

---

# AuxUtils

Manual

<manual@majoron.com>

Copyright © 2007 Majoron.com

Revision History

2007.11.14

First published revision

Revision 1.0

## Table of Contents

Chapter 1. Introduction .....	2
Overview .....	2
Scope of this document .....	2
Target audience .....	2
Chapter 2. Installing AuxUtils .....	2
General installation issues .....	2
Installation from binaries .....	6
Installation from sources .....	7
Installation from development tree .....	8
How to report a bug or problem .....	8
How to report a feature request .....	9
Chapter 3. Library overview .....	10
Dynamic module routines .....	10
EPT routines .....	10
IO routines .....	11
IPC routines .....	11
Utils routines .....	11
Chapter 4. Common problems and errors .....	12
Installation issues .....	12
Common problems .....	12
Known bugs .....	12
A. Examples .....	13
B. Frequently asked questions .....	14
C. Roadmap .....	14
Roadmap for version 0.5 .....	14
D. Change log .....	16
Change Log for version 0.5 .....	16
E. Credits .....	19
Developers at AuxUtils .....	19
Contributors to AuxUtils .....	19
Documenters and translators .....	19
Libraries used by AuxUtils .....	19
Tools that were used to create AuxUtils .....	19
Supporters of AuxUtils .....	19
F. Public license .....	20
How to apply these terms to you new programs .....	23
How to use this license for you documentations .....	23
Index .....	24

# Chapter 1. Introduction

## Overview

The project AuxUtils is an auxiliary C++ library that implements the functionality which doesn't exist or had limitations in the Boost libraries. The library provides some efficient functions and templates, wrap some platform dependent functions.

Officially supported platforms are Windows, and Linux. The AuxUtils library may be ported to other platforms by demand.

Supported language is C++.

## Scope of this document

This document describes version 0.5 of the AuxUtils library. It has list the goals and objectives to archive, contains library implementation overview and a few examples and tutorials in order to begin to use library in application easy.

## Target audience

AuxUtils library intended for designed and developers of the software applications that have some restriction or limitation at Boost functionality, which AuxUtils is eliminated.

## Glossary

EPT	Effective Programming Templates
IO	Input output
IPC	Inter process communication

## References

[1] *AuxUtils Doxygen APIs. Majoron (www.majoron.com). Available at www.majoron.com..*

[2] *AuxUtils Documentation. Majoron (www.majoron.com). Available at www.majoron.com..*

# Chapter 2. Installing AuxUtils

This section describes how to obtain and install AuxUtils library. A summary of the procedure follows and later sections provide the details.

## General installation issues

The AuxUtils installation procedure depends on distribution package type and platforms on which distribution you will install.

The immediate following sections contain the information necessary to choose, download and verify you distribution. The instructions in the later sections describe how to install the distribution that you choose. For binary distributions, see the instructions at section "Installation from binaries" for appropriate operation system or installation type (RPM or DEB). To build AuxUtils from source codes see the instruction at section "Installation from sources" for appropriate operation system. To install AuxUtils from development tree see the instruction at section "Installation from development tree".

## Supported operation systems

Officially supported platforms are:

- Windows
- Linux

GNU Autoconf and Automake are used to build AuxUtils on Unix-like operation systems, so it is possible to port AuxUtils to all moderns operation systems that have a C++ compiler and dependent packages.

## How to get AuxUtils

Check out our download page at [www.majoron.com](http://www.majoron.com) for information about the current version binary or source distributions and for downloading instructions. You can download these packages from primary site [www.majoron.com](http://www.majoron.com) or select appropriate mirrors at [www.sourceforge.net](http://www.sourceforge.net).

Obtaining the development three isn't possible for now. But will be added in the nearest future.

## Verifying the package integrity

After you have downloaded the AuxUtils package that suits you needs and before you attempt to install it, you should make sure that it is intact and has not been tampered with. There are a few means of the integrity checking:

- MD5 checksums
- For RPM packages, the built-in RPM integrity verification mechanism

## Verifying MD5 checksums

After you have downloaded a AuxUtils package, you should make sure that its MD5 checksum matches the one provided on the AuxUtils download pages. Each package has an individual checksum that you can verify with the following command, where `package_name` is the name of the package you downloaded:

```
shell> md5sum package_name
```

### Example 1. Verifying MD5 checksums

```
shell>md5sum libAuxUtils-src-0.5.0.tar.bz2
shell> aaab65abbec64d5e907dcd41b8699945 AuxUtils-src-0.5.0.tar.bz2
```

You should verify that the resulting checksum (the string of hexadecimal digits) matches the one displayed on the download page immediately below the respective package.

Note that not all operating systems support the `md5sum` command. On some, it is simply called `md5`, and others do not ship it at all. On Linux, it is part of the GNU Text Utilities package, which is available for a wide range of platforms. You can download the source code from <http://www.gnu.org/software/textutils/> as well. If you have OpenSSL installed, you can use the command `openssl md5 package_name` instead. A Windows implementation of the `md5` command line utility is available from <http://www.fourmilab.ch/md5/>. `winMd5Sum` is a graphical MD5 checking tool that can be obtained from <http://www.nullriver.com/index/products/winmd5sum>.

## Signature checking using RPM

For RPM packages, there is no separate signature. RPM packages have a built-in GPG signature and MD5 checksum. You can verify a package by running the following command::

```
shell> rpm --checksig package_name.rpm
```

## Example 2. Signature checking using RPM

```
shell> rpm --checksig libAuxUtils-0.5-linux-2.6-intel.rpm
libAuxUtils-0.5-linux-2.6-intel.rpm: sha1 md5 OK
```

## Installation layouts

This section describe the default layout of the directories created by installing binary or source distributions provided at Majoron. A distributions provided by another vendor might use layout different from those shown here.

