

KARLSRUHE INSTITUTE OF TECHNOLOGY

SOFTWARE ENGINEERING PRACTICE

WINTER TERM 2015/2016

# rootJS

Node.js bindings for ROOT 6

*Jonas Schwabe*

*Theo Beffart*

*Sachin Rajgopal*

*Christoph Wolff*

*Christoph Haas*

*Maximilian Früh*

supervised by  
Dr. Marek SZUBA

# Contents

<b>1</b>	<b>NodeApplication</b>	<b>3</b>
1.1	ctorCallback . . . . .	3
1.2	staticCtorCallback . . . . .	4
1.3	memberGetterCallback . . . . .	5
1.4	memberSetterCallback . . . . .	6
1.5	memberFunctionCallback . . . . .	7
1.6	staticGetterCallback . . . . .	8
1.7	staticSetterCallback . . . . .	9
1.8	staticFunctionCallback . . . . .	10
1.9	Initialize . . . . .	11
1.10	Instance . . . . .	12
1.11	getIsolate . . . . .	13
1.12	getExports . . . . .	14
1.13	getTemplateFactory . . . . .	15
1.14	getFunctionFactory . . . . .	16
1.15	getObjectFactory . . . . .	17
<b>2</b>	<b>TemplateFactory</b>	<b>18</b>
2.1	createTemplate . . . . .	18
<b>3</b>	<b>TemplateCache</b>	<b>19</b>
3.1	contains . . . . .	19
3.2	get . . . . .	20
3.3	store . . . . .	21
<b>4</b>	<b>ProxyFunctionFactory</b>	<b>22</b>
4.1	createProxyFunction . . . . .	22
4.2	fromArgs . . . . .	23
<b>5</b>	<b>ProxyFunction</b>	<b>24</b>
5.1	ProxyFunction . . . . .	24
5.2	convertArgs . . . . .	25
5.3	call . . . . .	26
5.4	isTemplateFunction . . . . .	27
<b>6</b>	<b>ProxyObjectFactory</b>	<b>28</b>
6.1	createProxyObject . . . . .	28
<b>7</b>	<b>ProxyObject</b>	<b>29</b>
7.1	ProxyObject . . . . .	29
7.2	getAddress . . . . .	30
7.3	getType . . . . .	31
7.4	set . . . . .	32
7.5	get . . . . .	33
7.6	isPrimitive . . . . .	34
<b>8</b>	<b>FunctionHelper</b>	<b>35</b>
8.1	GetCallFunc . . . . .	35
8.2	copyArgs . . . . .	36
8.3	FastCall . . . . .	37
8.4	CallV . . . . .	38
8.5	CallR . . . . .	39
8.6	CallS . . . . .	40

8.7	CallO . . . . .	41
8.8	CallConstructor . . . . .	42
8.9	CallDestructor . . . . .	43
8.10	IsConstructor . . . . .	44
8.11	IsPublicMethod . . . . .	45
8.12	IsStaticMethod . . . . .	46
8.13	IsConstMethod . . . . .	47
8.14	IsMethodTemplate . . . . .	48
8.15	GetMethodNumTemplateArgs . . . . .	49
8.16	GetMethodTemplateArgName . . . . .	50
8.17	GetNumMethods . . . . .	51
8.18	GetMethodIndexAt . . . . .	52
8.19	GetMethodsFromName . . . . .	53
8.20	GetMethod . . . . .	54
8.21	GetMethodName . . . . .	55
8.22	GetMethodResultType . . . . .	56
8.23	GetMethodNumArgs . . . . .	57
8.24	GetMethodReqArgs . . . . .	58
8.25	GetMethodArgName . . . . .	59
8.26	GetMethodArgType . . . . .	60
8.27	GetMethodArgDefault . . . . .	61
8.28	GetMethodSignature . . . . .	62
<b>9</b>	<b>ClassHelper</b>	<b>63</b>
9.1	IsNamespace . . . . .	63
9.2	IsAbstract . . . . .	64
9.3	IsEnum . . . . .	65
9.4	IsStruct . . . . .	66
9.5	GetFinalName . . . . .	67
9.6	GetScopedFinalName . . . . .	68
9.7	GetNumBases . . . . .	69
9.8	GetBaseName . . . . .	70
9.9	IsSubtype . . . . .	71
9.10	GetNumDatamembers . . . . .	72
9.11	GetDatamemberName . . . . .	73
9.12	GetDatamemberType . . . . .	74
9.13	GetDatamemberOffset . . . . .	75
9.14	GetDatamemberIndex . . . . .	76
9.15	IsPublicData . . . . .	77
9.16	IsStaticData . . . . .	78
9.17	IsConstData . . . . .	79
9.18	IsEnumData . . . . .	80
9.19	resolveAddress . . . . .	81
9.20	resolveAddress . . . . .	82

