quasar{Stack}

QUASAR-STACK LIBRARY OFFICIAL DOCUMENTATION

CREATED BY KAYKY VITOR CRUZ

quasarStack is a library focused on simplifying application development by providing quick solutions for data processing and specific modules for generalizing methods. Requires .NET Core 3.1+

DEVLOG [UPDATES/INTERNAL CHANGES/NEW FEATURES]

<table>
<tr><th colspan="2">[04/2021] BUILD 3 (Day 19) – FIRST FULL RELEASE [v0.3]</th></tr>
<tr><td colspan="2">This version has a new error dictionary for debugging the application with support for various types of errors: logic, mathematics, data creation, conversion, internet connection and server connection. The version also has new functions for manipulating data, creating standards and integrating methods for the NetKit module.</td></tr>
<tr><td><b>[NEW] Error Dictionary</b></td><td>Error dictionary to improve application development with quasarStack. Consult the "Error Dictionary" at the end of the documentation</td></tr>
<tr><td><b>[NEW] quasarStack.Data.Standardify Module</b></td><td>Module focused on creating automated standards to facilitate data operations</td></tr>
<tr><td><b>[NEW] EasyData.StringArrJoin()</b></td><td>Method that joins two arrays of strings into one</td></tr>
<tr><td><b>[NEW] EasyData.IntArrJoin()</b></td><td>Method that joins two arrays of integers into one</td></tr>
<tr><td><b>[NEW] EasyData.LongArrJoin()</b></td><td>Method that joins two arrays of longs into one</td></tr>
<tr><td><b>[NEW] EasyData.StringLoop()</b></td><td>Create an array looping the desired string</td></tr>
<tr><td><b>[NEW] EasyData.IntLoop()</b></td><td>Create an array looping the desired integer</td></tr>
<tr><td><b>[NEW] NetKit.PublicIP()</b></td><td>Returns the public IP of the device using the Ipfy.org API</td></tr>
<tr><td><b>[NEW] NetKit.IsWebConnected()</b></td><td>Performs varied tests on generic websites to test if the device is connected to the internet</td></tr>
<tr><td><b>[NEW] NetKit.TestConnection()</b></td><td>Check the connection to a target address</td></tr>
</table>

<table>
<tr><th colspan="2">[04/2021] BUILD 2 (Day 13) – SECONDARY TEST RELEASE [v0.2]</th></tr>
<tr><td colspan="2">This version has specific bug fixes in some methods and the addition of a new module called NetKit and new methods in the EasyData module. <b>The documentation now also clearly presents quasarStack's error management functions.</b></td></tr>
<tr><td><b>[NEW] EasyData.ArrayImp() Method</b></td><td>Implode an array and convert it to a string. Methods: <b>default</b>: standard method / <b>list-array</b>: add line breaks between items</td></tr>
<tr><td><b>[NEW] EasyData.StringExp() Method</b></td><td>Make an array based on a string</td></tr>
<tr><td><b>[CHANGE] EasyData.IsDate() Method</b></td><td>Now supports date common character (-)

(Now the definition method is "DefaultChars" that
determines whether major date characters are required or free analysis is performed with any character).</td></tr>
<tr><td><b>[NEW] quasarStack.NetView Module</b></td><td>-</td></tr>
</table>

<table>
<tr><th>[04/2021] BUILD 1 (Day 12) – INITIAL TEST RELEASE [v0.1]</th></tr>
<tr><td>This version is being made available for testing and has several basic functions for converting, integrating and creating data as well as the "Inventory" function that is being developed in the testing phase with the aim of storing complex data and allowing them to can be integrated directly between other programming languages.</td></tr>
</table>

EasyData Module

The EasyData module contains analysis and data conversion functions with the aim of promoting fast and convenient conversions between types of data that would require manual conversions and could violate the DRY method. Implementation: using quasarStack.Data

