

MazBotV4 Helpers API

- Cexplode
- Atomic operations
- Linked list

Copyright

```
/* ***** */
/*
 *      Implementation of php's explode written in C
 *      Written by Maz (2008)
 *      Added Atomic operations for x86 architecture and
 *      Linked list implementation.
 *      Written by Maz (2009-2010)
 *      http://maz-programmersdiary.blogspot.com/
 *
 *      You're free to use this piece of code.
 *      You can also modify it freely, but if you
 *      improve this, you must write the improved code
 *      in comments at:
 *      http://maz-programmersdiary.blogspot.com/
 *      or at:
 *      http://c-ohjelmoi.janajatuksia.blogspot.com/
 *      or mail the corrected version to me at
 *      Mazziesaccount@gmail.com
 *
 *      Revision History:
 *
 *      - 0.0.6 15.08.2009/Maz Fixed atomic CAS
 *      - 0.0.5 11.08.2009/Maz Added Cexplode_free_allButPieces
 *      - 0.0.4 11.08.2009/Maz Added atomic ops and
 *                               mbot_ll
 *      -v0.0.3 31.07.2009/Maz Added Cexplode_concat
 *                               (untested)
 *      -v0.0.2 21.07.2009/Maz Some additions for better
 *                               usability in MazBotV4
 *      -v0.0.1 16.09.2008/Maz
 *
 * ***** */
```

Data Structures

Here are the data structures with brief descriptions:

CexplodeStrings Struct for Cexplode object

mbot_linkedList

MbotAtomic32 Struct for 32bit wide integer type used in atomic operations

Most recent version

Bug tracker

Maz - programmer's diary

Contact: Mazziesaccount@gmail.com

MazBotV4 Helpers API

- Cexplode
- Atomic operations
- Linked list

Copyright

```
/* *****  
/*  
/*      Implementation of php's explode written in C      *  
/*      Written by  Maz (2008)                             *  
/*      Added Atomic operations for x86 architecture and  *  
/*      Linked list implementation.                       *  
/*      Written by  Maz (2009-2010)                       *  
/*      http://maz-programmersdiary.blogspot.com/         *  
/*  
/*      You're free to use this piece of code.           *  
/*      You can also modify it freely, but if you        *  
/*      improve this, you must write the improved code   *  
/*      in comments at:                                   *  
/*      http://maz-programmersdiary.blogspot.com/         *  
/*      or at:                                           *  
/*      http://c-ohjelmoi.janajatuksia.blogspot.com/     *  
/*      or mail the corrected version to me at          *  
/*      Mazziesaccount@gmail.com                         *  
/*  
/*      Revision History:                                *  
/*  
/*      - 0.0.6 15.08.2009/Maz  Fixed atomic CAS          *  
/*      - 0.0.5 11.08.2009/Maz  Added Cexplode_free_allButPieces *  
/*      - 0.0.4 11.08.2009/Maz  Added atomic ops and      *  
/*                               mbot_ll                   *  
/*      -v0.0.3 31.07.2009/Maz  Added Cexplode_concat     *  
/*                               (untested)                *  
/*      -v0.0.2 21.07.2009/Maz  Some additions for better *  
/*                               usability in MazBotV4     *  
/*      -v0.0.1 16.09.2008/Maz                               *  
/*  
/* *****
```

Data Structure Index

C | M

C	CexplodeStrings	M	mbot_linkedList	MbotAtomic32
---	-----------------	---	-----------------	--------------

C | M

Most recent version

Bug tracker

Maz - programmer's diary

Contact: Mazziesaccount@gmail.com

MazBotV4 Helpers API

- Cexplode
- Atomic operations
- Linked list

Copyright

```
/* *****  
/*  
/*      Implementation of php's explode written in C      *  
/*      Written by  Maz (2008)                             *  
/*      Added Atomic operations for x86 architecture and  *  
/*      Linked list implementation.                       *  
/*      Written by  Maz (2009-2010)                       *  
/*      http://maz-programmersdiary.blogspot.com/         *  
/*  
/*      You're free to use this piece of code.           *  
/*      You can also modify it freely, but if you        *  
/*      improve this, you must write the improved code   *  
/*      in comments at:                                  *  
/*      http://maz-programmersdiary.blogspot.com/         *  
/*      or at:                                           *  
/*      http://c-ohjelmoi.janajatuksia.blogspot.com/     *  
/*      or mail the corrected version to me at          *  
/*      Mazziesaccount@gmail.com                         *  
/*  
/*      Revision History:                                *  
/*  
/*      - 0.0.6 15.08.2009/Maz  Fixed atomic CAS          *  
/*      - 0.0.5 11.08.2009/Maz  Added Cexplode_free_allButPieces *  
/*      - 0.0.4 11.08.2009/Maz  Added atomic ops and      *  
/*                               mbot_ll                   *  
/*      -v0.0.3 31.07.2009/Maz  Added Cexplode_concat     *  
/*                               (untested)                *  
/*      -v0.0.2 21.07.2009/Maz  Some additions for better *  
/*                               usability in MazBotV4     *  
/*      -v0.0.1 16.09.2008/Maz                               *  
/*  
/* *****
```

File List

Here is a list of all files with brief descriptions:

helpers.h [code]

Most recent version

Bug tracker

Maz - programmer's diary

Contact: Mazziesaccount@gmail.com

MazBotV4 Helpers API

- Cexplode
- Atomic operations
- Linked list

Copyright

```
/* ***** */
/*
 *      Implementation of php's explode written in C
 *      Written by  Maz (2008)
 *      Added Atomic operations for x86 architecture and
 *      Linked list implementation.
 *      Written by  Maz (2009-2010)
 *      http://maz-programmersdiary.blogspot.com/
 *
 *      You're free to use this piece of code.
 *      You can also modify it freely, but if you
 *      improve this, you must write the improved code
 *      in comments at:
 *      http://maz-programmersdiary.blogspot.com/
 *      or at:
 *      http://c-ohjelmoi.janajatuksia.blogspot.com/
 *      or mail the corrected version to me at
 *      Mazziesaccount@gmail.com
 *
 *      Revision History:
 *
 *      - 0.0.6 15.08.2009/Maz  Fixed atomic CAS
 *      - 0.0.5 11.08.2009/Maz  Added Cexplode_free_allButPieces
 *      - 0.0.4 11.08.2009/Maz  Added atomic ops and
 *                               mbot_ll
 *      -v0.0.3 31.07.2009/Maz  Added Cexplode_concat
 *                               (untested)
 *      -v0.0.2 21.07.2009/Maz  Some additions for better
 *                               usability in MazBotV4
 *      -v0.0.1 16.09.2008/Maz
 *
/* ***** */
```

- All
- Variables

Here is a list of all struct and union fields with links to the structures/unions they belong to:

- amnt : CexplodeStrings
- data : mbot_linkedList
- head : mbot_linkedList
- index : CexplodeStrings
- next : mbot_linkedList
- prev : mbot_linkedList
- sem : MbotAtomic32
- separator : CexplodeStrings
- sepwasatend : CexplodeStrings
- startedWdelim : CexplodeStrings