### Windows installation layouts

Denote AUXUTILS\_ROOT as installation directory of the AuxUtils project. The installation directory has the following subdirectories:

**Table 1. Windows installation layouts**

Directory	Content of the directory
AUXUTILS_ROOT/docs	AuxUtils documentation.
AUXUTILS_ROOT/include	Header files to include at you projects.
AUXUTILS_ROOT/include/dynmod	Header files to work with dynamical modules.
AUXUTILS_ROOT/include/ept	Header files to work thread pools, timers and object pools.
AUXUTILS_ROOT/include/io	Header files to work with IO routines.
AUXUTILS_ROOT/include/ipc	Header files to work with IPC routines.
AUXUTILS_ROOT/include/utils	Header files that contains some utilities.
AUXUTILS_ROOT/lib	Libraries files to link with you applications.

### Unix-like installation layouts

On Unix-like operation systems sources distribution after building and installing, binary distribution and RPM or DEB distributions after installing create similar directory layouts. By default, but you can configure installation location, the installation step installs files under /usr/local, in the following subdirectories:

**Table 2. Unix-like installation layouts**

Directory	Content of the directory
/usr/local/include/AuxUtils	Header files to include at you projects.
/usr/local/include/AuxUtils/dynmod	Header files to work with dynamical modules.
/usr/local/include/AuxUtils/ept	Header files to work thread pools, timers and object pools.
/usr/local/include/AuxUtils/io	Header files to work with IO routines.
/usr/local/include/AuxUtils/ipc	Header files to work with IPC routines.
/usr/local/include/AuxUtils/utils	Header files that contains some utilities.
/usr/local/lib/AuxUtils	Libraries files to link with you applications.

### Sources distribution layouts

After extracting a sources distribution to AUXUTILS\_ROOT, it will have the following subdirectories:

**Table 3. Sources distribution layouts**

Directory	Content of the directory
AUXUTILS_ROOT/AuxUtils/	AuxUtils project files needed to use it in own projects.
AUXUTILS_ROOT/Documents/	AuxUtils manual and others documentations.
AUXUTILS_ROOT/TestDynmod/	The simplest dynamical module to test AuxUtils library.
AUXUTILS_ROOT/UnitTests/	Unit test to check AuxUtils library.
AUXUTILS_ROOT/Workspaces/	Workspaces for building projects.

To use AuxUtils at you project you need files located at AUXUTILS\_ROOT/AuxUtils and has the following subdirectories:

**Table 4. AuxUtils layouts**

Directory	Content of the directory
AUXUTILS_ROOT/AuxUtils/docs	Doxygen files to create APIs.
AUXUTILS_ROOT/AuxUtils/include	Header files to include at you projects.
AUXUTILS_ROOT/AuxUtils/include/dynmod	Header files to work with dynamical modules.
AUXUTILS_ROOT/AuxUtils/include/ept	Header files to work thread pools, timers and object pools.
AUXUTILS_ROOT/AuxUtils/include/io	Header files to work with IO routines.
AUXUTILS_ROOT/AuxUtils/include/ipc	Header files to work with IPC routines.
AUXUTILS_ROOT/AuxUtils/include/utills	Header files that contains some utilities.
AUXUTILS_ROOT/AuxUtils/res	Resources file dependents on operation system.
AUXUTILS_ROOT/AuxUtils/src	Source files of the AuxUtils project.
AUXUTILS_ROOT/AuxUtils/src/dynmod	Source files to work with dynamical modules.
AUXUTILS_ROOT/AuxUtils/src/ept	Source files to work thread pools, timers and object pools.
AUXUTILS_ROOT/AuxUtils/src/io	Source files to work with IO routines.
AUXUTILS_ROOT/AuxUtils/src/ipc	Source files to work with IPC routines.
AUXUTILS_ROOT/AuxUtils/src/utills	Source files that contains some utilities.

The simplest dynamical module is located at AUXUTILS\_ROOT/TestDynmod and has the following subdirectories:

**Table 5. TestDynmod layouts**

Directory	Content of the directory
AUXUTILS_ROOT/TestDynmod/docs	Doxygen files to create APIs.
AUXUTILS_ROOT/TestDynmod/include	Header files to include at you projects.
AUXUTILS_ROOT/TestDynmod/res	Resources file dependents on operation system.
AUXUTILS_ROOT/TestDynmod/src	Sources file of the AuxUtils project.

Unit tests to verify library located at AUXUTILS\_ROOT/UnitTests and has the following subdirectories:

**Table 6. UnitTest layouts**

Directory	Content of the directory
AUXUTILS_ROOT/UnitTests/docs	Doxygen files to create APIs.
AUXUTILS_ROOT/UnitTests/include	Header files to include at you projects.
AUXUTILS_ROOT/UnitTests/res	Resources file dependents on operation system.
AUXUTILS_ROOT/UnitTests/src	Sources file of the AuxUtils project.

Workspaces to build library from source located at AUXUTILS\_ROOT/Workspaces and has the following subdirectories:

**Table 7. Workspaces layouts**

Directory	Content of the directory
AUXUTILS_ROOT/Workspaces/autotools	Workspaces for building projects by GNU Autoconf and Automake tools.
AUXUTILS_ROOT/Workspaces/distrib	Workspaces for building projects and create distribution packages.
AUXUTILS_ROOT/Workspaces/eclipse	Workspaces for building projects by Eclipse development IDE.
AUXUTILS_ROOT/Workspaces/msvc8	Workspaces for building projects by MSVC 8.x.
AUXUTILS_ROOT/Workspaces/msvc9	Workspaces for building projects by MSVC 9.x.

## Installation from binaries

The immediate following sections describe how to install AuxUtils library from binary distribution under different platforms.

### Windows installation from binaries

Unpack distribution package using archiver, for example WinZip, to destination directory.

