

# ZeroMQ

[return to the de facto Standards bodies](#)

**Source:** [ØMQ - The Guide](#)

Table 1: Data Sheet for ZeroMQ (ØMQ).

Characteristic	Value
Original author(s)	Pieter Hintjens
Developer(s)	iMatrix
Initial release	2010
Stable release	4.3.1
API Documentation	<a href="http://api.zeromq.org/4-3:_start">http://api.zeromq.org/4-3:_start</a>
Repository	<a href="http://zeromq.org/intro:get-the-software">http://zeromq.org/intro:get-the-software</a> or <a href="https://github.com/zeromq/libzmq">https://github.com/zeromq/libzmq</a>
Written in	Mainly in C, but also in PHP, Java, Python, Lua, and Haxe
Example Languages	C++, C#, CL, Delphi, Erlang, F#, Felix, Haskell, Objective-C, Ruby, Ada, Basic, Clojure, Go, Haxe, Node.js, ooc, Perl, and Scala
Operating system	runs on most operating systems
Guide	<a href="http://zguide.zeromq.org/page:all">http://zguide.zeromq.org/page:all</a>
Available in	English and 27 languages other language
Type	Messaging Service
License	LGPLv3
Website	<a href="http://zeromq.org/">http://zeromq.org/</a>

## Abstract

*ZeroMQ (also known as ØMQ, 0MQ, or zmq) looks like an embeddable networking library but acts like a concurrency framework. It gives you sockets that carry atomic messages across various transports like in-process, inter-process, TCP, and multicast. You can connect sockets N-to-N with patterns like fan-out, pub-sub, task distribution, and request-reply. It's fast enough to be the fabric for clustered products. Its asynchronous I/O model gives you scalable multicore applications, built as asynchronous message-processing tasks. It has a score of language APIs and runs on most operating systems. ZeroMQ is from iMatix and is LGPLv3 open source.*

From:  
<https://www.omgwiki.org/dido/> - **DIDO Wiki**

Permanent link:  
[https://www.omgwiki.org/dido/doku.php?id=dido:public:ra:xapend:xapend.b\\_stds:defact:zeromq:start&rev=1559611427](https://www.omgwiki.org/dido/doku.php?id=dido:public:ra:xapend:xapend.b_stds:defact:zeromq:start&rev=1559611427)

Last update: **2019/06/03 21:23**