- strings : CexplodeStrings
- value : MbotAtomic32

Most recent version

Bug tracker

Maz - programmer's diary

Contact: *Mazziesaccount@gmail.com*

MazBotV4 Helpers API

- Cexplode
- Atomic operations
- Linked list

Copyright

```
/* *****  
/*  
/*      Implementation of php's explode written in C      *  
/*      Written by  Maz (2008)                             *  
/*      Added Atomic operations for x86 architecture and  *  
/*      Linked list implementation.                       *  
/*      Written by  Maz (2009-2010)                       *  
/*      http://maz-programmersdiary.blogspot.com/         *  
/*  
/*      You're free to use this piece of code.            *  
/*      You can also modify it freely, but if you        *  
/*      improve this, you must write the improved code   *  
/*      in comments at:                                   *  
/*      http://maz-programmersdiary.blogspot.com/         *  
/*      or at:                                            *  
/*      http://c-ohjelmoi.janajatuksia.blogspot.com/     *  
/*      or mail the corrected version to me at          *  
/*      Mazziesaccount@gmail.com                         *  
/*  
/*      Revision History:                                 *  
/*  
/*      - 0.0.6 15.08.2009/Maz  Fixed atomic CAS          *  
/*      - 0.0.5 11.08.2009/Maz  Added Cexplode_free_allButPieces *  
/*      - 0.0.4 11.08.2009/Maz  Added atomic ops and      *  
/*                               mbot_ll                  *  
/*      -v0.0.3 31.07.2009/Maz  Added Cexplode_concat     *  
/*                               (untested)                *  
/*      -v0.0.2 21.07.2009/Maz  Some additions for better *  
/*                               usability in MazBotV4     *  
/*      -v0.0.1 16.09.2008/Maz                               *  
/*  
/* *****
```

- All
- Variables

- amnt : CexplodeStrings
- data : mbot_linkedList
- head : mbot_linkedList
- index : CexplodeStrings
- next : mbot_linkedList
- prev : mbot_linkedList
- sem : MbotAtomic32
- separator : CexplodeStrings
- sepwasatend : CexplodeStrings
- startedWdelim : CexplodeStrings

- strings : CexplodeStrings
- value : MbotAtomic32

Most recent version

Bug tracker

Maz - programmer's diary

Contact: *Mazziesaccount@gmail.com*

MazBotV4 Helpers API

- Cexplode
- Atomic operations
- Linked list

Copyright

```
/* *****  
/*  
/*      Implementation of php's explode written in C  
/*      Written by  Maz (2008)  
/*      Added Atomic operations for x86 architecture and  
/*      Linked list implementation.  
/*      Written by  Maz (2009-2010)  
/*      http://maz-programmersdiary.blogspot.com/  
/*  
/*      You're free to use this piece of code.  
/*      You can also modify it freely, but if you  
/*      improve this, you must write the improved code  
/*      in comments at:  
/*      http://maz-programmersdiary.blogspot.com/  
/*      or at:  
/*      http://c-ohjelmoi.janajatuksia.blogspot.com/  
/*      or mail the corrected version to me at  
/*      Mazziesaccount@gmail.com  
/*  
/*      Revision History:  
/*  
/*      - 0.0.6 15.08.2009/Maz  Fixed atomic CAS  
/*      - 0.0.5 11.08.2009/Maz  Added Cexplode_free_allButPieces  
/*      - 0.0.4 11.08.2009/Maz  Added atomic ops and  
/*                               mbot_ll  
/*      -v0.0.3 31.07.2009/Maz  Added Cexplode_concat  
/*                               (untested)  
/*      -v0.0.2 21.07.2009/Maz  Some additions for better  
/*                               usability in MazBotV4  
/*      -v0.0.1 16.09.2008/Maz  
/*  
/* *****
```

- All
- Functions
- Enumerations
- Enumerator
- Defines

- CEXPLODE_LAST_ITEM : helpers.h

Most recent version

Bug tracker

Maz - programmer's diary

Contact: Mazziesaccount@gmail.com

MazBotV4 Helpers API

- Cexplode
- Atomic operations
- Linked list

Copyright

```
/* *****  
/*  
/*      Implementation of php's explode written in C  
/*      Written by  Maz (2008)  
/*      Added Atomic operations for x86 architecture and  
/*      Linked list implementation.  
/*      Written by  Maz (2009-2010)  
/*      http://maz-programmersdiary.blogspot.com/  
/*  
/*      You're free to use this piece of code.  
/*      You can also modify it freely, but if you  
/*      improve this, you must write the improved code  
/*      in comments at:  
/*      http://maz-programmersdiary.blogspot.com/  
/*      or at:  
/*      http://c-ohjelmoi.janajatuksia.blogspot.com/  
/*      or mail the corrected version to me at  
/*      Mazziesaccount@gmail.com  
/*  
/*      Revision History:  
/*  
/*      - 0.0.6 15.08.2009/Maz  Fixed atomic CAS  
/*      - 0.0.5 11.08.2009/Maz  Added Cexplode_free_allButPieces  
/*      - 0.0.4 11.08.2009/Maz  Added atomic ops and  
/*                               mbot_ll  
/*      -v0.0.3 31.07.2009/Maz  Added Cexplode_concat  
/*                               (untested)  
/*      -v0.0.2 21.07.2009/Maz  Some additions for better  
/*                               usability in MazBotV4  
/*      -v0.0.1 16.09.2008/Maz  
/*  
/* *****
```

- All
- Functions
- Enumerations
- Enumerator
- Defines

- ECexplodeRet : helpers.h

Most recent version

Bug tracker

Maz - programmer's diary

Contact: Mazziesaccount@gmail.com

MazBotV4 Helpers API

- Cexplode
- Atomic operations
- Linked list

Copyright

```
/* *****  
/*  
/*      Implementation of php's explode written in C  
/*      Written by  Maz (2008)  
/*      Added Atomic operations for x86 architecture and  
/*      Linked list implementation.  
/*      Written by  Maz (2009-2010)  
/*      http://maz-programmersdiary.blogspot.com/  
/*  
/*      You're free to use this piece of code.  
/*      You can also modify it freely, but if you  
/*      improve this, you must write the improved code  
/*      in comments at:  
/*      http://maz-programmersdiary.blogspot.com/  
/*      or at:  
/*      http://c-ohjelmoi.janajatuksia.blogspot.com/  
/*      or mail the corrected version to me at  
/*      Mazziesaccount@gmail.com  
/*  
/*      Revision History:  
/*  
/*      - 0.0.6 15.08.2009/Maz  Fixed atomic CAS  
/*      - 0.0.5 11.08.2009/Maz  Added Cexplode_free_allButPieces  
/*      - 0.0.4 11.08.2009/Maz  Added atomic ops and  
/*                               mbot_ll  
/*      -v0.0.3 31.07.2009/Maz  Added Cexplode_concat  
/*                               (untested)  
/*      -v0.0.2 21.07.2009/Maz  Some additions for better  
/*                               usability in MazBotV4  
/*      -v0.0.1 16.09.2008/Maz  
/*  
/* *****
```

- All
- Functions
- Enumerations
- Enumerator
- Defines

- ECexplodeRet_InternalFailure : helpers.h
- ECexplodeRet_InvalidParams : helpers.h

Most recent version

Bug tracker

Maz - programmer's diary

Contact: *Mazziesaccount@gmail.com*

MazBotV4 Helpers API

- Cexplode
- Atomic operations
- Linked list

Copyright

```
/* *****  
/*  
/*      Implementation of php's explode written in C  
/*      Written by  Maz (2008)  
/*      Added Atomic operations for x86 architecture and  
/*      Linked list implementation.  
/*      Written by  Maz (2009-2010)  
/*      http://maz-programmersdiary.blogspot.com/  
/*  
/*      You're free to use this piece of code.  
/*      You can also modify it freely, but if you  
/*      improve this, you must write the improved code  
/*      in comments at:  
/*      http://maz-programmersdiary.blogspot.com/  
/*      or at:  
/*      http://c-ohjelmoi.janajatuksia.blogspot.com/  
/*      or mail the corrected version to me at  
/*      Mazziesaccount@gmail.com  
/*  
/*      Revision History:  
/*  
/*      - 0.0.6 15.08.2009/Maz  Fixed atomic CAS  
/*      - 0.0.5 11.08.2009/Maz  Added Cexplode_free_allButPieces  
/*      - 0.0.4 11.08.2009/Maz  Added atomic ops and  
/*                               mbot_ll  
/*      -v0.0.3 31.07.2009/Maz  Added Cexplode_concat  
/*                               (untested)  
/*      -v0.0.2 21.07.2009/Maz  Some additions for better  
/*                               usability in MazBotV4  
/*      -v0.0.1 16.09.2008/Maz  
/*  
/* *****
```

