# XNA Math Cheatsheet

*Version 1.1 © 2010-06-13 Adam Sawicki*
*Based on DirectX SDK June 2010*

#include <**Xnamath.h**>

## Types

| | |
|---|---|
| **XMVECTOR** | __m128 / __vector4 |
| **XMVECTORF32** | union { float f[4]; XMVECTOR v; }; |
| **XMVECTORU32** | union { UINT u[4]; XMVECTOR v; }; |
| **XMVECTORI32** | union { INT i[4]; XMVECTOR v; }; |
| **XMVECTORU8** | union { BYTE u[16]; XMVECTOR v; }; |
| **HALF** | USHORT |

## Calling Conventions

| | | |
|---|---|---|
| **FXMVECTOR** | const XMVECTOR | Up to the first three arguments |
| **CXMVECTOR** | const XMVECTOR& | Any remaining arguments |

## Macros

**XMASSERT**(Expression)
**XMGLOBALCONST**

**XMComparisonAllFalse**(CR)
**XMComparisonAllTrue**(CR)
**XMComparisonAnyFalse**(CR)
**XMComparisonAnyTrue**(CR)
**XMComparisonAllInBounds**(CR)
**XMComparisonAnyOutOfBounds**(CR)
**XMComparisonMixed**(CR)

**XMMin**(a, b)
**XMMax**(a, b)

## Constants

| | |
|---|---|
| **XM_PI** | π |
| **XM_2PI** | 2π |
| **XM_PIDIV2** | π / 2 |
| **XM_PIDIV4** | π / 4 |
| **XM_1DIVPI** | 1 / π |
| **XM_1DIV2PI** | 1 / (2π) |

XM_PERMUTE_0X, XM_PERMUTE_0Y, XM_PERMUTE_0Z, XM_PERMUTE_0W
XM_PERMUTE_1X, XM_PERMUTE_1Y, XM_PERMUTE_1Z, XM_PERMUTE_1W

XM_SELECT_0, XM_SELECT_1

XM_CRMASK_CR6
XM_CRMASK_CR6FALSE
XM_CRMASK_CR6TRUE
XM_CRMASK_CR6BOUNDS

## Compiler Directives

| | |
|---|---|
| **_XM_NO_INTRINSICS_** | Default: No |
| **_XM_SSE_INTRINSICS_** | Default: Yes (Windows) |
| **_XM_VMX128_INTRINSICS_** | Default: Yes (Xbox 360) |
| **XM_NO_ALIGNMENT** | Default: No |
| **XM_NO_MISALIGNED_VECTOR_ACCESS** | Default: No |
| **XM_NO_OPERATOR_OVERLOADS** | Default: No |
| **XM_STRICT_VECTOR4** | Default: No |

| Structure Name | Fields | Type | Bits | DXGI_FORMAT |
|---|---|---|---|---|
| **XMCOLOR** | union { struct { UINT a : 8; UINT r : 8; UINT g : 8; UINT b : 8; }; UINT c; }; | unsigned int | 8+8+8+8 = 32 | DXGI_FORMAT_B8G8R8A8_UNORM |
| **XMUBYTE4**<br>**XMUBYTEN4** | union { struct { BYTE x; BYTE y; BYTE z; BYTE w; }; UINT v; }; | unsigned int | 8+8+8+8 = 32 | DXGI_FORMAT_x8x8x8x8_UINT<br>DXGI_FORMAT_x8x8x8x8_UNORM |
| **XMBYTE4**<br>**XMBYTEN4** | union { struct { CHAR x; CHAR y; CHAR z; CHAR w; }; UINT v; }; | signed int | 8+8+8+8 = 32 | DXGI_FORMAT_x8x8x8x8_SINT<br>DXGI_FORMAT_x8x8x8x8_SNORM |
| **XMUSHORT2**<br>**XMUSHORTN2** | USHORT x; USHORT y; | unsigned int | 2 * 16 | DXGI_FORMAT_R16G16_UINT<br>DXGI_FORMAT_R16G16_UNORM |
| **XMUSHORT4**<br>**XMUSHORTN4** | USHORT x; USHORT y; USHORT z; USHORT w; | unsigned int | 4 * 16 | DXGI_FORMAT_R16G16B16A16_UINT<br>DXGI_FORMAT_R16G16B16A16_UNORM |
| **XMSHORT2**<br>**XMSHORTN2** | SHORT x; SHORT y; | signed int | 2 * 16 | DXGI_FORMAT_R16G16_SINT<br>DXGI_FORMAT_R16G16_SNORM |
| **XMSHORT4**<br>**XMSHORTN4** | SHORT x; SHORT y; SHORT z; SHORT w; | signed int | 4 * 16 | DXGI_FORMAT_R16G16B16A16_SINT<br>DXGI_FORMAT_R16G16B16A16_SNORM |
| **XMHALF2** | HALF x; HALF y; | float | 2 * 16 | DXGI_FORMAT_R16G16_FLOAT |
| **XMHALF4** | HALF x; HALF y; HALF z; HALF w; | float | 4 * 16 | DXGI_FORMAT_R16G16B16A16_FLOAT |
| **XMFLOAT2**<br>**XMFLOAT2A** | FLOAT x; FLOAT y; | float | 2 * 32 | DXGI_FORMAT_R32G32_FLOAT |
| **XMFLOAT3**<br>**XMFLOAT3A** | FLOAT x; FLOAT y; FLOAT z; | float | 3 * 32 | DXGI_FORMAT_R32G32B32_FLOAT |
| **XMFLOAT4**<br>**XMFLOAT4A** | FLOAT x; FLOAT y; FLOAT z; FLOAT w; | float | 4 * 32 | DXGI_FORMAT_R32G32B32A32_FLOAT |
| **XMMATRIX** | union { XMVECTOR r[4];<br>struct { FLOAT _11; FLOAT _12; FLOAT _13; FLOAT _14;<br>FLOAT _21; FLOAT _22; FLOAT _23; FLOAT _24;<br>FLOAT _31; FLOAT _32; FLOAT _33; FLOAT _34;<br>FLOAT _41; FLOAT _42; FLOAT _43; FLOAT _44; };<br>FLOAT m[4][4]; }; | float | 4x4 * 32 | |
| **XMFLOAT3X3** | union { struct { FLOAT _11; FLOAT _12; FLOAT _13;<br>FLOAT _21; FLOAT _22; FLOAT _23;<br>FLOAT _31; FLOAT _32; FLOAT _33; };<br>struct { FLOAT _m00; FLOAT _m01; FLOAT _m02;<br>FLOAT _m10; FLOAT _m11; FLOAT _m12;<br>FLOAT _m20; FLOAT _m21; FLOAT _m22; };<br>FLOAT m[3][3]; }; | float | 3x3 * 32 | |

