The current list of supported System Methods and Functions available from KPL code, divided by category of function:

MATH FUNCTIONS

Function Abs (Value AS INT) AS Int
Returns the absolute (positive) value of the integral number stored in ***Value***

Function Acos (d AS Decimal) AS Decimal
Returns the arccosine or inverse cosine in degrees for the cosine value specified in ***d***

Function ArcTan (d AS Decimal) AS Decimal
Returns the angle whose tangent is specified in ***d***

Function ArcTan2 (X AS Decimal, Y as Decimal) AS Decimal
Returns the angle in degrees that is the arctangent of ***Y/X***

Function Asin **(** d **AS Decimal) AS Decimal**
Returns the angle in degrees that is the arcsine of the value in ***d***

Function Ceiling (d AS Decimal) AS Decimal
Returns the smallest integer that is not smaller than the value ***d***

Function Cos (d AS DECIMAL) AS Decimal
Returns the cosine of the angle specified in ***d***

Function CosH (d AS DECIMAL) AS Decimal
Returns the hyperbolic cosine of the value specified in ***d***

Function Exp (d AS DECIMAL) AS Decimal
Returns ***e*** *raised to the power* ***d***

Function Floor (d AS Decimal) AS Decimal
Returns the largest integer that is not larger than the value ***d***

Function Log (d AS DECIMAL) AS Decimal
Returns the natural (base ***e****) logarithm of the number specified in* ***d***

Function Log10 (d AS DECIMAL) AS Decimal
Returns the base ***10*** *logarithm of the number specified in* ***d***

Function LogBase (value AS DECIMAL, base AS DECIMAL) AS Decimal
Returns the logarithm of the ***value*** *in the base of* ***base***

Function Max (ValueA AS Any, ValueB as Any) AS Any
Returns whichever of ***ValueA*** *or* ***ValueB*** *is the larger value*

Function Min (ValueA AS Any, ValueB as Any) AS Any
*Returns whichever of **ValueA** or **ValueB** is the smaller value*

Function Power (value AS DECIMAL, power AS DECIMAL) AS Decimal
*Returns the result of **value** raised to the power of **power***

Function Random (MIN AS INT, MAX AS INT) AS Int
*Returns a random Integer (whole) number in the range from **MIN** to **MAX**, inclusive. For example: **Random (1, 10)** returns a random number between 1 and 10.*

Function Round (d AS Decimal) AS Decimal
*Returns the integer that is closest to the value **d***

Function RoundToPlace (d AS Decimal, decimalPlaces As Decimal) AS Decimal
*Rounds a decimal **d** to **decimalPlaces** decimal places.*

Function Sin (d AS DECIMAL) AS DECIMAL
*Returns the sine of the angle specified in **d***

Function Sqrt (d AS DECIMAL) AS DECIMAL
Returns the square root of **d**

Function Tan (d AS DECIMAL) AS DECIMAL
*Returns the tangent of the angle specified in **d***

Function TanH (d AS DECIMAL) AS DECIMAL
*Returns the hyperbolic tangent of the value specified in **d***

Function TickCount ( ) AS DECIMAL
Returns the number of seconds since KPL started running – useful as a timer

DATA FUNCTIONS

Function ArrayLength (List as ARRAY) as INT
*Returns the length of the array stored in **List***

Function ConvertToBool (value as ANY) as BOOL
*Converts **value** to a BOOL*

Function ConvertToDecimal (value as ANY) as DECIMAL
*Converts **value** to a DECIMAL*

Function ConvertToInt (value as ANY) as INT
*Converts **value** to an INT*

Function ConvertToString (value as ANY) as STRING
*Converts **value** to a STRING*

Function FormatString (FormatString AS STRING, Value AS ANY) AS STRING

Function IndexOf (Value AS STRING, Pattern AS STRING) AS INT
*Returns the position in **Value** of the first occurrence of the string stored in **Pattern***

