, y1 As Integer, x2 As Integer, y2 As
Integer)
```

Draws a line between the given coordinates on the canvas.

Parameters:
- `x1` - x coordinate of first point
- `y1` - y coordinate of first point
- `x2` - x coordinate of second point
- `y2` - y coordinate of second point

namespace com.google.devtools.simple.runtime.components

CheckBox

Two-state button that can either be checked or unchecked.

Events

- Initialize - Initialization event.
- Changed - Change event.
- GotFocus - Focus received event.
- LostFocus - Focus lost event.

Properties

- BackgroundColor - Property for controlling the component's background color.
- Column - Property for controlling the column position when using a table layout.
- Height - Property for controlling the component's height.
- Row - Property for controlling the row position when using a table layout.
- Width - Property for controlling the component's width.
- FontBold - Property for controlling the component's font weight.
- FontItalic - Property for controlling the component's font style.
- FontSize - Property for controlling the component's font size.
- FontTypeface - Property for controlling the component's font typeface.
- Justification - Property for controlling the component's text justification.
- Text - Property for controlling the component's text.
- TextColor - Property for controlling the component's text color.
- Enabled - Property for controlling whether the component is enabled.
- Value - Property for controlling the component's value.

Initialize

```
Event Initialize()
```

Event raised upon component initialization.

Changed

```
Event Changed()
```

Event raised after the checkbox's value changed.

GotFocus

```
Event GotFocus()
```

Event raised after the component gains focus.

LostFocus

```
Event LostFocus()
```

Event raised after the component lost focus.

BackgroundColor

```
Property BackgroundColor As Integer
```

Reading from the BackgroundColor property returns the current background color of the component. The color value is encoded as &Haarrggbb where aa represents the alpha value (&H00 - transparent to &HFF - opaque), rr represents the red, gg the green and bb the blue component of the color. Writing to the BackgroundColor property will set the background color of the component.
There are a number of predefined color constants: `Component.COLOR_NONE`, `Component.COLOR_BLACK`, `Component.COLOR_BLUE`,`Component.COLOR_CYAN`, `Component.COLOR_DKGRAY`, `Component.COLOR_GRAY`, `Component.COLOR_GREEN`, `Component.COLOR_LTGRAY`, `Component.COLOR_MAGENTA`, `Component.COLOR_RED`, `Component.COLOR_WHITE` and `Component.COLOR_YELLOW`.

Column

```
Property Column As Integer
```

Reading from the Column property returns the current column position within a table layout. Writing to the Column property will set the column position of the component within a table layout. This property has no meaning for any other layout.

Height

```
Property Height As Integer
```

Reading from the Height property returns the current height of the component in pixels. Writing to the Height property changes the height of the component to the given value in pixels. There are two special values. `Component.LENGTH_PREFERRED` sets the preferred height of the component which depends on the contents of the component. `Component.LENGTH_FILL_PARENT` sets the height of the component to its maximum to fill the height of its parent container.

Row

```
Property Row As Integer
```

Reading from the Row property returns the current row position within a table layout. Writing to the Row property will set the row position of the component within a table layout. This property has no meaning for any other layout.

Width

```
Property Width As Integer
```

Reading from the Width property returns the current width of the component in pixels. Writing to the Width property changes the width of the component to the given value in pixels. There are two special values. `Component.LENGTH_PREFERRED` sets the preferred width of the component which depends on the contents of the component. `Component.LENGTH_FILL_PARENT` sets the width of the component to its maximum to fill the width of its parent container.

FontBold

```
Property FontBold As Boolean
```

Reading from this property indicates the font weight. A value of `True` means that the component font is bold, `False` means normal. Writing to this property changes the changes the font weight.
The default value of this property is False.

FontItalic

```
Property FontItalic As Boolean
```

Reading from this property indicates the font style. A value of `True` means that the component font is italic, `False` means normal. Writing to this property changes the changes the font style.
The default value of this property is False.

FontSize

```
Property FontSize As Single
```

Reading from this property returns the font height in points. Writing to this property changes the changes the font height.
The default value of this property is 14 points.

FontTypeface

```
Property FontTypeface As Integer
```

Reading from this property returns the font typeface. The value must be one of `Component.TYPEFACE_DEFAULT`, `Component.TYPEFACE_SERIF`, `Component.TYPEFACE_SANSSERIF` or `Component.TYPEFACE_MONOSPACE`. Writing to this property changes the changes the font typeface.

The default value of this property is `Component.TYPEFACE_DEFAULT`.

Justification

```
Property Justification As Integer
```

Reading from this property returns the text justification. The value must be one of `Component.JUSTIFY_LEFT`, `Component.JUSTIFY_CENTER` or `Component.JUSTIFY_RIGHT`.
Writing to this property changes the changes the text justification.
The default value of this property is `Component.JUSTIFY_LEFT`.

Text

```
Property Text As String
```

Reading from this property returns the text displayed by the component. Writing to this property changes the changes the text displayed.

TextColor

```
Property TextColor As Integer
```

Reading from the TextColor property returns the current color of the text displayed by this component. The color value is encoded as &Haarrggbb where aa represents the alpha value (&H00 - transparent to &HFF - opaque), rr represents the red, gg the green and bb the blue component of the color. Writing to the TextColor property will set the color for the text of the component.
There are a number of predefined color constants: `Component.COLOR_NONE`, `Component.COLOR_BLACK`, `Component.COLOR_BLUE`,`Component.COLOR_CYAN`, `Component.COLOR_DKGRAY`, `Component.COLOR_GRAY`, `Component.COLOR_GREEN`, `Component.COLOR_LTGRAY`, `Component.COLOR_MAGENTA`, `Component.COLOR_RED`, `Component.COLOR_WHITE` and `Component.COLOR_YELLOW`.

Enabled

```
Property Enabled As Boolean
```

Reading from the Enabled property indicates whether the checkbox is enabled. Writing to the Enabled property will enabled or disable the checkbox.
Checkboxes are enabled by default.

Value

```
Property Value As Boolean
```

Reading from the Value property indicates the current state of the checkbox. Writing to the Value property will either check or uncheck the checkbox.
Checkboxes are unchecked by default.

namespace com.google.devtools.simple.runtime.components

EmailPicker

Editable text box using auto-completion to pick out an email address from contacts.

Events

- Initialize - Initialization event.
- GotFocus - Focus received event.
- LostFocus - Focus lost event.

Properties

- BackgroundColor - Property for controlling the component's background color.
- Column - Property for controlling the column position when using a table layout.
- Height - Property for controlling the component's height.
- Row - Property for controlling the row position when using a table layout.
- Width - Property for controlling the component's width.
- FontBold - Property for controlling the component's font weight.
- FontItalic - Property for controlling the component's font style.
- FontSize - Property for controlling the component's font size.
- FontTypeface - Property for controlling the component's font typeface.
- Justification - Property for controlling the component's text justification.
- Text - Property for controlling the component's text.
- TextColor - Property for controlling the component's text color.
- Enabled - Property for controlling whether the component is enabled.

Initialize

