vecLib Reference Update

Performance > Vector Engines

2005-06-04

Contents

4

Introduction to vecLib Reference Update

This document summarizes the symbols that have been added to the vecLib framework. The full reference documentation notes in what version a symbol was introduced, but sometimes it's useful to see only the new symbols for a given release.

If you are not familiar with this framework you should refer to the complete framework reference documentation.

Organization of This Document

Symbols are grouped by class or protocol for Objective-C and by header file for C. For each symbol there is a link to complete documentation, if available, and a brief description, if available.

See Also

See "vDSP Library."

vecLib.framework

Various sub-framework APIs have been updated with new functionality.

C Symbols

All of the header files with new symbols in `vecLib.framework` are listed alphabetically, with their new symbols described.

cblas.h

A number of new functions are introduced.

Functions

All of the new functions in this header file are listed alphabetically, with links to documentation and abstracts, if available.

`ATLU_DestroyThreadMemory`	
`catlas_caxpby`	
`catlas_cset`	
`catlas_daxpby`	
`catlas_dset`	
`catlas_saxpby`	
`catlas_sset`	
`catlas_zaxpby`	
`catlas_zset`	

clapack.h

A number of new functions are introduced.

Functions

All of the new functions in this header file are listed alphabetically, with links to documentation and abstracts, if available.

`cgelsd_`	
`cgelss_`	
`cgesdd_`	
`cgesvd_`	
`chbgvd_`	
`chetd2_`	
`cladiv_`	
`clalsd_`	
`clangb_`	
`clange_`	
`clangt_`	
`clanhb_`	
`clanhe_`	
`clanhp_`	
`clanhs_`	
`clanht_`	
`clansb_`	
`clansp_`	
`clansy_`	
`clantb_`	
`clantp_`	
`clantr_`	
`cpteqr_`	
`cstegr_`	
`dlamc3_`	
`dlamch_`	

dlangb_	
dlange_	
dlangt_	
dlanhs_	
dlansb_	
dlansp_	
dlanst_	
dlansy_	
dlantb_	
dlantp_	
dlantr_	
dlapy2_	
dlapy3_	
dsecnd_	
dzsum1_	
lsame_	
lsamen_	
scsum1_	
second_	
slamc3_	
slamch_	
slangb_	
slange_	
slangt_	
slanhs_	
slansb_	
slansp_	
slanst_	

slansy_	
slantb_	
slantp_	
slantr_	
slapy2_	
slapy3_	
zgelsd_	
zgesdd_	
zgesvd_	
zhbgvd_	
zhetd2_	
zladiv_	
zlalsd_	
zlangb_	
zlange_	
zlangt_	
zlanhb_	
zlanhe_	
zlanhp_	
zlanhs_	
zlanht_	
zlansb_	
zlansp_	
zlansy_	
zlantb_	
zlantp_	
zlantr_	
zpteqr_	

```
zstegr_
```

vDSP.h

Many new functions are introduced to provide new functionality.

Functions

All of the new functions in this header file are listed alphabetically, with links to documentation and abstracts, if available.

`vDSP_acor`	
`vDSP_acorD`	Autocorrelation with automatic selection of domain.
`vDSP_acorf`	
`vDSP_acorfD`	Frequency-domain autocorrelation.
`vDSP_acort`	
`vDSP_acortD`	Time-domain autocorrelation.
`vDSP_blkman`	
`vDSP_blkmanD`	
`vDSP_blkman_window`	Creates a Blackman window.
`vDSP_blkman_windowD`	Creates a Blackman window.
`vDSP_ccor`	
`vDSP_ccorD`	
`vDSP_ccorf`	
`vDSP_ccorfD`	
`vDSP_ccort`	
`vDSP_ccortD`	
`vDSP_deq22`	Difference equation, 2 poles, 2 zeros.
`vDSP_deq22D`	Difference equation, 2 poles, 2 zeros.
`vDSP_desamp`	Convolution with decimation.
`vDSP_desampD`	Convolution with decimation.
`vDSP_hamm`	