| | | | | |
|---|---|---|---|---|
| **XMFLOAT4X3**<br>**XMFLOAT4X3A** | union { struct { FLOAT _11; FLOAT _12; FLOAT _13;<br>FLOAT _21; FLOAT _22; FLOAT _23;<br>FLOAT _31; FLOAT _32; FLOAT _33;<br>FLOAT _41; FLOAT _42; FLOAT _43; };<br>struct { FLOAT _m00; FLOAT _m01; FLOAT _m02;<br>FLOAT _m10; FLOAT _m11; FLOAT _m12;<br>FLOAT _m20; FLOAT _m21; FLOAT _m22;<br>FLOAT _m30; FLOAT _m31; FLOAT _m32; };<br>FLOAT m[4][3]; }; | float | 4x3 * 32 | |
| **XMFLOAT4X4**<br>**XMFLOAT4X4A** | union { struct { FLOAT _11; FLOAT _12; FLOAT _13; FLOAT _14;<br>FLOAT _21; FLOAT _22; FLOAT _23; FLOAT _24;<br>FLOAT _31; FLOAT _32; FLOAT _33; FLOAT _34;<br>FLOAT _41; FLOAT _42; FLOAT _43; FLOAT _44; };<br>struct { FLOAT _m00; FLOAT _m01; FLOAT _m02; FLOAT _m03;<br>FLOAT _m10; FLOAT _m11; FLOAT _m12; FLOAT _m13;<br>FLOAT _m20; FLOAT _m21; FLOAT _m22; FLOAT _m23;<br>FLOAT _m30; FLOAT _m31; FLOAT _m32; FLOAT _m33; };<br>FLOAT m[4][4]; }; | float | 4x4 * 32 | |
| **XMUNIBBLE4** | union { struct { USHORT x : 4; USHORT y : 4; USHORT z : 4; USHORT w : 4; };<br>USHORT v; }; | unsigned int | 4+4+4+4 = 16 | |
| **XMU555** | union { struct { USHORT x : 5; USHORT y : 5; USHORT z : 5; USHORT w : 1; };<br>USHORT v; }; | unsigned int | 5+5+5+1 = 16 | DXGI_FORMAT_B5G5R5A1_UNORM |
| **XMU565** | union { struct { USHORT x : 5; USHORT y : 6; USHORT z : 5; };<br>USHORT v; }; | unsigned int | 5+6+5 = 16 | DXGI_FORMAT_B5G6R5_UNORM |
| **XMUDHEN3**<br>**XMUDHENN3** | union { struct { UINT x : 10; UINT y : 11; UINT z : 11; }; UINT v; }; | unsigned int | 10+11+11 = 32 | |
| **XMUHEND3**<br>**XMUHENDN3** | union { struct { UINT x : 11; UINT y : 11; UINT z : 10; }; UINT v; }; | unsigned int | 11+11+10 = 32 | |
| **XMDHEN3**<br>**XMDHENN3** | union { struct { INT x : 10; INT y : 11; INT z : 11; }; UINT v; }; | signed int | 10+11+11 = 32 | |
| **XMHEND3**<br>**XMHENDN3** | union { struct { INT x : 11; INT y : 11; INT z : 10; }; UINT v; }; | signed int | 11+11+10 = 32 | |
| **XMPACKED4** | union { struct { UINT w : 2; INT z : 10; INT y : 10; INT x : 10; }; UINT v; }; | (un)signed int | 2+10+10+10 = 32 | |
| **XMDEC4**<br>**XMDECN4** | union { struct { INT x : 10; INT y : 10; INT z : 10; INT w : 2; }; UINT v; }; | signed int | 10+10+10+2 = 32 | |
| **XMUDEC4**<br>**XMUDECN4** | union { struct { UINT x : 10; UINT y : 10; UINT z : 10; UINT w : 2; }; UINT v; }; | unsigned int | 10+10+10+2 = 32 | DXGI_FORMAT_R10G10B10A2_UINT<br>DXGI_FORMAT_R10G10B10A2_UNORM |
| **XMXDEC4**<br>**XMXDECN4** | union { struct { INT x : 10; INT y : 10; INT z : 10; UINT w : 2; }; UINT v; }; | (un)signed int | 10+10+10+2 = 32 | |

| | | | | |
|---|---|---|---|---|
| **XMUICO4**<br>**XMUICON4** | union { struct { UINT64 x : 20; UINT64 y : 20; UINT64 z : 20; UINT64 w : 4; };<br>UINT64 v; }; | unsigned int | 20+20+20+4 = 64 | |
| **XMICO4**<br>**XMICON4** | union { struct { INT64 x : 20; INT64 y : 20; INT64 z : 20; INT64 w : 4; };<br>UINT64 v; }; | signed int | 20+20+20+4 = 64 | |
| **XMXICO4**<br>**XMXICON4** | union { struct { INT64 x : 20; INT64 y : 20; INT64 z : 20; UINT64 w : 4; };<br>UINT64 v; }; | (un)signed int | 20+20+20+4 = 64 | |
| **XMFLOAT3PK** | union { struct { UINT xm : 6; UINT xe : 5;<br>UINT ym : 6; UINT ye : 5;<br>UINT zm : 5; UINT ze : 5; };<br>UINT v; }; | float | 11+11+10 = 32 | DXGI_FORMAT_R11G11B10_FLOAT |
| **XMFLOAT3SE** | union { struct { UINT xm : 9; UINT ym : 9; UINT zm : 9; UINT e : 5; };<br>UINT v; }; | float | 9+9+9+(5) = 32 | DXGI_FORMAT_R9G9B9E5_SHAREDEXP |

## Functions

### Color

XMVECTOR **XMColorNegative**(XMVECTOR C)
XMVECTOR **XMColorModulate**(XMVECTOR C1, XMVECTOR C2)
XMVECTOR **XMColorAdjustContrast**(XMVECTOR C, FLOAT Contrast)
XMVECTOR **XMColorAdjustSaturation**(XMVECTOR C, FLOAT Saturation)
BOOL **XMColorEqual**(XMVECTOR C1, XMVECTOR C2)
BOOL **XMColorNotEqual**(XMVECTOR C1, XMVECTOR C2)
BOOL **XMColorGreater**(XMVECTOR C1, XMVECTOR C2)
BOOL **XMColorGreaterOrEqual**(XMVECTOR C1, XMVECTOR C2)
BOOL **XMColorLess**(XMVECTOR C1, XMVECTOR C2)
BOOL **XMColorLessOrEqual**(XMVECTOR C1, XMVECTOR C2)
BOOL **XMColorIsInfinite**(XMVECTOR C)
BOOL **XMColorIsNaN**(XMVECTOR C)

### Plane

XMVECTOR **XMPlaneDot**(XMVECTOR P, XMVECTOR V)
XMVECTOR **XMPlaneDotNormal**(XMVECTOR P, XMVECTOR V)
XMVECTOR **XMPlaneDotCoord**(XMVECTOR P, XMVECTOR V)
BOOL **XMPlaneEqual**(XMVECTOR P1, XMVECTOR P2)
BOOL **XMPlaneNotEqual**(XMVECTOR P1, XMVECTOR P2)
BOOL **XMPlaneNearEqual**(XMVECTOR P1, XMVECTOR P2,
XMVECTOR Epsilon)
XMVECTOR **XMPlaneNormalize**(XMVECTOR P)
XMVECTOR **XMPlaneNormalizeEst**(XMVECTOR P)
BOOL **XMPlaneIsInfinite**(XMVECTOR P)
BOOL **XMPlaneIsNaN**(XMVECTOR P)
XMVECTOR **XMPlaneTransform**(XMVECTOR P, XMMATRIX M)
XMFLOAT4* **XMPlaneTransformStream**(

### Conversion

HALF **XMConvertFloatToHalf**(FLOAT Value)
HALF* **XMConvertFloatToHalfStream**(
HALF *pOutputStream, UINT OutputStride,
CONST FLOAT *pInputStream, UINT InputStride, UINT FloatCount)
FLOAT **XMConvertHalfToFloat**(HALF Value)
FLOAT* **XMConvertHalfToFloatStream**(
FLOAT *pOutputStream, UINT OutputStride,
CONST HALF *pInputStream, UINT InputStride, UINT HalfCount)
FLOAT **XMConvertToDegrees**(FLOAT fRadians)
FLOAT **XMConvertToRadians**(FLOAT fDegrees)
XMVECTOR **XMConvertVectorFloatToUInt**(XMVECTOR VFloat, UINT MulExponent)
XMVECTOR **XMConvertVectorFloatToInt**(XMVECTOR VFloat, UINT MulExponent)
XMVECTOR **XMConvertVectorUIntToFloat**(XMVECTOR VUInt, UINT DivExponent)
XMVECTOR **XMConvertVectorIntToFloat**(XMVECTOR VInt, UINT DivExponent)