Function ParseInt (Value AS STRING) AS INT
*Converts the string stored in **Value** into an Integer value*

Function Split (Text AS STRING, Delimiter AS STRING) AS ARRAY
*Splits the string stored in **Text** into an array of strings. Splits **Text** at each occurrence of **Delimiter***

Function StringLength (Text AS STRING) AS INT
Returns the length of the string stored in **Text**

Function StringReplace (Text AS STRING, Pattern AS STRING, NewPattern AS STRING) AS STRING
*Replaces any occurrence within **Text** of the string in **Pattern** with the string in **NewPattern***

Function Substring (Text AS STRING, StartPos AS INT, Length AS INT) AS STRING
*Returns the part of **Text** starting at **StartPos**, to a maximum of **Length** letters*

Function ToLower (Value AS STRING) AS STRING
Converts the string stored in to all lowercase letters

Function ToUpper (Value AS STRING) AS STRING
*Converts the string stored in **Value** to all uppercase letters*

DATE FUNCTIONS

Function AddDays (Date as String, NumberOfDays as Int) as String
*Returns result of **Date** value plus **NumberOfDays** days*

Function AddHours (Date as String, NumberOfHours as Int) as String
*Returns result of **Date** value plus **NumberOfHours** hours*

Function AddMinutes (Date as String, NumberOfMinutes as Int) as String
*Returns result of **Date** value plus **NumberOfMinutes** minutes*

Function AddMonths (Date as String, NumberOfMonths as Int) as String
*Returns result of **Date** value plus **NumberOfMonths** months*

Function AddSeconds (Date as String, NumberOfSeconds as Int) as String
*Returns result of **Date** value plus **NumberOfSeconds** seconds*

Function AddYears (Date as String, NumberOfYears as Int) as String
*Returns result of **Date** value plus **NumberOfYears** years*

Function DateDifference (Date1 as String, Date2 as String) as Int
Returns an integer value representing the number of days difference between two date strings

Function Day (Date as String) as INT
Returns day of month integer from a Date string

Function DayName (DayOfWeek as Int, Abbreviated as Bool) as String
Returns local-language-specific day of week name, either abbreviated or not

Function DayOfWeek (Date as String) as Int
*Returns numeric day of week based on input **Date** string*

Function DayOfYear (Date as String) as Int
*Returns numeric day of year based on input **Date** string*

Function DaysInMonth (Year as Int, Month as Int) as Int
Returns number of days in the specified Month of the specified Year

Function FirstDayOfWeek () as Int
Returns the index of the first day of the week, based on local culture settings

Function FormatDate (Year As Int, Month as Int, Day as Int, UseLongFormat as Bool) As String
Returns a long or short format of a Date string, using the local language and culture settings

Function FormatTime (Hour As Int, Minute as Int, Second as Int, UseLongFormat as Bool) As String
Returns a long or short format of a Time string, using the local language and culture settings

Function Hour (Time as String) as INT
Returns hour integer from a Time string

Function IsLeapDay (Date as String) as Bool
*Returns **True** if specified **Date** is a leap day*

Function IsLeapMonth (Date as String) as Bool
*Returns **True** if specified **Date** is in a leap month*

Function IsLeapYear (Date as String) as Bool
*Returns **True** if specified **Date** is in a leap year*

Function Minute (Time as String) as Int
Returns minute integer from a Time string

Function Month (Date as String) as Int
Returns month integer from a Date string

Function MonthName (Month as Int, Abbreviated as Bool) as String
Returns local-language-specific month name, either abbreviated or not

Function Second (Time as String) as Int
Returns second integer from a Time string

Function TimeDifference (Time1 as String, Time2 as String) as String
Returns a time string representing the difference between two time strings, in hours, minutes and seconds

Function TimeNow () as String
Returns a time string specifying the time of day right now

Function Today () as String
Returns a date string specifying today's date

Function WeekOfYear (Date as String) as Int
Returns week of the year integer from a Date string

Function Year (Date as String) as Int
Returns year integer from a Date string

KEYBOARD FUNCTIONS

Function Confirm(Message As STRING, Title As STRING) As BOOL

Function GetKey ( ) AS STRING
Returns the last key that the user pressed, or an empty string ("") if no key has been pressed.