`vDSP_hammD`	
`vDSP_hamm_window`	Creates a Hamming window.
`vDSP_hamm_windowD`	Creates a Hamming window.
`vDSP_hann`	
`vDSP_hannD`	
`vDSP_hann_window`	Creates a Hanning window.
`vDSP_hann_windowD`	Creates a Hanning window.
`vDSP_maxmgv`	Vector maximum magnitude.
`vDSP_maxmgvD`	Vector maximum magnitude.
`vDSP_maxmgvi`	Vector maximum magnitude with index.
`vDSP_maxmgviD`	Vector maximum magnitude with index.
`vDSP_maxv`	Vector maximum value.
`vDSP_maxvD`	Vector maximum value.
`vDSP_maxvi`	Vector maximum value with index.
`vDSP_maxviD`	Vector maximum value with index.
`vDSP_meamgv`	Vector mean magnitude.
`vDSP_meamgvD`	Vector mean magnitude.
`vDSP_meanv`	Vector mean value.
`vDSP_meanvD`	Vector mean value.
`vDSP_measqv`	Vector mean square value.
`vDSP_measqvD`	Vector mean square value.
`vDSP_minmgv`	Vector minimum magnitude.
`vDSP_minmgvD`	Vector minimum magnitude.
`vDSP_minmgvi`	Vector minimum magnitude with index.
`vDSP_minmgviD`	Vector minimum magnitude with index.
`vDSP_minv`	Vector minimum value.
`vDSP_minvD`	Vector minimum value.
`vDSP_minvi`	Vector minimum value with index.

vDSP_minviD	Vector minimum value with index.
vDSP_mmov	The contents of a submatrix are copied to another submatrix.
vDSP_mmovD	The contents of a submatrix are copied to another submatrix.
vDSP_mvessq	Vector mean of signed squares.
vDSP_mvessqD	Vector mean of signed squares.
vDSP_nzcros	Find zero crossings.
vDSP_nzcrosD	Find zero crossings.
vDSP_polar	Rectangular to polar conversion.
vDSP_polarD	Rectangular to polar conversion.
vDSP_rect	Polar to rectangular conversion.
vDSP_rectD	Polar to rectangular conversion.
vDSP_rmsqv	Vector root-mean-square.
vDSP_rmsqvD	Vector root-mean-square.
vDSP_svdiv	Divide scalar by vector.
vDSP_svdivD	Divide scalar by vector.
vDSP_sve	Vector sum.
vDSP_sveD	Vector sum.
vDSP_svemg	Vector sum of magnitudes.
vDSP_svemgD	Vector sum of magnitudes.
vDSP_svesq	Vector sum of squares.
vDSP_svesqD	Vector sum of squares.
vDSP_svs	Vector sum of signed squares.
vDSP_svsD	Vector sum of signed squares.
vDSP_vaam	Vector add, add, and multiply.
vDSP_vaamD	Vector add, add, and multiply.
vDSP_vabs	Vector absolute values.
vDSP_vabsD	Vector absolute values.
vDSP_vabsi	Integer vector absolute values.

`vDSP_vasbm`	Vector add, subtract, and multiply.
`vDSP_vasbmD`	Vector add, subtract, and multiply.
`vDSP_vasm`	Vector add and scalar multiply.
`vDSP_vasmD`	Vector add and scalar multiply.
`vDSP_vavlin`	Vector linear average.
`vDSP_vavlinD`	Vector linear average.
`vDSP_vclip`	Vector clip.
`vDSP_vclipD`	Vector clip.
`vDSP_vclipc`	Vector clip and count.
`vDSP_vclipcD`	Vector clip and count.
`vDSP_vclr`	Vector clear.
`vDSP_vclrD`	Vector clear.
`vDSP_vcmprs`	Vector compress.
`vDSP_vcmprsD`	Vector compress.
`vDSP_vdbcon`	Vector convert power or amplitude to decibels.
`vDSP_vdbconD`	Vector convert power or amplitude to decibels.
`vDSP_vdist`	Vector distance.
`vDSP_vdistD`	Vector distance.
`vDSP_vdiv`	Vector divide.
`vDSP_vdivD`	Vector divide.
`vDSP_vdivi`	Vector divide.
`vDSP_vdpsp`	Vector convert double-precision to single-precision.
`vDSP_venvlp`	Vector envelope.
`vDSP_venvlpD`	Vector envelope.
`vDSP_veqvi`	Vector equivalence, 32-bit logical.
`vDSP_vfill`	Vector fill.
`vDSP_vfillD`	Vector fill.
`vDSP_vfilli`	Integer vector fill.

vDSP_vfix16	
vDSP_vfix16D	
vDSP_vfix32	
vDSP_vfix32D	
vDSP_vfix8	
vDSP_vfix8D	
vDSP_vfixr16	
vDSP_vfixr16D	
vDSP_vfixr32	
vDSP_vfixr32D	
vDSP_vfixr8	
vDSP_vfixr8D	
vDSP_vfixru16	
vDSP_vfixru16D	
vDSP_vfixru32	
vDSP_vfixru32D	
vDSP_vfixru8	
vDSP_vfixru8D	
vDSP_vfixu16	
vDSP_vfixu16D	
vDSP_vfixu32	
vDSP_vfixu32D	
vDSP_vfixu8	
vDSP_vfixu8D	
vDSP_vflt16	
vDSP_vflt16D	
vDSP_vflt32	
vDSP_vflt32D	