# 1. NodeApplication

describe class NodeApplication here

## 1.1. ctorCallback

<i>Name</i>	NodeApplication::ctorCallback(args: FunctionCallbackInfo<Value>)
<i>Visibility</i>	
<i>Parameters</i>	args: FunctionCallbackInfo<Value>
<i>Return value</i>	<b>none</b>
<i>behavior</i>	describe beahviour

## 1.2. staticCtorCallback

<i>Name</i>	<code>NodeApplication::staticCtorCallback(args: FunctionCallbackInfo&lt;Value&gt;)</code>
<i>Visibility</i>	
<i>Parameters</i>	<code>args: FunctionCallbackInfo&lt;Value&gt;</code>
<i>Return value</i>	<b>none</b>
<i>behavior</i>	describe beahviour

### 1.3. memberGetterCallback

<i>Name</i>	<code>NodeApplication::memberGetterCallback(property: Local&lt;String&gt;, info: PropertyCallbackInfo&lt;Value&gt;)</code>
<i>Visibility</i>	
<i>Parameters</i>	<code>property: Local&lt;String&gt;, info: PropertyCallbackInfo&lt;Value&gt;</code>
<i>Return value</i>	<b>none</b>
<i>behavior</i>	describe beahviour

## 1.4. memberSetterCallback

<i>Name</i>	<code>NodeApplication::memberSetterCallback(property: Local&lt;String&gt;, value: Local&lt;Value&gt;, info: PropertyCallbackInfo&lt;Value&gt;)</code>
<i>Visibility</i>	
<i>Parameters</i>	<code>property: Local&lt;String&gt;, value: Local&lt;Value&gt;, info: PropertyCallbackInfo&lt;Value&gt;</code>
<i>Return value</i>	<b>none</b>
<i>behavior</i>	describe beahviour

## 1.5. memberFunctionCallback

<i>Name</i>	<code>NodeApplication::memberFunctionCallback(args: FunctionCallbackInfo&lt;Value&gt;)</code>
<i>Visibility</i>	
<i>Parameters</i>	<code>args: FunctionCallbackInfo&lt;Value&gt;</code>
<i>Return value</i>	<b>none</b>
<i>behavior</i>	describe beahviour



## 1.6. staticGetterCallback

<i>Name</i>	<code>NodeApplication::staticGetterCallback(property: Local&lt;String&gt;, info: PropertyCallbackInfo&lt;Value&gt;)</code>
<i>Visibility</i>	
<i>Parameters</i>	<code>property: Local&lt;String&gt;, info: PropertyCallbackInfo&lt;Value&gt;</code>
<i>Return value</i>	<b>none</b>
<i>behavior</i>	describe beahviour

## 1.7. staticSetterCallback

<i>Name</i>	<code>NodeApplication::staticSetterCallback(property: Local&lt;String&gt;, value: Local&lt;Value&gt;, info: PropertyCallbackInfo&lt;Value&gt;)</code>
<i>Visibility</i>	
<i>Parameters</i>	<code>property: Local&lt;String&gt;, value: Local&lt;Value&gt;, info: PropertyCallbackInfo&lt;Value&gt;</code>
<i>Return value</i>	<b>none</b>
<i>behavior</i>	describe beahviour

## 1.8. staticFunctionCallback

<i>Name</i>	<code>NodeApplication::staticFunctionCallback(args: FunctionCallbackInfo&lt;Value&gt;)</code>
<i>Visibility</i>	
<i>Parameters</i>	<code>args: FunctionCallbackInfo&lt;Value&gt;</code>
<i>Return value</i>	<b>none</b>
<i>behavior</i>	describe beahviour