### Unix-like installation from binaries

Unpack distribution package using archiver to destination directory. You can use tar command to unpack distribution, where package\_name is the name of the package you downloaded.

```
shell> tar -xjvf package_name
```

#### Example 3. Unpacking binaries

```
shell>tar -xjvf libAuxUtils-bin-linux-i386-0.5.0.tar.bz2
```

### Unix-like installation from RPM

The recommended way to install AuxUtils on RPM-based Linux distributions is by using the RPM packages. The RPMs provided by Majoron to the community should work on all versions of Linux that support RPM packages. To obtain RPM packages, see section "How to get AuxUtils".

To install RPM distribution you can use rpm command, where package\_name is the name of the package you downloaded:

```
shell> rpm -i package_name
```

**Example 4. Installation from RPM**

```
shell>rmp -i libAuxUtils-0.5-linux-2.6-intel.rpm
```

**Example 5. Viewing the content of the RPM**

```
shell>rmp -qpl libAuxUtils-0.5-linux-2.6-intel.rpm
```

**Unix-like installation from DEB**

The recommended way to install AuxUtils on DEB-based Linux distributions is by using the DEB packages. The DEBs provided by Majoron to the community should work on all versions of Linux that support DEB packages. To obtain DEB packages, see section "How to get AuxUtils".

To install DEB distribution you can use dpkg command, where package\_name is the name of the package you downloaded:

```
shell> dpk - i package_name
```

**Example 6. Installation from DEB**

```
shell>dpkg -i libAuxUtils-0.5-linux-2.6-intel.deb
```

**Example 7. Viewing the content of the DEB**

```
shell>dkpg -l libAuxUtils-0.5-linux-2.6-intel.deb
```

**Installation from sources**

The immediate following sections describe how to build and install AuxUtils library from source distribution under different platforms.

**Windows installation from sources**

Unpack distribution package using archivator, for example WinZip, to destination directory. After unpacking distribution you need a build library. In order to do it, select suitable projects file from Workspaces directory and build library.

**Unix-like installation from sources**

Unpack distribution package using archivator to destination directory. You can use tar command to unpack distribution, where package\_am is the name of the package you downloaded.

```
shell> tar -xjvf package_name
```

**Example 8. Unpacking sources**

```
shell>tar -xjvf libAuxUtils-src-0.5.0.tar.bz2
```

After unpacking a distribution package you need to build it. To do it use GNU Autoconf and Automake files that located at Worspaces/autotools directory. After a building you may optionally to install header files and libraries.

## Typical configure options

**Table 8. Typical configure options**

Configure option	Option description
--enable-debug	Building a binaries with debug information.
--with-cppflags	User specific compiler's flags.
--with-ldflags	User specific linker's flags.

## Installation from development tree

Obtaining the development tree and hence installation based on development tree isn't possible for now. But will be added in the nearest future.

## How to report a bug or problem

Before posting a bug report about a problem, please try to verify that it is a bug and that it has not been reported already:

- Start by searching the AuxUtils online manual at [www.majoron.com](http://www.majoron.com). We try to keep the manual up to date by updating it frequently with solutions to newly found problems. The change history located at [www.majoron.com](http://www.majoron.com) can be particularly useful since it is quite possible that a newer version contains a solution to your problem.
- For solutions to some common problems, see chapter 7: "Problems and common errors". This chapter contains information about well-known problems.
- Search the bugs database at [www.majoron.com](http://www.majoron.com) to see whether the bug has been reported and fixed.
- Search the forums database at [www.majoron.com](http://www.majoron.com) to see related threads. Here you could discuss the problems with other user and get a help.

If you can't find an answer in the manual, the bugs database, or the forum, please use the following guidelines for reporting the bug. The normal way to report bugs is to visit [www.majoron.com/](http://www.majoron.com/), which is the address for our bugs database. This database is public and can be browsed and searched by anyone. If you log in to the system, you can enter new reports to the bugs database.

Writing a good bug report takes patience, but doing it right the first time saves time both for us and for yourself. A good bug report, containing a full test case for the bug, makes it very likely that we will fix the bug in the next release. This section helps you write your report correctly so that you don't waste your time doing things that may not help us much or at all. Please read this section carefully and make sure that all the information described here is included in your report.

It is most helpful when a good description of the problem is included in the bug report. That is, give a good example of everything you did that led to the problem and describe, in exact detail, the problem itself. The best reports are those that include a full example showing how to reproduce the bug or problem.

Remember that it is possible for us to respond to a report containing too much information, but not to one containing too little. People often omit facts because they think they know the cause of a problem and assume that some details don't matter. A good principle to follow is that if you are in doubt about stating something, state it. It is faster and less troublesome to write a couple more lines in your report than to wait longer for the answer if we must ask you to provide information that was missing from the initial report.

The most common errors made in bug reports are (a) not including the version number of the AuxUtils distribution that you use, and (b) not fully describing the platform on which the AuxUtils library is installed (including the platform type and version number). These are highly relevant pieces of information, and in 99 cases out of 100, the bug report is useless without them. Very often we get

questions like, "Why doesn't this work for me?" Then we find that the feature requested wasn't implemented in that AuxUtils version, or that a bug described in a report has been fixed in newer AuxUtils versions. Errors often are platform-dependent. In such cases, it is next to impossible for us to fix anything without knowing the operating system and the version number of the platform.

If you compiled AuxUtils from source, remember also to provide information about your compiler if it is related to the problem. Often people find bugs in compilers and think the problem is AuxUtils-related. Most compilers are under development all the time and become better version by version. To determine whether your problem depends on your compiler, we need to know what compiler you used. Note that every compiling problem should be regarded as a bug and reported accordingly.