```
Event Initialize()
```

Event raised upon component initialization.

GotFocus

```
Event GotFocus()
```

Event raised after the component gains focus.

LostFocus

```
Event LostFocus()
```

Event raised after the component lost focus.

BackgroundColor

```
Property BackgroundColor As Integer
```

Reading from the BackgroundColor property returns the current background color of the component. The color value is encoded as &Haarrggbb where aa represents the alpha value (&H00 - transparent to &HFF - opaque), rr represents the red, gg the green and bb the blue component of the color. Writing to the BackgroundColor property will set the background color of the component.
There are a number of predefined color constants: `Component.COLOR_NONE`, `Component.COLOR_BLACK`, `Component.COLOR_BLUE`, `Component.COLOR_CYAN`, `Component.COLOR_DKGRAY`, `Component.COLOR_GRAY`, `Component.COLOR_GREEN`, `Component.COLOR_LTGRAY`, `Component.COLOR_MAGENTA`, `Component.COLOR_RED`, `Component.COLOR_WHITE` and `Component.COLOR_YELLOW`.

Column

```
Property Column As Integer
```

Reading from the Column property returns the current column position within a table layout. Writing to the Column property will set the column position of the component within a table layout. This property has no meaning for any other layout.

Height

```
Property Height As Integer
```

Reading from the Height property returns the current height of the component in pixels.

Writing to the Height property changes the height of the component to the given value in pixels. There are two special values. `Component.LENGTH_PREFERRED` sets the preferred height of the component which depends on the contents of the component. `Component.LENGTH_FILL_PARENT` sets the height of the component to its maximum to fill the height of its parent container.

Row

```
Property Row As Integer
```

Reading from the Row property returns the current row position within a table layout. Writing to the Row property will set the row position of the component within a table layout. This property has no meaning for any other layout.

Width

```
Property Width As Integer
```

Reading from the Width property returns the current width of the component in pixels. Writing to the Width property changes the width of the component to the given value in pixels. There are two special values. `Component.LENGTH_PREFERRED` sets the preferred width of the component which depends on the contents of the component. `Component.LENGTH_FILL_PARENT` sets the width of the component to its maximum to fill the width of its parent container.

FontBold

```
Property FontBold As Boolean
```

Reading from this property indicates the font weight. A value of `True` means that the component font is bold, `False` means normal. Writing to this property changes the changes the font weight.
The default value of this property is `False`.

FontItalic

```
Property FontItalic As Boolean
```

Reading from this property indicates the font style. A value of `True` means that the component font is italic, `False` means normal. Writing to this property changes the changes the font style.
The default value of this property is `False`.

FontSize

```
Property FontSize As Single
```

Reading from this property returns the font height in points. Writing to this property changes the changes the font height.
The default value of this property is 14 points.

FontTypeface

```
Property FontTypeface As Integer
```

Reading from this property returns the font typeface. The value must be one of `Component.TYPEFACE_DEFAULT`, `Component.TYPEFACE_SERIF`, `Component.TYPEFACE_SANSSERIF` or `Component.TYPEFACE_MONOSPACE`. Writing to this property changes the changes the font typeface.
The default value of this property is `Component.TYPEFACE_DEFAULT`.

Justification

```
Property Justification As Integer
```

Reading from this property returns the text justification. The value must be one of `Component.JUSTIFY_LEFT`, `Component.JUSTIFY_CENTER` or `Component.JUSTIFY_RIGHT`. Writing to this property changes the changes the text justification.

The default value of this property is `Component.JUSTIFY_LEFT`.

Text

```
Property Text As String
```

Reading from this property returns the text displayed by the component. Writing to this property changes the changes the text displayed.

TextColor

```
Property TextColor As Integer
```

Reading from the TextColor property returns the current color of the text displayed by this component. The color value is encoded as &Haarrggbb where aa represents the alpha value (&H00 - transparent to &HFF - opaque), rr represents the red, gg the green and bb the blue component of the color. Writing to the TextColor property will set the color for the text of the component.
There are a number of predefined color constants: `Component.COLOR_NONE`, `Component.COLOR_BLACK`, `Component.COLOR_BLUE`, `Component.COLOR_CYAN`, `Component.COLOR_DKGRAY`, `Component.COLOR_GRAY`, `Component.COLOR_GREEN`, `Component.COLOR_LTGRAY`, `Component.COLOR_MAGENTA`, `Component.COLOR_RED`, `Component.COLOR_WHITE` and `Component.COLOR_YELLOW`.

Enabled

```
Property Enabled As Boolean
```

Reading from the Enabled property indicates whether the email picker is enabled. Writing to the Enabled property will enabled or disable the email picker.
Email pickers are enabled by default.

namespace com.google.devtools.simple.runtime.components

Form

Form is the base object of all forms defined by applications. A form is the root container for all components on it.

Events

- Initialize - Initialization event.
- Keyboard - Keyboard input event.
- MenuSelected - Menu selection event.
- TouchGesture - Touch gesture event.

Properties

- BackgroundColor - Property for controlling the form's background color.
- Height - Property for reading the forms's height.
- Width - Property for reading the forms's width.
- BackgroundImage - Property for controlling the form's background image.
- Layout - Property for controlling the form's layout.
- Scrollable - Property for controlling whether the content is scrollable.
- Title - Property for controlling the form's title.

Initialize

```
Event Initialize()
```

Event raised upon form initialization. Inside of an event handler for this event is the best place for adding components dynamically to a form. For more information of dynamic forms see How To Write A Simple Application.

Keyboard