- All
- Functions
- Enumerations
- Enumerator
- Defines
- c
- m

- c -

- Cexplode() : helpers.h
- Cexplode_concat() : helpers.h
- Cexplode_free() : helpers.h
- Cexplode_free_allButPieces() : helpers.h
- Cexplode_getAmnt() : helpers.h
- Cexplode_getfirst() : helpers.h
- Cexplode_getlast() : helpers.h
- Cexplode_getlentilllast() : helpers.h
- Cexplode_getnext() : helpers.h
- Cexplode_getNth() : helpers.h
- Cexplode_nextexists() : helpers.h
- Cexplode_removeCurrent() : helpers.h
- Cexplode_removeNth() : helpers.h
- Cexplode_sepwasatend() : helpers.h

- m -

- mbot_atomicAdd() : helpers.h
- mbot_atomicCAS() : helpers.h
- mbot_atomicDec() : helpers.h
- mbot_atomicDecIfGreater() : helpers.h
- mbot_atomicDecIfSmaller() : helpers.h
- mbot_atomicGet() : helpers.h
- mbot_atomicIncIfGreater() : helpers.h
- mbot_atomicIncIfSmaller() : helpers.h
- mbot_ll_add() : helpers.h
- mbot_ll_copylist_wdata() : helpers.h
- mbot_ll_dataGet() : helpers.h
- mbot_ll_dataSet() : helpers.h
- mbot_ll_destroy() : helpers.h
- mbot_ll_get_first() : helpers.h
- mbot_ll_get_last() : helpers.h
- mbot_ll_get_next() : helpers.h
- mbot_ll_get_prev() : helpers.h
- mbot_ll_head_get() : helpers.h
- mbot_ll_init() : helpers.h
- mbot_ll_release() : helpers.h
- mbot_ll_safe_release() : helpers.h
- mbot_ll_seek() : helpers.h
- mbot_ltrim() : helpers.h
- mbot_ltrim() : helpers.h
- mbot_rtrim() : helpers.h
- mbot_trimall() : helpers.h
- MbotAtomic32Init() : helpers.h
- MbotAtomic32Uninit() : helpers.h

Most recent version

Bug tracker

Maz - programmer's diary

Contact: *Mazziesaccount@gmail.com*

MazBotV4 Helpers API

- Cexplode
- Atomic operations
- Linked list

Copyright

```
/* ***** */
/*
 *      Implementation of php's explode written in C
 *      Written by  Maz (2008)
 *      Added Atomic operations for x86 architecture and
 *      Linked list implementation.
 *      Written by  Maz (2009-2010)
 *      http://maz-programmersdiary.blogspot.com/
 *
 *      You're free to use this piece of code.
 *      You can also modify it freely, but if you
 *      improve this, you must write the improved code
 *      in comments at:
 *      http://maz-programmersdiary.blogspot.com/
 *      or at:
 *      http://c-ohjelmoi.janajatuksia.blogspot.com/
 *      or mail the corrected version to me at
 *      Mazziesaccount@gmail.com
 *
 *      Revision History:
 *
 *      - 0.0.6 15.08.2009/Maz  Fixed atomic CAS
 *      - 0.0.5 11.08.2009/Maz  Added Cexplode_free_allButPieces
 *      - 0.0.4 11.08.2009/Maz  Added atomic ops and
 *                               mbot_ll
 *      -v0.0.3 31.07.2009/Maz  Added Cexplode_concat
 *                               (untested)
 *      -v0.0.2 21.07.2009/Maz  Some additions for better
 *                               usability in MazBotV4
 *      -v0.0.1 16.09.2008/Maz
 *
/* ***** */
```

- All
- Functions
- Enumerations
- Enumerator
- Defines

- c
- e
- m

Here is a list of all functions, variables, defines, enums, and typedefs with links to the files they belong to:

- c -

- Cexplode() : helpers.h
- Cexplode_concat() : helpers.h
- Cexplode_free() : helpers.h
- Cexplode_free_allButPieces() : helpers.h
- Cexplode_getAmnt() : helpers.h
- Cexplode_getfirst() : helpers.h
- Cexplode_getlast() : helpers.h
- Cexplode_getlentilllast() : helpers.h
- Cexplode_getnext() : helpers.h
- Cexplode_getNth() : helpers.h
- CEXPLODE_LAST_ITEM : helpers.h
- Cexplode_nextexists() : helpers.h
- Cexplode_removeCurrent() : helpers.h
- Cexplode_removeNth() : helpers.h
- Cexplode_sepwasatend() : helpers.h

- e -

- ECexplodeRet : helpers.h
- ECexplodeRet_InternalFailure : helpers.h
- ECexplodeRet_InvalidParams : helpers.h

- m -

- mbot_atomicAdd() : helpers.h
- mbot_atomicCAS() : helpers.h
- mbot_atomicDec() : helpers.h
- mbot_atomicDecIfGreater() : helpers.h
- mbot_atomicDecIfSmaller() : helpers.h
- mbot_atomicGet() : helpers.h
- mbot_atomicIncIfGreater() : helpers.h
- mbot_atomicIncIfSmaller() : helpers.h
- mbot_ll_add() : helpers.h
- mbot_ll_copylist_wdata() : helpers.h
- mbot_ll_dataGet() : helpers.h
- mbot_ll_dataSet() : helpers.h
- mbot_ll_destroy() : helpers.h
- mbot_ll_get_first() : helpers.h
- mbot_ll_get_last() : helpers.h
- mbot_ll_get_next() : helpers.h
- mbot_ll_get_prev() : helpers.h
- mbot_ll_head_get() : helpers.h
- mbot_ll_init() : helpers.h
- mbot_ll_release() : helpers.h
- mbot_ll_safe_release() : helpers.h

- `mbot_ll_seek()` : `helpers.h`
- `mbot_lrtrim()` : `helpers.h`
- `mbot_ltrim()` : `helpers.h`
- `mbot_rtrim()` : `helpers.h`
- `mbot_trimall()` : `helpers.h`
- `MbotAtomic32Init()` : `helpers.h`
- `MbotAtomic32Uninit()` : `helpers.h`

Most recent version

Bug tracker

Maz - programmer's diary

Contact: *Mazziesaccount@gmail.com*

MazBotV4 Helpers API

- Cexplode
- Atomic operations
- Linked list

Copyright

```
/* *****  
/*  
/*      Implementation of php's explode written in C  
/*      Written by  Maz (2008)  
/*      Added Atomic operations for x86 architecture and  
/*      Linked list implementation.  
/*      Written by  Maz (2009-2010)  
/*      http://maz-programmersdiary.blogspot.com/  
/*  
/*      You're free to use this piece of code.  
/*      You can also modify it freely, but if you  
/*      improve this, you must write the improved code  
/*      in comments at:  
/*      http://maz-programmersdiary.blogspot.com/  
/*      or at:  
/*      http://c-ohjelmoi.janajatuksia.blogspot.com/  
/*      or mail the corrected version to me at  
/*      Mazziesaccount@gmail.com  
/*  
/*      Revision History:  
/*  
/*      - 0.0.6 15.08.2009/Maz  Fixed atomic CAS  
/*      - 0.0.5 11.08.2009/Maz  Added Cexplode_free_allButPieces  
/*      - 0.0.4 11.08.2009/Maz  Added atomic ops and  
/*                               mbot_ll  
/*      -v0.0.3 31.07.2009/Maz  Added Cexplode_concat  
/*                               (untested)  
/*      -v0.0.2 21.07.2009/Maz  Some additions for better  
/*                               usability in MazBotV4  
/*      -v0.0.1 16.09.2008/Maz  
/*  
/* *****
```

helpers.h File Reference

```
#include <stdio.h>  
#include <string.h>  
#include <stdlib.h>  
#include <semaphore.h>
```

Go to the source code of this file.