## 1.9. Initialize

<i>Name</i>	<code>NodeApplication::Initialize(exports: Local&lt;Object&gt;, module: Local&lt;Object&gt;)</code>
<i>Visibility</i>	
<i>Parameters</i>	<code>exports: Local&lt;Object&gt;, module: Local&lt;Object&gt;</code>
<i>Return value</i>	<b>none</b>
<i>behavior</i>	describe beahviour

## 1.10. Instance

<i>Name</i>	<code>NodeApplication::Instance()</code>
<i>Visibility</i>	
<i>Parameters</i>	<i>none</i>
<i>Return value</i>	<b>NodeApplication</b> describe return value
<i>behavior</i>	describe beahviour

## 1.11. getIsolate

<i>Name</i>	NodeApplication::getIsolate()
<i>Visibility</i>	
<i>Parameters</i>	<i>none</i>
<i>Return value</i>	<b>Isolate*</b> describe return value
<i>behavior</i>	describe beahviour

## 1.12. getExports

<i>Name</i>	NodeApplication::getExports()
<i>Visibility</i>	
<i>Parameters</i>	<i>none</i>
<i>Return value</i>	<b>Local&lt;Object&gt;</b> describe return value
<i>behavior</i>	describe beahviour

### 1.13. getTemplateFactory

<i>Name</i>	NodeApplication::getTemplateFactory()
<i>Visibility</i>	
<i>Parameters</i>	<i>none</i>
<i>Return value</i>	<b>TemplateFactory</b> describe return value
<i>behavior</i>	describe beahviour



## 1.14. getFunctionFactory

<i>Name</i>	<code>NodeApplication::getFunctionFactory()</code>
<i>Visibility</i>	
<i>Parameters</i>	<i>none</i>
<i>Return value</i>	<b>ProxyFunctionFactory</b> describe return value
<i>behavior</i>	describe beahviour

## 1.15. getObjectFactory

<i>Name</i>	<code>NodeApplication::getObjectFactory()</code>
<i>Visibility</i>	
<i>Parameters</i>	<i>none</i>
<i>Return value</i>	<b>ProxyObjectFactory</b> describe return value
<i>behavior</i>	describe beahviour

## 2. TemplateFactory

describe class TemplateFactory here

### 2.1. createTemplate

<i>Name</i>	TemplateFactory::createTemplate(clazz: TClassRef)
<i>Visibility</i>	
<i>Parameters</i>	clazz: TClassRef
<i>Return value</i>	<b>Local&lt;FunctionTemplate&gt;</b> describe return value
<i>behavior</i>	describe beahviour

## 3. TemplateCache

describe class TemplateCache here

### 3.1. contains

<i>Name</i>	TemplateCache::contains(type: TClassRef)
<i>Visibility</i>	
<i>Parameters</i>	<i>type: TClassRef</i>
<i>Return value</i>	<b>bool</b> describe return value
<i>behavior</i>	describe beahviour

### 3.2. get

<i>Name</i>	<code>TemplateCache::get(type: TClassRef)</code>
<i>Visibility</i>	
<i>Parameters</i>	<i>type: TClassRef</i>
<i>Return value</i>	<b>Local&lt;FunctionTemplate&gt;</b> describe return value
<i>behavior</i>	describe beahviour

### 3.3. store

<i>Name</i>	<code>TemplateCache::store(type: TClassRef, tpl: Local&lt;FunctionTemplate&gt;)</code>
<i>Visibility</i>	
<i>Parameters</i>	<i>type: TClassRef, tpl: Local&lt;FunctionTemplate&gt;</i>
<i>Return value</i>	<b>none</b>
<i>behavior</i>	describe beahviour

## 4. ProxyFunctionFactory

describe class ProxyFunctionFactory here

### 4.1. createProxyFunction

<i>Name</i>	ProxyFunctionFactory::createProxyFunction(info: TMethod)
<i>Visibility</i>	
<i>Parameters</i>	<i>info: TMethod</i>
<i>Return value</i>	<b>ProxyFunction</b> describe return value
<i>behavior</i>	describe beahviour

## 4.2. fromArgs

<i>Name</i>	<code>ProxyFunctionFactory::fromArgs(name: string, clazz: TClassRef, args: FunctionCallbackInfo)</code>
<i>Visibility</i>	
<i>Parameters</i>	<code>name: string, clazz: TClassRef, args: FunctionCallbackInfo</code>
<i>Return value</i>	<b>ProxyFunction</b> describe return value
<i>behavior</i>	describe beahviour



