**BlastAssetModelSimple : BlastAssetModel : BlastAsset**

**BlastAssetModelSimple**

**RenderMaterial**

**BlastAssetModel**

**BlastModel**

**Material(\*.mtl)**

**Chunk(\*.obj)**

**Mesh**

**SimpleMesh**

**Vertex**

**BlastAsset**

**Renderer**

**ExtPxAsset(\*.bpxa)**

**ExtPxAsset(ExtPxAssetImpl)**

**TkAsset(TkAssetImpl)**

**NvBlastAsset(Nv::Blast::Asset)**

**ExtPxChunk**

**ExtPxSubchunk**

ExtPxAsset::deserialize

ExtPxAssetImpl::DataHeader header;

stream.read(&header, sizeof(header));

TkFrameworkImpl::deserialize

if (stream.readDword() != ClassID)

const auto it = m\_typeIDToIndex.find(stream.readDword());

if (stream.readDword() != type->getVersionInternal())

NvBlastID id;

stream.read(&id, sizeof(NvBlastID));

TkAssetImpl::deserialize

const uint32\_t assetSize = stream.readDword();

stream.read(asset->m\_assetLL, assetSize);

const uint32\_t jointDescCount = stream.readDword();

ExtPxAsset::deserialize

stream.read(asset->m\_chunks.begin(), sizeof(ExtPxChunk) \* asset->m\_chunks.size());

const uint32\_t subchunkCount = stream.readDword();

stream.read(&subChunk.transform, sizeof(PxTransform));

stream.read(&subChunk.geometry.scale, sizeof(PxMeshScale));