### Scalar

FLOAT **XMScalarSin**(FLOAT Value)
FLOAT **XMScalarSinEst**(FLOAT Value)
FLOAT **XMScalarCos**(FLOAT Value)
FLOAT **XMScalarCosEst**(FLOAT Value)
VOID **XMScalarSinCos**(FLOAT *pSin, FLOAT *pCos, FLOAT Value)
VOID **XMScalarSinCosEst**(FLOAT *pSin, FLOAT *pCos, FLOAT Value)
FLOAT **XMScalarASin**(FLOAT Value)
FLOAT **XMScalarASinEst**(FLOAT Value)
FLOAT **XMScalarACos**(FLOAT Value)
FLOAT **XMScalarACosEst**(FLOAT Value)
FLOAT **XMScalarModAngle**(FLOAT Value)

XMFLOAT4 *pOutputStream, UINT OutputStride,
CONST XMFLOAT4 *pInputStream, UINT InputStride,
UINT PlaneCount, XMMATRIX M)
XMVECTOR **XMPlaneFromPointNormal**(XMVECTOR Point, XMVECTOR Normal)
XMVECTOR **XMPlaneFromPoints**(
XMVECTOR Point1, XMVECTOR Point2, XMVECTOR Point3)
XMVECTOR **XMPlaneIntersectLine**(
XMVECTOR P, XMVECTOR LinePoint1, XMVECTOR LinePoint2)
VOID **XMPlaneIntersectPlane**(
XMVECTOR *pLinePoint1, XMVECTOR *pLinePoint2,
XMVECTOR P1, XMVECTOR P2)

## Vector – Arithmetic

XMVECTOR **XMVectorNegate**(XMVECTOR V)
XMVECTOR **XMVectorScale**(XMVECTOR V, FLOAT ScaleFactor)
XMVECTOR **XMVectorAdd**(XMVECTOR V1, XMVECTOR V2)
XMVECTOR **XMVectorSubtract**(XMVECTOR V1, XMVECTOR V2)
XMVECTOR **XMVectorMultiply**(XMVECTOR V1, XMVECTOR V2)
XMVECTOR **XMVectorDivide**(XMVECTOR V1, XMVECTOR V2)
XMVECTOR **XMVectorMod**(XMVECTOR V1, XMVECTOR V2)
XMVECTOR **XMVectorMultiplyAdd**(
XMVECTOR V1, XMVECTOR V2, XMVECTOR V3)
XMVECTOR **XMVectorNegativeMultiplySubtract**(
XMVECTOR V1, XMVECTOR V2, XMVECTOR V3)
XMVECTOR **XMVectorPow**(XMVECTOR V1, XMVECTOR V2)
XMVECTOR **XMVectorPowEst**(XMVECTOR V1, XMVECTOR V2)
XMVECTOR **XMVectorSqrt**(XMVECTOR V)
XMVECTOR **XMVectorSqrtEst**(XMVECTOR V)
XMVECTOR **XMVectorReciprocal**(XMVECTOR V)
XMVECTOR **XMVectorReciprocalEst**(XMVECTOR V)
XMVECTOR **XMVectorReciprocalSqrt**(XMVECTOR V)
XMVECTOR **XMVectorReciprocalSqrtEst**(XMVECTOR V)
XMVECTOR **XMVectorFloor**(XMVECTOR V)
XMVECTOR **XMVectorCeiling**(XMVECTOR V)
XMVECTOR **XMVectorRound**(XMVECTOR V)
XMVECTOR **XMVectorAbs**(XMVECTOR V)
XMVECTOR **XMVectorSaturate**(XMVECTOR V)
XMVECTOR **XMVectorClamp**(XMVECTOR V, XMVECTOR Min, XMVECTOR Max)
XMVECTOR **XMVectorTruncate**(XMVECTOR V)
XMVECTOR **XMVectorMin**(XMVECTOR V1, XMVECTOR V2)
XMVECTOR **XMVectorMax**(XMVECTOR V1, XMVECTOR V2)
XMVECTOR **XMVectorAddAngles**(XMVECTOR V1, XMVECTOR V2)
XMVECTOR **XMVectorSubtractAngles**(XMVECTOR V1, XMVECTOR V2)
XMVECTOR **XMVectorModAngles**(XMVECTOR Angles)
XMVECTOR **XMVectorIsInfinite**(XMVECTOR V)
XMVECTOR **XMVectorIsNaN**(XMVECTOR V)
BOOL **XMScalarNearEqual**(FLOAT S1, FLOAT S2, FLOAT Epsilon)

## Vector – Bit-Wise

XMVECTOR **XMVectorAndInt**(XMVECTOR V1, XMVECTOR V2)
XMVECTOR **XMVectorAndCInt**(XMVECTOR V1, XMVECTOR V2)
XMVECTOR **XMVectorOrInt**(XMVECTOR V1, XMVECTOR V2)
XMVECTOR **XMVectorXorInt**(XMVECTOR V1, XMVECTOR V2)
XMVECTOR **XMVectorNorInt**(XMVECTOR V1, XMVECTOR V2)
XMVECTOR **XMVectorNotEqual**(XMVECTOR V1, XMVECTOR V2)
XMVECTOR **XMVectorNotEqualInt**(XMVECTOR V1, XMVECTOR V2)

## Vector – Comparison

XMVECTOR **XMVectorEqual**(XMVECTOR V1, XMVECTOR V2)
XMVECTOR **XMVectorEqualR**(UINT *pCR, XMVECTOR V1, XMVECTOR V2)
XMVECTOR **XMVectorEqualInt**(XMVECTOR V1, XMVECTOR V2)
XMVECTOR **XMVectorEqualIntR**(UINT *pCR, XMVECTOR V1, XMVECTOR V2)
XMVECTOR **XMVectorGreater**(XMVECTOR V1, XMVECTOR V2)
XMVECTOR **XMVectorGreaterR**(UINT *pCR, XMVECTOR V1, XMVECTOR V2)
XMVECTOR **XMVectorGreaterOrEqual**(XMVECTOR V1, XMVECTOR V2)
XMVECTOR **XMVectorGreaterOrEqualR**(UINT *pCR,
XMVECTOR V1, XMVECTOR V2)
XMVECTOR **XMVectorLess**(XMVECTOR V1, XMVECTOR V2)
XMVECTOR **XMVectorLessOrEqual**(XMVECTOR V1, XMVECTOR V2)
XMVECTOR **XMVectorNearEqual**(XMVECTOR V1, XMVECTOR V2,
XMVECTOR Epsilon)

## Vector – Component-Wise

XMVECTOR **XMVectorInsert**(
XMVECTOR VD, XMVECTOR VS, UINT VSLeftRotateElements,
UINT Select0, UINT Select1, UINT Select2, UINT Select3)
XMVECTOR **XMVectorMergeXY**(XMVECTOR V1, XMVECTOR V2)
XMVECTOR **XMVectorMergeZW**(XMVECTOR V1, XMVECTOR V2)
XMVECTOR **XMVectorPermute**(XMVECTOR V1, XMVECTOR V2,
XMVECTOR Control)
XMVECTOR **XMVectorPermuteControl**(
UINT ElementIndex0, UINT ElementIndex1,
UINT ElementIndex2, UINT ElementIndex3)
XMVECTOR **XMVectorSwizzle**(XMVECTOR V, UINT E0, UINT E1, UINT E2, UINT E3)
XMVECTOR **XMVectorSelect**(XMVECTOR V1, XMVECTOR V2, XMVECTOR Control)
XMVECTOR **XMVectorSelectControl**(
UINT VectorIndex0, UINT VectorIndex1, UINT VectorIndex2, UINT VectorIndex3)
XMVECTOR **XMVectorSplatX**(XMVECTOR V)
XMVECTOR **XMVectorSplatY**(XMVECTOR V)
XMVECTOR **XMVectorSplatZ**(XMVECTOR V)
XMVECTOR **XMVectorSplatW**(XMVECTOR V)
XMVECTOR **XMVectorRotateLeft**(XMVECTOR V, UINT Elements)