## 5. ProxyFunction

describe class ProxyFunction here

### 5.1. ProxyFunction

<i>Name</i>	ProxyFunction::ProxyFunction(address: void*, info: TFunction)
<i>Visibility</i>	
<i>Parameters</i>	address: void*, info: TFunction
<i>Return value</i>	« <b>constructor</b> » describe return value
<i>behavior</i>	describe beahviour

## 5.2. convertArgs

<i>Name</i>	<code>ProxyFunction::convertArgs(args: FunctionCallbackInfo)</code>
<i>Visibility</i>	
<i>Parameters</i>	<i>args: FunctionCallbackInfo</i>
<i>Return value</i>	<b>ProxyObject[]</b> describe return value
<i>behavior</i>	describe beahviour

### 5.3. call

<i>Name</i>	<code>ProxyFunction::call(args: ProxyObject[])</code>
<i>Visibility</i>	
<i>Parameters</i>	<code>args: ProxyObject[]</code>
<i>Return value</i>	<b>ProxyObject</b> describe return value
<i>behavior</i>	describe beahviour

## 5.4. isTemplateFunction

<i>Name</i>	<code>ProxyFunction::isTemplateFunction()</code>
<i>Visibility</i>	
<i>Parameters</i>	<i>none</i>
<i>Return value</i>	<b>bool</b> describe return value
<i>behavior</i>	describe beahviour

## 6. ProxyObjectFactory

describe class ProxyObjectFactory here

### 6.1. createProxyObject

<i>Name</i>	ProxyObjectFactory::createProxyObject(type: TDataMember, holder: ProxyObject)
<i>Visibility</i>	
<i>Parameters</i>	<i>type: TDataMember, holder: ProxyObject</i>
<i>Return value</i>	<b>ProxyObject</b> describe return value
<i>behavior</i>	describe beahviour

## 7. ProxyObject

describe class ProxyObject here

### 7.1. ProxyObject

<i>Name</i>	ProxyObject::ProxyObject(address: void*, type: TDataMember)
<i>Visibility</i>	
<i>Parameters</i>	address: void*, type: TDataMember
<i>Return value</i>	« <b>constructor</b> » describe return value
<i>behavior</i>	describe beahviour

## 7.2. getAddress

<i>Name</i>	<code>ProxyObject::getAddress()</code>
<i>Visibility</i>	
<i>Parameters</i>	<i>none</i>
<i>Return value</i>	<b>void*</b> describe return value
<i>behavior</i>	describe beahviour

### 7.3. getType

<i>Name</i>	ProxyObject::getType()
<i>Visibility</i>	
<i>Parameters</i>	none
<i>Return value</i>	<b>TDataMember</b> describe return value
<i>behavior</i>	describe beahviour



## 7.4. set

<i>Name</i>	<code>ProxyObject::set(value: ProxyObject)</code>
<i>Visibility</i>	
<i>Parameters</i>	<code>value: ProxyObject</code>
<i>Return value</i>	<b>none</b>
<i>behavior</i>	describe beahviour

## 7.5. get

<i>Name</i>	<code>ProxyObject::get()</code>
<i>Visibility</i>	
<i>Parameters</i>	<i>none</i>
<i>Return value</i>	<b>Local</b> <Value> describe return value
<i>behavior</i>	describe beahviour

## 7.6. isPrimitive

<i>Name</i>	<code>ProxyObject::isPrimitive()</code>
<i>Visibility</i>	
<i>Parameters</i>	<i>none</i>
<i>Return value</i>	<b>bool</b> describe return value
<i>behavior</i>	describe beahviour

## 8. FunctionHelper

describe class FunctionHelper here

### 8.1. GetCallFunc

<i>Name</i>	<code>FunctionHelper::GetCallFunc(method: TCppMethod)</code>
<i>Visibility</i>	
<i>Parameters</i>	<i>method: TCppMethod</i>
<i>Return value</i>	<b>CallFunc*</b> describe return value
<i>behavior</i>	describe beahviour