```
Event Keyboard(keycode As Integer)
```

Event raised after keyboard input.

Parameters:

- `keycode` - constant identifying pressed key (one of
  `Component.KEYCODE_0, Component.KEYCODE_1,`
  `Component.KEYCODE_2, Component.KEYCODE_3,`
  `Component.KEYCODE_4, Component.KEYCODE_5,`
  `Component.KEYCODE_6, Component.KEYCODE_7,`
  `Component.KEYCODE_8, Component.KEYCODE_9,`
  `Component.KEYCODE_A, Component.KEYCODE_APOSTROPHE,`
  `Component.KEYCODE_AT, Component.KEYCODE_B,`
  `Component.KEYCODE_BACK, Component.KEYCODE_BACKSLASH,`
  `Component.KEYCODE_C, Component.KEYCODE_CALL,`
  `Component.KEYCODE_CAMERA, Component.KEYCODE_CLEAR,`
  `Component.KEYCODE_COMMA, Component.KEYCODE_D,`
  `Component.KEYCODE_DEL, Component.KEYCODE_E,`
  `Component.KEYCODE_ENDCALL, Component.KEYCODE_ENTER,`
  `Component.KEYCODE_ENVELOPE, Component.KEYCODE_EQUALS,`
  `Component.KEYCODE_EXPLORER, Component.KEYCODE_F,`
  `Component.KEYCODE_FOCUS, Component.KEYCODE_G,`
  `Component.KEYCODE_GRAVE, Component.KEYCODE_H,`
  `Component.KEYCODE_HEADSETHOOK, Component.KEYCODE_HOME,`
  `Component.KEYCODE_I, Component.KEYCODE_J,`
  `Component.KEYCODE_K, Component.KEYCODE_L,`
  `Component.KEYCODE_LEFT, Component.KEYCODE_LEFT_ALT,`
  `Component.KEYCODE_LEFT_BRACKET, Component.KEYCODE_LEFT_SHIFT,`
  `Component.KEYCODE_M, Component.KEYCODE_MEDIA_FAST_FORWARD,`
  `Component.KEYCODE_MEDIA_NEXT,`
  `Component.KEYCODE_MEDIA_PLAY_PAUSE,`
  `Component.KEYCODE_MEDIA_PREVIOUS,`
  `Component.KEYCODE_MEDIA_REWIND, Component.KEYCODE_MEDIA_STOP,`
  `Component.KEYCODE_MENU, Component.KEYCODE_MINUS,`
  `Component.KEYCODE_MUTE, Component.KEYCODE_N,`
  `Component.KEYCODE_NOTIFICATION, Component.KEYCODE_NUM,`
  `Component.KEYCODE_O, Component.KEYCODE_P,`
  `Component.KEYCODE_PAD_CENTER, Component.KEYCODE_PAD_DOWN,`
  `Component.KEYCODE_PAD_LEFT, Component.KEYCODE_PAD_RIGHT,`
  `Component.KEYCODE_PAD_UP, Component.KEYCODE_PERIOD,`
  `Component.KEYCODE_PLUS, Component.KEYCODE_POUND,`
  `Component.KEYCODE_POWER, Component.KEYCODE_Q,`
  `Component.KEYCODE_R, Component.KEYCODE_RIGHT,`
  `Component.KEYCODE_RIGHT_ALT, Component.KEYCODE_RIGHT_BRACKET,`
  `Component.KEYCODE_RIGHT_SHIFT, Component.KEYCODE_S,`
  `Component.KEYCODE_SEARCH, Component.KEYCODE_SEMICOLON,`
  `Component.KEYCODE_SLASH, Component.KEYCODE_SPACE,`
  `Component.KEYCODE_STAR, Component.KEYCODE_SYM,`
  `Component.KEYCODE_T, Component.KEYCODE_TAB,`
  `Component.KEYCODE_U, Component.KEYCODE_V,`
  `Component.KEYCODE_VOLUME_DOWN, Component.KEYCODE_VOLUME_UP,`

Component.KEYCODE_W, Component.KEYCODE_X, Component.KEYCODE_Y or Component.KEYCODE_Z)

MenuSelected

```
Event MenuSelected(caption As String)
```

Event raised after a menu entry was selected.

Parameters:

- caption - string identifying selected menu item

Also see Application.AddMenuItem().

TouchGesture

```
Event TouchGesture(direction As Integer)
```

Event raised after input of a gesture on the touch screen was recognized.

Parameters:

- direction - constant identifying direction of touch gesture (one of Component.TOUCH_DOUBLETAP, Component.TOUCH_FLINGDOWN, Component.TOUCH_FLINGLEFT, Component.TOUCH_FLINGRIGHT, Component.TOUCH_FLINGUP, Component.TOUCH_MOVEDOWN, Component.TOUCH_MOVELEFT, Component.TOUCH_MOVERIGHT, Component.TOUCH_MOVEUP or Component.TOUCH_TAP)

BackgroundColor

```
Property BackgroundColor As Integer
```

Reading from the BackgroundColor property returns the current background color of the form. The color value is encoded as &Haarrggbb where aa represents the alpha value (&H00 - transparent to &HFF - opaque), rr represents the red, gg the green and bb the blue component of the color. Writing to the BackgroundColor property will set the background color of the form.
There are a number of predefined color constants: `Component.COLOR_NONE`, `Component.COLOR_BLACK`, `Component.COLOR_BLUE`, `Component.COLOR_CYAN`, `Component.COLOR_DKGRAY`, `Component.COLOR_GRAY`, `Component.COLOR_GREEN`, `Component.COLOR_LTGRAY`, `Component.COLOR_MAGENTA`, `Component.COLOR_RED`, `Component.COLOR_WHITE` and `Component.COLOR_YELLOW`.
The default background color for forms is `Component.COLOR_WHITE`.

Height

```
Property Height As Integer
```

Reading from the Height property returns the current height of the form in pixels. For forms the Height property is a read-only property.

Width

```
Property Width As Integer
```

Reading from the Width property returns the current width of the form in pixels. For forms the Width property is a read-only property.

BackgroundImage

```
Property BackgroundImage As String
```

Writing to this property changes the background image shown on the component. The value assigned to this property should be the name of a file in the project's assets directory. This property is write-only.

Layout

```
Property Layout As Variant
```

Reading from the Layout property returns the current layout object instance. For more information about layouts see LinearLayout, TableLayout and FrameLayout. Writing to the Layout property changes the layout to a different layout. The following predefined constants can be used: `Component.LAYOUT_LINEAR`, `Component.LAYOUT_TABLE` or `Component.LAYOUT_FRAME`.
Note that once components have been added to the form its layout cannot be changed any longer!

Scrollable

```
Property Scrollable As Boolean
```

Reading from the this property indicates whether the contents of the form are scrollable. Writing to this property will make the contents of the form scrollable in case the components of the form do not fit within the height or width of the form.

Title

```
Property Title As String
```

Reading from the this property returns the title shown at the top of the form. Writing to this property changes the title of the form.

namespace com.google.devtools.simple.runtime.components

Image

Component for displaying images.

Events

- Initialize - Initialization event.

Properties

- BackgroundColor - Property for controlling the component's background color.
- Column - Property for controlling the column position when using a table layout.
- Height - Property for controlling the component's height.
- Row - Property for controlling the row position when using a table layout.
- Width - Property for controlling the component's width.
- Picture - Property for controlling the component's image.

Initialize

```
Event Initialize()
```

Event raised upon component initialization.

BackgroundColor

```
Property BackgroundColor As Integer
```

Reading from the BackgroundColor property returns the current background color of the component. The color value is encoded as &Haarrggbb where aa represents the alpha value (&H00 - transparent to &HFF - opaque), rr represents the red, gg the green and bb the blue component of the color. Writing to the BackgroundColor property will set the background color of the component.
There are a number of predefined color constants: `Component.COLOR_NONE`, `Component.COLOR_BLACK`, `Component.COLOR_BLUE`, `Component.COLOR_CYAN`, `Component.COLOR_DKGRAY`, `Component.COLOR_GRAY`, `Component.COLOR_GREEN`, `Component.COLOR_LTGRAY`, `Component.COLOR_MAGENTA`, `Component.COLOR_RED`, `Component.COLOR_WHITE` and `Component.COLOR_YELLOW`.

Column