Data Structures

```
struct MbotAtomic32  
    Struct for 32bit wide integer type used in atomic operations. More...  
struct CexplodeStrings  
    Struct for Cexplode object. More...
```

```
struct mbot_linkedList
```

Defines

```
#define CEXPLODE_LAST_ITEM 0xFFFFFFFF
```

Enumerations

```
enum ECexplodeRet { ECexplodeRet_InternalFailure = -666,  
ECexplodeRet_InvalidParams = -667 }  
enumeration for Cexplodei's error return values
```

More...

Functions

```
int Cexplode_removeCurrent (CexplodeStrings *exp_obj)  
    Removes the previously returned piece.  
char * Cexplode_removeNth (int nro, CexplodeStrings *exp_obj)  
    Removes Nth piece from cexplode Must not be called before calling Cexplode If  
    removed item is last piece, the "sepwasatend" flag will be set true! Note, you can use  
    special CEXPLODE_LAST_ITEM define to remove the last item.  
int Cexplode_getAmnt (CexplodeStrings exp_obj)  
    Get the amount of pieces in exploded object Must not be called before calling  
    Cexplode.  
int Cexplode (const char *string, const char *delim, CexplodeStrings *exp_obj)  
    Explodes string to pieces according to delimiter. Result is stored in exp_obj and can  
    be retrieved using functions below The results of explosion are stored in same order  
    as they occurred in initial string, eg. if string "1 2 3 4" would be exploded with space  
    (" ") as delimiter, Cexplode_getfirst() would return 1, Cexplode_getNth() with n  
    being 4, would return 4.  
int Cexplode_nextexists (CexplodeStrings exp_obj)  
    Peeks if there's another result in exp_obj. Must not be called before calling Cexplode.  
char * Cexplode_getNth (int index, CexplodeStrings *exp_obj)  
    Retrieve's Nth exploded piece - first is first (index starts from 1, not from 0) Updates  
    internal iterator, IE following call to Cexplode_getnext will retrieve index+1th piece.  
char * Cexplode_getfirst (CexplodeStrings *exp_obj)  
    Get's the first exploded piece. Same as Cexplode_getNth(1,*exp_obj);.  
char * Cexplode_getnext (CexplodeStrings *exp_obj)  
    Get's next piece. Returns NULL if no more pieces are around.  
char * Cexplode_getlast (CexplodeStrings *exp_obj)  
    Gets last exploded piece.  
void Cexplode_free (CexplodeStrings exp_obj)  
    Frees resources allocated by call to Cexplode() - BEWARE frees also splitted pieces.  
void Cexplode_free_allButPieces (CexplodeStrings exp_obj)  
    Frees resources allocated by call to Cexplode() - does not free splitted pieces.  
size_t Cexplode_getlentilllast (CexplodeStrings exp_obj)  
    Gets the amount of chars from the start of the original string to the beginning of last  
    found delimiter.  
int Cexplode_sepwasatend (CexplodeStrings exp_obj)  
    returns 1 if last chars in original string were the separator - else returns 0  
int Cexplode_concat (CexplodeStrings *first, CexplodeStrings *second)  
    Concatenates two exp_objs into one. Modifies the first argument to contain new  
    exp_obj. Does not modify second argument.
```

int mbot_ltrim (char *text, char trimchar)
 removes trimchars from the beginning of a string.

int mbot_rtrim (char *text, char trimchar)
 removes trailing trimchars from a string.

int mbot_lrtrim (char *text, char trimchar)
 removes trailing trimchars as well as trimchars from the beginning of a string.

int mbot_trimall (char *text, char trimchar)
 removes all trimchars from a string.

MbotAtomic32

* MbotAtomic32Init ()
 Creates 32bit atomic variable, compatible with mbot_atomic* operations.

void MbotAtomic32Uninit (MbotAtomic32 *_this_)
 Uninitializes MbotAtomic32. This must not be called when it is possible someone is using the variable.

unsigned int mbot_atomicGet (MbotAtomic32 *atomic)
 Get the value atomically.

unsigned int mbot_atomicAdd (MbotAtomic32 *atomic, unsigned int addition)
 Increase value atomically - returns value before increment.

unsigned int mbot_atomicDec (MbotAtomic32 *atomic, unsigned int decrement)
 Decrease value atomically - returns value before decrement.

unsigned int mbot_atomicDecIfGreater (MbotAtomic32 *atomic, unsigned int decrement, unsigned int cmp)
 Decrease value atomically, if original value is greater than cmp. Returns original value. (If returnval<cmp, no decrement occurred.

unsigned int mbot_atomicDecIfSmaller (MbotAtomic32 *atomic, unsigned int decrement, unsigned int cmp)
 Decrease value atomically, if original value is smaller than cmp. Returns original value. (If returnval>cmp, no decrement occurred.

unsigned int mbot_atomicIncIfGreater (MbotAtomic32 *atomic, unsigned int decrement, unsigned int cmp)
 Increase value atomically, if original value is greater than cmp. Returns original value. (If returnval<cmp, no increment occurred.

unsigned int mbot_atomicIncIfSmaller (MbotAtomic32 *atomic, unsigned int decrement, unsigned int cmp)
 Increase value atomically, if original value is smaller than cmp. Returns original value. (If returnval>cmp, no increment occurred.

unsigned int mbot_atomicCAS (MbotAtomic32 *atomic, unsigned int old, unsigned int newval)

mbot_linkedList

* mbot_ll_init ()
 Initializes linked list for use - returns ptr to list head.

mbot_linkedList

* mbot_ll_get_prev (mbot_linkedList *_this)
 Gets previous list item. - returns previous item, or NULL if error occurred/first item given as param.

mbot_linkedList

* mbot_ll_head_get (mbot_linkedList *_this)
 Get the head of the list Head can be used to maintain the location of empty list.

mbot_linkedList

* mbot_ll_get_next (mbot_linkedList *_this)

Get's next element - NULL if error occurred, or last element was provided as argument.

`mbot_linkedList`

* `mbot_ll_get_first (mbot_linkedList *_this)`

Get's the first list element - returns first element or NULL if no elements stored, or if an error occurred.

`mbot_linkedList`

* `mbot_ll_get_last (mbot_linkedList *_this)`

Gets the last element in list.

`mbot_linkedList`

* `mbot_ll_add (mbot_linkedList *_this, void *data)`

Adds item to list (data). Does not do a copy of data. Any list item (including head) can be used as `_this`.

`mbot_linkedList`

* `mbot_ll_release (mbot_linkedList *_this)`

removes given item from list - does not free memory.

`mbot_linkedList`

* `mbot_ll_safe_release (mbot_linkedList *_this, void *data)`

removes list item which holds data pointed by data. Any list item can be given in `_this`. Does not free memory. Returns removed list entry, and user must call free upon entry and stored data.

`void * mbot_ll_dataGet (mbot_linkedList *_this)`

Gets data stored to an entry - entry and data are left untouched.

`void * mbot_ll_dataSet (mbot_linkedList *_this, void *data)`

Sets data to an list,.

`mbot_linkedList`

* `mbot_ll_seek (mbot_linkedList *_this, void *data, size_t datasize)`

Searchs through the list and returns element in which the held data matches data specified in params.

`mbot_linkedList`

* `mbot_ll_copypolist_wdata (mbot_linkedList *old, size_t itemsize)`

Copies given list and itemsize bytes of data from each container to new list, and returns a pointer to the copypolist.

`void mbot_ll_destroy (mbot_linkedList **_this)`

Frees all entries from list, and destroys the list - does not free stored data. `_this` is NULled upon return.