## Vector – Initialization

XMVECTOR **XMVectorZero**()
XMVECTOR **XMVectorFalseInt**()
XMVECTOR **XMVectorTrueInt**()
XMVECTOR **XMVectorSplatOne**()
XMVECTOR **XMVectorSplatEpsilon**()
XMVECTOR **XMVectorSplatInfinity**()
XMVECTOR **XMVectorSplatQNaN**()
XMVECTOR **XMVectorSplatSignMask**()
XMVECTOR **XMVectorSplatConstant**(UINT IntConstant, UINT DivExponent)
XMVECTOR **XMVectorSplatConstantInt**(UINT IntConstant)
XMVECTOR **XMVectorSet**(FLOAT x, FLOAT y, FLOAT z, FLOAT w)
XMVECTOR **XMVectorSetInt**(UINT x, UINT y, UINT z, UINT w)
XMVECTOR **XMVectorSetBinaryConstant**(UINT C0, UINT C1, UINT C2, UINT C3)
XMVECTOR **XMVectorReplicate**(FLOAT Value)
XMVECTOR **XMVectorReplicatePtr**(Const FLOAT *Value)
XMVECTOR **XMVectorReplicateInt**(UINT Value)
XMVECTOR **XMVectorReplicateIntPtr**(Const UINT *Value)

## Utility

VOID **XMAssert**(CONST CHAR *pExpression,
CONST CHAR *pFileName, UNIT LineNumber)
XMVECTOR **XMFresnelTerm**(XMVECTOR CosIncidentAngle,
XMVECTOR RefractionIndex)
BOOL **XMVerifyCPUSupport**()

XMVECTOR **XMVectorRotateRight**(XMVECTOR V, UINT Elements)
XMVECTOR **XMVectorShiftLeft**(XMVECTOR V1, XMVECTOR V2, UINT Elements)

## Vector – Transcendental

XMVECTOR **XMVectorSin**(XMVECTOR V)
XMVECTOR **XMVectorSinEst**(XMVECTOR V)
XMVECTOR **XMVectorCos**(XMVECTOR V)
XMVECTOR **XMVectorCosEst**(XMVECTOR V)
VOID **XMVectorSinCos**(XMVECTOR *pSin, XMVECTOR *pCos,
XMVECTOR V)
VOID **XMVectorSinCosEst**(XMVECTOR *pSin, XMVECTOR *pCos,
XMVECTOR V)
XMVECTOR **XMVectorTan**(XMVECTOR V)
XMVECTOR **XMVectorTanEst**(XMVECTOR V)
XMVECTOR **XMVectorASin**(XMVECTOR V)
XMVECTOR **XMVectorASinEst**(XMVECTOR V)
XMVECTOR **XMVectorACos**(XMVECTOR V)
XMVECTOR **XMVectorACosEst**(XMVECTOR V)
XMVECTOR **XMVectorATan**(XMVECTOR V)
XMVECTOR **XMVectorATanEst**(XMVECTOR V)
XMVECTOR **XMVectorATan2**(XMVECTOR Y, XMVECTOR X)
XMVECTOR **XMVectorATan2Est**(XMVECTOR Y, XMVECTOR X)
XMVECTOR **XMVectorSinH**(XMVECTOR V)
XMVECTOR **XMVectorSinHEst**(XMVECTOR V)
XMVECTOR **XMVectorCosH**(XMVECTOR V)
XMVECTOR **XMVectorCosHEst**(XMVECTOR V)
XMVECTOR **XMVectorTanH**(XMVECTOR V)
XMVECTOR **XMVectorTanHEst**(XMVECTOR V)
XMVECTOR **XMVectorExp**(XMVECTOR V)
XMVECTOR **XMVectorExpEst**(XMVECTOR V)
XMVECTOR **XMVectorLog**(XMVECTOR V)
XMVECTOR **XMVectorLogEst**(XMVECTOR V)

## Vector – Geometric

XMVECTOR **XMVectorBaryCentric**(XMVECTOR Position0, XMVECTOR Position1, XMVECTOR Position2, FLOAT f, FLOAT g)
XMVECTOR **XMVectorBaryCentricV**(XMVECTOR Position0, XMVECTOR Position1, XMVECTOR Position2, XMVECTOR F, XMVECTOR G)
XMVECTOR **XMVectorCatmullRom**(XMVECTOR Position0, XMVECTOR Position1, XMVECTOR Position2, XMVECTOR Position3, FLOAT t)
XMVECTOR **XMVectorCatmullRomV**(XMVECTOR Position0, XMVECTOR Position1, XMVECTOR Position2, XMVECTOR Position3, XMVECTOR T)
XMVECTOR **XMVectorHermite**(XMVECTOR Position0, XMVECTOR Tangent0, XMVECTOR Position1, XMVECTOR Tangent1, FLOAT t)
XMVECTOR **XMVectorHermiteV**(XMVECTOR Position0, XMVECTOR Tangent0, XMVECTOR Position1, XMVECTOR Tangent1, XMVECTOR T)
XMVECTOR **XMVectorInBounds**(XMVECTOR V, XMVECTOR Bounds)
XMVECTOR **XMVectorInBoundsR**(UINT *pCR, XMVECTOR V, XMVECTOR Bounds)
XMVECTOR **XMVectorLerp**(XMVECTOR V0, XMVECTOR V1, FLOAT t)
XMVECTOR **XMVectorLerpV**(XMVECTOR V0, XMVECTOR V1, XMVECTOR T)

## 2D Vector – Comparison

BOOL **XMVector2Equal**(XMVECTOR V1, XMVECTOR V2)
UINT **XMVector2EqualR**(XMVECTOR V1, XMVECTOR V2)
BOOL **XMVector2EqualInt**(XMVECTOR V1, XMVECTOR V2)
UINT **XMVector2EqualIntR**(XMVECTOR V1, XMVECTOR V2)
BOOL **XMVector2NotEqual**(XMVECTOR V1, XMVECTOR V2)
BOOL **XMVector2NotEqualInt**(XMVECTOR V1, XMVECTOR V2)
BOOL **XMVector2Greater**(XMVECTOR V1, XMVECTOR V2)
UINT **XMVector2GreaterR**(XMVECTOR V1, XMVECTOR V2)
BOOL **XMVector2GreaterOrEqual**(XMVECTOR V1, XMVECTOR V2)
UINT **XMVector2GreaterOrEqualR**(XMVECTOR V1, XMVECTOR V2)
BOOL **XMVector2Less**(XMVECTOR V1, XMVECTOR V2)
BOOL **XMVector2LessOrEqual**(XMVECTOR V1, XMVECTOR V2)
BOOL **XMVector2NearEqual**(XMVECTOR V1, XMVECTOR V2, XMVECTOR Epsilon)
BOOL **XMVector2IsInfinite**(XMVECTOR V)
BOOL **XMVector2IsNaN**(XMVECTOR V)