```
Property Column As Integer
```

Reading from the Column property returns the current column position within a table layout. Writing to the Column property will set the column position of the component within a table layout. This property has no meaning for any other layout.

Height

```
Property Height As Integer
```

Reading from the Height property returns the current height of the component in pixels. Writing to the Height property changes the height of the component to the given value in pixels. There are two special values. `Component.LENGTH_PREFERRED` sets the preferred height of the component which depends on the contents of the component. `Component.LENGTH_FILL_PARENT` sets the height of the component to its maximum to fill the height of its parent container.

Row

```
Property Row As Integer
```

Reading from the Row property returns the current row position within a table layout. Writing to the Row property will set the row position of the component within a table layout. This property has no meaning for any other layout.

Width

```
Property Width As Integer
```

Reading from the Width property returns the current width of the component in pixels. Writing to the Width property changes the width of the component to the given value in

pixels. There are two special values. `Component.LENGTH_PREFERRED` sets the preferred width of the component which depends on the contents of the component. `Component.LENGTH_FILL_PARENT` sets the width of the component to its maximum to fill the width of its parent container.

Picture

```
Property Picture As String
```

Writing to this property changes the image shown on the component. The value assigned to this property should be the name of a file in the project's assets directory. This property is write-only.

namespace com.google.devtools.simple.runtime.components

Label

Displays a non-editable text.

Events

- Initialize - Initialization event.

Properties

- BackgroundColor - Property for controlling the component's background color.
- Column - Property for controlling the column position when using a table layout.
- Height - Property for controlling the component's height.
- Row - Property for controlling the row position when using a table layout.
- Width - Property for controlling the component's width.
- FontBold - Property for controlling the component's font weight.
- FontItalic - Property for controlling the component's font style.
- FontSize - Property for controlling the component's font size.
- FontTypeface - Property for controlling the component's font typeface.
- Justification - Property for controlling the component's text justification.
- Text - Property for controlling the component's text.
- TextColor - Property for controlling the component's text color.

Initialize

```
Event Initialize()
```

Event raised upon component initialization.

BackgroundColor

```
Property BackgroundColor As Integer
```

Reading from the BackgroundColor property returns the current background color of the component. The color value is encoded as &Haarrggbb where aa represents the alpha value (&H00 - transparent to &HFF - opaque), rr represents the red, gg the green and bb the blue component of the color. Writing to the BackgroundColor property will set the background color of the component.
There are a number of predefined color constants: `Component.COLOR_NONE`,

Component.COLOR_BLACK, Component.COLOR_BLUE, Component.COLOR_CYAN, Component.COLOR_DKGRAY, Component.COLOR_GRAY, Component.COLOR_GREEN, Component.COLOR_LTGRAY, Component.COLOR_MAGENTA, Component.COLOR_RED, Component.COLOR_WHITE and Component.COLOR_YELLOW.

Column

```
Property Column As Integer
```

Reading from the Column property returns the current column position within a table layout. Writing to the Column property will set the column position of the component within a table layout. This property has no meaning for any other layout.

Height

```
Property Height As Integer
```

Reading from the Height property returns the current height of the component in pixels. Writing to the Height property changes the height of the component to the given value in pixels. There are two special values. Component.LENGTH_PREFERRED sets the preferred height of the component which depends on the contents of the component. Component.LENGTH_FILL_PARENT sets the height of the component to its maximum to fill the height of its parent container.

Row

```
Property Row As Integer
```

Reading from the Row property returns the current row position within a table layout. Writing to the Row property will set the row position of the component within a table layout. This property has no meaning for any other layout.

Width

```
Property Width As Integer
```

Reading from the Width property returns the current width of the component in pixels. Writing to the Width property changes the width of the component to the given value in pixels. There are two special values. `Component.LENGTH_PREFERRED` sets the preferred width of the component which depends on the contents of the component. `Component.LENGTH_FILL_PARENT` sets the width of the component to its maximum to fill the width of its parent container.

FontBold

```
Property FontBold As Boolean
```

Reading from this property indicates the font weight. A value of `True` means that the component font is bold, `False` means normal. Writing to this property changes the changes the font weight.
The default value of this property is `False`.

FontItalic

```
Property FontItalic As Boolean
```

Reading from this property indicates the font style. A value of `True` means that the component font is italic, `False` means normal. Writing to this property changes the changes the font style.
The default value of this property is `False`.

FontSize

```
Property FontSize As Single
```

Reading from this property returns the font height in points. Writing to this property

changes the changes the font height.
The default value of this property is 14 points.

FontTypeface

```
Property FontTypeface As Integer
```

Reading from this property returns the font typeface. The value must be one of `Component.TYPEFACE_DEFAULT`, `Component.TYPEFACE_SERIF`, `Component.TYPEFACE_SANSSERIF` or `Component.TYPEFACE_MONOSPACE`. Writing to this property changes the changes the font typeface.
The default value of this property is Component.TYPEFACE_DEFAULT.

Justification

```
Property Justification As Integer
```

Reading from this property returns the text justification. The value must be one of `Component.JUSTIFY_LEFT`, `Component.JUSTIFY_CENTER` or `Component.JUSTIFY_RIGHT`. Writing to this property changes the changes the text justification.
The default value of this property is `Component.JUSTIFY_LEFT`.

Text

```
Property Text As String
```

Reading from this property returns the text displayed by the component. Writing to this property changes the changes the text displayed.

TextColor

```
Property TextColor As Integer
```

Reading from the TextColor property returns the current color of the text displayed by this component. The color value is encoded as &Haarrggbb where aa represents the alpha value (&H00 - transparent to &HFF - opaque), rr represents the red, gg the green and bb the blue component of the color. Writing to the TextColor property will set the color for the text of the component.
There are a number of predefined color constants: `Component.COLOR_NONE`, `Component.COLOR_BLACK`, `Component.COLOR_BLUE`, `Component.COLOR_CYAN`, `Component.COLOR_DKGRAY`, `Component.COLOR_GRAY`, `Component.COLOR_GREEN`, `Component.COLOR_LTGRAY`, `Component.COLOR_MAGENTA`, `Component.COLOR_RED`, `Component.COLOR_WHITE` and `Component.COLOR_YELLOW`.

namespace com.google.devtools.simple.runtime.components

LocationSensor

Sensor that can determines the current location (longitude, latitude, altitude).

Events

- Initialize - Initialization event.
- Changed - Event indicating a change in location.

Properties

- Available - Available property (read-only property).
- Enabled - Enabled property.
- HasAltitude - Property indicating whether the sensor supports altitude information (read-only property).
- Latitude - Latitude property (read-only property).
- Longitude - Longitude property (read-only property).
- Altitude - Altitude property (read-only property).
- CurrentAddress - Property containing the current street address (read-only property).

Initialize