Define Documentation

`#define CEXPLODE_LAST_ITEM 0xFFFFFFFF`

Enumeration Type Documentation

`enum ECexplodeRet`

enumeration for Cexplodei's error return values

Enumerator:

ECexplodeRet_InternalFailure

ECexplodeRet_InvalidParams

Function Documentation

```
int Cexplode (    const char *    string,
                 const char *    delim,
                 CexplodeStrings * exp_obj
               )
```

Explodes string to pieces according to delimiter. Result is stored in exp_obj and can be retrieved using functions below. The results of explosion are stored in same order as they occurred in initial string, eg. if string "1 2 3 4" would be exploded with space (" ") as delimiter, Cexplode_getfirst() would return 1, Cexplode_getNth() with n being 4, would return 4.

Parameters:

const char *string pointer to C string being exploded
const char *delim pointer to C string used as delimiter for cutting original string
CexplodeStrings *exp_obj pointer to CexplodeStrings type object, which will be filled to contain results of explosion.

Returns:

amount of pieces - number smaller than 1 if an error occurs

See also:

CexplodeStrings, Cexplode_removeCurrent, Cexplode_removeNth, Cexplode_getAmnt, Cexplode_nextexists, Cexplode_getNth, Cexplode_getfirst, Cexplode_getnext, Cexplode_getlast, Cexplode_free, Cexplode_free_allButPieces, Cexplode_getlentilllast, Cexplode_sepwasatend, Cexplode_concat

```
int Cexplode_concat (    CexplodeStrings *    first,
                        CexplodeStrings *    second
                      )
```

Concatenates two exp_objs into one. Modifies the first argument to contain new exp_obj. Does not modify second argument.

Parameters:

CexplodeStrings *first pointer to CexplodeStrings type object, filled by call to Cexplode() to be combined with another CexplodeStrings object. This will contain new CexplodeStrings object holding results for both of the original CexplodeStrings objects.

CexplodeStrings *second pointer to CexplodeStrings type object, filled by call to Cexplode() to be combined with another CexplodeStrings object - this will not be modified during call.

Returns:

the amount of pieces in new exp_obj - negative number upon error.

Warning:

Must not be called before calling Cexplode for both first and second argument.

```
void Cexplode_free ( CexplodeStrings exp_obj )
```

Frees resources allocated by call to Cexplode() - BEWARE frees also splitted pieces.

Parameters:

CexplodeStrings exp_obj CexplodeStrings type object, filled by call to Cexplode()

Warning:

Must not be called before calling Cexplode

BEWARE frees also splitted pieces, in which the returned pointers by Cexplode_get* points.

See also:

Cexplode_free_allButPieces, Cexplode, Cexplode_getNth, Cexplode_getnext, Cexplode_getfirst, Cexplode_getlast

```
void Cexplode_free_allButPieces ( CexplodeStrings exp_obj )
```

Frees resources allocated by call to Cexplode() - does not free splitted pieces.

Parameters:

CexplodeStrings exp_obj CexplodeStrings type object, filled by call to Cexplode()

Warning:

Must not be called before calling Cexplode

See also:

Cexplode_free, Cexplode, Cexplode_getNth, Cexplode_getnext, Cexplode_getfirst, Cexplode_getlast

```
int Cexplode_getAmnt ( CexplodeStrings exp_obj )
```

Get the amount of pieces in exploded object Must not be called before calling Cexplode.

Parameters:

CexplodeStrings *exp_obj pointer to CexplodeStrings type object, filled by call to Cexplode()

Returns:

amount of exploded pieces stored in CexplodeStrings container

See also:

Cexplode

```
char* Cexplode_getfirst ( CexplodeStrings * exp_obj )
```

Get's the first exploded piece. Same as Cexplode_getNth(1,*exp_obj);.

Parameters:

CexplodeStrings *exp_obj pointer to CexplodeStrings type object, filled by call to Cexplode()

Returns:

NULL on error, othervice a pointer to result stored in Cexplode object

Warning:

Must not be called before calling Cexplode

See also:

Cexplode, Cexplode_getNth, Cexplode_getnext, Cexplode_getlast

```
char* Cexplode_getlast ( CexplodeStrings * exp_obj )
```

Gets last exploded piece.

Parameters:

CexplodeStrings *exp_obj pointer to CexplodeStrings type object, filled by call to Cexplode()

Returns:

NULL on error, othervice a pointer to result stored in Cexplode object

Warning:

Must not be called before calling Cexplode

See also:

Cexplode, Cexplode_getNth, Cexplode_getnext, Cexplode_getfirst

```
size_t Cexplode_getlentilllast ( CexplodeStrings exp_obj )
```

Gets the amount of chars from the start of the original string to the beginning of last found delimiter.

Parameters:

CexplodeStrings exp_obj CexplodeStrings type object, filled by call to Cexplode()

Returns:

amount of chars from the start of the original string to the beginning of last found delimiter

Warning:

Must not be called before calling Cexplode

See also:

Cexplode, Cexplode_sepwasatend

```
char* Cexplode_getnext ( CexplodeStrings * exp_obj )
```

Get's next piece. Returns NULL if no more pieces are around.

Parameters:

CexplodeStrings *exp_obj pointer to CexplodeStrings type object, filled by call to Cexplode()

Returns:

NULL on error, otherwice a pointer to result stored in Cexplode object

Warning:

Must not be called before calling Cexplode

See also:

Cexplode, Cexplode_getNth, Cexplode_getfirst, Cexplode_getlast

```
char* Cexplode_getNth ( int index,
                        CexplodeStrings * exp_obj
                      )
```

Retrieve's Nth exploded piece - first is first (index starts from 1, not from 0) Updates internal iterator, IE following call to Cexplode_getnext will retrieve index+1th piece.

Parameters:

int index index number of result to be retrieved. first is first (index starts from 1, not from 0)

CexplodeStrings *exp_obj pointer to CexplodeStrings type object, filled by call to Cexplode()

Returns:

NULL on error, otherwice a pointer to result stored in Cexplode object

Warning:

Must not be called before calling Cexplode

See also:

Cexplode, Cexplode_getfirst, Cexplode_getnext, Cexplode_getlast, Cexplode_getAmnt

```
int Cexplode_nextexists ( CexplodeStrings exp_obj )
```

Peeks if there's another result in exp_obj. Must not be called before calling Cexplode.

Parameters:

CexplodeStrings exp_obj CexplodeStrings type object, filled by call to Cexplode()

Returns:

1 if next piece exists (Eg. if Cexplode_getnext et al. can be safely used), 0 if there's no next result in object.

See also:

Cexplode, Cexplode_getnext

```
int Cexplode_removeCurrent ( CexplodeStrings * exp_obj )
```

Removes the previously returned piece.

Must not be called before calling Cexplode If removed item is last piece, the "sepwasatend" flag will be set true

Parameters:

CexplodeStrings *exp_obj pointer to CexplodeStrings type object, filled by call to Cexplode()

Returns:

0 at success, -1 at failure

See also:

Cexplode, Cexplode_removeNth, Cexplode_getAmnt, Cexplode_nextexists

```
char* Cexplode_removeNth ( int nro,
                          CexplodeStrings * exp_obj
                          )
```

Removes Nth piece from cexplode Must not be called before calling Cexplode If removed item is last piece, the "sepwasatend" flag will be set true! Note, you can use special CEXPLODE_LAST_ITEM define to remove the last item.

Parameters:

int nro number of exploded piece to be removed from the CexplodeStrings containing results

CexplodeStrings *exp_obj pointer to CexplodeStrings type object, filled by call to Cexplode()

Returns:

ptr to removed string

See also:

Cexplode, Cexplode_removeCurrent, Cexplode_getAmnt, Cexplode_nextexists

int Cexplode_sepwasatend (*CexplodeStrings* *exp_obj*)

returns 1 if last chars in original string were the separator - else returns 0

Parameters:

CexplodeStrings exp_obj CexplodeStrings type object, filled by call to Cexplode()

Returns:

1 if last chars in original string were the separator - else returns 0

Warning:

Must not be called before calling Cexplode

See also:

Cexplode, Cexplode_getlentilllast

unsigned int mbot_atomicAdd (*MbotAtomic32* * *atomic*,
 unsigned int *addition*
)

Increase value atomically - returns value before increment.