Please include the following information in your report:

- The version number of the AuxUtils distribution you are using (for example, AuxUtils-0.5). You can find out which version you are using at the name of the distribution.
- The manufacturer and model of the machine on which you experience the problem.
- The operating system name and version. If you work with Windows, you can usually get the name and version number by double-clicking your My Computer icon and pulling down the "Help/About Windows" menu. For most Unix-like operating systems, you can get this information by executing the command **uname -a**.
- Sometimes the amount of memory (real and virtual) is relevant. If in doubt, include these values.
- If you are using a source distribution of the AuxUtils, include the name and version number of the compiler that you used. If you have a binary distribution, include the distribution name.
- If the problem occurs during compilation, include the exact error messages and also a few lines of context around the offending code in the file where the error occurs.
- If you have a patch for a bug, do include it. But don't assume that the patch is all we need, or that we can use it, if you don't provide some necessary information such as test cases showing the bug that your patch fixes. We might find problems with your patch or we might not understand it at all. If so, we can't use it. If we can't verify the exact purpose of the patch, we won't use it. Test cases help us here. Show that the patch handles all the situations that may occur. If we find a borderline case (even a rare one) where the patch won't work, it may be useless.
- Guesses about what the bug is, why it occurs, or what it depends on are usually wrong. Even the AuxUtils team can't guess such things without first using a debugger to determine the real cause of a bug.
- If possible, download and install the most recent version of AuxUtils and check whether it solves your problem. All versions of the AuxUtils library are thoroughly tested and should work without problems. We believe in making everything as backward-compatible as possible, and you should be able to switch AuxUtils versions without difficulty.

## How to report a feature request

Before posting a feature request, please try to verify that it has not been reported already:

- See appendix: "Project Roadmap" to that feature request has not been included at roadmap already. Also please visit online roadmap at [www.majoron.com](http://www.majoron.com) to check at the latest version..
- Search the forums database at [www.majoron.com](http://www.majoron.com) to see has this feature request discussed already.
- Search the bugs database at [www.majoron.com](http://www.majoron.com) to see has this feature request discussed already.

In case this feature request is not included at roadmap yet and after researching the bugs database and forums you are sure that it is useful than submit it bugs database located at [www.majoron.com](http://www.majoron.com). Don't forget setup Severity at enhancement value. Please include the following information in your feature request demand:

- Use case of the feature request.
- Why do you think that it is a responsibility of the AuxUtils?
- How are you going to use it?
- Why do you think it will be useful for others users?

## Chapter 3. Library overview

This chapter contains information about base library objects. It covers what exists at AuxUtils library and if necessary that it covers implementation topics. AuxUtils library consists a few parts, which are discussed in the following sections.

### Dynamic module routines

This section overview a dynamic modules routines implemented at AuxUtils.

#### Dynamic module overview

Dynamic module routines server to load a dynamic module. The module is a library that implemented as a dynamic(.dll) on Windows or shared(.so) on Unix-like system library .

The functionality implemented in a follow classes:

- **IModule** This class declare interface to work this dynamic module.
- **ModuleImpl** This class implements interface to work with dynamic module which was declare at IModule.
- **ModulesFactory** This class is a module factory and serve to load specified module to get declare at module interface.

### EPT routines

This section overview a EPT routines implemented at AuxUtils.

#### EPT overview

EPT routines server to work with a times, thread pools and pool objects. These routines allow to execute task on thread pool and subscribe timers.

The functionality implemented in a follow classes:

- **IObjectPoolItemFactory** This class is an interface to pol item factory. It is declare create, reset and destroy interface.
- **ObjectPool** Objects pool shares the object instances for more efficiently using memory management. Objects instance in pool are never recreated.
- **ObjectPoolPtrItemFactory** implement IObjectPoolItemFactory interface to work with object which was created at memory heap.
- **thread pool** This class encapsulate working with tasks. It allow to reserve a some threads to execute specified task. Also it has functionality to manage increase and decrease the quantity of the thread inside. Also it has flexible mechanism to assign a specified task thread to reduce hardware loading.
- **Timer** Class time implement itself times and allow to work with times: create new timer, receive notification about time-out and cancel subscribed before timer.

## IO routines

This section overview a IO routines implemented at AuxUtils.

### IO overview

IO routines server to work with a data to transfers, streams.

The functionality implemented in a follow classes:

- **ByteArrayInputStream** This class implement `InputStream` and `IOOutput` stream interface. It is convenient as duplex stream: it allow to read data from a stream and write data into stream. Such types of the behaviour useful for using in protocols tasks .
- **ByteBuffer** This class is a buffer and has method to work with data inside: append new data or cut current data and etc. Also it allow to manage increasing and decreasing the memory consumption and reserving memory. This class allow to work with external memory buffers or can manage own buffer inside.
- **InputStream** This class declare interface to work with input stream. It allow to write a portions of data into stream. It allow to write a portion of the data to stream.
- **OutputStream** This class declare interface to work with output stream. It allow to read a portion of the data from stream. Also class declare appropriate interface to check is it some data inside steam to read.
- **StreamReframer** This class allow to re framer a stream. For example one source send packets with size 1024Kb at network, but destination require packet with size at 2048Kb. There is a task to re framer received packets before sent. This class allow to do it.

## IPC routines

This section overview a IPC routines implemented at AuxUtils.

### IPC overview

IPC routines server to work with a inter process communication. These routines wrap primitives to work with a shared memory to cross platforms classes.

The functionality implemented in a follow classes:

- **SharedCondition** This class encapsulate shared condition inside. It has boost like interface to work with. There isn't implementation at boos library such synchronization primitive.
- **SharedMemory** This class encapsulate shared memory inside. It has boost like interface to work with. There isn't implementation at boos library such synchronization primitive.
- **SharedMutex** This class encapsulate shared mutex inside. It has boost like interface to work with. There isn't implementation at boos library such synchronization primitive.
- **SharedSemaphore** This class encapsulate shared semaphore inside. It has boost like interface to work with. There isn't implementation at boos library such synchronization primitive.

## Utils routines

This section overview a utility routines implemented at AuxUtils.

### Utils overview

Utils routines server to work with sequence generator, operation system process and time.