```
Event Initialize()
```

Event raised upon component initialization.

Changed

```
Event Changed(latitude As Double, longitude As Double, altitude As Double)
```

Event raised when the location changes.

Parameters:
- `latitude` - latitude
- `longitude` - longitude

- `altitude` - altitude in feet.

Available

```
Property Available As Boolean
```

This property indicates whether the sensor is available on the device running the application. This property is read-only.

Enabled

```
Property Enabled As Boolean
```

Reading from the Enabled property indicates whether the sensor is generating data. Writing to the Enabled property will turn sensor data generation on or off. Data generation is enabled by default.

HasAltitude

```
Property HasAltitude As Boolean
```

Indicates whether the location sensor provides altitude information. This property is read-only.

Longitude

```
Property Longitude As Double
```

Reading the value of this property returns the most recent longitude value of the device. In order for this property to supply meaningful values, the sensor needs to be available and

enabled. This property is read-only.

Latitude

```
Property Latitude As Double
```

Reading the value of this property returns the most recent latitude value of the device. In order for this property to supply meaningful values, the sensor needs to be available and enabled. This property is read-only.

Altitude

```
Property Altitude As Double
```

Reading the value of this property returns the most recent altitude value of the device. In order for this property to supply meaningful values, the sensor needs to be available and enabled. This property is read-only.

CurrentAddress

```
Property CurrentAddress As String
```

Provides a street address for the current location. If no street address can be found for the current location, an empty string will be returned. In order for this property to supply meaningful values, the sensor needs to be available and enabled. This property is read-only.

namespace com.google.devtools.simple.runtime.components

OrientationSensor

Sensor that can measure absolute orientation in 3 dimensions.

Events

- Initialize - Initialization event.
- OrientationChanged - Event indicating a change in orientation.

Properties

- Available - Available property (read-only property).
- Enabled - Enabled property.
- Yaw - Yaw property (read-only property).
- Pitch - Pitch property (read-only property).
- Roll - Roll property (read-only property).
- Angle - Property indicating the angle of the current device tilt (read-only property).
- Magnitude - Property indicating the magnitude of the current device tilt (read-only property).

Initialize

```
Event Initialize()
```

Event raised upon component initialization.

OrientationChanged

```
Event OrientationChanged(yaw As Single, pitch As Single, roll As Single)
```

Event raised when the orientation in any of the 3 dimensions changes.

Parameters:

- `yaw` - angle between the magnetic north direction and the Y axis, around the Z axis (0 to 359). 0=North, 90=East, 180=South, 270=West

- `pitch` - rotation around X axis (-180 to 180), with positive values when the z-axis moves toward the y-axis.
- `roll` - rotation around Y axis (-90 to 90), with positive values when the x-axis moves away from the z-axis.

Available

```
Property Available As Boolean
```

This property indicates whether the sensor is available on the device running the application. This property is read-only.

Enabled

```
Property Enabled As Boolean
```

Reading from the Enabled property indicates whether the sensor is generating data. Writing to the Enabled property will turn sensor data generation on or off. Data generation is enabled by default.

Pitch

```
Property Pitch As Single
```

Reading the value of this property returns the most recent pitch value of the device. In order for this property to supply meaningful values, the sensor needs to be available and enabled. This property is read-only.

Roll

```
Property Roll As Single
```

Reading the value of this property returns the most recent pitch value of the device. In order for this property to supply meaningful values, the sensor needs to be available and enabled. This property is read-only.

Yaw

```
Property Yaw As Single
```

Reading the value of this property returns the most recent pitch value of the device. In order for this property to supply meaningful values, the sensor needs to be available and enabled. This property is read-only.

Angle

```
Property Angle As Single
```

Reading the value of this property returns the angle in which the device is tilted in degrees. For the magnitude of the tilt, use the Magnitude property. In order for this property to supply meaningful values, the sensor needs to be available and enabled. This property is read-only.

Magnitude

```
Property Magnitude As Single
```

Reading the value of this property returns a number between 0 and 1, inclusive, indicating how far the device is tilted. For the angle of the tilt, use the Angle property. In order for this property to supply meaningful values, the sensor needs to be available and enabled. This property is read-only.

namespace com.google.devtools.simple.runtime.components

Panel

A panel is a container for other components including other nested panels.

Events

- Initialize - Initialization event.

Properties

- BackgroundColor - Property for controlling the component's background color.
- Column - Property for controlling the column position when using a table layout.
- Height - Property for controlling the component's height.
- Row - Property for controlling the row position when using a table layout.
- Width - Property for controlling the component's width.
- Layout - Property for controlling the component's layout.

Initialize

```
Event Initialize()
```

Event raised upon component initialization.

BackgroundColor

```
Property BackgroundColor As Integer
```

Reading from the BackgroundColor property returns the current background color of the component. The color value is encoded as &Haarrggbb where aa represents the alpha value (&H00 - transparent to &HFF - opaque), rr represents the red, gg the green and bb the blue component of the color. Writing to the BackgroundColor property will set the background color of the component.
There are a number of predefined color constants: `Component.COLOR_NONE`, `Component.COLOR_BLACK`, `Component.COLOR_BLUE`, `Component.COLOR_CYAN`, `Component.COLOR_DKGRAY`, `Component.COLOR_GRAY`, `Component.COLOR_GREEN`, `Component.COLOR_LTGRAY`, `Component.COLOR_MAGENTA`, `Component.COLOR_RED`, `Component.COLOR_WHITE` and `Component.COLOR_YELLOW`.

Column

```
Property Column As Integer
```

Reading from the Column property returns the current column position within a table layout. Writing to the Column property will set the column position of the component within a table layout. This property has no meaning for any other layout.

Height

```
Property Height As Integer
```

Reading from the Height property returns the current height of the component in pixels. Writing to the Height property changes the height of the component to the given value in pixels. There are two special values. `Component.LENGTH_PREFERRED` sets the preferred height of the component which depends on the contents of the component. `Component.LENGTH_FILL_PARENT` sets the height of the component to its maximum to fill the height of its parent container.

Row

```
Property Row As Integer
```

Reading from the Row property returns the current row position within a table layout. Writing to the Row property will set the row position of the component within a table layout. This property has no meaning for any other layout.

Width

```
Property Width As Integer
```

Reading from the Width property returns the current width of the component in pixels. Writing to the Width property changes the width of the component to the given value in pixels. There are two special values. `Component.LENGTH_PREFERRED` sets the preferred width of the component which depends on the contents of the component.

`Component.LENGTH_FILL_PARENT` sets the width of the component to its maximum to fill the width of its parent container.

Layout