## 8.2. copyArgs

<i>Name</i>	FunctionHelper::copyArgs(args: void*, vars: void**)
<i>Visibility</i>	
<i>Parameters</i>	args: void*, vars: void**
<i>Return value</i>	<b>void</b> describe return value
<i>behavior</i>	describe beahviour

### 8.3. FastCall

<i>Name</i>	<code>FunctionHelper::FastCall(method: TCppMethod, args: void*, self: void*, result: void*)</code>
<i>Visibility</i>	
<i>Parameters</i>	<code>method: TCppMethod, args: void*, self: void*, result: void*</code>
<i>Return value</i>	<b>bool</b> describe return value
<i>behavior</i>	describe beahviour

## 8.4. CallV

<i>Name</i>	FunctionHelper::CallV(method: TCppMethod, self: TCppObject, args: void*)
<i>Visibility</i>	
<i>Parameters</i>	method: TCppMethod, self: TCppObject, args: void*
<i>Return value</i>	<b>void</b> describe return value
<i>behavior</i>	describe beahviour

## 8.5. CallR

<i>Name</i>	FunctionHelper::CallR(method: TCppMethod, self: TCppObject, args: void*)
<i>Visibility</i>	
<i>Parameters</i>	method: TCppMethod, self: TCppObject, args: void*
<i>Return value</i>	<b>void*</b> describe return value
<i>behavior</i>	describe beahviour



## 8.6. CallS

<i>Name</i>	FunctionHelper::CallS(method: TCppMethod, self: TCppObject, args: void*)
<i>Visibility</i>	
<i>Parameters</i>	method: TCppMethod, self: TCppObject, args: void*
<i>Return value</i>	<b>Char*</b> describe return value
<i>behavior</i>	describe beahviour

## 8.7. CallO

<i>Name</i>	FunctionHelper::CallO(method: TCppMethod, self: TCppObject, args: void*, resultype: TCppType)
<i>Visibility</i>	
<i>Parameters</i>	method: TCppMethod, self: TCppObject, args: void*, resultype: TCppType
<i>Return value</i>	<b>TCppObject</b> describe return value
<i>behavior</i>	describe beahviour

## 8.8. CallConstructor

<i>Name</i>	<code>FunctionHelper::CallConstructor(method: TCppMethod, klass: TCppType, args: void*)</code>
<i>Visibility</i>	
<i>Parameters</i>	<i>method: TCppMethod, klass: TCppType, args: void*</i>
<i>Return value</i>	<b>TCppObject</b> describe return value
<i>behavior</i>	describe beahviour

## 8.9. CallDestructor

<i>Name</i>	<code>FunctionHelper::CallDestructor(type: TCppType, self: TCppObject)</code>
<i>Visibility</i>	
<i>Parameters</i>	<i>type: TCppType, self: TCppObject</i>
<i>Return value</i>	<b>void</b> describe return value
<i>behavior</i>	describe beahviour

## 8.10. IsConstructor

<i>Name</i>	<code>FunctionHelper::IsConstructor(method: TCppMethod)</code>
<i>Visibility</i>	
<i>Parameters</i>	<i>method: TCppMethod</i>
<i>Return value</i>	<b>bool</b> describe return value
<i>behavior</i>	describe beahviour

## 8.11. IsPublicMethod

<i>Name</i>	<code>FunctionHelper::IsPublicMethod(method: TCppMethod)</code>
<i>Visibility</i>	
<i>Parameters</i>	<i>method: TCppMethod</i>
<i>Return value</i>	<b>bool</b> describe return value
<i>behavior</i>	describe beahviour

## 8.12. IsStaticMethod

<i>Name</i>	<code>FunctionHelper::IsStaticMethod(method: TCppMethod)</code>
<i>Visibility</i>	
<i>Parameters</i>	<i>method: TCppMethod</i>
<i>Return value</i>	<b>bool</b> describe return value
<i>behavior</i>	describe beahviour

### 8.13. IsConstMethod

<i>Name</i>	<code>FunctionHelper::IsConstMethod(method: TCppMethod)</code>
<i>Visibility</i>	
<i>Parameters</i>	<i>method: TCppMethod</i>
<i>Return value</i>	<b>bool</b> describe return value
<i>behavior</i>	describe beahviour