## 2D Vector – Geometric

XMVECTOR **XMVector2AngleBetweenNormals**(XMVECTOR N1, XMVECTOR N2)
XMVECTOR **XMVector2AngleBetweenNormalsEst**(XMVECTOR N1, XMVECTOR N2)
XMVECTOR **XMVector2AngleBetweenVectors**(XMVECTOR V1, XMVECTOR V2)
XMVECTOR **XMVector2ClampLength**(XMVECTOR V, FLOAT LengthMin, FLOAT LengthMax)
XMVECTOR **XMVector2ClampLengthV**(XMVECTOR V, XMVECTOR LengthMin, XMVECTOR LengthMax)
XMVECTOR **XMVector2Cross**(XMVECTOR V1, XMVECTOR V2)
XMVECTOR **XMVector2Dot**(XMVECTOR V1, XMVECTOR V2)
BOOL **XMVector2InBounds**(XMVECTOR V, XMVECTOR Bounds)
UINT **XMVector2InBoundsR**(XMVECTOR V, XMVECTOR Bounds)
XMVECTOR **XMVector2IntersectLine**(XMVECTOR Line1Point1, XMVECTOR Line1Point2, XMVECTOR Line2Point1, XMVECTOR Line2Point2)
XMVECTOR **XMVector2Length**(XMVECTOR V)
XMVECTOR **XMVector2LengthEst**(XMVECTOR V)
XMVECTOR **XMVector2LengthSq**(XMVECTOR V)
XMVECTOR **XMVector2LinePointDistance**(XMVECTOR LinePoint1, XMVECTOR LinePoint2, XMVECTOR Point)
XMVECTOR **XMVector2Normalize**(XMVECTOR V)
XMVECTOR **XMVector2NormalizeEst**(XMVECTOR V)
XMVECTOR **XMVector2Orthogonal**(XMVECTOR V)
XMVECTOR **XMVector2ReciprocalLength**(XMVECTOR V)
XMVECTOR **XMVector2ReciprocalLengthEst**(XMVECTOR V)
XMVECTOR **XMVector2Reflect**(XMVECTOR Incident, XMVECTOR Normal)
XMVECTOR **XMVector2Refract**(XMVECTOR Incident, XMVECTOR Normal, FLOAT RefractionIndex)
XMVECTOR **XMVector2RefractV**(XMVECTOR Incident, XMVECTOR Normal, XMVECTOR RefractionIndex)

## 2D Vector – Transformation

XMVECTOR **XMVector2Transform**(XMVECTOR V, XMMATRIX M)
XMVECTOR **XMVector2TransformNormal**(XMVECTOR V, XMMATRIX M)
XMVECTOR **XMVector2TransformCoord**(XMVECTOR V, XMMATRIX M)
XMFLOAT4* **XMVector2TransformStream**( XMFLOAT4 *pOutputStream, UINT OutputStride,

CONST XMFLOAT2 *pInputStream, UINT InputStride, UINT VectorCount, XMMATRIX M)

XMFLOAT4* **XMVector2TransformStreamNC**( XMFLOAT4 *pOutputStream, UINT OutputStride,
CONST XMFLOAT2 *pInputStream, UINT InputStride, UINT VectorCount, XMMATRIX M)

XMFLOAT2* **XMVector2TransformNormalStream**( XMFLOAT2 *pOutputStream, UINT OutputStride,
CONST XMFLOAT2 *pInputStream, UINT InputStride, UINT VectorCount, XMMATRIX M)

XMFLOAT2* **XMVector2TransformCoordStream**( XMFLOAT2 *pOutputStream, UINT OutputStride,
CONST XMFLOAT2 *pInputStream, UINT InputStride, UINT VectorCount, XMMATRIX M)

## 3D Vector – Comparison

BOOL **XMVector3Equal**(XMVECTOR V1, XMVECTOR V2)
UINT **XMVector3EqualR**(XMVECTOR V1, XMVECTOR V2)
BOOL **XMVector3EqualInt**(XMVECTOR V1, XMVECTOR V2)
UINT **XMVector3EqualIntR**(XMVECTOR V1, XMVECTOR V2)
BOOL **XMVector3NotEqual**(XMVECTOR V1, XMVECTOR V2)
BOOL **XMVector3NotEqualInt**(XMVECTOR V1, XMVECTOR V2)
BOOL **XMVector3Greater**(XMVECTOR V1, XMVECTOR V2)
UINT **XMVector3GreaterR**(XMVECTOR V1, XMVECTOR V2)
BOOL **XMVector3GreaterOrEqual**(XMVECTOR V1, XMVECTOR V2)
UINT **XMVector3GreaterOrEqualR**(XMVECTOR V1, XMVECTOR V2)
BOOL **XMVector3Less**(XMVECTOR V1, XMVECTOR V2)
BOOL **XMVector3LessOrEqual**(XMVECTOR V1, XMVECTOR V2)
BOOL **XMVector3NearEqual**(XMVECTOR V1, XMVECTOR V2, XMVECTOR Epsilon)
BOOL **XMVector3IsInfinite**(XMVECTOR V)
BOOL **XMVector3IsNaN**(XMVECTOR V)

## 3D Vector – Geometric

XMVECTOR **XMVector3AngleBetweenNormals**(XMVECTOR N1, XMVECTOR N2)
XMVECTOR **XMVector3AngleBetweenNormalsEst**(XMVECTOR N1, XMVECTOR N2)
XMVECTOR **XMVector3AngleBetweenVectors**(XMVECTOR V1, XMVECTOR V2)
XMVECTOR **XMVector3ClampLength**(XMVECTOR V, FLOAT LengthMin, FLOAT LengthMax)
XMVECTOR **XMVector3ClampLengthV**(XMVECTOR V, XMVECTOR LengthMin, XMVECTOR LengthMax)
VOID **XMVector3ComponentsFromNormal**(XMVECTOR *pParallel, XMVECTOR *pPerpendicular, XMVECTOR V, XMVECTOR Normal)
XMVECTOR **XMVector3Cross**(XMVECTOR V1, XMVECTOR V2)
XMVECTOR **XMVector3Dot**(XMVECTOR V1, XMVECTOR V2)
BOOL **XMVector3InBounds**(XMVECTOR V, XMVECTOR Bounds)
UINT **XMVector3InBoundsR**(XMVECTOR V, XMVECTOR Bounds)
XMVECTOR **XMVector3Length**(XMVECTOR V)
XMVECTOR **XMVector3LengthEst**(XMVECTOR V)
XMVECTOR **XMVector3LengthSq**(XMVECTOR V)
XMVECTOR **XMVector3LinePointDistance**(XMVECTOR LinePoint1, XMVECTOR LinePoint2, XMVECTOR Point)
XMVECTOR **XMVector3Normalize**(XMVECTOR V)
XMVECTOR **XMVector3NormalizeEst**(XMVECTOR V)
XMVECTOR **XMVector3Orthogonal**(XMVECTOR V)
XMVECTOR **XMVector3ReciprocalLength**(XMVECTOR V)
XMVECTOR **XMVector3ReciprocalLengthEst**(XMVECTOR V)
XMVECTOR **XMVector3Reflect**(XMVECTOR Incident, XMVECTOR Normal)

XMVECTOR **XMVector3Refract**(XMVECTOR Incident, XMVECTOR Normal, FLOAT RefractionIndex)
XMVECTOR **XMVector3RefractV**(XMVECTOR Incident, XMVECTOR Normal, XMVECTOR RefractionIndex)