```
Property Layout As Variant
```

Reading from the Layout property returns the current layout object instance. For more information about layouts see LinearLayout, TableLayout and FrameLayout. Writing to the Layout property changes the layout to a different layout. The following predefined constants can be used: `Component.LAYOUT_LINEAR`, `Component.LAYOUT_TABLE` or `Component.LAYOUT_FRAME`.
Note that once components have been added to the panel its layout cannot be changed any longer!

namespace com.google.devtools.simple.runtime.components

PasswordTextBox

Editable text box for entering passwords.

Events

- Initialize - Initialization event.
- GotFocus - Focus received event.
- LostFocus - Focus lost event.

Properties

- BackgroundColor - Property for controlling the component's background color.
- Column - Property for controlling the column position when using a table layout.
- Height - Property for controlling the component's height.
- Row - Property for controlling the row position when using a table layout.
- Width - Property for controlling the component's width.
- FontBold - Property for controlling the component's font weight.
- FontItalic - Property for controlling the component's font style.
- FontSize - Property for controlling the component's font size.
- FontTypeface - Property for controlling the component's font typeface.
- Justification - Property for controlling the component's text justification.
- Text - Property for controlling the component's text.
- TextColor - Property for controlling the component's text color.
- Enabled - Property for controlling whether the component is enabled.
- Hint - Property for controlling display of a hint.

Initialize

```
Event Initialize()
```

Event raised upon component initialization.

GotFocus

```
Event GotFocus()
```

Event raised after the component gains focus.

LostFocus

```
Event LostFocus()
```

Event raised after the component lost focus.

BackgroundColor

```
Property BackgroundColor As Integer
```

Reading from the BackgroundColor property returns the current background color of the component. The color value is encoded as &Haarrggbb where aa represents the alpha value (&H00 - transparent to &HFF - opaque), rr represents the red, gg the green and bb the blue component of the color. Writing to the BackgroundColor property will set the background color of the component.
There are a number of predefined color constants: `Component.COLOR_NONE`, `Component.COLOR_BLACK`, `Component.COLOR_BLUE`, `Component.COLOR_CYAN`, `Component.COLOR_DKGRAY`, `Component.COLOR_GRAY`, `Component.COLOR_GREEN`, `Component.COLOR_LTGRAY`, `Component.COLOR_MAGENTA`, `Component.COLOR_RED`, `Component.COLOR_WHITE` and `Component.COLOR_YELLOW`.

Column

```
Property Column As Integer
```

Reading from the Column property returns the current column position within a table layout. Writing to the Column property will set the column position of the component within a table layout. This property has no meaning for any other layout.

Height

```
Property Height As Integer
```

Reading from the Height property returns the current height of the component in pixels. Writing to the Height property changes the height of the component to the given value in pixels. There are two special values. `Component.LENGTH_PREFERRED` sets the preferred height of the component which depends on the contents of the component. `Component.LENGTH_FILL_PARENT` sets the height of the component to its maximum to fill the height of its parent container.

Row

```
Property Row As Integer
```

Reading from the Row property returns the current row position within a table layout. Writing to the Row property will set the row position of the component within a table layout. This property has no meaning for any other layout.

Width

```
Property Width As Integer
```

Reading from the Width property returns the current width of the component in pixels. Writing to the Width property changes the width of the component to the given value in pixels. There are two special values. `Component.LENGTH_PREFERRED` sets the preferred width of the component which depends on the contents of the component. `Component.LENGTH_FILL_PARENT` sets the width of the component to its maximum to fill the width of its parent container.

FontBold

```
Property FontBold As Boolean
```

Reading from this property indicates the font weight. A value of `True` means that the component font is bold, `False` means normal. Writing to this property changes the changes the font weight.
The default value of this property is `False`.

FontItalic

```
Property FontItalic As Boolean
```

Reading from this property indicates the font style. A value of `True` means that the component font is italic, `False` means normal. Writing to this property changes the changes the font style.
The default value of this property is `False`.

FontSize

```
Property FontSize As Single
```

Reading from this property returns the font height in points. Writing to this property changes the changes the font height.
The default value of this property is 14 points.

FontTypeface

```
Property FontTypeface As Integer
```

Reading from this property returns the font typeface. The value must be one of `Component.TYPEFACE_DEFAULT`, `Component.TYPEFACE_SERIF`, `Component.TYPEFACE_SANSSERIF` or `Component.TYPEFACE_MONOSPACE`. Writing to this property changes the changes the font typeface.
The default value of this property is `Component.TYPEFACE_DEFAULT`.

Justification

```
Property Justification As Integer
```

Reading from this property returns the text justification. The value must be one of `Component.JUSTIFY_LEFT`, `Component.JUSTIFY_CENTER` or `Component.JUSTIFY_RIGHT`. Writing to this property changes the changes the text justification.

The default value of this property is `Component.JUSTIFY_LEFT`.

Text

```
Property Text As String
```

Reading from this property returns the text displayed by the component. Writing to this property changes the changes the text displayed.

TextColor

```
Property TextColor As Integer
```

Reading from the TextColor property returns the current color of the text displayed by this component. The color value is encoded as &Haarrggbb where aa represents the alpha value (&H00 - transparent to &HFF - opaque), rr represents the red, gg the green and bb the blue component of the color. Writing to the TextColor property will set the color for the text of the component.
There are a number of predefined color constants: `Component.COLOR_NONE`, `Component.COLOR_BLACK`, `Component.COLOR_BLUE`, `Component.COLOR_CYAN`, `Component.COLOR_DKGRAY`, `Component.COLOR_GRAY`, `Component.COLOR_GREEN`, `Component.COLOR_LTGRAY`, `Component.COLOR_MAGENTA`, `Component.COLOR_RED`, `Component.COLOR_WHITE` and `Component.COLOR_YELLOW`.

Enabled

```
Property Enabled As Boolean
```

Reading from the Enabled property indicates whether the password textbox is enabled. Writing to the Enabled property will enabled or disable the password textbox. Password textboxes are enabled by default.

Hint

```
Property Hint As String
```

Reading from the Hint property returns the text of the hint that will be shown for the password textbox. Writing to the Hint property will change the hint being shown for the password textbox.

namespace com.google.devtools.simple.runtime.components

Phone

Component providing phone-related functionality. There should be only one phone component per form.

Events

- Initialize - Initialization event.

Properties

- Available - Available property (read-only property).

Functions

- Call - Places a call to the given phone number.
- Vibrate - Vibrates the phone.

Initialize

```
Event Initialize()
```

Event raised upon component initialization.

Available

```
Property Available As Boolean
```

This property indicates whether the sensor is available on the device running the application. This property is read-only.

Call

```
Sub Call(phoneNumber As String)
```

Places a call to the given phone number.

Parameters:
- `phoneNumber` - phone number in the form of numbers only (no spaces, no dashes etc.)

Vibrate