Warning:

If non x86 arch is used, these atomic ops are ineffective dummies using a huge semaphore (provided only for compatibility). On x86 arch compile with define ARCH_x86

unsigned int mbot_atomicCAS (*MbotAtomic32* * *atomic*,
 unsigned int *old*,
 unsigned int *newval*
)

```

unsigned int mbot_atomicDec    (    MbotAtomic32 *    atomic,
                                unsigned int          decrement
                                )

```

Decrease value atomically - returns value before decrement.

Warning:

If non x86 arch is used, these atomic ops are ineffective dummies using a huge semaphore (provided only for compatibility). On x86 arch compile with define ARCH_x86

```

unsigned int mbot_atomicDecIfGreater    (    MbotAtomic32 *    atomic,
                                            unsigned int        decrement,
                                            unsigned int        cmp
                                            )

```

Decrease value atomically, if original value is greater than cmp. Returns original value. (If returnval<cmp, no decrement occurred.

Warning:

If non x86 arch is used, these atomic ops are ineffective dummies using a huge semaphore (provided only for compatibility). On x86 arch compile with define ARCH_x86

```

unsigned int mbot_atomicDecIfSmaller    (    MbotAtomic32 *    atomic,
                                            unsigned int        decrement,
                                            unsigned int        cmp
                                            )

```

Decrease value atomically, if original value is smaller than cmp. Returns original value. (If returnval>cmp, no decrement occurred.

Warning:

If non x86 arch is used, these atomic ops are ineffective dummies using a huge semaphore (provided only for compatibility). On x86 arch compile with define ARCH_x86

```

unsigned int mbot_atomicGet    (    MbotAtomic32 *    atomic    )

```

Get the value atomically.

Warning:

If non x86 arch is used, these atomic ops are ineffective dummies using a huge semaphore (provided only for compatibility). On x86 arch compile with define ARCH_x86

```

unsigned int mbot_atomicIncIfGreater (  MbotAtomic32 *   atomic,
                                         unsigned int    decrement,
                                         unsigned int    cmp
                                         )

```

Increase value atomically, if original value is greater than *cmp*. Returns original value. (If *returnval*<*cmp*, no increment occurred.

Warning:

If non x86 arch is used, these atomic ops are ineffective dummies using a huge semaphore (provided only for compatibility). On x86 arch compile with define ARCH_x86

```

unsigned int mbot_atomicIncIfSmaller (  MbotAtomic32 *   atomic,
                                         unsigned int    decrement,
                                         unsigned int    cmp
                                         )

```

Increase value atomically, if original value is smaller than *cmp*. Returns original value. (If *returnval*>*cmp*, no increment occurred.

Warning:

If non x86 arch is used, these atomic ops are ineffective dummies using a huge semaphore (provided only for compatibility). On x86 arch compile with define ARCH_x86

```

mbot_linkedList* mbot_ll_add (  mbot_linkedList *   _this,
                                void *               data
                                )

```

Adds item to list (*data*). Does not do a copy of *data*. Any list item (including head) can be used as *_this*.

Returns:

list entry corresponding to stored *data*

```

mbot_linkedList* mbot_ll_copylist_wdata (  mbot_linkedList *   old,
                                             size_t              itemsiz
                                             )

```

Copies given list and *itemsiz* bytes of *data* from each container to new list, and returns a pointer to the copylist.

Returns:

a pointer to the copylist and NULL on error

Warning:

This assumes that each "container" in list holds at least itemsize bytes of data - and copies exactly itemsize bytes.

Usable really only for lists which hold fixed size items!

```
void* mbot_ll_dataGet ( mbot_linkedList * _this )
```

Gets data stored to an entry - entry and data are left untouched.

```
void* mbot_ll_dataSet ( mbot_linkedList * _this,  
                        void * data  
                        )
```

Sets data to an list,.

Returns:

previous data

Warning:

- this should be avoided. Malicious use may corrupt the list!

```
void mbot_ll_destroy ( mbot_linkedList ** _this )
```

Frees all entries from list, and destroys the list - does not free stored data. _this is NULLED upon return.

```
mbot_linkedList* mbot_ll_get_first ( mbot_linkedList * _this )
```

Get's the first list element - returns first element or NULL if no elements stored, or if an error occurred.

```
mbot_linkedList* mbot_ll_get_last ( mbot_linkedList * _this )
```

Gets the last element in list.

```
mbot_linkedList* mbot_ll_get_next ( mbot_linkedList * _this )
```

Get's next element - NULL if error occurred, or last element was provided as argument.

```
mbot_linkedList* mbot_ll_get_prev ( mbot_linkedList * _this )
```

Gets previous list item. - returns previous item, or NULL if error occurred/first item given as param.

```
mbot_linkedList* mbot_ll_head_get ( mbot_linkedList * _this )
```

Get the head of the list Head can be used to maintain the location of empty list.

Returns:

the head, and NULL on error

Warning:

HEAD IS NOT SUPPOSED TO BE USED AS STORING ELEMENT!

```
mbot_linkedList* mbot_ll_init ( )
```

Initializes linked list for use - returns ptr to list head.

```
mbot_linkedList* mbot_ll_release ( mbot_linkedList * _this )
```

removes given item from list - does not free memory.

Returns:

removed list entry, and user must call free upon entry and stored data.

```
mbot_linkedList* mbot_ll_safe_release ( mbot_linkedList * _this,  
                                         void * data  
                                         )
```

removes list item which holds data pointed by data. Any list item can be given in *_this*. Does not free memory. Returns removed list entry, and user must call free upon entry and stored data.

Returns:

removed list entry

```
mbot_linkedList* mbot_ll_seek ( mbot_linkedList * _this,  
                                void * data,  
                                size_t datasize  
                                )
```

Searchs through the list and returns element in which the held data matches data specified in params.

Warning:

, all elements must contain at least as much data as specified in size_t *datasize*!

```
int mbot_ltrim ( char * text,
                char trimchar
              )
```

removes trailing trimchars as well as trimchars from the beginning of a string.

Returns:

number of characters removed

```
int mbot_ltrim ( char * text,
                char trimchar
              )
```

removes trimchars from the beginning of a string.

Returns:

number of characters removed

```
int mbot_rtrim ( char * text,
                char trimchar
              )
```

removes trailing trimchars from a string.

Returns:

number of characters removed

```
int mbot_trimall ( char * text,
                  char trimchar
                )
```

removes all trimchars from a string.

Returns:

number of characters removed

```
MbotAtomic32* MbotAtomic32Init ( )
```

Creates 32bit atomic variable, compatible with mbot_atomic* operations.

```
void MbotAtomic32Uninit ( MbotAtomic32 ** _this_ )
```

Uninitializes MbotAtomic32. This must not be called when it is possible someone is using the variable.

