

[Accelerate](#) / [vecLib](#)

vecLib

Perform computations on large vectors.

Overview

The vecLib framework contains nine C header files (not counting `vecLib.h` which merely includes the others). Two of them, `vDSP.h` and `vDSP_translate.h`, are covered in [vDSP Programming Guide](#) and [vDSP](#).

Three of the header files are Apple's versions of well-known libraries which are described in detail in external references:

- `cbLAS.h` and `vbLAS.h` are the interfaces to Apple's implementations of BLAS. You can find reference documentation in [BLAS](#). Additional documentation on the BLAS standard, including reference implementations, can be found on the web starting from the BLAS FAQ page at these URLs: <http://www.netlib.org/blas/faq.html> and <http://www.netlib.org/blas/blast-forum/blast-forum.html>.
- `clapack.h` is the interface to Apple's implementation of LAPACK. Documentation of the LAPACK interfaces, including reference implementations, can be found on the web starting from the LAPACK FAQ page at this URL: <http://netlib.org/lapack/faq.html>

This document describes the functions declared in the remaining header files: `vecLibTypes.h`, `vfp.h`, `vForce.h`, `vBasicOps.h`, `vectorOps.h`, and `vBigNum.h`. These files support the vector mathematical functions library (also called "vMathLib"), which runs on vector processing hardware if available.

This library abstracts the vector processing capability so that code written for it will execute appropriate instructions for the processor available at runtime. For this reason, unless you are writing specialized code that targets a single CPU, you should generally use these functions rather than directly using vector instructions. By using these functions, your code does not have to deal with subtle differences in vector instruction availability between different microarchitectures, freeing you to focus on the problem you are trying to solve. Also, code written using vecLib tends to be easier to port to different CPU architectures (porting your core libraries to ARM on iOS, for example).

Topics

vecLibTypes

- ⋮ vecLibTypes
Create and work with 128-bit vector data types.

vBasicOps

- ⋮ vBasicOps
Perform basic arithmetic and logical functions on 128-bit vectors.

vfp

- ⋮ vfp
Perform floating-point arithmetic, transcendental, and trigonometric functions on 128-bit vectors.

vForce

- ⋮ vForce
Perform transcendental and trigonometric functions on vectors of any length.

vectorOps

- ⋮ vectorOps
Perform vector and matrix BLAS functions on arrays of 128-bit vectors.

vBigNum

- ⋮ vBigNum
Perform arithmetic and logical functions on large integers.

Macros