```
Sub Vibrate(duration As Integer)
```

Vibrates the phone.

Parameters:
- `duration` - duration in milliseconds

namespace com.google.devtools.simple.runtime.components

RadioButton

A radio button is a two-state button that can be checked or unchecked. I can only be used within a panel with a linear layout. Checking a radio button will automatically uncheck any previously checked radio button within the same panel.

Events

- Initialize - Initialization event.
- Changed - Change event.
- GotFocus - Focus received event.
- LostFocus - Focus lost event.

Properties

- BackgroundColor - Property for controlling the component's background color.
- Column - Property for controlling the column position when using a table layout.
- Height - Property for controlling the component's height.
- Row - Property for controlling the row position when using a table layout.
- Width - Property for controlling the component's width.
- FontBold - Property for controlling the component's font weight.
- FontItalic - Property for controlling the component's font style.
- FontSize - Property for controlling the component's font size.
- FontTypeface - Property for controlling the component's font typeface.
- Justification - Property for controlling the component's text justification.
- Text - Property for controlling the component's text.
- TextColor - Property for controlling the component's text color.
- Enabled - Property for controlling whether the component is enabled.
- Value - Property for controlling the component's value.

Initialize

```
Event Initialize()
```

Event raised upon component initialization.

Changed

```
Event Changed()
```

Event raised after the radio button's value changed.

GotFocus

```
Event GotFocus()
```

Event raised after the component gains focus.

LostFocus

```
Event LostFocus()
```

Event raised after the component lost focus.

BackgroundColor

```
Property BackgroundColor As Integer
```

Reading from the BackgroundColor property returns the current background color of the component. The color value is encoded as &Haarrggbb where aa represents the alpha value (&H00 - transparent to &HFF - opaque), rr represents the red, gg the green and bb the blue component of the color. Writing to the BackgroundColor property will set the background color of the component.
There are a number of predefined color constants: `Component.COLOR_NONE`, `Component.COLOR_BLACK`, `Component.COLOR_BLUE`, `Component.COLOR_CYAN`, `Component.COLOR_DKGRAY`, `Component.COLOR_GRAY`, `Component.COLOR_GREEN`, `Component.COLOR_LTGRAY`, `Component.COLOR_MAGENTA`, `Component.COLOR_RED`, `Component.COLOR_WHITE` and `Component.COLOR_YELLOW`.

Column

```
Property Column As Integer
```

Reading from the Column property returns the current column position within a table layout. Writing to the Column property will set the column position of the component within a table layout. This property has no meaning for any other layout.

Height

```
Property Height As Integer
```

Reading from the Height property returns the current height of the component in pixels. Writing to the Height property changes the height of the component to the given value in pixels. There are two special values. `Component.LENGTH_PREFERRED` sets the preferred height of the component which depends on the contents of the component. `Component.LENGTH_FILL_PARENT` sets the height of the component to its maximum to fill the height of its parent container.

Row

```
Property Row As Integer
```

Reading from the Row property returns the current row position within a table layout. Writing to the Row property will set the row position of the component within a table layout. This property has no meaning for any other layout.

Width

```
Property Width As Integer
```

Reading from the Width property returns the current width of the component in pixels. Writing to the Width property changes the width of the component to the given value in pixels. There are two special values. `Component.LENGTH_PREFERRED` sets the preferred width of the component which depends on the contents of the component. `Component.LENGTH_FILL_PARENT` sets the width of the component to its maximum to fill the width of its parent container.

FontBold

```
Property FontBold As Boolean
```

Reading from this property indicates the font weight. A value of `True` means that the component font is bold, `False` means normal. Writing to this property changes the changes the font weight.
The default value of this property is `False`.

FontItalic

```
Property FontItalic As Boolean
```

Reading from this property indicates the font style. A value of `True` means that the component font is italic, `False` means normal. Writing to this property changes the changes the font style.
The default value of this property is `False`.

FontSize

```
Property FontSize As Single
```

Reading from this property returns the font height in points. Writing to this property changes the changes the font height.
The default value of this property is 14 points.

FontTypeface

```
Property FontTypeface As Integer
```

Reading from this property returns the font typeface. The value must be one of `Component.TYPEFACE_DEFAULT`, `Component.TYPEFACE_SERIF`, `Component.TYPEFACE_SANSSERIF` or `Component.TYPEFACE_MONOSPACE`. Writing to this property changes the changes the font typeface.

The default value of this property is `Component.TYPEFACE_DEFAULT`.

Justification

```
Property Justification As Integer
```

Reading from this property returns the text justification. The value must be one of `Component.JUSTIFY_LEFT`, `Component.JUSTIFY_CENTER` or `Component.JUSTIFY_RIGHT`. Writing to this property changes the changes the text justification.
The default value of this property is `Component.JUSTIFY_LEFT`.

Text

```
Property Text As String
```

Reading from this property returns the text displayed by the component. Writing to this property changes the changes the text displayed.

TextColor

```
Property TextColor As Integer
```

Reading from the TextColor property returns the current color of the text displayed by this component. The color value is encoded as &Haarrggbb where aa represents the alpha value (&H00 - transparent to &HFF - opaque), rr represents the red, gg the green and bb the blue component of the color. Writing to the TextColor property will set the color for the text of the component.
There are a number of predefined color constants: `Component.COLOR_NONE`, `Component.COLOR_BLACK`, `Component.COLOR_BLUE`, `Component.COLOR_CYAN`, `Component.COLOR_DKGRAY`, `Component.COLOR_GRAY`, `Component.COLOR_GREEN`, `Component.COLOR_LTGRAY`, `Component.COLOR_MAGENTA`, `Component.COLOR_RED`, `Component.COLOR_WHITE` and `Component.COLOR_YELLOW`.

Enabled

```
Property Enabled As Boolean
```

Reading from the Enabled property indicates whether the radio button is enabled. Writing to the Enabled property will enabled or disable the radio button.
Radio buttons are enabled by default.

Value

```
Property Value As Boolean
```

Reading from the Value property indicates the current state of the radio button. Writing to the Value property will either check or uncheck the radio button.
Radio buttons are unchecked by default.

namespace com.google.devtools.simple.runtime.components

TextBox

Editable text box.

Events

- Initialize - Initialization event.
- GotFocus - Focus received event.
- LostFocus - Focus lost event.
- Validate - Input validation event.

Properties

- BackgroundColor - Property for controlling the component's background color.
- Column - Property for controlling the column position when using a table layout.
- Height - Property for controlling the component's height.
- Row - Property for controlling the row position when using a table layout.
- Width - Property for controlling the component's width.
- FontBold - Property for controlling the component's font weight.
- FontItalic - Property for controlling the component's font style.
- FontSize - Property for controlling the component's font size.
- FontTypeface - Property for controlling the component's font typeface.
- Justification - Property for controlling the component's text justification.
- Text - Property for controlling the component's text.
- TextColor - Property for controlling the component's text color.
- Enabled - Property for controlling whether the component is enabled.
- Hint - Property for controlling display of a hint.

Initialize

```
Event Initialize()
```

Event raised upon component initialization.

GotFocus

```
Event GotFocus()
```

Event raised after the component gains focus.

LostFocus

```
Event LostFocus()
```

Event raised after the component lost focus.

Validate

