Utils

1.9.0.12

by Teodor Krastev for Imperial College London

Chapter 1

Namespace Index

1.1 Packages

Here are the packages with brief descriptions (if available):

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

Chapter 3

Class Index

3.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 4

File Index

4.1 File List

Here is a list of all files with brief descriptions:

Chapter 5

Namespace Documentation

5.1 PipesServerNS Namespace Reference

Classes

- class PipeServer

Functions

- delegate void DelegateMessage (string Reply)

5.1.1 Function Documentation

5.1.1.1 DelegateMessage()

```
delegate void PipesServerNS.DelegateMessage (
             string Reply )
```

5.2 UtilsNS Namespace Reference

Classes

- class AutoFileLogger

  This is an obsolete version, please use FileLogger below instead !!! Async data storage device first you set the full path of the file, otherwise it will save in data dir under date-time file name when you want the logging to start you set Enabled to true at the end you set Enabled to false (that will flush the buffer to HD)
- class baseMMscan

  Properties part of MMscan
- class DictFileLogger

creates and logs in multicollumn table; record structure is defined in record List of string the column names must be set when created dictLog will extract only the keys with these names in that order

- class DictFileReader

  Read dictionary from multi-column text file (tab separated) format first line header (for conditions) - optional next line column names; header, subheaders & col names are read when instance is created if _record = null then read this row in record if _record has items the record will be the cross-section of _record and column names list (fileRecord)
- class FileLogger

  Async data storage device - new (dec.2018) optimized for speed (7x faster to AutoFilelogger) logger first you set the full path of the file, otherwise it will save in data dir under date-time file name when you want the logging to start you set Enabled to true at the end you set Enabled to false (that will flush the buffer to HD)
- class memLog

  In memory meassage log - mostly for debug
- class MMexec

  Class encapsulating one formated by "Book of JaSON" message
- class MMscan

  Class encapsulating a scan of a parameter
- class RemoteMessaging

  Messaging service using Windows messages (quickest way possible)
- class **Utils**
- class WaitCursor

  Hour-glass cursor while waiting for Godot

Chapter 6

Class Documentation

6.1 UtilsNS.AutoFileLogger Class Reference

This is an obsolete version, please use FileLogger below instead !!! Async data storage device first you set the full path of the file, otherwise it will save in data dir under date-time file name when you want the logging to start you set Enabled to true at the end you set Enabled to false (that will flush the buffer to HD)

Public Member Functions

- AutoFileLogger (string _prefix="", string Filename="")
- int log (List$<$ string $>$ newItems)
- int log (string newItem)
- void DropLastChar ()
- Task Flush ()

Public Attributes

- string header = ""
- string defaultExt = ".ahf"
- int bufferLimit = 256
- Stopwatch stw

Properties

- int bufferSize `[get]`
- int bufferCharSize `[get]`
- string prefix `[get]`
- bool writing `[get]`
- bool missingData `[get]`
- bool Enabled `[get, set]`
- string AutoSaveFileName `[get, set]`

6.1.1 Detailed Description

This is an obsolete version, please use FileLogger below instead !!! Async data storage device first you set the full path of the file, otherwise it will save in data dir under date-time file name when you want the logging to start you set Enabled to true at the end you set Enabled to false (that will flush the buffer to HD)

Definition at line 434 of file UtilsLib.cs.

6.1.2 Constructor & Destructor Documentation

6.1.2.1 AutoFileLogger()

```
UtilsNS.AutoFileLogger.AutoFileLogger (
             string _prefix = "",
             string Filename = "" )
```

Definition at line 449 of file UtilsLib.cs.

6.1.3 Member Function Documentation

6.1.3.1 DropLastChar()

```
void UtilsNS.AutoFileLogger.DropLastChar ( )
```

Definition at line 473 of file UtilsLib.cs.

6.1.3.2 Flush()

```
Task UtilsNS.AutoFileLogger.Flush ( )
```

Definition at line 485 of file UtilsLib.cs.

6.1.3.3 log() [1/2]

```
int UtilsNS.AutoFileLogger.log (
             List< string > newItems )
```

Definition at line 458 of file UtilsLib.cs.

6.1.3.4 log() [2/2]

```
int UtilsNS.AutoFileLogger.log (
             string newItem )
```

Definition at line 465 of file UtilsLib.cs.

6.1.4 Member Data Documentation

6.1.4.1 bufferLimit

```
int UtilsNS.AutoFileLogger.bufferLimit = 256
```

Definition at line 440 of file UtilsLib.cs.

6.1.4.2 defaultExt

```
string UtilsNS.AutoFileLogger.defaultExt = ".ahf"
```

Definition at line 438 of file UtilsLib.cs.

6.1.4.3 header

```
string UtilsNS.AutoFileLogger.header = ""
```

Definition at line 437 of file UtilsLib.cs.

6.1.4.4 stw

```
Stopwatch UtilsNS.AutoFileLogger.stw
```

Definition at line 447 of file UtilsLib.cs.