`vDSP_vflt8`	
`vDSP_vflt8D`	
`vDSP_vfltu16`	
`vDSP_vfltu16D`	
`vDSP_vfltu32`	
`vDSP_vfltu32D`	
`vDSP_vfltu8`	
`vDSP_vfltu8D`	
`vDSP_vfrac`	Vector truncate to fraction.
`vDSP_vfracD`	Vector truncate to fraction.
`vDSP_vgathr`	Vector gather.
`vDSP_vgathrD`	Vector gather.
`vDSP_vgathra`	Vector gather, absolute pointers.
`vDSP_vgathraD`	Vector gather, absolute pointers.
`vDSP_vgen`	Vector tapered ramp.
`vDSP_vgenD`	Vector tapered ramp.
`vDSP_vgenp`	Vector generate by extrapolation and interpolation.
`vDSP_vgenpD`	Vector generate by extrapolation and interpolation.
`vDSP_viclip`	Vector inverted clip.
`vDSP_viclipD`	Vector inverted clip.
`vDSP_vindex`	Vector index.
`vDSP_vindexD`	Vector index.
`vDSP_vintb`	Vector linear interpolation between vectors.
`vDSP_vintbD`	Vector linear interpolation between vectors.
`vDSP_vlim`	Vector test limit.
`vDSP_vlimD`	Vector test limit.
`vDSP_vlint`	Vector linear interpolation between neighboring values.
`vDSP_vlintD`	Vector linear interpolation between neighboring values.

vDSP_vma	Vector multiply and add.
vDSP_vmaD	Vector multiply and add.
vDSP_vmax	Vector maxima.
vDSP_vmaxD	Vector maxima.
vDSP_vmaxmg	Vector maximum magnitudes.
vDSP_vmaxmgD	Vector maximum magnitudes.
vDSP_vmin	Vector minima.
vDSP_vminD	Vector minima.
vDSP_vminmg	Vector minimum magnitudes.
vDSP_vminmgD	Vector minimum magnitudes.
vDSP_vmma	Vector multiply, multiply, and add.
vDSP_vmmaD	Vector multiply, multiply, and add.
vDSP_vmmsb	Vector multiply, multiply, and subtract.
vDSP_vmmsbD	Vector multiply, multiply, and subtract.
vDSP_vmsa	Vector multiply and scalar add.
vDSP_vmsaD	Vector multiply and scalar add.
vDSP_vmsb	Vector multiply and subtract.
vDSP_vmsbD	Vector multiply and subtract.
vDSP_vnabs	Vector negative absolute value.
vDSP_vnabsD	Vector negative absolute value.
vDSP_vneg	Vector negative value.
vDSP_vnegD	Vector negative value.
vDSP_vpoly	Vector polynomial.
vDSP_vpolyD	Vector polynomial.
vDSP_vpythg	Vector pythagoras.
vDSP_vpythgD	Vector pythagoras.
vDSP_vqint	Vector quadratic interpolation.
vDSP_vqintD	Vector quadratic interpolation.

`vDSP_vramp`	Build ramped vector.
`vDSP_vrampD`	Build ramped vector.
`vDSP_vrsum`	Vector running sum integration.
`vDSP_vrsumD`	Vector running sum integration.
`vDSP_vrvrs`	Vector reverse order, in place.
`vDSP_vrvrsD`	Vector reverse order, in place.
`vDSP_vsadd`	Vector scalar add.
`vDSP_vsaddD`	Vector scalar add.
`vDSP_vsaddi`	Integer vector scalar add.
`vDSP_vsbm`	Vector subtract and multiply.
`vDSP_vsbmD`	Vector subtract and multiply.
`vDSP_vsbsbm`	Vector subtract, subtract, and multiply.
`vDSP_vsbsbmD`	Vector subtract, subtract, and multiply.
`vDSP_vsbsm`	Vector subtract and scalar multiply.
`vDSP_vsbsmD`	Vector subtract and scalar multiply.
`vDSP_vsdiv`	Vector scalar divide.
`vDSP_vsdivD`	Vector scalar divide.
`vDSP_vsdivi`	Integer vector scalar divide.
`vDSP_vsimps`	Simpson integration.
`vDSP_vsimpsD`	Simpson integration.
`vDSP_vsma`	Vector scalar multiply and vector add.
`vDSP_vsmaD`	Vector scalar multiply and vector add.
`vDSP_vsmsa`	Vector scalar multiply and scalar add.
`vDSP_vsmsaD`	Vector scalar multiply and scalar add.
`vDSP_vsmsb`	Vector scalar multiply and vector subtract.
`vDSP_vsmsbD`	Vector scalar multiply and vector subtract.
`vDSP_vsort`	Vector in-place sort.
`vDSP_vsortD`	Vector in-place sort.