The functionality implemented in a follow classes:

- **OSProcTools** This routines encapsulate working with processes and port across different platforms. These routines allow to start new process and wait until started before process will be finished.
- **OSTimeTools** This routines simplify to working with boost time functions. They are adapters to boost on which they are based.
- **SequenceGenerator** This class is a sequence generator to create new id. It is flexible to configure strategy to receive a new identity. This allow to created new id one by one, or with some gaps. As template parameter it accepted any class which has right interface. Also this class has default template parameter which implement default behaviour.
- **Version** This class is an auxiliary and allow to work with module versions.

## Chapter 4. Common problems and errors

This chapter contains information about common problems and errors and has three sections. The first section contains common and well-know problems and solutions for it. The second section contains information about installation problems and solution for it. The last one contain information about known bugs at current release.

### Installation issues

Here is installation issues problems and it's solutions.

#### Doesn't pass all unit tests.

On some operation system shared memory test may fail due to operation system configuration. Read section about shared memory in detail and check how to configure you operation system shared memory relative options.

### Common problems

Here is well-known problems and it's solutions.

#### Unable create a SharedMemory object

Some operation system shared memory has limitation due to operation system configuration. Read section about shared memory in detail and check how to configure you operation system shared memory relative options.

### Known bugs

To see the latest list of the known please visit the Kwon bugs page at [www.majoron.com](http://www.majoron.com). This page contain the latest information.

Follow notation is used at change log. Each entry begin with a version, for which entry for created. Then follow component at { }. Then follow category of the bug at []. It can be bug report, feature request and etc. Than follow bug number at bug tracking system between // signs. And then follow a description of the entry. For example entry 1.0 { AntHill Core } [ Feature Request ] / 380 / STLport support mean than entry was created for 1.0 version of the AntHill Core component and entry is feature request with 380 number at bug tracking system to support STLPort library.

#### Known bugs for release 0.5

- 1.0 { AntHill Core } [ --- ] / 241 / Hwm & Lwm Coordinator

- 1.0 { AntHill Core } [ --- ] / 234 / Add a reuse port at query for tcp & udp
- 1.0 { AntHill Core } [ --- ] / 207 / Add support a projects files for msvc 6,7 & 9
- 1.0 { AntHill Core } [ --- ] / 133 / Add keep alive library support
- 1.0 { AntHill Core } [ --- ] / 126 / Port library to java
- 1.0 { AntHill Core } [ --- ] / 125 / Add xot & x25 transport
- 1.0 { AntHill Core } [ --- ] / 124 / Add a endpoint or list of endpoints for acceptor
- 1.0 { AntHill Core } [ --- ] / 123 / Add hmw & lwm support on acceptor
- 1.0 { Setup } [ --- ] / 245 / Include eclipse project to source distribution
- 0.5 { AntHill Core } [ --- ] / 275 / Reorganize a unit test set
- 0.5 { AntHill Core } [ --- ] / 272 / Unable to close connection from IReliableConnection
- 0.5 { AntHill Core } [ --- ] / 261 / Create interface to protocol coder.
- 0.5 { AntHill Core } [ --- ] / 147 / Stop connection where Acceptor::stop
- 0.5 { Documentation } [ --- ] / 273 / Reorganize documentation set
- 0.5 { Documentation } [ --- ] / 266 / Write about diameter protocol
- 0.5 { Documentation } [ --- ] / 264 / Write about fictive udp session
- 0.5 { Setup } [ --- ] / 276 / Install diameter include files during make - install
- 0.5 { Setup } [ --- ] / 274 / Reorganize installation set
- 0.5 { Setup } [ --- ] / 248 / Implement md5 check sum & rpm signature
- 0.5 { Transport UDP } [ --- ] / 265 / Default start without any parameter for udp
- 0.5 { Transport UDP } [ --- ] / 137 / Add a custom errors
- 0.5 { Samples } [ --- ] / 267 / Write sample for using diameter protocol
- 0.5 { Samples } [ --- ] / 236 / Create a sample how to write a protocol
- 0.5 { Protocol Diameter } [ --- ] / 278 / Changes names for commands
- 0.5 { Protocol Diameter } [ --- ] / 268 / Add a support for custom avp

## A. Examples

AuxUtils is a well-documented library. A lot of examples provided at [1] document. Also the good way to understand how these entities are working is unit test. Almost all functionality has appropriate unit test. This examples illustrate how to use a AuxUtils library at different aspects. Among examples are:

- Working with IO object such as ByteBuffer, InputStream, OutputStream.
- Working with modules such as ModuleFactory and IModule.
- Working with shared objects such as SharedMemory, SharedMutex and SharedSemaphore.
- Working with timers, tasks and tread pool objects such as Timer and ThreadPool.
- Working with sequence identity objects such as SequenceGenerator.

## B. Frequently asked questions

To see the latest list of the frequently asked questions please visit the F.A..Q. page at [www.majoron.com](http://www.majoron.com). This page contains the latest information.

- *What is the target group? Who are users?*

AuxUtils library intended for designed and developers of the software applications that have some restriction or limitation at Boost functionality, which AuxUtils is eliminated.

- *Is the AuxUtils thread-safe?*

Yes. The AuxUtils library was designed as thread safe library until you are using thread safe STL implementation.

- *All unit tests dont' pass.*

On some operation system shared memory test may fail due to operation system configuration. Read section about shared memory in detail and check how to configure you operation system shared memory relative options.

## C. Roadmap

To see the latest list of the roadmap please visit the Roadmap page at [www.majoron.com](http://www.majoron.com). This page contain the latest information.

Follow notation is used at change log. Each entry begin with a version, for which entry for created. Then follow component at { }. Then follow category of the bug at []. It can be bug report, feature request and etc. Than follow bug number at bug tracking system between // signs. And then follow a description of the entry. For example entry 1.0 { AntHill Core } [ Feature Request ] / 380 / STLport support mean than entry was created for 1.0 version of the AntHill Core component and entry is feature request with 380 number at bug tracking system to support STLPort library.