Function IsKeyDown (Key AS STRING) AS BOOL
*Returns **True** if the named **Key** is currently being pressed by the user. For example, **IsKeyDown("Right")** will return **True** if the user is pressing the right arrow key*

CONSOLE FUNCTIONS

Function ConsoleReadDecimal(Prompt As String, EchoToConsole As Bool) As DECIMAL

Function ConsoleReadInt(Prompt As String, EchoToConsole As Bool) As INT

Function ConsoleReadKey(EchoToConsole As Bool) As STRING

Function ConsoleReadLine(Prompt As String, EchoToConsole As Bool) As STRING

CONSOLE METHODS

Method ConsoleWrite(Message As STRING)

Method ConsoleWriteLine(Message As STRING)

Method ClearConsole( )

Method HideConsole( )

Method MaximizeConsole( )

Method SetConsoleBackgroundColor(Color As INT)

Method SetConsoleBulletMode(IsBulletModeOn As BOOL)

Method SetConsoleFont(Name As STRING, Size As INT)

Method SetConsoleFontColor(Color As INT)

Method SetConsoleFontSize(Size As DECIMAL)

Method SetConsoleFontStyle(Bold As BOOL, Italic As BOOL, Underline As BOOL)

Method SetConsoleHangingIndent(Indent As INT)

Method SetConsoleHeight(Height As INT)

Method SetConsoleIndent(Indent As INT)

Method SetConsoleTextAlignment(Alignment As STRING)

Method ShowConsole( )

SCREEN FUNCTIONS

Function ScreenHeight ( ) AS INT
Returns the height of the KPL screen, in pixels

Function ScreenWidth ( ) AS INT
Returns the width of the KPL screen, in pixels

SCREEN METHODS

Method BeginFrame ( )
Causes KPL to not refresh the screen until RefreshScreen() is called

Method Clear ( )
This method clears the screen

Method Delay (Milliseconds AS INT)
*Causes the running KPL program to wait for **Milliseconds** milliseconds.*

Method Circle (Size AS INT, Filled AS BOOL)
Draws a circle on screen using the current pen location and pen color

Method Maximize ( )
Makes the running KPL program take up the whole screen

Method Print (text AS STRING)
*Prints the contents of **text** at the location of the drawing pen, using the current pen color. Numeric variables, if passed to this method, will be automatically formatted and printed as strings.*

Method PrintInRect (text AS STRING, X AS INT, Y AS INT, Width AS INT, Height AS INT)
*Prints the contents of **text** withing the specified rectangle. Words will be wrapped to fit horizontally, and cropped vertically if they go beyond the bottom of the rectangle. Numeric variables, if passed to this method, will be automatically formatted and printed as strings.*

Method PrintLine (text AS STRING)
*Prints the contents of **text** at the location of the drawing pen, using the current pen color, then adds a **Carraige Return** to move the pen to the next line. Numeric variables, if passed to this method, will be automatically formatted and printed as strings.*

Method RefreshScreen ( )
Causes KPL to refresh all sprites and graphics on the screen

Method ScrollBackground (X AS INT, Y AS INT, Width AS INT, Height AS INT, ScrollX AS INT, ScrollY AS INT)
*Scrolls a rectangular background region defined by **X, Y** as the upper left corner, and the specified **Width** and **Height**. **ScrollX** and **ScrollY** specify the amount to scroll along each axis.*

Method SetAlgebraCoordinates ( )
Causes KPL to use algebraic coordinates, with the (X, Y) origin of (0, 0) in the center the KPL window