6.1.5 Property Documentation

6.1.5.1 AutoSaveFileName

```
string UtilsNS.AutoFileLogger.AutoSaveFileName  [get], [set]
```

Definition at line 571 of file UtilsLib.cs.

6.1.5.2 bufferCharSize

```
int UtilsNS.AutoFileLogger.bufferCharSize  [get]
```

Definition at line 443 of file UtilsLib.cs.

6.1.5.3 bufferSize

```
int UtilsNS.AutoFileLogger.bufferSize  [get]
```

Definition at line 442 of file UtilsLib.cs.

6.1.5.4 Enabled

```
bool UtilsNS.AutoFileLogger.Enabled  [get], [set]
```

Definition at line 529 of file UtilsLib.cs.

6.1.5.5 missingData

```
bool UtilsNS.AutoFileLogger.missingData  [get]
```

Definition at line 446 of file UtilsLib.cs.

6.1.5.6 prefix

```
string UtilsNS.AutoFileLogger.prefix  [get]
```

Definition at line 444 of file UtilsLib.cs.

6.1.5.7 writing

```
bool UtilsNS.AutoFileLogger.writing  [get]
```

Definition at line 445 of file UtilsLib.cs.

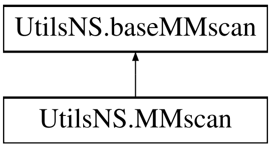
The documentation for this class was generated from the following file:

- E:/VSprojects/AxelSuite/Utils/UtilsLib.cs

6.2 UtilsNS.baseMMscan Class Reference

Properties part of MMscan

Inheritance diagram for UtilsNS.baseMMscan:



Public Member Functions

- string getAsString ()

Properties

- string sParam [get, set]
- double sFrom [get, set]
- double sTo [get, set]
- double sBy [get, set]
- double Value [get, set]

6.2.1 Detailed Description

Properties part of MMscan

Definition at line 527 of file RemoteMessaging.cs.

6.2.2 Member Function Documentation

6.2.2.1 getAsString()

```
string UtilsNS.baseMMscan.getAsString ( )
```

Definition at line 535 of file RemoteMessaging.cs.

6.2.3 Property Documentation

6.2.3.1 sBy

```
double UtilsNS.baseMMscan.sBy  [get], [set]
```

Definition at line 532 of file RemoteMessaging.cs.

6.2.3.2 sFrom

```
double UtilsNS.baseMMscan.sFrom  [get], [set]
```

Definition at line 530 of file RemoteMessaging.cs.

6.2.3.3 sParam

```
string UtilsNS.baseMMscan.sParam  [get], [set]
```

Definition at line 529 of file RemoteMessaging.cs.

6.2.3.4 sTo

```
double UtilsNS.baseMMscan.sTo  [get], [set]
```

Definition at line 531 of file RemoteMessaging.cs.

6.2.3.5 Value

```
double UtilsNS.baseMMscan.Value  [get], [set]
```

Definition at line 533 of file RemoteMessaging.cs.

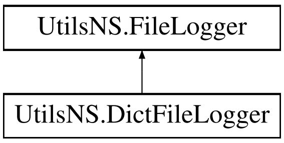
The documentation for this class was generated from the following file:

- E:/VSprojects/AxelSuite/Utils/RemoteMessaging.cs

6.3 UtilsNS.DictFileLogger Class Reference

creates and logs in multicollumn table; record structure is defined in record List of string the column names must be set when created dictLog will extract only the keys with these names in that order

Inheritance diagram for UtilsNS.DictFileLogger:



Public Member Functions

- DictFileLogger (string[ ] _record, string _prefix="", string _reqFilename="")

  Class constructor
- void setMMexecAsHeader (MMexec mme)

  When the group MMexec is known
- override void writeHeader ()

  Write the header & subheaders and colunm names line
- void dictLog (Dictionary< string, string > dict)

  This main methods in three variations, but that is the basic one
- void dictLog (Dictionary< string, object > dict)

  log that way with undefined Values type
- void dictLog (Dictionary< string, double > dict, string format="")

  log that way with double Values type with format

Additional Inherited Members

6.3.1 Detailed Description

creates and logs in multicollumn table; record structure is defined in record List of string the column names must be set when created dictLog will extract only the keys with these names in that order

Definition at line 742 of file UtilsLib.cs.

6.3.2 Constructor & Destructor Documentation

6.3.2.1 DictFileLogger()

```
UtilsNS.DictFileLogger.DictFileLogger (
             string[ ] _record,
             string _prefix = "",
             string _reqFilename = "" )
```

Class constructor

Parameters

_record	Required and fixed column names
_prefix	Ending in case of timestamp name
_reqFilename	If empty timestamp name is generated

Definition at line 751 of file UtilsLib.cs.