### Roadmap for version 0.5

- 0.5 { AntHill Core } [ --- ] / 256 / Invalid memory management in case restore connection return false.
- 0.5 { AntHill Core } [ --- ] / 255 / Deadlock while sending message
- 0.5 { AntHill Core } [ --- ] / 240 / StdScalingCoordinator with same weight
- 0.5 { AntHill Core } [ --- ] / 239 / StdFailOverCoordinator with same weight
- 0.5 { AntHill Core } [ --- ] / 235 / Change IReliableClient\* to IReliableClient::SharedPtr
- 0.5 { AntHill Core } [ --- ] / 218 / Extending interface IReliableConnectionListener for error reporting with OS specific message
- 0.5 { AntHill Core } [ --- ] / 217 / Scaling coordinator
- 0.5 { AntHill Core } [ --- ] / 210 / Add unit test: a sending a big buffers
- 0.5 { AntHill Core } [ --- ] / 194 / Erase asio macros \_WIN32\_WINNT from all projects
- 0.5 { AntHill Core } [ --- ] / 171 / Coding style: constant localtion
- 0.5 { AntHill Core } [ --- ] / 164 / Change license from majoron to latestbit.
- 0.5 { AntHill Core } [ --- ] / 145 / Replace typedef boost::shared\_ptr<Socket> SocketSharedPtr with pointer

- 0.5 { AntHill Core } [ --- ] / 142 / Remove Socket Shared Ptr at UDP & TCP
- 0.5 { AntHill Core } [ --- ] / 141 / Add resource with version for win32 for all libraries
- 0.5 { AntHill Core } [ --- ] / 135 / Implement ability set all options for tcp & udp by query
- 0.5 { AntHill Core } [ --- ] / 134 / enlarge interface allow to get error message
- 0.5 { AntHill Core } [ --- ] / 132 / Use po\_Socket->async\_read instead asio::async\_read(poSocket)
- 0.5 { AntHill Core } [ --- ] / 131 / Add atomic op on m\_nQueueSize
- 0.5 { AntHill Core } [ --- ] / 130 / Implement query parsing at EndpointURI
- 0.5 { AntHill Core } [ --- ] / 127 / Replace magic string on constan string
- 0.5 { AntHill Core } [ --- ] / 122 / AntHill::TransportFactoryBase create and impelent
- 0.5 { Documentation } [ --- ] / 262 / Add a thread model for scaling/fail-over layer
- 0.5 { Documentation } [ --- ] / 254 / place topic about restore connection
- 0.5 { Documentation } [ --- ] / 243 / Code format.
- 0.5 { Documentation } [ --- ] / 238 / Basic concepts for each layers
- 0.5 { Documentation } [ --- ] / 237 / Mixing a ray and connection inside coordinator
- 0.5 { Documentation } [ --- ] / 151 / hwm & lwm processing
- 0.5 { Documentation } [ --- ] / 149 / Depict connector & acceptor & connection id fro shmем
- 0.5 { Documentation } [ --- ] / 120 / Declare thread model and thread safe behavior at documentation.
- 0.5 { Setup } [ --- ] / 246 / Documentation(pdf) is not included at source distribution
- 0.5 { Setup } [ --- ] / 229 / add --with-debug-enabled options for configure to add -g
- 0.5 { Setup } [ --- ] / 154 / Create automake & autoconf installation for linux
- 0.5 { Transport UDP } [ --- ] / 230 / Select between sendto & connect way for UDP
- 0.5 { Transport UDP } [ --- ] / 214 / Rename output file as libAntHillUDP
- 0.5 { Transport UDP } [ --- ] / 212 / Rename output file as libAntHillTCP
- 0.5 { Transport UDP } [ --- ] / 139 / Collect query params at one place
- 0.5 { Transport TCP } [ --- ] / 233 / processing WSAEWOLDBLOCK
- 0.5 { Transport TCP } [ --- ] / 221 / localport & connnectorPort
- 0.5 { Transport TCP } [ --- ] / 209 / Extra schedule and hence invalid test
- 0.5 { Transport TCP } [ --- ] / 203 / Inv following to coding style:
- 0.5 { Transport TCP } [ --- ] / 143 / Coding styly incompatibility
- 0.5 { Transport TCP } [ --- ] / 129 / Not effective async\_read
- 0.5 { Transport Shmem } [ --- ] / 232 / Replace local variables oByteBuffer onto member( for performance issues?).
- 0.5 { Transport Shmem } [ --- ] / 231 / extract methods
- 0.5 { Transport Shmem } [ --- ] / 213 / Rename output file as libAntHillShmem
- 0.5 { Transport Shmem } [ --- ] / 201 / Create ShmemSharedResource class