Method SetDeviceCoordinates ( )
Causes KPL to use computer device coordinates (X, Y) origin of (0, 0) in the upper left of the KPL window

Method SetFont (FontName AS STRING, Size AS INT, Bold AS BOOL, Italic AS BOOL, Underline AS BOOL)
Sets the current font used for printing text

Method SetScreenSize (Width as INT, Height as INT)
*Causes KPL to set the runtime screen size to **Width** and **Height***

Method Status (Text AS STRING)
*Displays the string value stored in **Text** in the status bar at the bottom of the running program's window*

Method TileBitmap (SpriteName AS STRING, X AS INT, Y AS INT, Width AS INT, Height AS INT, SourceX AS INT, SourceY AS INT)

Method TileSprite (SpriteName AS STRING, X AS INT, Y AS INT, Width AS INT, Height AS INT, OffsetX AS INT, OffsetY AS INT)

PEN METHODS

Method Circle (Diameter AS INT, Filled AS BOOL)
Draws an circle on screen using the current pen location and pen color

Method Color (Color AS INT)
Sets the color of the drawing pen using color constants, which can be specified either by name or by number as shown. Examples: Color (Blue) or Color (10).

Transparent = 0
AliceBlue = 1
AntiqueWhite = 2
Aqua = 3
Aquamarine = 4
Azure = 5
Beige = 6
Bisque = 7
Black = 8
BlanchedAlmond = 9
Blue = 10
BlueViolet = 11
Brown = 12
BurlyWood = 13
CadetBlue = 14
Chartreuse = 15
Chocolate = 16
Coral = 17
CornflowerBlue = 18
Cornsilk = 19
Crimson = 20
Cyan = 21
DarkBlue = 22
DarkCyan = 23
DarkGoldenrod = 24
DarkGray = 25
DarkGreen = 26
DarkKhaki = 27
DarkMagenta = 28
DarkOliveGreen = 29
DarkOrange = 30
DarkOrchid = 31
DarkRed = 32
DarkSalmon = 33
DarkSeaGreen = 34
DarkSlateBlue = 35
DarkSlateGray = 36
DarkTurquoise = 37
DarkViolet = 38
DeepPink = 39
DeepSkyBlue = 40
DimGray = 41
DodgerBlue = 42
Firebrick = 43
FloralWhite = 44
ForestGreen = 45
Fuchsia = 46
Gainsboro = 47
GhostWhite = 48
Gold = 49
Goldenrod = 50
Gray = 51
Green = 52
GreenYellow = 53
Honeydew = 54
HotPink = 55
IndianRed = 56
Indigo = 57
Ivory = 58
Khaki = 59
Lavender = 60
LavenderBlush = 61
LawnGreen = 62
LemonChiffon = 63
LightBlue = 64
LightCoral = 65
LightCyan = 66
LightGoldenrodYellow = 67
LightGreen = 68
LightGray = 69
LightPink = 70
LightSalmon = 71
LightSeaGreen = 72
LightSkyBlue = 73
LightSlateGray = 74
LightSteelBlue = 75
LightYellow = 76
Lime = 77
LimeGreen = 78
Linen = 79
Magenta = 80
Maroon = 81
MediumAquamarine = 82
MediumBlue = 83
MediumOrchid = 84
MediumPurple = 85
MediumSeaGreen = 86
MediumSlateBlue = 87
MediumSpringGreen = 88
MediumTurquoise = 89
MediumVioletRed = 90
MidnightBlue = 91
MintCream = 92
MistyRose = 93
Moccasin = 94
NavajoWhite = 95
Navy = 96
OldLace = 97
Olive = 98
OliveDrab = 99
Orange = 100
OrangeRed = 101
Orchid = 102
PaleGoldenrod = 103
PaleGreen = 104
PaleTurquoise = 105
PaleVioletRed = 106
PapayaWhip = 107
PeachPuff = 108
Peru = 109
Pink = 110
Plum = 111
PowderBlue = 112
Purple = 113
Red = 114
RosyBrown = 115
RoyalBlue = 116
SaddleBrown = 117
Salmon = 118
SandyBrown = 119
SeaGreen = 120
SeaShell = 121
Sienna = 122
Silver = 123
SkyBlue = 124
SlateBlue = 125