6.3.3 Member Function Documentation

6.3.3.1 dictLog() [1/3]

```
void UtilsNS.DictFileLogger.dictLog (
            Dictionary< string, double > dict,
            string format = "" )
```

log that way with double Values type with format

Parameters

dict	
format	

Definition at line 828 of file UtilsLib.cs.

6.3.3.2 dictLog() [2/3]

```
void UtilsNS.DictFileLogger.dictLog (
            Dictionary< string, object > dict )
```

log that way with undefined Values type

Parameters

dict	

Definition at line 813 of file UtilsLib.cs.

6.3.3.3 dictLog() [3/3]

```
void UtilsNS.DictFileLogger.dictLog (
            Dictionary< string, string > dict )
```

This main methods in three variations, but that is the basic one

Parameters

dict	

Definition at line 796 of file UtilsLib.cs.

6.3.3.4 setMMexecAsHeader()

```
void UtilsNS.DictFileLogger.setMMexecAsHeader (
             MMexec mme )
```

When the group MMexec is known

Parameters

mme	

Definition at line 762 of file UtilsLib.cs.

6.3.3.5 writeHeader()

```
override void UtilsNS.DictFileLogger.writeHeader ( )  [virtual]
```

Write the header & subheaders and colunm names line

Reimplemented from UtilsNS.FileLogger.

Definition at line 780 of file UtilsLib.cs.

The documentation for this class was generated from the following file:

- E:/VSprojects/AxelSuite/Utils/UtilsLib.cs

6.4 UtilsNS.DictFileReader Class Reference

Read dictionary from multi-column text file (tab separated) format first line header (for conditions) - optional next line column names; header, subheaders & col names are read when instance is created if _record = null then read this row in record if _record has items the record will be the cross-section of _record and column names list (fileRecord)

Public Member Functions

- DictFileReader (string Filename, string[ ] strArr=null)

  Class constructor
- bool stringIterator (ref Dictionary< string, string > rslt)

  returns one line (row) as (column.name , cell.value) dictionary
- bool doubleIterator (ref Dictionary< string, double > rslt)

  same as above but Values are double

Public Attributes

- string header
- List< string > subheaders
- int counter = 0
- List< string > record

6.4.1 Detailed Description

Read dictionary from multi-column text file (tab separated) format first line header (for conditions) - optional next line column names; header, subheaders & col names are read when instance is created if _record = null then read this row in record if _record has items the record will be the cross-section of _record and column names list (fileRecord)

Definition at line 848 of file UtilsLib.cs.

6.4.2 Constructor & Destructor Documentation

6.4.2.1 DictFileReader()

```
UtilsNS.DictFileReader.DictFileReader (
             string Filename,
             string[ ] strArr = null )
```

Class constructor

Parameters

Filename	File must exits
strArr	Array of column names

Definition at line 859 of file UtilsLib.cs.

6.4.3 Member Function Documentation

6.4.3.1 doubleIterator()

```
bool UtilsNS.DictFileReader.doubleIterator (
            ref Dictionary< string, double > rslt )
```

same as above but Values are double

Parameters

rslt	

Returns

Definition at line 930 of file UtilsLib.cs.

6.4.3.2 stringIterator()

```
bool UtilsNS.DictFileReader.stringIterator (
            ref Dictionary< string, string > rslt )
```

returns one line (row) as (column.name , cell.value) dictionary

Parameters

rslt	one table row

Returns

if we can go again

Definition at line 900 of file UtilsLib.cs.

6.4.4 Member Data Documentation

6.4.4.1 counter

```
int UtilsNS.DictFileReader.counter = 0
```

Definition at line 852 of file UtilsLib.cs.

6.4.4.2 header

```
string UtilsNS.DictFileReader.header
```

Definition at line 850 of file UtilsLib.cs.

6.4.4.3 record

```
List<string> UtilsNS.DictFileReader.record
```

Definition at line 853 of file UtilsLib.cs.

6.4.4.4 subheaders

```
List<string> UtilsNS.DictFileReader.subheaders
```

Definition at line 851 of file UtilsLib.cs.

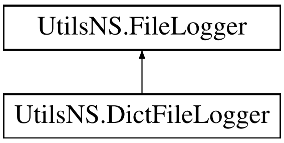
The documentation for this class was generated from the following file:

- E:/VSprojects/AxelSuite/Utils/UtilsLib.cs

6.5 UtilsNS.FileLogger Class Reference

Async data storage device - new (dec.2018) optimized for speed (7x faster to AutoFilelogger) logger first you set the full path of the file, otherwise it will save in data dir under date-time file name when you want the logging to start you set Enabled to true at the end you set Enabled to false (that will flush the buffer to HD)

Inheritance diagram for UtilsNS.FileLogger:



Public Member Functions

- FileLogger (string _prefix="", string _reqFilename="")

  Class constructor
- void log (List< string > newItems)

  The main call if you have List of strings
- void log (string newItem)

  That's the main method
- void CreateLogger (string filePath)

  Create actual asynchronious logger
- virtual void writeHeader ()