EasyData.StringConvert(object args) <RETURNS STRING>	Converts the value contained in the object to a string using the standard method "string.Join"
EasyData.StringConvert(object[] args) <RETURNS STRING>	Parses the values contained in the object array and returns a multi-line string
EasyData.IntConvert(object args) <RETURNS INT>	Parses an object's value from StringConvert and then creates an integer value based on the analyzed string
EasyData.LongConvert(object args) <RETURNS LONG>	Performs the same as the previous function but returns a long number instead of an integer
EasyData.DoubleConvert(object args) <RETURNS DOUBLE>	Converts the values of an object to double from the analysis of the result of its StringConvert counting its division from its first located point
EasyData.FloatConvert(object) <RETURNS FLOAT>	Performs the same action as the previous function but returns a float.
EasyData.FilterArray(object[] array, string content, int method) <RETURNS STRING[]> BugInfo #1: Due to several bugs the default return for this method will be string. This will be resolved in future versions of the library. BugInfo #2: Due to the previous bug and some problems derived, the filtering of numerical arrays should be performed by a specific function for numbers.	Returns the values of an array based on the given method. METHOD 1: Returns the values equal to the one informed. METHOD 2: Returns the values that contain the informed value. METHOD 3: Returns the values that do NOT contain the informed value. METHOD 4: Returns values other than the entered value.
EasyData.FilterIntArray(int[] array, int content) <RETURNS STRING[]>	Filters the numeric array of integers and returns a string with the corresponding numbers.
EasyData.FilterLongArray(long[] array, long content) <RETURNS STRING[]>	Filters the numeric array of longs and returns a string with the corresponding numbers.

EasyData.FilterFloatArray(float[] array, float content) <RETURNS STRING[]>	Filters the numeric array of floats and returns a string with the corresponding numbers.
EasyData.StringPush(string[] array, string content) <RETURNS STRING[]>	Returns the initial string array + the content added to the end of the array.
EasyData.StringStack(string[] array, string content) <RETURNS STRING[]>	Returns the initial string array + the content added to the top of the array.
EasyData.IntPush(int[] array, int content) <RETURNS STRING[]>	Returns the initial int array + the content added to the end of the array.
EasyData.IntStack(int[] array, int content) <RETURNS STRING[]>	Returns the initial int array + the content added to the top of the array.
EasyData.CharSplit(object content) <RETURNS CHAR[]>	Convert the Array of objects to a String Array using the StringConvert and divide it into an Array of characters.
EasyData.IsEmail(string args) <RETURNS BOOL>	Checks whether the informed entry has the e-mail format.
EasyData.IsNumber(object args) <RETURNS BOOL>	Checks whether the object is composed only of numbers.
EasyData.IsNumber(char args) <RETURNS BOOL>	Checks whether the char is a number.
EasyData.IsTel(char args) <RETURNS BOOL>	Checks whether the entry has a valid phone format.
EasyData.IsDate(object args, bool countBars) <RETURNS BOOL>	Checks whether the entry has a valid date format. The countBars (bool) method will check if the method will take forward slash characters in count.
EasyData.IsCPF(string args) <RETURNS BOOL>	Usable only for CPF (Brazilian numbering). Checks if the entered entry is a valid CPF.
EasyData.ArrayImp(object[] array, string method) <RETURNS STRING>	Implode an array in a string. METHOD default: Don’t add line breaks between items. METHOD list-array: Add line breaks between items.
EasyData.StringExp(string toExplode, string baseChar) <RETURNS STRING[]>	Make an array based on a string. baseChar is the split char.
EasyData.StringArrJoin(string[] arr1, string[] arr2)	Join two string arrays to make a new one.

<RETURNS STRING[]>	
EasyData.IntArrJoin(int[] arr1, int[] arr2) <RETURNS INT[]>	Join two int arrays to make a new one.
EasyData.LongArrJoin(long[] arr1, long[] arr2) <RETURNS LONG[]>	Join two long arrays to make a new one.
EasyData.StringLoop(string StringToRepeat, int Times) <RETURNS STRING[]>	Repeat a string in a string array.
EasyData.IntLoop(int IntToRepeat, int Times) <RETURNS INT[]>	Repeat a int in a int array.