SlateGray = 126
Snow = 127
SpringGreen = 128
SteelBlue = 129
Tan = 130
Teal = 131
Thistle = 132
Tomato = 133
Turquoise = 134
Violet = 135
Wheat = 136
White = 137
WhiteSmoke = 138
Yellow = 139
YellowGreen = 140

Method ColorRGB (Red AS INT, Blue AS INT, Green AS INT)
Sets the color of the drawing pen using combined intensities of Red, Green and Blue. Examples: ColorRGB (0, 0, 0) is black, and ColorRGB (255, 255, 255) is white.

Function DrawLine *(StartX AS INT, StartY AS INT) to (EndX AS INT, EndY AS INT)*
*Draws a line on the screen from (**StartX**,**StartY**) to (**EndX**,**EndY**), in the current pen color*

Method Ellipse (Width AS INT, Height AS INT, Filled AS BOOL)
Draws an ellipse on screen using the current pen location and pen color

Method MoveBy (DeltaX AS INT, DeltaY AS INT)
Moves the drawing pen/cursor by the specified amounts along the x and y axes

Method MoveTo (X AS INT, Y AS INT)
*Moves the drawing pen/cursor to the (**X**, **Y**) location*

Method Pen (IsPenDown AS BOOL)
Controls whether the drawing pen is 'down' and thus whether it will draw as it is moved

Method PenWidth (PenWidth AS INT)
Specified the drawing pen's drawing width in pixels

Method PutPixel (X AS DECIMAL, Y AS DECIMAL)
*Puts a pixel on the screen at the coordinates (**X**, **Y**), in the current pen color*

Method Rectangle (Width AS INT, Height AS INT, Filled AS BOOL)
*Draws a rectangle in the current pen color, starting at the current pen position, **Width** pixels wide and **Height** pixels high. If **Filled** is **True**, the rectangle will be filled.*

SPRITE FUNCTIONS

Function GetSpriteFrameCount (SpriteName AS STRING) AS INT
Returns the number of frames that can be shown for the sprite

Function GetSpriteHeight (SpriteName AS STRING) AS INT
Returns the current height of the named sprite's bounding box

Function GetSpriteLeft (SpriteName AS STRING) AS INT
Returns the X position of the named sprite's upper-left corner

Function GetSpritesThatIntersectWith (SpriteName AS STRING) AS UNKNOWN[]
Returns a string array with the names of all sprites that are touching (intersecting) with the named sprite

Function GetSpriteTop (SpriteName AS STRING) AS INT
Returns the Y position of the named sprite's upper-left corner

Function GetSpriteWidth (SpriteName AS STRING) AS INT
Returns the current width of the named sprite's bounding box

Function SpriteContainsPoint (SpriteName AS STRING, X AS INT, Y AS INT) AS BOOL
Returns ***True*** *if the point specified by (****X,Y****) is occupied by the sprite named by* ***SpriteName***

Function SpritesIntersect (SpriteName1 AS STRING, SpriteName2 AS STRING) AS BOOL

SPRITE METHODS

Method ClearSprites ( )
Clears all sprites from the screen

Method HideSprite (SpriteName AS STRING)
Causes the named sprite to no longer be visible

Method FlipSpriteHorizontally (SpriteName AS STRING)
Flips the displayed Sprite image, right to left

Method FlipSpriteVertically (SpriteName AS STRING)
Flips the displayed Sprite image, top to bottom

Method LoadSprite (SpriteName AS STRING, FileName AS STRING)
Creates a new sprite whose name is specified by ***SpriteName****. The sprite's picture will be loaded from* ***Filename****.*