## 3D Vector – Transformation

XMVECTOR **XMVector3Transform**(XMVECTOR V, XMMATRIX M)
XMVECTOR **XMVector3TransformNormal**(XMVECTOR V, XMMATRIX M)
XMVECTOR **XMVector3TransformCoord**(XMVECTOR V, XMMATRIX M)
XMVECTOR **XMVector3Rotate**(XMVECTOR V, XMVECTOR RotationQuaternion)
XMVECTOR **XMVector3InverseRotate**(XMVECTOR V, XMVECTOR RotationQuaternion)
XMFLOAT4* **XMVector3TransformStream**( XMFLOAT4 *pOutputStream, UINT OutputStride,
CONST XMFLOAT3 *pInputStream, UINT InputStride, UINT VectorCount, XMMATRIX M)
XMFLOAT4* **XMVector3TransformStreamNC**( XMFLOAT4 *pOutputStream, UINT OutputStride,
CONST XMFLOAT3 *pInputStream, UINT InputStride, UINT VectorCount, XMMATRIX M)
XMFLOAT3* **XMVector3TransformNormalStream**( XMFLOAT3 *pOutputStream, UINT OutputStride,
CONST XMFLOAT3 *pInputStream, UINT InputStride, UINT VectorCount, XMMATRIX M)
XMFLOAT3* **XMVector3TransformCoordStream**( XMFLOAT3 *pOutputStream, UINT OutputStride,
CONST XMFLOAT3 *pInputStream, UINT InputStride, UINT VectorCount, XMMATRIX M)
XMVECTOR **XMVector3Project**( XMVECTOR V, FLOAT ViewportX, FLOAT ViewportY, FLOAT ViewportWidth, FLOAT ViewportHeight,
FLOAT ViewportMinZ, FLOAT ViewportMaxZ, XMMATRIX Projection, XMMATRIX View, XMMATRIX World)
XMFLOAT3* **XMVector3ProjectStream**( XMFLOAT3 *pOutputStream, UINT OutputStride, CONST XMFLOAT3 *pInputStream, UINT InputStride, UINT VectorCount,
FLOAT ViewportX, FLOAT ViewportY, FLOAT ViewportWidth, FLOAT ViewportHeight,
FLOAT ViewportMinZ, FLOAT ViewportMaxZ, XMMATRIX Projection, XMMATRIX View, XMMATRIX World)
XMVECTOR **XMVector3Unproject**( XMVECTOR V, FLOAT ViewportX, FLOAT ViewportY, FLOAT ViewportWidth, FLOAT ViewportHeight,
FLOAT ViewportMinZ, FLOAT ViewportMaxZ, XMMATRIX Projection, XMMATRIX View, XMMATRIX World)
XMFLOAT3* **XMVector3UnprojectStream**( XMFLOAT3 *pOutputStream, UINT OutputStride, CONST XMFLOAT3 *pInputStream, UINT InputStride, UINT VectorCount,
FLOAT ViewportX, FLOAT ViewportY, FLOAT ViewportWidth, FLOAT ViewportHeight,
FLOAT ViewportMinZ, FLOAT ViewportMaxZ, XMMATRIX Projection, XMMATRIX View, XMMATRIX World)

## 4D Vector – Comparison

BOOL **XMVector4Equal**(XMVECTOR V1, XMVECTOR V2)
UINT **XMVector4EqualR**(XMVECTOR V1, XMVECTOR V2)
BOOL **XMVector4EqualInt**(XMVECTOR V1, XMVECTOR V2)
UINT **XMVector4EqualIntR**(XMVECTOR V1, XMVECTOR V2)
BOOL **XMVector4NotEqual**(XMVECTOR V1, XMVECTOR V2)
BOOL **XMVector4NotEqualInt**(XMVECTOR V1, XMVECTOR V2)
BOOL **XMVector4Greater**(XMVECTOR V1, XMVECTOR V2)
UINT **XMVector4GreaterR**(XMVECTOR V1, XMVECTOR V2)
BOOL **XMVector4GreaterOrEqual**(XMVECTOR V1, XMVECTOR V2)
UINT **XMVector4GreaterOrEqualR**(XMVECTOR V1, XMVECTOR V2)
BOOL **XMVector4Less**(XMVECTOR V1, XMVECTOR V2)
BOOL **XMVector4LessOrEqual**(XMVECTOR V1, XMVECTOR V2)
BOOL **XMVector4NearEqual**(XMVECTOR V1, XMVECTOR V2, XMVECTOR Epsilon)
BOOL **XMVector4IsInfinite**(XMVECTOR V)
BOOL **XMVector4IsNaN**(XMVECTOR V)

## 4D Vector – Geometric

XMVECTOR **XMVector4AngleBetweenNormals**(XMVECTOR N1, XMVECTOR N2)
XMVECTOR **XMVector4AngleBetweenNormalsEst**(XMVECTOR N1, XMVECTOR N2)
XMVECTOR **XMVector4AngleBetweenVectors**(XMVECTOR V1, XMVECTOR V2)
XMVECTOR **XMVector4ClampLength**(XMVECTOR V, FLOAT LengthMin, FLOAT LengthMax)
XMVECTOR **XMVector4ClampLengthV**(XMVECTOR V, XMVECTOR LengthMin, XMVECTOR LengthMax)
XMVECTOR **XMVector4Cross**(XMVECTOR V1, XMVECTOR V2, XMVECTOR V3)
XMVECTOR **XMVector4Dot**(XMVECTOR V1, XMVECTOR V2)
BOOL **XMVector4InBounds**(XMVECTOR V, XMVECTOR Bounds)
UINT **XMVector4InBoundsR**(XMVECTOR V, XMVECTOR Bounds)
XMVECTOR **XMVector4Length**(XMVECTOR V)
XMVECTOR **XMVector4LengthEst**(XMVECTOR V)
XMVECTOR **XMVector4LengthSq**(XMVECTOR V)
XMVECTOR **XMVector4Normalize**(XMVECTOR V)
XMVECTOR **XMVector4NormalizeEst**(XMVECTOR V)
XMVECTOR **XMVector4Orthogonal**(XMVECTOR V)
XMVECTOR **XMVector4ReciprocalLength**(XMVECTOR V)
XMVECTOR **XMVector4ReciprocalLengthEst**(XMVECTOR V)
XMVECTOR **XMVector4Reflect**(XMVECTOR Incident, XMVECTOR Normal)
XMVECTOR **XMVector4Refract**(XMVECTOR Incident, XMVECTOR Normal, FLOAT RefractionIndex)
XMVECTOR **XMVector4RefractV**(XMVECTOR Incident, XMVECTOR Normal, XMVECTOR RefractionIndex)

## 4D Vector – Transformation

XMVECTOR **XMVector4Transform**(XMVECTOR V, XMMATRIX M)
XMFLOAT4* **XMVector4TransformStream**( XMFLOAT4 *pOutputStream, UINT OutputStride,
CONST XMFLOAT4 *pInputStream, UINT InputStride, UINT VectorCount, XMMATRIX M)

## Vector Accessor Functions

FLOAT **XMVectorGetX**(XMVECTOR V)
FLOAT **XMVectorGetY**(XMVECTOR V)
FLOAT **XMVectorGetZ**(XMVECTOR V)
FLOAT **XMVectorGetW**(XMVECTOR V)
UNINT **XMVectorGetIntX**(XMVECTOR V)
UNINT **XMVectorGetIntY**(XMVECTOR V)
UNINT **XMVectorGetIntZ**(XMVECTOR V)
UNINT **XMVectorGetIntW**(XMVECTOR V)
FLOAT **XMVectorGetByIndex**(XMVECTOR V, UINT i)
UINT **XMVectorGetIntByIndex**(XMVECTOR V, UINT i)

VOID **XMVectorGetXPtr**(FLOAT *x, XMVECTOR V)
VOID **XMVectorGetYPtr**(FLOAT *y, XMVECTOR V)
VOID **XMVectorGetZPtr**(FLOAT *z, XMVECTOR V)
VOID **XMVectorGetWPtr**(FLOAT *w, XMVECTOR V)
VOID **XMVectorGetIntXPtr**(UINT *x, XMVECTOR V)
VOID **XMVectorGetIntYPtr**(UINT *y, XMVECTOR V)
VOID **XMVectorGetIntZPtr**(UINT *z, XMVECTOR V)
VOID **XMVectorGetIntWPtr**(UINT *w, XMVECTOR V)
VOID **XMVectorGetByIndexPtr**(FLOAT *f, XMVECTOR V, UINT i)
VOID **XMVectorGetIntByIndexPtr**(UINT *x, XMVECTOR V, UINT i)