## 8.14. IsMethodTemplate

<i>Name</i>	<code>FunctionHelper::IsMethodTemplate(method: TCppMethod)</code>
<i>Visibility</i>	
<i>Parameters</i>	<i>method: TCppMethod</i>
<i>Return value</i>	<b>bool</b> describe return value
<i>behavior</i>	describe beahviour

## 8.15. GetMethodNumTemplateArgs

<i>Name</i>	<code>FunctionHelper::GetMethodNumTemplateArgs(scope: TCppScope, imeth: TCppIndex)</code>
<i>Visibility</i>	
<i>Parameters</i>	<code>scope: TCppScope, imeth: TCppIndex</code>
<i>Return value</i>	<b>TCppIndex</b> describe return value
<i>behavior</i>	describe beahviour

## 8.16. GetMethodTemplateName

<i>Name</i>	<code>FunctionHelper::GetMethodTemplateName(scope: TCppScope, imeth: TCppIndex, iarg: TCppIndex)</code>
<i>Visibility</i>	
<i>Parameters</i>	<code>scope: TCppScope, imeth: TCppIndex, iarg: TCppIndex</code>
<i>Return value</i>	<b>string</b> describe return value
<i>behavior</i>	describe beahviour

## 8.17. GetNumMethods

<i>Name</i>	<code>FunctionHelper::GetNumMethods(scope: TCppScope)</code>
<i>Visibility</i>	
<i>Parameters</i>	<i>scope: TCppScope</i>
<i>Return value</i>	<b>TCppIndex</b> describe return value
<i>behavior</i>	describe beahviour

## 8.18. GetMethodIndexAt

<i>Name</i>	<code>FunctionHelper::GetMethodIndexAt(scope: TCppScope, imeth: TCppIndex)</code>
<i>Visibility</i>	
<i>Parameters</i>	<code>scope: TCppScope, imeth: TCppIndex</code>
<i>Return value</i>	<b>TCppIndex</b> describe return value
<i>behavior</i>	describe beahviour

## 8.19. GetMethodsFromName

<i>Name</i>	<code>FunctionHelper::GetMethodsFromName(scope: TCppScope, name: string)</code>
<i>Visibility</i>	
<i>Parameters</i>	<code>scope: TCppScope, name: string</code>
<i>Return value</i>	<code>vector&lt;TCppMethod&gt;</code> describe return value
<i>behavior</i>	describe beahviour

## 8.20. GetMethod

<i>Name</i>	<code>FunctionHelper::GetMethod(scope: TCppScope, imeth: TCppIndex)</code>
<i>Visibility</i>	
<i>Parameters</i>	<code>scope: TCppScope, imeth: TCppIndex</code>
<i>Return value</i>	<b>TCppMethod</b> describe return value
<i>behavior</i>	describe beahviour

## 8.21. GetMethodName

<i>Name</i>	<code>FunctionHelper::GetMethodName(method: TCppMethod)</code>
<i>Visibility</i>	
<i>Parameters</i>	<i>method: TCppMethod</i>
<i>Return value</i>	<b>string</b> describe return value
<i>behavior</i>	describe beahviour



## 8.22. GetMethodResultType

<i>Name</i>	<code>FunctionHelper::GetMethodResultType(method: TCppMethod)</code>
<i>Visibility</i>	
<i>Parameters</i>	<i>method: TCppMethod</i>
<i>Return value</i>	<b>string</b> describe return value
<i>behavior</i>	describe beahviour

## 8.23. GetMethodNumArgs

<i>Name</i>	<code>FunctionHelper::GetMethodNumArgs (method: TCppMethod)</code>
<i>Visibility</i>	
<i>Parameters</i>	<i>method: TCppMethod</i>
<i>Return value</i>	<b>TCppIndex</b> describe return value
<i>behavior</i>	describe beahviour

## 8.24. GetMethodReqArgs

<i>Name</i>	<code>FunctionHelper::GetMethodReqArgs (method: TCppMethod)</code>
<i>Visibility</i>	
<i>Parameters</i>	<i>method: TCppMethod</i>
<i>Return value</i>	<b>TCppIndex</b> describe return value
<i>behavior</i>	describe beahviour

## 8.25. GetMethodArgName

<i>Name</i>	<code>FunctionHelper::GetMethodArgName(method: TCppMethod, iarg: int)</code>
<i>Visibility</i>	
<i>Parameters</i>	<i>method: TCppMethod, iarg: int</i>
<i>Return value</i>	<b>string</b> describe return value
<i>behavior</i>	describe beahviour