- 0.5 { Transport Shmem } [ --- ] / 200 / Incorrect onConnected event
- 0.5 { Transport Shmem } [ --- ] / 199 / Data lost due to event
- 0.5 { Transport Shmem } [ --- ] / 198 / Lost data (need one mutex more?)
- 0.5 { Transport Shmem } [ --- ] / 197 / Memory Id Acceptor generation.
- 0.5 { Transport Shmem } [ --- ] / 196 / Double sending disconnect
- 0.5 { Transport Shmem } [ --- ] / 192 / boost::bind & events
- 0.5 { Transport Shmem } [ --- ] / 168 / Use boost serialization instead of memcpy
- 0.5 { Transport Shmem } [ --- ] / 167 / remove memory copy for deliver command
- 0.5 { Transport Shmem } [ --- ] / 166 / Add a timeout waiting connection & sending
- 0.5 { Transport Shmem } [ --- ] / 165 / Create a factory for ICommand interface
- 0.5 { Transport Shmem } [ --- ] / 155 / Create a full duplex instead of semi duplex communication
- 0.5 { Transport Shmem } [ --- ] / 150 / calculate and set acceptor & connector memory volume and setup it
- 0.5 { Transport Shmem } [ --- ] / 146 / pass memory at query to shared memory
- 0.5 { Transport Shmem } [ --- ] / 144 / communication between different languages based on ASN.1
- 0.5 { Samples } [ --- ] / 136 / Create example chat
- 0.5 { AuxLibrary Sockets } [ --- ] / 226 / Simplify gethostname\_r
- 0.5 { AuxLibrary Sockets } [ --- ] / 220 / new implementation setNoneBlockMode
- 0.5 { AuxLibrary Sockets } [ --- ] / 215 / Move constants
- 0.5 { AntHill Core } [ Feature Request ] / 219 / Needed some sample(s) for scaling/failover layer
- 0.5 { AntHill Core } [ Feature Request ] / 216 / HWM/LWM implementation for failover/scaling level
- 0.5 { AuxLibrary Sockets } [ Feature Request ] / 211 / Migration implementation from TCP transport to AntHillSockets library for reuse in UDP

## D. Change log

To see the latest list of the change log please visit the Change Log page at [www.majoron.com](http://www.majoron.com). This page contain the latest information.

Follow notation is used at change log. Each entry begin with a version, for which entry for created. Then follow component at { }. Then follow category of the bug at []. It can be bug report, feature request and etc. Than follow bug number at bug tracking system between // signs. And then follow a description of the entry. For example empty 1.0 { AntHill Core } [ Feature Request ] / 380 / STLport support mean than entry was created for 1.0 version of the AntHill Core component and empty is feature request with 380 number at bug tracking system to support STLPort library.

## Change Log for version 0.5

- 0.5 { AntHill Core } [ --- ] / 256 / Invalid memory managment in case restoreConnection return false.
- 0.5 { AntHill Core } [ --- ] / 255 / Deadlock while sending message

- 0.5 { AntHill Core } [ --- ] / 240 / StdScalingCoordinator with same weight
- 0.5 { AntHill Core } [ --- ] / 239 / StdFailOverCoordinator with same weight
- 0.5 { AntHill Core } [ --- ] / 235 / Change IReliableClient\* to IReliableClient::SharedPtr
- 0.5 { AntHill Core } [ --- ] / 218 / Extending interface IReliableConnectionListener for error reporting with OS specific message
- 0.5 { AntHill Core } [ --- ] / 217 / Scaling coordinator
- 0.5 { AntHill Core } [ --- ] / 210 / Add unit test: a sending a big buffers
- 0.5 { AntHill Core } [ --- ] / 194 / Erase asio macros \_WIN32\_WINNT from all projects
- 0.5 { AntHill Core } [ --- ] / 171 / Coding style: constant localtion
- 0.5 { AntHill Core } [ --- ] / 164 / Change license from majoron to latestbit.
- 0.5 { AntHill Core } [ --- ] / 145 / Replace typedef boost::shared\_ptr<Socket> SocketSharedPtr with pointer
- 0.5 { AntHill Core } [ --- ] / 142 / Remove Socket Shared Ptr at UDP & TCP
- 0.5 { AntHill Core } [ --- ] / 141 / Add resource with version for win32 for all libraries
- 0.5 { AntHill Core } [ --- ] / 135 / Implement ability set all options for tcp & udp by query
- 0.5 { AntHill Core } [ --- ] / 134 / enlarge interface allow to get error message
- 0.5 { AntHill Core } [ --- ] / 132 / Use po\_Socket->async\_read instead asio::async\_read(poSocket)
- 0.5 { AntHill Core } [ --- ] / 131 / Add atomic op on m\_nQueueSize
- 0.5 { AntHill Core } [ --- ] / 130 / Implement query parsing at EndpointURI
- 0.5 { AntHill Core } [ --- ] / 127 / Replace magic string on constan string
- 0.5 { AntHill Core } [ --- ] / 122 / AntHill::TransportFactoryBase create and impelent
- 0.5 { Documentation } [ --- ] / 262 / Add a thread model for scaling/fail-over layer
- 0.5 { Documentation } [ --- ] / 254 / place topic about restore connection
- 0.5 { Documentation } [ --- ] / 243 / Code format.
- 0.5 { Documentation } [ --- ] / 238 / Basic concepts for each layers
- 0.5 { Documentation } [ --- ] / 237 / Mixing a ray and connection inside coordinator
- 0.5 { Documentation } [ --- ] / 151 / hwm & lwm processing
- 0.5 { Documentation } [ --- ] / 149 / Depict connector & acceptor & connection id fro shmem
- 0.5 { Documentation } [ --- ] / 120 / Declare thread model and thread safe behavior at documentation.
- 0.5 { Setup } [ --- ] / 246 / Documentation(pdf) is not included at source distribution
- 0.5 { Setup } [ --- ] / 229 / add --with-debug-enabled options for configure to add -g
- 0.5 { Setup } [ --- ] / 154 / Create automake & autoconf installation for linux
- 0.5 { Transport UDP } [ --- ] / 230 / Select between sendto & connect way for UDP
- 0.5 { Transport UDP } [ --- ] / 214 / Rename output file as libAntHillUDP
- 0.5 { Transport UDP } [ --- ] / 212 / Rename output file as libAntHillTCP