XMVECTOR **XMVectorSetX**(XMVECTOR V, FLOAT x)
XMVECTOR **XMVectorSetY**(XMVECTOR V, FLOAT y)
XMVECTOR **XMVectorSetZ**(XMVECTOR V, FLOAT z)
XMVECTOR **XMVectorSetW**(XMVECTOR V, FLOAT w)
VOID **XMVectorSetIntX**(XMVECTOR V, UNINT x)
XMVECTOR **XMVectorSetIntY**(XMVECTOR V, UNINT y)
XMVECTOR **XMVectorSetIntZ**(XMVECTOR V, UNINT z)

XMVECTOR **XMVectorSetXPtr**(XMVECTOR V, CONST FLOAT *x)
XMVECTOR **XMVectorSetYPtr**(XMVECTOR V, CONST FLOAT *y)
XMVECTOR **XMVectorSetZPtr**(XMVECTOR V, CONST FLOAT *z)
XMVECTOR **XMVectorSetWPtr**(XMVECTOR V, CONST FLOAT *w)
XMVECTOR **XMVectorSetIntXPtr**(XMVECTOR V, CONST UINT *x)
XMVECTOR **XMVectorSetIntYPtr**(XMVECTOR V, CONST UINT *y)
XMVECTOR **XMVectorSetIntZPtr**(XMVECTOR V, CONST UINT *z)

XMVECTOR **XMVectorSetIntW**(XMVECTOR V, UNINT w)
XMVECTOR **XMVectorSetIntWPtr**(XMVECTOR V, CONST UINT *w)
XMVECTOR **XMVectorSetByIndex**(XMVECTOR V, FLOAT f, UINT i)
XMVECTOR **XMVectorSetByIndexPtr**(XMVECTOR V, CONST FLOAT *f, UINT i)
XMVECTOR **XMVectorSetIntByIndex**(XMVECTOR V, UINT f, UINT i)
XMVECTOR **XMVectorSetIntByIndexPtr**(XMVECTOR V, CONST UINT *x, UINT i)

## Matrix

XMMATRIX **XMMatrixIdentity**()
XMMATRIX **XMMatrixSet**( FLOAT m00, FLOAT m01, FLOAT m02, FLOAT m03,
FLOAT m10, FLOAT m11, FLOAT m12, FLOAT m13,
FLOAT m20, FLOAT m21, FLOAT m22, FLOAT m23,
FLOAT m30, FLOAT m31, FLOAT m32, FLOAT m33)
XMMATRIX **XMMatrixTranslation**(FLOAT OffsetX, FLOAT OffsetY, FLOAT OffsetZ)
XMMATRIX **XMMatrixTranslationFromVector**(XMVECTOR Offset)
XMMATRIX **XMMatrixScaling**(FLOAT ScaleX, FLOAT ScaleY, FLOAT ScaleZ)
XMMATRIX **XMMatrixScalingFromVector**(XMVECTOR Scale)
XMMATRIX **XMMatrixRotationX**(FLOAT Angle)
XMMATRIX **XMMatrixRotationY**(FLOAT Angle)
XMMATRIX **XMMatrixRotationZ**(FLOAT Angle)
XMMATRIX **XMMatrixRotationAxis**(XMVECTOR Axis, FLOAT Angle)
XMMATRIX **XMMatrixRotationNormal**(XMVECTOR NormalAxis, FLOAT Angle)
XMMATRIX **XMMatrixRotationQuaternion**(XMVECTOR Quaternion)
XMMATRIX **XMMatrixRotationRollPitchYaw**(FLOAT Pitch, FLOAT Yaw, FLOAT Roll)
XMMATRIX **XMMatrixRotationRollPitchYawFromVector**(XMVECTOR Angles)
XMMATRIX **XMMatrixLookAtLH**(XMVECTOR EyePosition, XMVECTOR FocusPosition, XMVECTOR UpDirection)
XMMATRIX **XMMatrixLookAtRH**(XMVECTOR EyePosition, XMVECTOR FocusPosition, XMVECTOR UpDirection)
XMMATRIX **XMMatrixLookToLH**(XMVECTOR EyePosition, XMVECTOR EyeDirection, XMVECTOR UpDirection)
XMMATRIX **XMMatrixLookToRH**(XMVECTOR EyePosition, XMVECTOR EyeDirection, XMVECTOR UpDirection)
XMMATRIX **XMMatrixOrthographicLH**(FLOAT ViewWidth, FLOAT ViewHeight, FLOAT NearZ, FLOAT FarZ)
XMMATRIX **XMMatrixOrthographicRH**(FLOAT ViewWidth, FLOAT ViewHeight, FLOAT NearZ, FLOAT FarZ)
XMMATRIX **XMMatrixOrthographicOffCenterLH**(FLOAT ViewLeft, FLOAT ViewRight, FLOAT ViewBottom, FLOAT ViewTop, FLOAT NearZ, FLOAT FarZ)
XMMATRIX **XMMatrixOrthographicOffCenterRH**(FLOAT ViewLeft, FLOAT ViewRight, FLOAT ViewBottom, FLOAT ViewTop, FLOAT NearZ, FLOAT FarZ)
XMMATRIX **XMMatrixPerspectiveLH**(FLOAT ViewWidth, FLOAT ViewHeight, FLOAT NearZ, FLOAT FarZ)
XMMATRIX **XMMatrixPerspectiveRH**(FLOAT ViewWidth, FLOAT ViewHeight, FLOAT NearZ, FLOAT FarZ)
XMMATRIX **XMMatrixPerspectiveFovLH**(FLOAT FovAngleY, FLOAT AspectHByW, FLOAT NearZ, FLOAT FarZ)
XMMATRIX **XMMatrixPerspectiveFovRH**(FLOAT FovAngleY, FLOAT AspectHByW, FLOAT NearZ, FLOAT FarZ)
XMMATRIX **XMMatrixPerspectiveOffCenterLH**(FLOAT ViewLeft, FLOAT ViewRight, FLOAT ViewBottom, FLOAT ViewTop, FLOAT NearZ, FLOAT FarZ)
XMMATRIX **XMMatrixPerspectiveOffCenterRH**(FLOAT ViewLeft, FLOAT ViewRight, FLOAT ViewBottom, FLOAT ViewTop, FLOAT NearZ, FLOAT FarZ)
XMMATRIX **XMMatrixReflect**(XMVECTOR ReflectionPlane)
XMMATRIX **XMMatrixShadow**(XMVECTOR ShadowPlane, XMVECTOR LightPosition)
XMMATRIX **XMMatrixTransformation2D**( XMVECTOR ScalingOrigin, FLOAT ScalingOrientation, XMVECTOR Scaling,
XMVECTOR RotationOrigin, FLOAT Rotation, XMVECTOR Translation)
XMMATRIX **XMMatrixTransformation**( XMVECTOR ScalingOrigin, XMVECTOR ScalingOrientationQuaternion, XMVECTOR Scaling,
XMVECTOR RotationOrigin, XMVECTOR RotationQuaternion, XMVECTOR Translation)
XMMATRIX **XMMatrixAffineTransformation2D**(XMVECTOR Scaling, XMVECTOR RotationOrigin, FLOAT Rotation, XMVECTOR Translation)
XMMATRIX **XMMatrixAffineTransformation**(XMVECTOR Scaling, XMVECTOR RotationOrigin, XMVECTOR RotationQuaternion, XMVECTOR Translation)
XMMATRIX **XMMatrixMultiply**(XMMATRIX M1, XMMATRIX M2)
XMMATRIX **XMMatrixMultiplyTranspose**(XMMATRIX M1, XMMATRIX M2)

