ZeroMQ Distributed Messaging

return to the de facto Standards area

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	http://api.zeromq.org/4-3:_start
Repository	http://zeromq.org/intro:get-the-software or https://github.com/zeromq/libzmq
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	http://zguide.zeromq.org/page:all
Available in	English and 27 languages other language
Type	Messaging Service
License	LGHPLv3
Website	http://zeromq.org/

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://omgwiki.org/dido/ - **DIDO Wiki**

Permanent link:
https://omgwiki.org/dido/doku.php?id=dido:public:ra:xapend:xapend.b_stds:defact:zeromq:start&rev=1605251733

Last update: **2020/11/13 02:15**