`vDSP_vsorti`	Vector integer in-place sort.
`vDSP_vsortiD`	Vector integer in-place sort.
`vDSP_vspdp`	Vector convert single-precision to double-precision.
`vDSP_vswap`	Vector swap.
`vDSP_vswapD`	Vector swap.
`vDSP_vswsum`	Vector sliding window sum.
`vDSP_vswsumD`	Vector sliding window sum.
`vDSP_vtabi`	Vector interpolation, table lookup.
`vDSP_vtabiD`	Vector interpolation, table lookup.
`vDSP_vthr`	Vector threshold.
`vDSP_vthrD`	Vector threshold.
`vDSP_vthres`	Vector threshold with zero fill.
`vDSP_vthresD`	Vector threshold with zero fill.
`vDSP_vthrsc`	Vector threshold with signed constant.
`vDSP_vthrscD`	Vector threshold with signed constant.
`vDSP_vtmerg`	Vector tapered merge of two vectors.
`vDSP_vtmergD`	Vector tapered merge of two vectors.
`vDSP_vtrapz`	Vector trapezoidal integration.
`vDSP_vtrapzD`	Vector trapezoidal integration.
`vDSP_wiener`	Wiener-Levinson general convolution.
`vDSP_wienerD`	Wiener-Levinson general convolution.
`vDSP_zaspec`	Computes an accumulating autospectrum.
`vDSP_zaspecD`	Computes an accumulating autospectrum.
`vDSP_zcoher`	Coherence function of two signals.
`vDSP_zcoherD`	Coherence function of two signals.
`vDSP_zcspec`	Accumulating cross-spectrum on two complex vectors.
`vDSP_zcspecD`	Accumulating cross-spectrum on two complex vectors.
`vDSP_zrdesamp`	Complex/real downsample with anti-aliasing.

`vDSP_zrdesampD`	Complex/real downsample with anti-aliasing.
`vDSP_zrvdiv`	Divides complex vector A by real vector B and leaves the result in vector C.
`vDSP_zrvdivD`	Divides complex vector A by real vector B and leaves the result in vector C.
`vDSP_ztrans`	Transfer function.
`vDSP_ztransD`	Transfer function.
`vDSP_ztransD`	Transfer function.
`vDSP_zvabs`	Complex vector absolute value.
`vDSP_zvabsD`	Complex vector absolute value.
`vDSP_zvcmul`	Complex vector conjugate and multiply.
`vDSP_zvcmulD`	Complex vector conjugate and multiply.
`vDSP_zvconj`	Complex vector conjugate.
`vDSP_zvconjD`	Complex vector conjugate.
`vDSP_zvdiv`	Complex vector divide.
`vDSP_zvdivD`	Complex vector divide.
`vDSP_zvfill`	Complex vector fill.
`vDSP_zvfillD`	Complex vector fill.
`vDSP_zvmags`	Complex vector magnitudes squared.
`vDSP_zvmagsD`	Complex vector magnitudes squared.
`vDSP_zvmgsa`	Complex vector magnitudes square and add.
`vDSP_zvmgsaD`	Complex vector magnitudes square and add.
`vDSP_zvmov`	Complex vector move.
`vDSP_zvmovD`	Complex vector move.
`vDSP_zvneg`	Complex vector negate.
`vDSP_zvnegD`	Complex vector negate.
`vDSP_zvphas`	Complex vector phase.
`vDSP_zvphasD`	Complex vector phase.
`vDSP_zvsma`	Complex vector scalar multiply and add.

vDSP_zvsmaD	Complex vector scalar multiply and add.
vDSP_zvzsml	Complex vector multiply by complex scalar.
vDSP_zvzsmlD	Complex vector multiply by complex scalar.

Data Types and Constants

All of the new data types and constants in this header file are listed alphabetically, with links to documentation and abstracts, if available.

FFTSetupD	

vForce.h

A number of new functions are introduced.