Method MoveSpriteBy (SpriteName AS STRING, DeltaX AS INT, DeltaY AS INT)
Moves the named sprite by adding ***DeltaX*** *to the sprite's current X value, and adding* ***DeltaY*** *to the sprite's current Y value.*

Method MoveSpriteTo (SpriteName AS STRING, X AS INT, Y AS INT)
*Moves the named sprite so that it's upper-left corner will be at (****X****,* ***Y****)*

Method MoveSpriteToPoint (SpriteName AS STRING, X AS INT, Y AS INT)
*Moves the named sprite so that it's upper-left corner will be at (****X****,* ***Y****)*

Method RotateSprite (SpriteName AS STRING, RotationAmount AS STRING)
Rotates the named sprite. ***RotationAmount*** *is the angle (in degrees) to rotate the sprite*

Method RotateSpriteBy (SpriteName AS STRING, AmountOfChange AS STRING)
Rotates the named sprite by changing it's current rotation by the amount specified in ***AmountOfChange****, which is in degrees*

Method ScaleSprite (SpriteName AS STRING, ScaleFactor AS DECIMAL)
Sets named sprite's size by scaling it to the indicated ***ScaleFactor*** *(as a percentage)*

Method SetSpriteActiveFrame (SpriteName AS STRING, FrameNumber AS INT)

Sets the active frame for the sprite. ***FrameNumber*** *must be less than or equal to the number of available frames*

Method SetSpriteAnimationTimeline (SpriteName AS STRING, AutoRewind AS BOOL, Timeline AS ARRAY)
Begins automatic animation of the named sprite by setting an **INT[]** array, where each element of the array specifies how long to show the frame with the same index as the element

Method SetSpriteCanCollide (SpriteName AS STRING, CanCollide AS BOOL)
Sets whether the named sprite can collide (intersect) with other sprites. This is useful for background sprites where collision information is not needed

Method SetSpriteClip (X AS INT, Y AS INT, Width AS INT, Height AS INT)
Sets a clipping region for the sprite which prevents any of it from being drawn outside of the specified rectangle

Method SetSpriteOpacity (SpriteName AS STRING, OpacityPercent AS INT)
Sets how transparent the named sprite is. An ***OpacityPercent*** *of 100 means that the sprite is fully visible, and a value of 0 means that it is completely invisible*

Method SetSpriteZIndex Method)
Sets the names sprite's Z-Index, which is the order in which a sprite is drawn. A sprite with a higher Z-Index will always be drawn on top of sprites with a lower Z-Index

Method ShowSprite (SpriteName AS STRING)
Makes the sprite whose name is specified by ***SpriteName*** *visible*

Method StampSprite (SpriteName AS STRING)
Causes the named sprite to be drawn permanently on the background canvas

Method StopSpriteAnimation (SpriteName AS STRING)
Stops the named sprite from being automatically animated. Used to stop animation after a call to

Method SetSpriteAnimationTimeline()

Method StretchSprite (SpriteName AS STRING, ScaleFactorX AS DECIMAL, ScaleFactorY AS DECIMAL)
Stetches the displayed image of a sprite using the specified ***ScaleFactors*** *(as percentages)*

Method UnloadSprite (SpriteName AS STRING)
Unloads the sprite whose name is specified by SpriteName

SOUND METHODS

Method PlaySound (FileName AS STRING)
Plays the sound file specified by Filename

TRACE AND RUNTIME METHODS

Method Alert(Message As STRING, Title As STRING)
Pops up a window that can be used to notify the user of something important

Method ClearTrace ()
Clears the ***Program Trace Messages*** *window*

Method ShowTrace ()
Displays the ***Program Trace Messages*** *window*

Method Stop ()
Stops execution of the running program

Method Trace (Text AS STRING)
Sends the string stored in ***Text*** *to the 'Program Trace' window. Use* ***ShowTrace()*** *or press the 'Trace Messages' button to view the trace window*