DynamicData Module

The EasyData module contains dynamic methods for creating complex data stored in simple containers in order to provide integration between programming languages and databases. The module is in the initial development phase and is not functional. Implementation: using quasarStack.Data

DynamicData.NewInv() <RETURNS STRING>	Creates a new inventory object. An inventory is a string-shaped container responsible for storing groupable items organized by names and content.
DynamicData.InvAdd(object itemName, object itemContent, int Amount, string[] inventory) <RETURNS STRING>	Returns the selected inventory + the new item that is added to it.
DynamicData.ShowInv(string[] inventory) <RETURNS STRING>	Returns the grouped content of the selected inventory.
DynamicData.MinusInv(string contentToRemove, string[] inventory) <RETURNS STRING> BugInfo #3: Use not recommended due to bugs that interfere with data processing.	Returns a specific item removal (name + space + content) from inventory.

They will be resolved in future versions of the library.	
DynamicData.RemInv(string contentToRemove, string[] inventory) <RETURNS STRING>	Returns an inventory that has no relation to the reported content.

Standardify Module

Standardify is a module focused on creating standards to improve data manipulation, creation and availability. Implementation: using quasarStack.Data

Standardify.GenInt(int Start, int Stop) <RETURNS INT[]>	Generate an integer array standard between the desired values.
Standardify.GenStandard(string type, int Start, int Stop) <RETURNS STRING[]>	Generates a pattern based on the type selected from the desired values. Available types: month: Generates months of the year. week: Generates days of the week.

NetView Module

The NetView module offers solutions for network analysis and connections, pings, obtaining information about adapters and their configurations and general information about connectivity. Implementation: using quasarStack.NetView

NetKit.ListTCP(string method) <RETURNS STRING[]>	Lists the active TCP connections on the device. METHOD show-comment-table: Alternative to leave without defined method, it shows informative comments about the connections in the return.

NetKit.FastPing(string target) <RETURNS FLOAT>	Performs a quick ping test to a target and returns latency.
NetKit.MultiPing(string target, string requestLength) <RETURNS FLOAT[]>	Performs a ping test to the target and returns an array of latency values based on the number of requests desired.
NetKit.CurrentDNS() <RETURNS STRING[]>	Return the current DNS servers configured in the device.
NetKit.PublicIP() <RETURNS STRING>	Return the current device public IP based on Ipfy.org API.
NetKit.IsWebConnected() <RETURNS BOOL>	Test if the device is connected to the internet checking common addresses.
NetKit.TestConnection(string target) <RETURNS BOOL>	Test the connection to specific target address.

General Module

The general module offers information about the current version of quasarStack as well as functions for error management. Implementation: using quasarStack

QuasarProductInfo.Name	quasarStack
QuasarProductInfo.Version	v0.2
General.QuasarStackGeneralLogRaw	Main quasarStack log
General.QuasarStackErrorLogRaw	Main quasarStack error log
General.ReportError(string error)	Report a custom error
GlobalErrorString() <RETURNS STRING>	Returns the global error log in string

Error Dictionary

The error dictionary contains a variety of errors to aid in the development of applications and also determines the severity of the errors detected. Implementation: using quasarStack

General.ERR_CONVERSION	An error occurred while performing the data conversion.
General.ERR_LOGIC	An undetermined logic error has occurred, review the logic used or the syntax implemented. If there's no visible problem in the runtime ignore the warning.
General.ERR_DATAMAKE	An undetermined Data Creation error has occurred.
General.ERR_MATH	A mathematical error has occurred, please review the implemented logic.
General.ERR_NETWORK	A connection error has occurred, check that the source device is connected to the internet and review the network, firewall and antivirus settings.
General.ERR_NETWORK_SERVER	A connection error has occurred, the target is inaccessible. Please try again.
General.ERR_NETWORK_SYSTEM_INFO	An error occurred when trying to find information about the local network, check the program logic, its permissions and if the system is properly configured.
General.ERR_UNKNOWN	An unknown error was detected while the program was running.
General.WARN_NOHTTP_RESPONSE	No HTTP response.