  Write a header and subheaders with #

Public Attributes

- string header = ""
- List< string > subheaders
- string defaultExt = ".ahf"
- Stopwatch stw

Properties

- string prefix `[get]`
- string reqFilename `[get]`
- bool writing `[get]`
- bool missingData `[get]`
- string LogFilename `[get]`
- bool Enabled `[get, set]`

  Switch this on to create the file and start to accept logs switch it off to flash the buffer and close the file

6.5.1 Detailed Description

Async data storage device - new (dec.2018) optimized for speed (7x faster to AutoFilelogger) logger first you set the full path of the file, otherwise it will save in data dir under date-time file name when you want the logging to start you set Enabled to true at the end you set Enabled to false (that will flush the buffer to HD)

Definition at line 590 of file UtilsLib.cs.

6.5.2 Constructor & Destructor Documentation

6.5.2.1 FileLogger()

```
UtilsNS.FileLogger.FileLogger (
            string _prefix = "",
            string _reqFilename = "" )
```

Class constructor

Parameters

_prefix	
_reqFilename	

Definition at line 608 of file UtilsLib.cs.

6.5.3 Member Function Documentation

6.5.3.1 CreateLogger()

```
void UtilsNS.FileLogger.CreateLogger (
            string filePath )
```

Create actual asynchronious logger

Parameters

filePath	

Definition at line 658 of file UtilsLib.cs.

6.5.3.2 log() [1/2]

```
void UtilsNS.FileLogger.log (
            List< string > newItems )
```

The main call if you have List of strings

Parameters

newItems	

Definition at line 621 of file UtilsLib.cs.

6.5.3.3 log() [2/2]

```
void UtilsNS.FileLogger.log (
            string newItem )
```

That's the main method

Parameters

newItem	

Definition at line 632 of file UtilsLib.cs.

6.5.3.4 writeHeader()

```
virtual void UtilsNS.FileLogger.writeHeader ( )  [virtual]
```

Write a header and subheaders with #

Reimplemented in UtilsNS.DictFileLogger.

Definition at line 687 of file UtilsLib.cs.

6.5.4 Member Data Documentation

6.5.4.1 defaultExt

```
string UtilsNS.FileLogger.defaultExt = ".ahf"
```

Definition at line 595 of file UtilsLib.cs.

6.5.4.2 header

```
string UtilsNS.FileLogger.header = ""
```

Definition at line 593 of file UtilsLib.cs.

6.5.4.3 stw

```
Stopwatch UtilsNS.FileLogger.stw
```

Definition at line 601 of file UtilsLib.cs.

6.5.4.4 subheaders

```
List<string> UtilsNS.FileLogger.subheaders
```

Definition at line 594 of file UtilsLib.cs.

6.5.5 Property Documentation

6.5.5.1 Enabled

```
bool UtilsNS.FileLogger.Enabled  [get], [set]
```

Switch this on to create the file and start to accept logs switch it off to flash the buffer and close the file

Definition at line 701 of file UtilsLib.cs.

6.5.5.2 LogFilename

```
string UtilsNS.FileLogger.LogFilename  [get]
```

Definition at line 675 of file UtilsLib.cs.

6.5.5.3 missingData

```
bool UtilsNS.FileLogger.missingData  [get]
```

Definition at line 600 of file UtilsLib.cs.

6.5.5.4 prefix

```
string UtilsNS.FileLogger.prefix  [get]
```

Definition at line 597 of file UtilsLib.cs.

6.5.5.5 reqFilename

```
string UtilsNS.FileLogger.reqFilename  [get]
```

Definition at line 598 of file UtilsLib.cs.

6.5.5.6 writing

```
bool UtilsNS.FileLogger.writing  [get]
```

Definition at line 599 of file UtilsLib.cs.

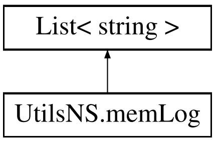
The documentation for this class was generated from the following file:

- E:/VSprojects/AxelSuite/Utils/UtilsLib.cs

6.6 UtilsNS.memLog Class Reference

In memory meassage log - mostly for debug

Inheritance diagram for UtilsNS.memLog:



Public Member Functions

- memLog (int depth=32)
- void log (string txt)

Public Attributes

- bool Enabled = true

6.6.1 Detailed Description

In memory meassage log - mostly for debug

Definition at line 20 of file RemoteMessaging.cs.

6.6.2 Constructor & Destructor Documentation

6.6.2.1 memLog()

```
UtilsNS.memLog.memLog (
                int depth = 32 )
```

Definition at line 24 of file RemoteMessaging.cs.

6.6.3 Member Function Documentation

6.6.3.1 log()

```
void UtilsNS.memLog.log (
             string txt )
```

Definition at line 28 of file RemoteMessaging.cs.

6.6.4 Member Data Documentation

6.6.4.1 Enabled

```
bool UtilsNS.memLog.Enabled = true
```

Definition at line 22 of file RemoteMessaging.cs.