```
Event Validate(text As String, ByRef accept As Boolean)
```

Event raised after each character input. A handler for this event may check the input text and validate the input by setting the accept reference parameter.

Parameters:

- `text` - proposed content for the text box
- `accept` - indicates whether to accept the input (default value is `True`)

BackgroundColor

```
Property BackgroundColor As Integer
```

Reading from the BackgroundColor property returns the current background color of the component. The color value is encoded as &Haarrggbb where aa represents the alpha value (&H00 - transparent to &HFF - opaque), rr represents the red, gg the green and bb the blue component of the color. Writing to the BackgroundColor property will set the background color of the component.
There are a number of predefined color constants: `Component.COLOR_NONE`, `Component.COLOR_BLACK`, `Component.COLOR_BLUE`, `Component.COLOR_CYAN`, `Component.COLOR_DKGRAY`, `Component.COLOR_GRAY`, `Component.COLOR_GREEN`, `Component.COLOR_LTGRAY`, `Component.COLOR_MAGENTA`, `Component.COLOR_RED`, `Component.COLOR_WHITE` and `Component.COLOR_YELLOW`.

Column

```
Property Column As Integer
```

Reading from the Column property returns the current column position within a table layout. Writing to the Column property will set the column position of the component within a table layout. This property has no meaning for any other layout.

Height

```
Property Height As Integer
```

Reading from the Height property returns the current height of the component in pixels. Writing to the Height property changes the height of the component to the given value in pixels. There are two special values. `Component.LENGTH_PREFERRED` sets the preferred height of the component which depends on the contents of the component. `Component.LENGTH_FILL_PARENT` sets the height of the component to its maximum to fill the height of its parent container.

Row

```
Property Row As Integer
```

Reading from the Row property returns the current row position within a table layout. Writing to the Row property will set the row position of the component within a table layout. This property has no meaning for any other layout.

Width

```
Property Width As Integer
```

Reading from the Width property returns the current width of the component in pixels.

Writing to the Width property changes the width of the component to the given value in pixels. There are two special values. `Component.LENGTH_PREFERRED` sets the preferred width of the component which depends on the contents of the component.
`Component.LENGTH_FILL_PARENT` sets the width of the component to its maximum to fill the width of its parent container.

FontBold

```
Property FontBold As Boolean
```

Reading from this property indicates the font weight. A value of `True` means that the component font is bold, `False` means normal. Writing to this property changes the changes the font weight.
The default value of this property is `False`.

FontItalic

```
Property FontItalic As Boolean
```

Reading from this property indicates the font style. A value of `True` means that the component font is italic, `False` means normal. Writing to this property changes the changes the font style.
The default value of this property is `False`.

FontSize

```
Property FontSize As Single
```

Reading from this property returns the font height in points. Writing to this property changes the changes the font height.
The default value of this property is 14 points.

FontTypeface

```
Property FontTypeface As Integer
```

Reading from this property returns the font typeface. The value must be one of `Component.TYPEFACE_DEFAULT`, `Component.TYPEFACE_SERIF`, `Component.TYPEFACE_SANSSERIF` or `Component.TYPEFACE_MONOSPACE`. Writing to this property changes the changes the font typeface.
The default value of this property is `Component.TYPEFACE_DEFAULT`.

Justification

```
Property Justification As Integer
```

Reading from this property returns the text justification. The value must be one of `Component.JUSTIFY_LEFT`, `Component.JUSTIFY_CENTER` or `Component.JUSTIFY_RIGHT`. Writing to this property changes the changes the text justification.
The default value of this property is `Component.JUSTIFY_LEFT`.

Text

```
Property Text As String
```

Reading from this property returns the text displayed by the component. Writing to this property changes the changes the text displayed.

TextColor

```
Property TextColor As Integer
```

Reading from the TextColor property returns the current color of the text displayed by this component. The color value is encoded as &Haarrggbb where aa represents the alpha value (&H00 - transparent to &HFF - opaque), rr represents the red, gg the green and bb the blue component of the color. Writing to the TextColor property will set the color for the text of

the component.
There are a number of predefined color constants: `Component.COLOR_NONE`, `Component.COLOR_BLACK`, `Component.COLOR_BLUE`, `Component.COLOR_CYAN`, `Component.COLOR_DKGRAY`, `Component.COLOR_GRAY`, `Component.COLOR_GREEN`, `Component.COLOR_LTGRAY`, `Component.COLOR_MAGENTA`, `Component.COLOR_RED`, `Component.COLOR_WHITE` and `Component.COLOR_YELLOW`.

Enabled

```
Property Enabled As Boolean
```

Reading from the Enabled property indicates whether the textbox is enabled. Writing to the Enabled property will enabled or disable the textbox.
Textboxes are enabled by default.

Hint

```
Property Hint As String
```

Reading from the Hint property returns the text of the hint that will be shown for the textbox. Writing to the Hint property will change the hint being shown for the textbox.

namespace com.google.devtools.simple.runtime.components

Timer

Component providing timer functionality.

Events

- Initialize - Initialization event.
- Timer - Timer expiration event.

Properties

- Enabled - Enabled property.
- Interval - Timer interval property.

Initialize

```
Event Initialize()
```

Event raised upon component initialization.

Timer

```
Event Timer()
```

Event raised upon Timer expiration. After completing an event handler for this event, the timer will be reset to the current interval value and restarted (unless it was disabled).

Enabled

```
Property Enabled As Boolean
```

Reading from the Enabled property indicates whether the timer is running and will be restarted after interval expiration. Writing to the Enabled property will turn timer on or off.

The timer is enabled by default.

Interval

```
Property Interval As Integer
```

Reading from the Interval property returns the number of milliseconds between timer events. Writing to the Interval property will changed the length of the interval between timer events. If the timer is enabled and the interval is being changed then the current timer run is aborted and the timer is immediately restarted with the new interval.
The default interval is 1000 ms.

namespace com.google.devtools.simple.runtime.components

FrameLayout

Layout for prominently showing a single component. If there are multiple components in the container then only the component last added will be shown.

namespace com.google.devtools.simple.runtime.components

LinearLayout

Layout for placing components horizontally or vertically. When choosing horizontal orientation, the components will wrap vertically if the width of the container is exceeded.

Properties

- Orientation - Property controlling the layout orientation.

Orientation

```
Property Orientation As Integer
```

This property sets the orientation of the linear layout. The assigned value must be either `Component.LAYOUT_ORIENTATION_HORIZONTAL` or `Component.LAYOUT_ORIENTATION_VERTICAL`. This property is write-only.

namespace com.google.devtools.simple.runtime.components

TableLayout

Layout for placing components in tabular form.

Properties

- Columns - Property controlling the number of columns.
- Rows - Property controlling the number of rows.

Columns

```
Property Columns As Integer
```

This property sets the number of columns used by the layout. The assigned value must be greater than zero. This property is write-only.

Rows

```
Property Rows As Integer
```

This property sets the number of rows used by the layout. The assigned value must be greater than zero. This property is write-only.