## 8.26. GetMethodArgType

<i>Name</i>	<code>FunctionHelper::GetMethodArgType(method: TCppMethod, iarg: int)</code>
<i>Visibility</i>	
<i>Parameters</i>	<i>method: TCppMethod, iarg: int</i>
<i>Return value</i>	<b>string</b> describe return value
<i>behavior</i>	describe beahviour

## 8.27. GetMethodArgDefault

<i>Name</i>	<code>FunctionHelper::GetMethodArgDefault(method: TCppMethod, iarg: int)</code>
<i>Visibility</i>	
<i>Parameters</i>	<i>method: TCppMethod, iarg: int</i>
<i>Return value</i>	<b>string</b> describe return value
<i>behavior</i>	describe beahviour

## 8.28. GetMethodSignature

<i>Name</i>	FunctionHelper::GetMethodSignature(scope: TCppScope, imeth: TCppIndex)
<i>Visibility</i>	
<i>Parameters</i>	scope: TCppScope, imeth: TCppIndex
<i>Return value</i>	<b>string</b> describe return value
<i>behavior</i>	describe beahviour

## 9. ClassHelper

describe class ClassHelper here

### 9.1. IsNamespace

<i>Name</i>	<code>ClassHelper::IsNamespace(scope: TCppScope)</code>
<i>Visibility</i>	
<i>Parameters</i>	<i>scope: TCppScope</i>
<i>Return value</i>	<b>bool</b> describe return value
<i>behavior</i>	describe beahviour



## 9.2. IsAbstract

<i>Name</i>	<code>ClassHelper::IsAbstract(klass: TCppType)</code>
<i>Visibility</i>	
<i>Parameters</i>	<i>klass: TCppType</i>
<i>Return value</i>	<b>bool</b> describe return value
<i>behavior</i>	describe beahviour

### 9.3. IsEnum

<i>Name</i>	<code>ClassHelper::IsEnum(typename: string)</code>
<i>Visibility</i>	
<i>Parameters</i>	<i>typename: string</i>
<i>Return value</i>	<b>bool</b> describe return value
<i>behavior</i>	describe beahviour

## 9.4. IsStruct

<i>Name</i>	<code>ClassHelper::IsStruct(typename: string)</code>
<i>Visibility</i>	
<i>Parameters</i>	<i>typename: string</i>
<i>Return value</i>	<b>bool</b> describe return value
<i>behavior</i>	describe beahviour

## 9.5. GetFinalName

<i>Name</i>	<code>ClassHelper::GetFinalName(klass: TCppType)</code>
<i>Visibility</i>	
<i>Parameters</i>	<i>klass: TCppType</i>
<i>Return value</i>	<b>string</b> describe return value
<i>behavior</i>	describe beahviour

## 9.6. GetScopedFinalName

<i>Name</i>	<code>ClassHelper::GetScopedFinalName(klass: TCppType)</code>
<i>Visibility</i>	
<i>Parameters</i>	<i>klass: TCppType</i>
<i>Return value</i>	<b>string</b> describe return value
<i>behavior</i>	describe beahviour

## 9.7. GetNumBases

<i>Name</i>	<code>ClassHelper::GetNumBases(klass: TCppType)</code>
<i>Visibility</i>	
<i>Parameters</i>	<i>klass: TCppType</i>
<i>Return value</i>	<b>TCppTypeIndex</b> describe return value
<i>behavior</i>	describe beahviour

## 9.8. GetBaseName

<i>Name</i>	<code>ClassHelper::GetBaseName(klass: TCppType, ibase: TCppTypeIndex)</code>
<i>Visibility</i>	
<i>Parameters</i>	<i>klass: TCppType, ibase: TCppTypeIndex</i>
<i>Return value</i>	<b>string</b> describe return value
<i>behavior</i>	describe beahviour

## 9.9. IsSubtype

<i>Name</i>	<code>ClassHelper::IsSubtype(derived: TCppType, base: TCppType)</code>
<i>Visibility</i>	
<i>Parameters</i>	<i>derived: TCppType, base: TCppType</i>
<i>Return value</i>	<b>bool</b> describe return value
<i>behavior</i>	describe beahviour