The documentation for this class was generated from the following file:

- E:/VSprojects/AxelSuite/Utils/RemoteMessaging.cs

6.7 UtilsNS.MMexec Class Reference

Class encapsulating one formated by "Book of JaSON" message

Public Member Functions

- MMexec (string Caption="", string Sender="", string Command="", int ID=-1)

  Class constructor
- void Clear ()

  Clean all up
- void Assign (MMexec src)

  Assign src to this
- MMexec Clone ()

  Clone this - copy all the props to a new instance
- string Abort (string Sender="")

  Standard Abort (Cancel) message

Public Attributes

- Dictionary< string, object > prms

Properties

- string mmexec `[get, set]`
- string sender `[get, set]`
- string cmd `[get, set]`
- int id `[get, set]`

6.7.1 Detailed Description

Class encapsulating one formated by "Book of JaSON" message

Definition at line 446 of file RemoteMessaging.cs.

6.7.2 Constructor & Destructor Documentation

6.7.2.1 MMexec()

```
UtilsNS.MMexec.MMexec (
             string Caption = "",
             string Sender = "",
             string Command = "",
             int ID = -1 )
```

Class constructor

Parameters

Caption	
Sender	
Command	
ID	

Definition at line 462 of file RemoteMessaging.cs.

6.7.3 Member Function Documentation

6.7.3.1 Abort()

```
string UtilsNS.MMexec.Abort (
             string Sender = "" )
```

Standard Abort (Cancel) message

Parameters

Sender	

Returns

Definition at line 514 of file RemoteMessaging.cs.

6.7.3.2 Assign()

```
void UtilsNS.MMexec.Assign (
             MMexec src )
```

Assign src to this

Parameters

src	

Definition at line 486 of file RemoteMessaging.cs.

6.7.3.3 Clear()

```
void UtilsNS.MMexec.Clear ( )
```

Clean all up

Definition at line 474 of file RemoteMessaging.cs.

6.7.3.4 Clone()

```
MMexec UtilsNS.MMexec.Clone ( )
```

Clone this - copy all the props to a new instance

Returns

Definition at line 498 of file RemoteMessaging.cs.

6.7.4 Member Data Documentation

6.7.4.1 prms

```
Dictionary<string, object> UtilsNS.MMexec.prms
```

Definition at line 453 of file RemoteMessaging.cs.

6.7.5 Property Documentation

6.7.5.1 cmd

```
string UtilsNS.MMexec.cmd  [get], [set]
```

Definition at line 451 of file RemoteMessaging.cs.

6.7.5.2 id

```
int UtilsNS.MMexec.id  [get], [set]
```

Definition at line 452 of file RemoteMessaging.cs.

6.7.5.3 mmexec

```
string UtilsNS.MMexec.mmexec  [get], [set]
```

Definition at line 449 of file RemoteMessaging.cs.

6.7.5.4 sender

```
string UtilsNS.MMexec.sender  [get], [set]
```

Definition at line 450 of file RemoteMessaging.cs.

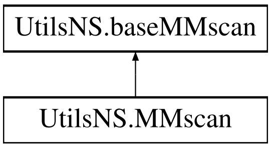
The documentation for this class was generated from the following file:

- E:/VSprojects/AxelSuite/Utils/RemoteMessaging.cs

6.8 UtilsNS.MMscan Class Reference

Class encapsulating a scan of a parameter

Inheritance diagram for UtilsNS.MMscan:



Public Member Functions

- MMscan (string _groupID="", string _sParam="", double _sFrom=double.NaN, double _sTo=double.NaN, double _sBy=double.NaN, double _Value=double.NaN)
- bool Check ()
- bool isFirstValue ()
- bool isLastValue ()
- bool Next ()

  The main call when scan, it works multiscan (chain by NextInChain of scans) mode The latter uses recursive (on class level) call the same Next down the chain
- void Assign (MMscan src)

  Assign src to this
- MMscan Clone ()

  Clone this - copy all the props to a new instance
- void TestInit ()

  Fill in something - only for tests
- void ToDictionary (ref Dictionary< string, object > dict)

  Fill in dictionary with props
- bool FromDictionary (Dictionary< string, object > dict)

  Update props from a dictionary

Public Attributes

- MMscan NextInChain = null

Properties

- string groupID `[get, set]`
- string AsString `[get, set]`

  Package / unpackage the scan params as string

6.8.1 Detailed Description

Class encapsulating a scan of a parameter

Definition at line 544 of file RemoteMessaging.cs.

6.8.2 Constructor & Destructor Documentation

6.8.2.1 MMscan()

```
UtilsNS.MMscan.MMscan (
             string _groupID = "",
             string _sParam = "",
             double _sFrom = double.NaN,
             double _sTo = double.NaN,
             double _sBy = double.NaN,
             double _Value = double.NaN )
```

Definition at line 548 of file RemoteMessaging.cs.