Warning:

If non x86 arch is used, these atomic ops are ineffective dummies using a huge semaphore (provided only for compatibility). On x86 arch compile with define ARCH_x86

Most recent version

Bug tracker

Maz - programmer's diary

Contact: *Mazziesaccount@gmail.com*

MazBotV4 Helpers API

- Cexplode
- Atomic operations
- Linked list

Copyright

```
/* ***** */
/*
 *      Implementation of php's explode written in C
 *      Written by  Maz (2008)
 *      Added Atomic operations for x86 architecture and
 *      Linked list implementation.
 *      Written by  Maz (2009-2010)
 *      http://maz-programmersdiary.blogspot.com/
 *
 *      You're free to use this piece of code.
 *      You can also modify it freely, but if you
 *      improve this, you must write the improved code
 *      in comments at:
 *      http://maz-programmersdiary.blogspot.com/
 *      or at:
 *      http://c-ohjelmoiJanajatuksia.blogspot.com/
 *      or mail the corrected version to me at
 *      Mazziesaccount@gmail.com
 *
 *      Revision History:
 *
 *      - 0.0.6 15.08.2009/Maz  Fixed atomic CAS
 *      - 0.0.5 11.08.2009/Maz  Added Cexplode_free_allButPieces
 *      - 0.0.4 11.08.2009/Maz  Added atomic ops and
 *                               mbot_ll
 *      -v0.0.3 31.07.2009/Maz  Added Cexplode_concat
 *                               (untested)
 *      -v0.0.2 21.07.2009/Maz  Some additions for better
 *                               usability in MazBotV4
 *      -v0.0.1 16.09.2008/Maz
 *
/* ***** */
```

helpers.h

Go to the documentation of this file.

```
00001
00002 /* ***** */
00003 /*
00004 *      Implementation of php's explode written in C
00005 *      Written by  Maz (2008)
00006 *      http://maz-programmersdiary.blogspot.com/
00007 *
00008 *      You're free to use this piece of code.
00009 *      You can also modify it freely, but if you
00010 *      improve this, you must write the improved code
00011 *      in comments at:
00012 *      http://maz-programmersdiary.blogspot.com/
00013 *      or at:
00014 *      http://c-ohjelmoiJanajatuksia.blogspot.com/
00015 *      or mail the corrected version to me at
00016 *      Mazziesaccount@gmail.com
00017 *
00018 *      Revision History:
```

```

00019 *
00020 * - 0.0.6 15.08.2009/Maz Fixed atomic CAS *
00021 * - 0.0.5 11.08.2009/Maz Added Cexplode_free_allButPieces *
00022 * - 0.0.4 11.08.2009/Maz Added atomic ops and *
00023 * mbot_ll *
00024 * -v0.0.3 31.07.2009/Maz Added Cexplode_concat *
00025 * (untested) *
00026 * -v0.0.2 21.07.2009/Maz Some additions for better *
00027 * usability in MazBotV4 *
00028 * -v0.0.1 16.09.2008/Maz *
00029 * */
00030 /* ***** */
00031
00032 #ifndef HELPERS_H
00033 #define HELPERS_H
00034
00035 /* Some Cexplode calls support using this special item define */
00036 #define CEXPLODE_LAST_ITEM 0xFFFFFFFF
00037
00038 #include <stdio.h>
00039 #include <string.h>
00040 #include <stdlib.h>
00041 #include <semaphore.h>
00042
00046 typedef struct MbotAtomic32
00047 {
00048     volatile unsigned int value;
00049     sem_t sem;
00050 }MbotAtomic32;
00051
00052
00056 typedef struct CexplodeStrings
00057 {
00058     int amnt;
00059     char **strings;
00060     char *separator;
00061     int sepwasatend;
00062     int startedWdelim;
00063     int index;
00064 }CexplodeStrings;
00065
00069 typedef enum ECexplodeRet
00070 {
00071     ECexplodeRet_InternalFailure = -666,
00072     ECexplodeRet_InvalidParams = -667
00073 }ECexplodeRet;
00074
00085 int Cexplode_removeCurrent(CexplodeStrings *exp_obj);
00086
00098 char *Cexplode_removeNth(int nro,CexplodeStrings *exp_obj);
00099
00108 int Cexplode_getAmnt(CexplodeStrings exp_obj);
00109
00122 int Cexplode(const char *string,const char *delim, CexplodeStrings *exp_obj );
00123
00132 int Cexplode_nextexists(CexplodeStrings exp_obj);
00133
00143 char *Cexplode_getNth(int index,CexplodeStrings *exp_obj);
00144
00152 char *Cexplode_getfirst(CexplodeStrings *exp_obj);
00153
00161 char *Cexplode_getnext(CexplodeStrings *exp_obj);
00169 char *Cexplode_getlast(CexplodeStrings *exp_obj);
00170
00178 void Cexplode_free(CexplodeStrings exp_obj);
00179
00186 void Cexplode_free_allButPieces(CexplodeStrings exp_obj);
00187
00195 size_t Cexplode_getlentilllast(CexplodeStrings exp_obj);
00196
00204 int Cexplode_sepwasatend(CexplodeStrings exp_obj);
00205
00214 int Cexplode_concat(CexplodeStrings *first,CexplodeStrings *second);
00215
00216
00221 int mbot_ltrim(char *text, char trimchar);
00222
00227 int mbot_rtrim(char *text, char trimchar);
00228
00233 int mbot_lrtrim(char *text, char trimchar);
00234
00239 int mbot_trimall(char *text, char trimchar);
00240
00244 MbotAtomic32 * MbotAtomic32Init();
00249 void MbotAtomic32Uninit(MbotAtomic32 **_this_);

```

```

00254 unsigned int mbot_atomicGet(MbotAtomic32* atomic);
00255
00260 unsigned int mbot_atomicAdd(MbotAtomic32* atomic,unsigned int addition);
00261
00266 unsigned int mbot_atomicDec(MbotAtomic32* atomic,unsigned int decrement);
00267
00272 unsigned int mbot_atomicDecIfGreater(MbotAtomic32* atomic,unsigned int decrement, unsigned int cmp);
00273
00278 unsigned int mbot_atomicDecIfSmaller(MbotAtomic32* atomic,unsigned int decrement, unsigned int cmp);
00279
00284 unsigned int mbot_atomicIncIfGreater(MbotAtomic32* atomic,unsigned int decrement, unsigned int cmp);
00285
00290 unsigned int mbot_atomicIncIfSmaller(MbotAtomic32* atomic,unsigned int decrement, unsigned int cmp);
00291
00292 #ifdef ARCH_x86
00293
00297 __inline__ unsigned int mbot_atomicCAS(MbotAtomic32* atomic, unsigned int old, unsigned int newval)
00298 {
00299     __asm__ __volatile__(
00300         "lock cmpxchgl %w0,%l" /* Swap value comp equals */
00301         : "+r"(newval)
00302         : "m"(atomic->value), "a"(old) /* According to Intel's manual, comp value must be in EAX register */
00303         : "memory");
00304
00305     return old;
00306 } /* AaCpuAtomicCmpSwap32 */
00307 #else
00308 unsigned int mbot_atomicCAS(MbotAtomic32* atomic, unsigned int old, unsigned int newval);
00309 #endif
00310
00311 /* Containers */
00312
00313 typedef struct mbot_linkedList
00314 {
00315     struct mbot_linkedList *head;
00316     struct mbot_linkedList *next;
00317     struct mbot_linkedList *prev;
00318     void *data;
00319 }mbot_linkedList;
00320
00324 mbot_linkedList *mbot_ll_init();
00328 mbot_linkedList * mbot_ll_get_prev(mbot_linkedList *_this);
00335 mbot_linkedList * mbot_ll_head_get(mbot_linkedList *_this);
00336
00340 mbot_linkedList * mbot_ll_get_next(mbot_linkedList *_this);
00341
00345 mbot_linkedList * mbot_ll_get_first(mbot_linkedList *_this);
00346
00350 mbot_linkedList * mbot_ll_get_last(mbot_linkedList *_this);
00351
00356 mbot_linkedList * mbot_ll_add(mbot_linkedList *_this,void *data);
00357
00362 mbot_linkedList * mbot_ll_release(mbot_linkedList *_this);
00363
00369 mbot_linkedList * mbot_ll_safe_release(mbot_linkedList *_this,void *data);
00370
00374 void * mbot_ll_dataGet(mbot_linkedList *_this);
00380 void * mbot_ll_dataSet(mbot_linkedList *_this,void *data);
00381
00386 mbot_linkedList * mbot_ll_seek(mbot_linkedList *_this, void *data, size_t datasize);
00387
00394 mbot_linkedList *mbot_ll_cpylist_wdata(mbot_linkedList *old,size_t itemsz);
00395
00399 void mbot_ll_destroy(mbot_linkedList **_this);
00400 #endif