## 9.10. GetNumDatamembers

<i>Name</i>	<code>ClassHelper::GetNumDatamembers(scope: TCppScope)</code>
<i>Visibility</i>	
<i>Parameters</i>	<i>scope: TCppScope</i>
<i>Return value</i>	<b>TCppIndex</b> describe return value
<i>behavior</i>	describe beahviour

## 9.11. GetDataMemberName

<i>Name</i>	<code>ClassHelper::GetDataMemberName(scope: TCppScope, idata: TCppIndex)</code>
<i>Visibility</i>	
<i>Parameters</i>	<i>scope: TCppScope, idata: TCppIndex</i>
<i>Return value</i>	<b>string</b> describe return value
<i>behavior</i>	describe beahviour

## 9.12. GetDataMemberType

<i>Name</i>	<code>ClassHelper::GetDataMemberType(scope: TCppScope, idata: TCppIndex)</code>
<i>Visibility</i>	
<i>Parameters</i>	<i>scope: TCppScope, idata: TCppIndex</i>
<i>Return value</i>	<b>string</b> describe return value
<i>behavior</i>	describe beahviour

### 9.13. GetDataMemberOffset

<i>Name</i>	<code>ClassHelper::GetDataMemberOffset(scope: TCppScope, idata: TCppIndex)</code>
<i>Visibility</i>	
<i>Parameters</i>	<i>scope: TCppScope, idata: TCppIndex</i>
<i>Return value</i>	<b>ptrdiff</b> describe return value
<i>behavior</i>	describe beahviour

## 9.14. GetDataMemberIndex

<i>Name</i>	<code>ClassHelper::GetDataMemberIndex(scope: TCppScope, name: string)</code>
<i>Visibility</i>	
<i>Parameters</i>	<code>scope: TCppScope, name: string</code>
<i>Return value</i>	<b>TCppIndex</b> describe return value
<i>behavior</i>	describe beahviour

## 9.15. IsPublicData

<i>Name</i>	<code>ClassHelper::IsPublicData(scope: TCppScope, idata: TCppIndex)</code>
<i>Visibility</i>	
<i>Parameters</i>	<i>scope: TCppScope, idata: TCppIndex</i>
<i>Return value</i>	<b>bool</b> describe return value
<i>behavior</i>	describe beahviour

## 9.16. IsStaticData

<i>Name</i>	<code>ClassHelper::IsStaticData(scope: TCppScope, idata: TCppIndex)</code>
<i>Visibility</i>	
<i>Parameters</i>	<i>scope: TCppScope, idata: TCppIndex</i>
<i>Return value</i>	<b>bool</b> describe return value
<i>behavior</i>	describe beahviour

## 9.17. IsConstData

<i>Name</i>	<code>ClassHelper::IsConstData(scope: TCppScope, idata: TCppIndex)</code>
<i>Visibility</i>	
<i>Parameters</i>	<i>scope: TCppScope, idata: TCppIndex</i>
<i>Return value</i>	<b>bool</b> describe return value
<i>behavior</i>	describe beahviour



## 9.18. IsEnumData

<i>Name</i>	<code>ClassHelper::IsEnumData(scope: TCppScope, idata: TCppIndex)</code>
<i>Visibility</i>	
<i>Parameters</i>	<i>scope: TCppScope, idata: TCppIndex</i>
<i>Return value</i>	<b>bool</b> describe return value
<i>behavior</i>	describe beahviour

## 9.19. resolveAddress

<i>Name</i>	<code>ClassHelper::resolveAddress(staticMember: TDataMember, clazz: TClassRef)</code>
<i>Visibility</i>	
<i>Parameters</i>	<i>staticMember: TDataMember, clazz: TClassRef</i>
<i>Return value</i>	<b>void*</b> describe return value
<i>behavior</i>	describe beahviour

## 9.20. resolveAddress

<i>Name</i>	<code>ClassHelper::resolveAddress(staticMember: TDataMember, clazz: TClassRef)</code>
<i>Visibility</i>	
<i>Parameters</i>	<i>staticMember: TDataMember, clazz: TClassRef</i>
<i>Return value</i>	<b>void*</b> describe return value
<i>behavior</i>	describe beahviour