6.8.3 Member Function Documentation

6.8.3.1 Assign()

```
void UtilsNS.MMscan.Assign (
             MMscan src )
```

Assign src to this

Parameters

src	

Definition at line 622 of file RemoteMessaging.cs.

6.8.3.2 Check()

```
bool UtilsNS.MMscan.Check ( )
```

Definition at line 552 of file RemoteMessaging.cs.

6.8.3.3 Clone()

```
MMscan UtilsNS.MMscan.Clone ( )
```

Clone this - copy all the props to a new instance

Returns

Definition at line 630 of file RemoteMessaging.cs.

6.8.3.4 FromDictionary()

```
bool UtilsNS.MMscan.FromDictionary (
             Dictionary< string, object > dict )
```

Update props from a dictionary

Parameters

dict	

Returns

Definition at line 664 of file RemoteMessaging.cs.

6.8.3.5 isFirstValue()

```
bool UtilsNS.MMscan.isFirstValue ( )
```

Definition at line 560 of file RemoteMessaging.cs.

6.8.3.6 isLastValue()

```
bool UtilsNS.MMscan.isLastValue ( )
```

Definition at line 565 of file RemoteMessaging.cs.

6.8.3.7 Next()

```
bool UtilsNS.MMscan.Next ( )
```

The main call when scan, it works multiscan (chain by NextInChain of scans) mode The latter uses recursive (on class level) call the same Next down the chain

Returns

Definition at line 577 of file RemoteMessaging.cs.

6.8.3.8 TestInit()

```
void UtilsNS.MMscan.TestInit ( )
```

Fill in something - only for tests

Definition at line 638 of file RemoteMessaging.cs.

6.8.3.9 ToDictionary()

```
void UtilsNS.MMscan.ToDictionary (
             ref Dictionary< string, object > dict )
```

Fill in dictionary with props

Parameters

dict	

Definition at line 650 of file RemoteMessaging.cs.

6.8.4 Member Data Documentation

6.8.4.1 NextInChain

```
MMscan UtilsNS.MMscan.NextInChain = null
```

Definition at line 570 of file RemoteMessaging.cs.

6.8.5 Property Documentation

6.8.5.1 AsString

```
string UtilsNS.MMscan.AsString  [get], [set]
```

Package / unpackage the scan params as string

Definition at line 601 of file RemoteMessaging.cs.

6.8.5.2 groupID

```
string UtilsNS.MMscan.groupID  [get], [set]
```

Definition at line 546 of file RemoteMessaging.cs.

The documentation for this class was generated from the following file:

- E:/VSprojects/AxelSuite/Utils/RemoteMessaging.cs

6.9 PipesServerNS.PipeServer Class Reference

Public Member Functions

- void Listen (string PipeName)

Events

- DelegateMessage PipeMessage

6.9.1 Detailed Description

Definition at line 14 of file PipeServer.cs.

6.9.2 Member Function Documentation

6.9.2.1 Listen()

```
void PipesServerNS.PipeServer.Listen (
             string PipeName )
```

Definition at line 19 of file PipeServer.cs.

6.9.3 Event Documentation

6.9.3.1 PipeMessage

```
DelegateMessage PipesServerNS.PipeServer.PipeMessage
```

Definition at line 16 of file PipeServer.cs.

The documentation for this class was generated from the following file:

- E:/VSprojects/AxelSuite/Utils/PipeServer.cs

6.10 UtilsNS.RemoteMessaging Class Reference

Messaging service using Windows messages (quickest way possible)

Classes

- struct **COPYDATASTRUCT**

  The COPYDATASTRUCT structure contains data to be passed to another application by the WM_COPYDATA message.
- struct **MyStruct**
- class **NativeMethod**

Public Member Functions

- RemoteMessaging (string Partner, int _keyID=666)

  Establish communication channel
- double elapsedTime ()

  Precise timer for accurate time stamps
- void DoEvents ()
- object ExitFrame (object f)
- bool synchroClock (bool force=false)

  Synchronizing stopwatches of two apps with precision less than 15us
- delegate bool ReceiveHandler (string msg)
- delegate void ActiveCommHandler (bool active, bool forced)
- delegate void AsyncSentHandler (bool OK, string json2send)
- bool sendCommand (string msg, int delay=0)

  The wrapper around the Windows messaging send part
- bool CheckConnection (bool forced=false)

  ping<->pong to check the connection

Public Attributes

- memLog Log
- Stopwatch stopwatch
- DispatcherTimer dTimer
- bool Enabled = true

Protected Member Functions

- bool Receive (string msg)

    Receive message
- void ActiveComm (bool active, bool forced)

    When the channel opens/closes
- void AsyncSent (bool OK, string json2send)

Properties

- string partner `[get]`
- bool silentPartner `[get, set]`
- string lastRcvMsg `[get]`
- string lastSndMsg `[get]`
- int autoCheckPeriod `[get, set]`
- int keyID `[get]`
- bool Connected `[get]`
- bool partnerPresent `[get]`

Events

- ReceiveHandler OnReceive
- ActiveCommHandler OnActiveComm
- AsyncSentHandler OnAsyncSent

6.10.1 Detailed Description

Messaging service using Windows messages (quickest way possible)

Definition at line 39 of file RemoteMessaging.cs.