- 0.5 { Transport UDP } [ --- ] / 139 / Collect query params at one place
- 0.5 { Transport TCP } [ --- ] / 233 / processing WSAEWOLDBLOCK
- 0.5 { Transport TCP } [ --- ] / 221 / localport & connectorPort
- 0.5 { Transport TCP } [ --- ] / 209 / Extra schedule and hence invalid test
- 0.5 { Transport TCP } [ --- ] / 203 / Inv following to coding style:
- 0.5 { Transport TCP } [ --- ] / 143 / Coding styly incompatibility
- 0.5 { Transport TCP } [ --- ] / 129 / Not effective async\_read
- 0.5 { Transport Shmem } [ --- ] / 232 / Replace local variables oByteBuffer onto member( for performance issues?).
- 0.5 { Transport Shmem } [ --- ] / 231 / extract methods
- 0.5 { Transport Shmem } [ --- ] / 213 / Rename output file as libAntHillShmem
- 0.5 { Transport Shmem } [ --- ] / 201 / Create ShmemSharedResource class
- 0.5 { Transport Shmem } [ --- ] / 200 / Incorrect onConnected event
- 0.5 { Transport Shmem } [ --- ] / 199 / Data lost due to event
- 0.5 { Transport Shmem } [ --- ] / 198 / Lost data (need one mutex more?)
- 0.5 { Transport Shmem } [ --- ] / 197 / Memory Id Acceptor generation.
- 0.5 { Transport Shmem } [ --- ] / 196 / Double sending disconnect
- 0.5 { Transport Shmem } [ --- ] / 192 / boost::bind & events
- 0.5 { Transport Shmem } [ --- ] / 168 / Use boost serialization instead of memcpy
- 0.5 { Transport Shmem } [ --- ] / 167 / remove memory copy for deliver command
- 0.5 { Transport Shmem } [ --- ] / 166 / Add a timeout waiting connection & sending
- 0.5 { Transport Shmem } [ --- ] / 165 / Create a factory for ICommand interface
- 0.5 { Transport Shmem } [ --- ] / 155 / Create a full duplex instead of semi duplex communication
- 0.5 { Transport Shmem } [ --- ] / 150 / calculate and set acceptor & connector memory volume and setup it
- 0.5 { Transport Shmem } [ --- ] / 146 / pass memory at query to shared memory
- 0.5 { Transport Shmem } [ --- ] / 144 / communication between different languages based on ASN.1
- 0.5 { Samples } [ --- ] / 136 / Create example chat
- 0.5 { AuxLibrary Sockets } [ --- ] / 226 / Simplify gethostbyname\_r
- 0.5 { AuxLibrary Sockets } [ --- ] / 220 / new implementation setNoneBlockMode
- 0.5 { AuxLibrary Sockets } [ --- ] / 215 / Move constants
- 0.5 { AntHill Core } [ Feature Request ] / 219 / Needed some sample(s) for scaling/failover layer
- 0.5 { AntHill Core } [ Feature Request ] / 216 / HWM/LWM implementation for failover/scaling level
- 0.5 { AuxLibrary Sockets } [ Feature Request ] / 211 / Migration implementation from TCP transport to AntHillSockets library for reuse in UDP

## E. Credits

This appendix contains lists the developers, contributors, and supporters that have helped to make AuxUtils what it is today.

### Developers at AuxUtils

The following people have created AuxUtils.

- **Abdulla Abdurakhmanov** Lead developer and main author of the AuxUtils. The main ideas and the first version of the library was created by Abdulla Abdurakhmanov
- **Artem Rufanov** Lead developer of the AuxUtils. He is support the library and responsible for future development and adjusting to requirement for AntHill and other libraries which were and will be create by Majoron.

### Contributors to AuxUtils

The following people have helped as to do AuxUtils better.

- **You can be here.**

### Documenters and translators

The following people have helped as to write documentation for AuxUtils.

- **Artem Rufanov.** Main write of the AuxUtils manual.
- **Roman Rufanov** Active contributor to the AuxUtils manual.

### Libraries used by AuxUtils

AuxUtils is a tiny library but meantime it is using other open source libraries. The following is a list of some of the libraries we have used to create AuxUtils. We use this to express our thanks to those that has created them as without these we could not have made AuxUtils what it is today.

- **Boost** From whom we got an excellent boost library.

### Tools that were used to create AuxUtils

The following is a list of some of the tools we have used to create AuxUtils. We use this to express our thanks to those that has created them as without these we could not have made AuxUtils what it is today.

- **Free Software Foundation** From whom we got an excellent compiler (gcc), an excellent debugger (gdb).
- **Doxygen** This development tools allow to create documentation for C++. It is used to create a well-formated html API documentation.

### Supporters of AuxUtils

We wish to thanks the following companies, which helped us finance the development of the AuxUtils library such as by paying us for developing a new feature or giving us hardware for development of the AuxUtils server.

- **You can be here**

## F. Public license

Apache License  
Version 2.0, January 2004  
<http://www.apache.org/licenses/>

### TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

#### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent

to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
  - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
  - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
  - (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
  - (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or

documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.
6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf

of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

## How to apply these terms to you new programs

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

```
Copyright [yyyy] [name of copyright owner]
```

```
Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at
```

```
http://www.apache.org/licenses/LICENSE-2.0
```

```
Unless required by applicable law or agreed to in writing, software  
distributed under the License is distributed on an "AS IS" BASIS,  
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.  
See the License for the specific language governing permissions and  
limitations under the License.
```

## How to use this license for you documentations

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

```
Copyright [yyyy] [name of copyright owner]
```

```
Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at
```

```
http://www.apache.org/licenses/LICENSE-2.0
```

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

# Index

## A

AuxUtils, 2

## B

ByteArrayInputStream, 11

ByteBuffer, 11

## I

InputStream, 11

IModule, 10

IObjectPoolItemFactory, 10

OutputStream, 11

## M

ModuleImpl, 10

ModulesFactory, 10

## O

ObjectPool, 10

ObjectPoolPtrItemFactory, 10

OSProcTools, 12

OSTimeTools, 12

## S

SequenceGenerator, 12

SharedCondition, 11

SharedMemory, 11, 11

SharedMutex, 11

SharedSemaphore, 11

StreamReframer, 11

## T

ThreadPool, 10

Timer, 10

## V

Version, 12