**Directxmath.h**

XMVECTOR XM\_CALLCONV XMLoadFloat2(const XMFLOAT2 \*pSource);

XMVECTOR XM\_CALLCONV XMLoadFloat3(const XMFLOAT3 \*pSource);

XMVECTOR XM\_CALLCONV XMLoadFloat4(const XMFLOAT4 \*pSource);

void XM\_CALLCONV XMStoreFloat2(XMFLOAT2 \*pDestination,FXMVECTOR V);

void XM\_CALLCONV XMStoreFloat3(XMFLOAT3 \*pDestination,FXMVECTOR V);

void XM\_CALLCONV XMStoreFloat4(XMFLOAT4 \*pDestination,FXMVECTOR V);

float XM\_CALLCONV XMVectorGetX(FXMVECTOR V);

float XM\_CALLCONV XMVectorGetY(FXMVECTOR V);

float XM\_CALLCONV XMVectorGetZ(FXMVECTOR V);

float XM\_CALLCONV XMVectorGetW(FXMVECTOR V);

XMVECTOR XM\_CALLCONV XMVectorSetX(FXMVECTOR V,float x);

XMVECTOR XM\_CALLCONV XMVectorSetY(FXMVECTOR V,float y);

XMVECTOR XM\_CALLCONV XMVectorSetZ(FXMVECTOR V,float z);

XMVECTOR XM\_CALLCONV XMVectorSetW(FXMVECTOR V,float w);

XM\_CONSTEXPR float XMConvertToRadians(float fDegrees);

XM\_CONSTEXPR float XMConvertToDegrees(float fRadians);

XMVECTOR XM\_CALLCONV XMVectorZero();

XMVECTOR XM\_CALLCONV XMVectorSplatOne();

XMVECTOR XM\_CALLCONV XMVectorSet(float x,float y,float z,float w);

XMVECTOR XM\_CALLCONV XMVectorReplicate(float Value);

XMVECTOR XM\_CALLCONV XMVectorSplatX(FXMVECTOR V);

XMVECTOR XM\_CALLCONV XMVectorSplatY(FXMVECTOR V);

XMVECTOR XM\_CALLCONV XMVectorSplatZ(FXMVECTOR V);

XMVECTOR XM\_CALLCONV XMVector3Length(FXMVECTOR V);

XMVECTOR XM\_CALLCONV XMVector3LengthSq(FXMVECTOR V);

XMVECTOR XM\_CALLCONV XMVector3Dot(FXMVECTOR V1,FXMVECTOR V2);

XMVECTOR XM\_CALLCONV XMVector3Cross(FXMVECTOR V1,FXMVECTOR V2);

XMVECTOR XM\_CALLCONV XMVector3Normalize(FXMVECTOR V);

XMVECTOR XM\_CALLCONV XMVector3Orthogonal(FXMVECTOR V);

XMVECTOR XM\_CALLCONV XMVector3AngleBetweenVectors(FXMVECTOR V1,FXMVECTOR V2);

void XM\_CALLCONV XMVector3ComponentsFromNormal(XMVECTOR \*pParallel,XMVECTOR \*pPerpendicular,FXMVECTOR V,FXMVECTOR Normal);

bool XM\_CALLCONV XMVector3Equal(FXMVECTOR V1,FXMVECTOR V2);

bool XM\_CALLCONV XMVector3NotEqual(FXMVECTOR V1,FXMVECTOR V2);