6.10.2 Constructor & Destructor Documentation

6.10.2.1 RemoteMessaging()

```
UtilsNS.RemoteMessaging.RemoteMessaging (
            string Partner,
            int _keyID = 666 )
```

Establish communication channel

Parameters

Partner	The title (caption) of the application
_keyID	Similar to port - one app can have more than one channel with diff. keyID

Definition at line 74 of file RemoteMessaging.cs.

6.10.3 Member Function Documentation

6.10.3.1 ActiveComm()

```
void UtilsNS.RemoteMessaging.ActiveComm (
             bool active,
             bool forced )  [protected]
```

When the channel opens/closes

Parameters

active	
forced	

Definition at line 185 of file RemoteMessaging.cs.

6.10.3.2 ActiveCommHandler()

```
delegate void UtilsNS.RemoteMessaging.ActiveCommHandler (
             bool active,
             bool forced )
```

6.10.3.3 AsyncSent()

```
void UtilsNS.RemoteMessaging.AsyncSent (
             bool OK,
             string json2send )  [protected]
```

Definition at line 269 of file RemoteMessaging.cs.

6.10.3.4 AsyncSentHandler()

```
delegate void UtilsNS.RemoteMessaging.AsyncSentHandler (
             bool OK,
             string json2send )
```

6.10.3.5 CheckConnection()

```
bool UtilsNS.RemoteMessaging.CheckConnection (
             bool forced = false )
```

ping<->pong to check the connection

Parameters

forced	

Returns

Definition at line 352 of file RemoteMessaging.cs.

6.10.3.6 DoEvents()

```
void UtilsNS.RemoteMessaging.DoEvents ( )
```

Definition at line 121 of file RemoteMessaging.cs.

6.10.3.7 elapsedTime()

```
double UtilsNS.RemoteMessaging.elapsedTime ( )
```

Precise timer for accurate time stamps

Returns

Definition at line 115 of file RemoteMessaging.cs.

6.10.3.8 ExitFrame()

```
object UtilsNS.RemoteMessaging.ExitFrame (
             object f )
```

Definition at line 129 of file RemoteMessaging.cs.

6.10.3.9 Receive()

```
bool UtilsNS.RemoteMessaging.Receive (
             string msg )  [protected]
```

Receive message

Parameters

msg	

Returns

Definition at line 172 of file RemoteMessaging.cs.

6.10.3.10 ReceiveHandler()

```
delegate bool UtilsNS.RemoteMessaging.ReceiveHandler (
             string msg )
```

6.10.3.11 sendCommand()

```
bool UtilsNS.RemoteMessaging.sendCommand (
             string msg,
             int delay = 0 )
```

The wrapper around the Windows messaging send part

Parameters

msg	
delay	

Returns

Definition at line 280 of file RemoteMessaging.cs.

6.10.3.12 synchroClock()

```
bool UtilsNS.RemoteMessaging.synchroClock (
            bool force = false )
```

Synchronizing stopwatches of two apps with precision less than 15us

Parameters

force	

Returns

Definition at line 140 of file RemoteMessaging.cs.

6.10.4 Member Data Documentation

6.10.4.1 dTimer

```
DispatcherTimer UtilsNS.RemoteMessaging.dTimer
```

Definition at line 49 of file RemoteMessaging.cs.

6.10.4.2 Enabled

```
bool UtilsNS.RemoteMessaging.Enabled = true
```

Definition at line 57 of file RemoteMessaging.cs.

6.10.4.3 Log

```
memLog UtilsNS.RemoteMessaging.Log
```

Definition at line 46 of file RemoteMessaging.cs.

6.10.4.4 stopwatch

```
Stopwatch UtilsNS.RemoteMessaging.stopwatch
```

Definition at line 47 of file RemoteMessaging.cs.

6.10.5 Property Documentation

6.10.5.1 autoCheckPeriod

```
int UtilsNS.RemoteMessaging.autoCheckPeriod  [get], [set]
```

Definition at line 52 of file RemoteMessaging.cs.

6.10.5.2 Connected

```
bool UtilsNS.RemoteMessaging.Connected  [get]
```

Definition at line 59 of file RemoteMessaging.cs.

6.10.5.3 keyID

```
int UtilsNS.RemoteMessaging.keyID  [get]
```

Definition at line 58 of file RemoteMessaging.cs.

6.10.5.4 lastRcvMsg

```
string UtilsNS.RemoteMessaging.lastRcvMsg  [get]
```

Definition at line 44 of file RemoteMessaging.cs.