Functions

All of the new functions in this header file are listed alphabetically, with links to documentation and abstracts, if available.

vvacos	For each double-precision array element, sets y to the arccosine of x.
vvacosf	For each single-precision array element, sets y to the arccosine of x.
vvacosh	For each double-precision array element, sets y to the inverse hyperbolic cosine of x.
vvacoshf	For each single-precision array element, sets y to the inverse hyperbolic cosine of x.
vvasin	For each double-precision array element, sets y to the arcsine of x.
vvasinf	For each single-precision array element, sets y to the arcsine of x.
vvasinh	For each double-precision array element, sets y to the inverse hyperbolic sine of x.
vvasinhf	For each single-precision array element, sets y to the inverse hyperbolic sine of x.
vvatan	For each double-precision array element, sets y to the arctangent of x.
vvatan2	For each double-precision array element, sets z to the arctangent of y/x.
vvatan2f	For each single-precision array element, sets z to the arctangent of y/x.
vvatanf	For each single-precision array element, sets y to the arctangent of x.

vvatanh	For each double-precision array element, sets y to the inverse hyperbolic tangent of x.
vvatanhf	For each single-precision array element, sets y to the inverse hyperbolic tangent of x.
vvceil	For each double-precision array element, sets y to the ceiling of x.
vvceilf	For each single-precision array element, sets y to the ceiling of x.
vvcos	For each double-precision array element, sets y to the cosine of x.
vvcosf	For each single-precision array element, sets y to the cosine of x.
vvcosh	For each double-precision array element, sets y to the hyperbolic cosine of x.
vvcoshf	For each single-precision array element, sets y to the hyperbolic cosine of x.
vvcosisin	For each double-precision array element, sets the real part of C to the sine of x and the imaginary part of C to the cosine of x.
vvcosisinf	For each single-precision array element, sets the real part of C to the sine of x and the imaginary part of C to the cosine of x.
vvdiv	For each double-precision array element, sets z to y/x.
vvdivf	For each single-precision array element, sets z to y/x.
vvexp	For each double-precision array element, sets y to the exponential of x.
vvexpf	For each single-precision array element, sets y to the exponential of x.
vvfloor	For each double-precision array element, sets y to the floor of x.
vvfloorf	For each single-precision array element, sets y to the floor of x.
vvint	For each double-precision array element, sets y to the integer truncation of x.
vvintf	For each single-precision array element, sets y to the integer truncation of x.
vvlog	For each double-precision array element, sets y to the natural logarithm of x.
vvlog10	For each double-precision array element, sets y to the base 10 logarithm of x.
vvlog10f	For each single-precision array element, sets y to the base 10 logarithm of x.
vvlogf	For each single-precision array element, sets y to the natural logarithm of x.
vvnint	For each double-precision array element, sets y to the nearest integer to x.
vvnintf	For each single-precision array element, sets y to the nearest integer to x.
vvpow	For each double-precision array element, sets z to x raised to the power of y.
vvpowf	For each single-precision array element, sets z to x raised to the power of y.

C Symbols

`vvrec`	For each double-precision array element, sets y to the reciprocal of y.
`vvrecf`	For each single-precision array element, sets y to the reciprocal of y.
`vvrsqrt`	For each double-precision array element, sets y to the reciprocal of the square root of x.
`vvrsqrtf`	For each single-precision array element, sets y to the reciprocal of the square root of x.
`vvsin`	For each double-precision array element, sets y to the sine of x.
`vvsincos`	For each double-precision array element, sets z to the sine of x and y to the cosine of x.
`vvsincosf`	For each single-precision array element, sets z to the sine of x and y to the cosine of x.
`vvsinf`	For each single-precision array element, sets y to the sine of x.
`vvsinh`	For each double-precision array element, sets y to the hyperbolic sine of x.
`vvsinhf`	For each single-precision array element, sets y to the hyperbolic sine of x.
`vvsqrt`	For each double-precision array element, sets y to the square root of x.
`vvsqrtf`	For each single-precision array element, sets y to the square root of x.
`vvtan`	For each double-precision array element, sets y to the tangent of x.
`vvtanf`	For each single-precision array element, sets y to the tangent of x.
`vvtanh`	For each double-precision array element, sets y to the hyperbolic tangent of x.
`vvtanhf`	For each single-precision array element, sets y to the hyperbolic tangent of x.

Document Revision History

This table describes the changes to *vecLib Reference Update*.

Date	Notes
2005-06-04	New document that summarizes the symbols added to the vecLib framework in Mac OS X v10.4.