

# Maya Bifrost SDK

Dexter Studios R&D  
Wanho Choi

**Bifrost::API::StateServer**



**Bifrost::API::Component**

**Bifrost::API::Channel**

**Raw Data**

**Bifrost::API::String filePathName**



**Bifrost::API::StateServer stateServer**



**Bifrost::API::Component component**



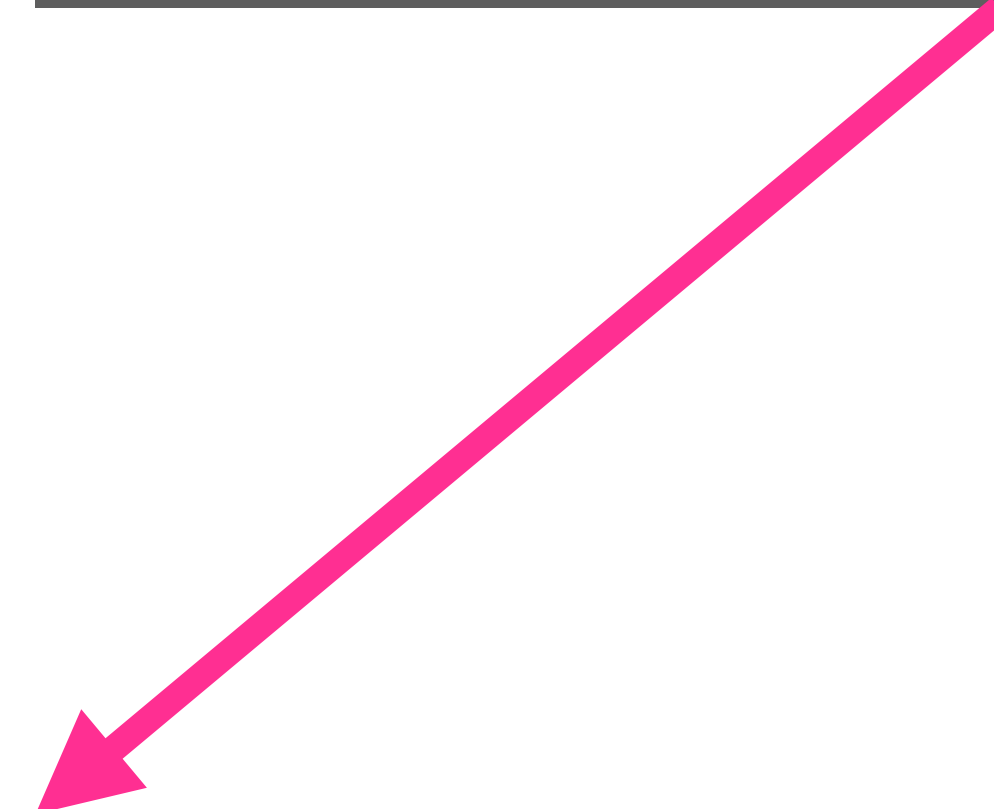
**Bifrost::API::Layout layout**



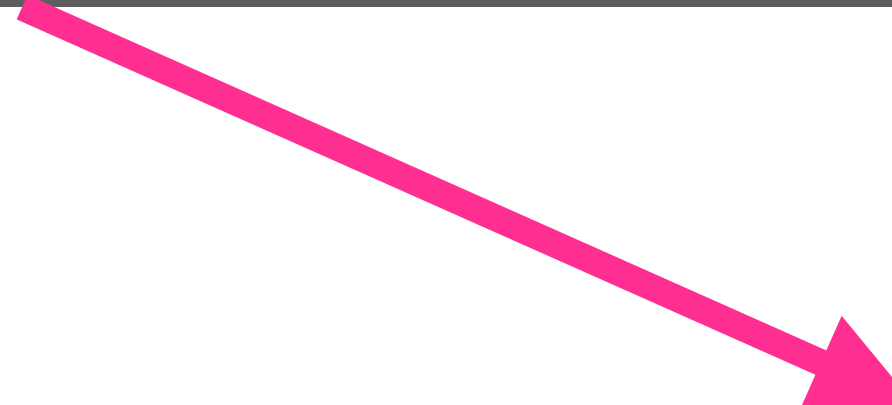
**Bifrost::API::Channel channel**



**Bifrost::API::TreeIndex treeIndex**



**Raw Data**



```
#typedef Bifrost::API::amino::Math::vec3f V3f

Bifrost::API:: ObjectModel om;
Bifrost::API:: FileIO fio = om.createFileIO( "particles.bif" );
Bifrost::API:: StateServer ss = fio.load( );

Bifrost::API:: Component component = ss.components()[0];
Bifrost::API:: Channel ch = component.findChannel( "position" );

Bifrost::API:: Layout layout = component.layout();

for( size_t d=0; d<layout.depthCount(); ++d ) {
    for( size_t t=0; t<layout.tileCount(d); ++t ) {
        Bifrost::API::TreeIndex tIndex(t,d);
        if ( !ch.elementCount(tIndex) ) { continue; }
        const Bifrost::API::TileData<V3f>& vArray = ch.tileData<V3f>(tIndex);
        for( size_t i=0; i<vArray.count(); ++i ) {
            const V3f& V = vArray[i];
            const float& x = V[i][0];
            const float& y = V[i][1];
            const float& z = V[i][2];
        }
    }
}
}
```