

v8::internal::wasm  
::anonymous\_namespace  
{liftoff-compiler::cc}  
::LiftoffCompiler::GlobalGet

v8::internal::wasm  
::anonymous\_namespace  
{liftoff-compiler::cc}  
::LiftoffCompiler::LoadAtomicObjectField

v8::internal::wasm  
::anonymous\_namespace  
{liftoff-compiler::cc}  
::LiftoffCompiler::LoadObjectField

v8::internal::wasm  
::LoadType::ForValueKind

```
graph LR; A["v8::internal::wasm::anonymous_namespace {liftoff-compiler::cc}::LiftoffCompiler::GlobalGet"] --> D["v8::internal::wasm::LoadType::ForValueKind"]; B["v8::internal::wasm::anonymous_namespace {liftoff-compiler::cc}::LiftoffCompiler::LoadAtomicObjectField"] --> D; C["v8::internal::wasm::anonymous_namespace {liftoff-compiler::cc}::LiftoffCompiler::LoadObjectField"] --> D;
```

The diagram illustrates a code transformation or linking process. Three source functions, each within a V8 internal WASM namespace and a liftoff-compiler anonymous namespace, are mapped to a single target function. The target function is also in the same namespace but is a static member of the LoadType struct. The source functions are GlobalGet, LoadAtomicObjectField, and LoadObjectField. The target function is ForValueKind. Arrows indicate the mapping from each source to the target.