XMMATRIX **XMMatrixTranspose**(XMMATRIX M)
XMMATRIX **XMMatrixInverse**(XMVECTOR *pDeterminant, XMMATRIX M)
XMVECTOR **XMMatrixDeterminant**(XMMATRIX M)
BOOL **XMMatrixDecompose**(XMVECTOR *outScale, XMVECTOR *outRotQuat, XMVECTOR *outTrans, XMMATRIX M)
BOOL **XMMatrixIsIdentity**(XMMATRIX M)
BOOL **XMMatrixIsInfinite**(XMMATRIX M)
BOOL **XMMatrixIsNaN**(XMMATRIX M)

## Quaternion

XMVECTOR **XMQuaternionIdentity**()
XMVECTOR **XMQuaternionRotationMatrix**(XMMATRIX M)
XMVECTOR **XMQuaternionRotationAxis**(XMVECTOR Axis, FLOAT Angle)
XMVECTOR **XMQuaternionRotationNormal**(XMVECTOR NormalAxis, FLOAT Angle)
XMVECTOR **XMQuaternionRotationRollPitchYaw**(FLOAT Pitch, FLOAT Yaw, FLOAT Roll)
XMVECTOR **XMQuaternionRotationRollPitchYawFromVector**(XMVECTOR Angles)
XMVECTOR **XMQuaternionBaryCentric**(XMVECTOR Q0, XMVECTOR Q1, XMVECTOR Q2, FLOAT f, FLOAT g)
XMVECTOR **XMQuaternionBaryCentricV**(XMVECTOR Q0, XMVECTOR Q1, XMVECTOR Q2, XMVECTOR F, XMVECTOR G)
XMVECTOR **XMQuaternionConjugate**(XMVECTOR Q)
XMVECTOR **XMQuaternionInverse**(XMVECTOR Q)
XMVECTOR **XMQuaternionExp**(XMVECTOR Q)
XMVECTOR **XMQuaternionLn**(XMVECTOR Q)
XMVECTOR **XMQuaternionMultiply**(XMVECTOR Q1, XMVECTOR Q2)
XMVECTOR **XMQuaternionDot**(XMVECTOR Q1, XMVECTOR Q2)
XMVECTOR **XMQuaternionLength**(XMVECTOR Q)
XMVECTOR **XMQuaternionLengthSq**(XMVECTOR Q)
XMVECTOR **XMQuaternionReciprocalLength**(XMVECTOR Q)
XMVECTOR **XMQuaternionNormalize**(XMVECTOR Q)
XMVECTOR **XMQuaternionNormalizeEst**(XMVECTOR Q)
XMVECTOR **XMQuaternionSlerp**(XMVECTOR Q0, XMVECTOR Q1, FLOAT t)
XMVECTOR **XMQuaternionSlerpV**(XMVECTOR Q0, XMVECTOR Q1, XMVECTOR T)
XMVECTOR **XMQuaternionSquad**(XMVECTOR Q0, XMVECTOR Q1, XMVECTOR Q2, XMVECTOR Q3, FLOAT t)
XMVECTOR **XMQuaternionSquadV**(XMVECTOR Q0, XMVECTOR Q1, XMVECTOR Q2, XMVECTOR Q3, XMVECTOR T)
VOID **XMQuaternionSquadSetup**(XMVECTOR *pA, XMVECTOR *pB, XMVECTOR *pC, XMVECTOR Q0, XMVECTOR Q1, XMVECTOR Q2, XMVECTOR Q3)
BOOL **XMQuaternionEqual**(XMVECTOR Q1, XMVECTOR Q2)
BOOL **XMQuaternionNotEqual**(XMVECTOR Q1, XMVECTOR Q2)
BOOL **XMQuaternionIsIdentity**(XMVECTOR Q)
BOOL **XMQuaternionIsInfinite**(XMVECTOR Q)
BOOL **XMQuaternionIsNaN**(XMVECTOR Q)
VOID **XMQuaternionToAxisAngle**(XMVECTOR *pAxis, FLOAT *pAngle, XMVECTOR Q)

## Vector Load, Vector Store

Example signatures: XMVECTOR **XMLoadByte4**(CONST XMBYTE4 *pSource)
VOID **XMStoreByte4**(XMBYTE4 *pDestination, XMVECTOR V)

| | | | | |
|---|---|---|---|---|
| XMLoadByteN4 | XMStoreByteN4 | | | |
| XMLoadColor | XMStoreColor | | | |
| XMLoadDec4 | XMStoreDec4 | XMLoadDecN4 | XMStoreDecN4 | |
| XMLoadDHen3 | XMStoreDHen3 | XMLoadDHenN3 | XMStoreDHenN3 | |
| XMLoadFloat | XMStoreFloat | | | |
| XMLoadFloat2 | XMStoreFloat2 | XMLoadFloat2A | XMStoreFloat2A | |
| XMLoadFloat3 | XMStoreFloat3 | XMLoadFloat3A | XMStoreFloat3A | |
| XMLoadFloat3PK | XMStoreFloat3PK | | | |
| XMLoadFloat3SE | XMStoreFloat3SE | | | |
| XMLoadFloat3x3 | XMStoreFloat3x3 | XMStoreFloat3x3NC | | |
| XMLoadFloat4 | XMStoreFloat4 | XMLoadFloat4A | XMStoreFloat4A | XMStoreFloat4NC |
| XMLoadFloat4x3 | XMStoreFloat4x3 | XMLoadFloat4x3A | XMStoreFloat4x3A | XMStoreFloat4x3NC |
| XMLoadFloat4x4 | XMStoreFloat4x4 | XMLoadFloat4x4A | XMStoreFloat4x4A | XMStoreFloat4x4NC |
| XMLoadHalf2 | XMStoreHalf2 | | | |
| XMLoadHalf4 | XMStoreHalf4 | | | |
| XMLoadHenD3 | XMStoreHenD3 | XMLoadHenDN3 | XMStoreHenDN3 | |
| XMLoadIco4 | XMStoreIco4 | XMLoadIcoN4 | XMStoreIcoN4 | |
| XMLoadInt | XMStoreInt | | | |
| XMLoadInt2 | XMStoreInt2 | XMLoadInt2A | XMStoreInt2A | |
| XMLoadInt3 | XMStoreInt3 | XMLoadInt3A | XMStoreInt3A | |
| XMLoadInt4 | XMStoreInt4 | XMLoadInt4A | XMStoreInt4A | XMStoreInt4NC |
| XMLoadPacked4 | XMStorePacked4 | | | |
| XMLoadShort2 | XMStoreShort2 | XMLoadShortN2 | XMStoreShortN2 | |
| XMLoadShort4 | XMStoreShort4 | XMLoadShortN4 | XMStoreShortN4 | |
| XMLoadU555 | XMStoreU555 | | | |
| XMLoadU565 | XMStoreU565 | | | |
| XMLoadUByte4 | XMStoreUByte4 | XMLoadUByteN4 | XMStoreUByteN4 | |
| XMLoadUDec4 | XMStoreUDec4 | XMLoadUDecN4 | XMStoreUDecN4 | |
| XMLoadUDHen3 | XMStoreUDHen3 | XMLoadUDHenN3 | XMStoreUDHenN3 | |
| XMLoadUHenD3 | XMStoreUHenD3 | XMLoadUHenDN3 | XMStoreUHenDN3 | |
| XMLoadUIco4 | XMStoreUIco4 | XMLoadUIcoN4 | XMStoreUIcoN4 | |
| XMLoadUNibble4 | XMStoreUNibble4 | | | |
| XMLoadUShort2 | XMStoreUShort2 | XMLoadUShortN2 | XMStoreUShortN2 | |
| XMLoadUShort4 | XMStoreUShort4 | XMLoadUShortN4 | XMStoreUShortN4 | |
| XMLoadXDec4 | XMStoreXDec4 | XMLoadXDecN4 | XMStoreXDecN4 | |
| XMLoadXIco4 | XMStoreXIco4 | XMLoadXIcoN4 | XMStoreXIcoN4 | |