6.10.5.5 lastSndMsg

```
string UtilsNS.RemoteMessaging.lastSndMsg  [get]
```

Definition at line 45 of file RemoteMessaging.cs.

6.10.5.6 partner

```
string UtilsNS.RemoteMessaging.partner  [get]
```

Definition at line 41 of file RemoteMessaging.cs.

6.10.5.7 partnerPresent

```
bool UtilsNS.RemoteMessaging.partnerPresent  [get]
```

Definition at line 61 of file RemoteMessaging.cs.

6.10.5.8 silentPartner

```
bool UtilsNS.RemoteMessaging.silentPartner  [get], [set]
```

Definition at line 42 of file RemoteMessaging.cs.

6.10.6 Event Documentation

6.10.6.1 OnActiveComm

```
ActiveCommHandler UtilsNS.RemoteMessaging.OnActiveComm
```

Definition at line 179 of file RemoteMessaging.cs.

6.10.6.2 OnAsyncSent

```
AsyncSentHandler UtilsNS.RemoteMessaging.OnAsyncSent
```

Definition at line 268 of file RemoteMessaging.cs.

6.10.6.3 OnReceive

```
ReceiveHandler UtilsNS.RemoteMessaging.OnReceive
```

Definition at line 166 of file RemoteMessaging.cs.

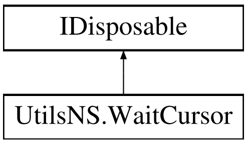
The documentation for this class was generated from the following file:

- E:/VSprojects/AxelSuite/Utils/RemoteMessaging.cs

6.11 UtilsNS.WaitCursor Class Reference

Hour-glass cursor while waiting for Godot

Inheritance diagram for UtilsNS.WaitCursor:



Public Member Functions

- WaitCursor ()
- void Dispose ()

6.11.1 Detailed Description

Hour-glass cursor while waiting for Godot

Definition at line 955 of file UtilsLib.cs.

6.11.2 Constructor & Destructor Documentation

6.11.2.1 WaitCursor()

```
UtilsNS.WaitCursor.WaitCursor ( )
```

Definition at line 958 of file UtilsLib.cs.

6.11.3 Member Function Documentation

6.11.3.1 Dispose()

```
void UtilsNS.WaitCursor.Dispose ( )
```

Definition at line 966 of file UtilsLib.cs.

The documentation for this class was generated from the following file:

- E:/VSprojects/AxelSuite/Utils/UtilsLib.cs

Chapter 7

File Documentation

7.1 E:/VSprojects/AxelSuite/Utils/PipeServer.cs File Reference

Classes

- class PipesServerNS.PipeServer

Namespaces

- namespace PipesServerNS

Functions

- delegate void PipesServerNS.DelegateMessage (string Reply)

7.2 E:/VSprojects/AxelSuite/Utils/RemoteMessaging.cs File Reference

Classes

- class UtilsNS.memLog

  In memory meassage log - mostly for debug
- class UtilsNS.RemoteMessaging

  Messaging service using Windows messages (quickest way possible)
- struct **UtilsNS.RemoteMessaging.MyStruct**
- struct **UtilsNS.RemoteMessaging.COPYDATASTRUCT**

  The COPYDATASTRUCT structure contains data to be passed to another application by the WM_COPYDATA message.
- class **UtilsNS.RemoteMessaging.NativeMethod**
- class UtilsNS.MMexec

  Class encapsulating one formated by "Book of JaSON" message
- class UtilsNS.baseMMscan

  Properties part of MMscan
- class UtilsNS.MMscan

  Class encapsulating a scan of a parameter

Namespaces

- namespace UtilsNS

7.3 E:/VSprojects/AxelSuite/Utils/UtilsLib.cs File Reference

Classes

- class **UtilsNS.Utils**
- class UtilsNS.AutoFileLogger

  This is an obsolete version, please use FileLogger below instead !!! Async data storage device first you set the full path of the file, otherwise it will save in data dir under date-time file name when you want the logging to start you set Enabled to true at the end you set Enabled to false (that will flush the buffer to HD)
- class UtilsNS.FileLogger

  Async data storage device - new (dec.2018) optimized for speed (7x faster to AutoFilelogger) logger first you set the full path of the file, otherwise it will save in data dir under date-time file name when you want the logging to start you set Enabled to true at the end you set Enabled to false (that will flush the buffer to HD)
- class UtilsNS.DictFileLogger

  creates and logs in multicollumn table; record structure is defined in record List of string the column names must be set when created dictLog will extract only the keys with these names in that order
- class UtilsNS.DictFileReader

  Read dictionary from multi-column text file (tab separated) format first line header (for conditions) - optional next line column names; header, subheaders & col names are read when instance is created if _record = null then read this row in record if _record has items the record will be the cross-section of _record and column names list (fileRecord)
- class UtilsNS.WaitCursor

  Hour-glass cursor while waiting for Godot

Namespaces

- namespace UtilsNS

Index