# rootJS Architecture

**ROOT::TApplication**

**NodeHandler**

-rootJS: Persistent<Object>

-exposeROOT()
-exposeGlobalFunctions()
-exposeGlobals()
-exposeMacros()
-exposeClasses()
-exposeClass(clazz: TClassRef)
+getExports(): Local<Object>

**NodeApplication**

-Exit(args: void*)

+NodeApplication(acn: char*, argc: int*, argv: char**) «constructor»
-initROOTGlobals()
-initROOTMessageCallback()

**CallbackHandler**

+ctorCallback(args: FunctionCallbackInfo<Value>)
+staticCtorCallback(args: FunctionCallbackInfo<Value>)
+memberGetterCallback(property: Local<String>, info: PropertyCallbackInfo<Value>)
+memberSetterCallback(property: Local<String>, value: Local<Value>, info: PropertyCallbackInfo<Value>)
+memberFunctionCallback(args: FunctionCallbackInfo<Value>)
+staticGetterCallback(property: Local<String>, info: PropertyCallbackInfo<Value>)
+staticSetterCallback(property: Local<String>, value: Local<Value>, info: PropertyCallbackInfo<Value>)
+staticFunctionCallback(args: FunctionCallbackInfo<Value>)

**ROOT**

**EnumProxy**

**StructProxy**

**ArrayProxy**

**PointerProxy**

**ReferenceProxy**

*uses*

**ProxyFunctionFactory**

-ProxyFunctionFactory() «constructor»
+createProxyFunction(function: TFunction, scope: TClassRef): ProxyFunciton
+fromArgs(name: string, scope: TClassRef, args: FunctionCallbackInfo): ProxyFunction

**ProxyObjectFactory**

-ProxyObjectFactory() «constructor»
+createProxyObject(type: TDataMember, scope: TClassRef, holder: ProxyObject): ProxyObject

**TemplateFactory**

-cache: map<string, Persistent<FunctionTemplate>>

-TemplateFactory() «constructor»
+createTemplate(clazz: TClassRef): Local<FunctionTemplate>

*creates*

**ProxyFunction**

-functions: map<TFunction*, CallFunc*>

-processCall(method: TFunction*, args: void*, self: void*, result: void*): bool
-callConstructor(method: TFunction*, type: TClassRef, args: void*): void*
-callDestructor(type: TClassRef, self: void*): void
-callObject(method: TFunction*, self: void*, args: void*, resType: TClassRef): void*
<T> -callPrimitive(TFunction* method, void* self, void* args): T
+getCallFunc(method: TFunction*): CallFunc*
+getMethodsFromName(scope: TClassRef, name: string): vector<TFunction*>
+ProxyFunction(address: void*, function: TFunction, scope: TClassRef) «constructor»
+getType(): TFunction
+validateArgs(args: FunctionCallbackInfo): ProxyObject[]
+call(args: ProxyObject[]): ProxyObject

**Proxy**

#address: void*
#type: TObject
#scope: TClassRef

#Proxy(address: void*, type: TObject, scope: TClassRef) «constructor»
+setAddress(address: void*)
+getAddress(): void*
+getType(): TObject
+getScope(): TClassRef
+isGlobal(): bool
+isTemplate(): bool
+isConst(): bool
+isStatic(): bool

**ProxyObject**

#proxy: Persistent<Object>

+ProxyObject(type: TDataMember, scope: TClassRef) «constructor»
+getType(): TDataMember
+set(value: ProxyObject)
+get(): Local<Value>
+setProxy(proxy: Local<Object>)
+getProxy(): Local<Object>
+isPrimitive(): bool

**TemplateProxy**

**PrimitiveProxy**

**NumberProxy**

**StringProxy**

**BooleanProxy**

*uses*

*creates*