```

Most recent version

Bug tracker

Maz - programmer's diary

Contact: Mazziesaccount@gmail.com

MazBotV4 Helpers API

- Cexplode
- Atomic operations
- Linked list

Copyright

```
/* *****  
/*  
/*      Implementation of php's explode written in C  
/*      Written by  Maz (2008)  
/*      Added Atomic operations for x86 architecture and  
/*      Linked list implementation.  
/*      Written by  Maz (2009-2010)  
/*      http://maz-programmersdiary.blogspot.com/  
/*  
/*      You're free to use this piece of code.  
/*      You can also modify it freely, but if you  
/*      improve this, you must write the improved code  
/*      in comments at:  
/*      http://maz-programmersdiary.blogspot.com/  
/*      or at:  
/*      http://c-ohjelmoi.janajatuksia.blogspot.com/  
/*      or mail the corrected version to me at  
/*      Mazziesaccount@gmail.com  
/*  
/*      Revision History:  
/*  
/*      - 0.0.6 15.08.2009/Maz  Fixed atomic CAS  
/*      - 0.0.5 11.08.2009/Maz  Added Cexplode_free_allButPieces  
/*      - 0.0.4 11.08.2009/Maz  Added atomic ops and  
/*                               mbot_ll  
/*      -v0.0.3 31.07.2009/Maz  Added Cexplode_concat  
/*                               (untested)  
/*      -v0.0.2 21.07.2009/Maz  Some additions for better  
/*                               usability in MazBotV4  
/*      -v0.0.1 16.09.2008/Maz  
/*  
/* *****
```

MazBotHelpers_Cexplode_etc Documentation

0.0.7 28.08.2009

- Main Page
- Data Structures
 - Data Structures
 - Data Fields
- Files
 - File List
 - Globals

Most recent version

Bug tracker

Maz - programmer's diary

Contact: *Mazziesaccount@gmail.com*

MazBotV4 Helpers API

- Cexplode
- Atomic operations
- Linked list

Copyright

```
/* *****  
/*  
/*      Implementation of php's explode written in C  
/*      Written by  Maz (2008)  
/*      Added Atomic operations for x86 architecture and  
/*      Linked list implementation.  
/*      Written by  Maz (2009-2010)  
/*      http://maz-programmersdiary.blogspot.com/  
/*  
/*      You're free to use this piece of code.  
/*      You can also modify it freely, but if you  
/*      improve this, you must write the improved code  
/*      in comments at:  
/*      http://maz-programmersdiary.blogspot.com/  
/*      or at:  
/*      http://c-ohjelmoi.janajatuksia.blogspot.com/  
/*      or mail the corrected version to me at  
/*      Mazziesaccount@gmail.com  
/*  
/*      Revision History:  
/*  
/*      - 0.0.6 15.08.2009/Maz  Fixed atomic CAS  
/*      - 0.0.5 11.08.2009/Maz  Added Cexplode_free_allButPieces  
/*      - 0.0.4 11.08.2009/Maz  Added atomic ops and  
/*                               mbot_ll  
/*      -v0.0.3 31.07.2009/Maz  Added Cexplode_concat  
/*                               (untested)  
/*      -v0.0.2 21.07.2009/Maz  Some additions for better  
/*                               usability in MazBotV4  
/*      -v0.0.1 16.09.2008/Maz  
/*  
/* *****
```

CexplodeStrings Struct Reference

Struct for Cexplode object. More...

```
#include <helpers.h>
```

Data Fields

```
    int amnt  
char ** strings  
    char * separator  
    int sepwasatend  
    int startedWdelim  
    int index
```

Detailed Description

Struct for Cexplode object.

Field Documentation

int amnt

int index

char* separator

int sepwasatend

int startedWdelim

char** strings

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

- `helpers.h`

Most recent version

Bug tracker

Maz - programmer's diary

Contact: *Mazziesaccount@gmail.com*

MazBotV4 Helpers API

- Cexplode
- Atomic operations
- Linked list

Copyright

```
/* *****  
/*  
/*      Implementation of php's explode written in C  
/*      Written by  Maz (2008)  
/*      Added Atomic operations for x86 architecture and  
/*      Linked list implementation.  
/*      Written by  Maz (2009-2010)  
/*      http://maz-programmersdiary.blogspot.com/  
/*  
/*      You're free to use this piece of code.  
/*      You can also modify it freely, but if you  
/*      improve this, you must write the improved code  
/*      in comments at:  
/*      http://maz-programmersdiary.blogspot.com/  
/*      or at:  
/*      http://c-ohjelmoi.janajatuksia.blogspot.com/  
/*      or mail the corrected version to me at  
/*      Mazziesaccount@gmail.com  
/*  
/*      Revision History:  
/*  
/*      - 0.0.6 15.08.2009/Maz  Fixed atomic CAS  
/*      - 0.0.5 11.08.2009/Maz  Added Cexplode_free_allButPieces  
/*      - 0.0.4 11.08.2009/Maz  Added atomic ops and  
/*                               mbot_ll  
/*      -v0.0.3 31.07.2009/Maz  Added Cexplode_concat  
/*                               (untested)  
/*      -v0.0.2 21.07.2009/Maz  Some additions for better  
/*                               usability in MazBotV4  
/*      -v0.0.1 16.09.2008/Maz  
/*  
/* *****
```

MbotAtomic32 Struct Reference

Struct for 32bit wide integer type used in atomic operations. More...

```
#include <helpers.h>
```

Data Fields

volatile unsigned int value

sem_t sem

If non x86 arch is used, these atomic ops are dummies using semaphore.

Detailed Description

Struct for 32bit wide integer type used in atomic operations.

Field Documentation

sem_t sem

If non x86 arch is used, these atomic ops are dummies using semaphore.

volatile unsigned int value

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

- [helpers.h](#)

Most recent version

Bug tracker

Maz - programmer's diary

Contact: Mazziesaccount@gmail.com

MazBotV4 Helpers API

- Cexplode
- Atomic operations
- Linked list

Copyright

```
/* *****  
/*  
/*      Implementation of php's explode written in C      *  
/*      Written by  Maz (2008)                             *  
/*      Added Atomic operations for x86 architecture and  *  
/*      Linked list implementation.                       *  
/*      Written by  Maz (2009-2010)                       *  
/*      http://maz-programmersdiary.blogspot.com/         *  
/*  
/*      You're free to use this piece of code.           *  
/*      You can also modify it freely, but if you        *  
/*      improve this, you must write the improved code   *  
/*      in comments at:                                   *  
/*      http://maz-programmersdiary.blogspot.com/         *  
/*      or at:                                           *  
/*      http://c-ohjelmoi.janajatuksia.blogspot.com/     *  
/*      or mail the corrected version to me at          *  
/*      Mazziesaccount@gmail.com                         *  
/*  
/*      Revision History:                                *  
/*  
/*      - 0.0.6 15.08.2009/Maz  Fixed atomic CAS          *  
/*      - 0.0.5 11.08.2009/Maz  Added Cexplode_free_allButPieces *  
/*      - 0.0.4 11.08.2009/Maz  Added atomic ops and      *  
/*                               mbot_ll                   *  
/*      -v0.0.3 31.07.2009/Maz  Added Cexplode_concat     *  
/*                               (untested)                 *  
/*      -v0.0.2 21.07.2009/Maz  Some additions for better *  
/*                               usability in MazBotV4      *  
/*      -v0.0.1 16.09.2008/Maz                               *  
/*  
/* *****
```

mbot_linkedList Struct Reference

```
#include <helpers.h>
```

Data Fields

```
struct mbot_linkedList * head  
struct mbot_linkedList * next  
struct mbot_linkedList * prev  
void * data
```

Field Documentation

`void* data`

`struct mbot_linkedList* head [read]`

`struct mbot_linkedList* next [read]`

`struct mbot_linkedList* prev [read]`

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

- `helpers.h`

[Most recent version](#)

[Bug tracker](#)

[Maz - programmer's diary](#)

Contact: Mazziesaccount@gmail.com