

Namespace PascalABCCompiler

Classes

[CompilationUnit](#)

[CompilationUnitHashTable](#)

[Compiler](#)

[CompilerInternalDebug](#)

[CompilerOptions](#)

Опции компиляции

[CompilerOptions.StandardModule](#)

[ConsoleCompilerConstants](#)

[DocXmlManager](#)

[ProjectInfo](#)

[ReferenceInfo](#)

[RemoteCompiler](#)

[RemoteCompilerError](#)

[RemoteCompilerInternalError](#)

[RemoteCompilerWarning](#)

[ResourceInfo](#)

[SourceCodeFileInfo](#)

[SupportedSourceFile](#)

[TooOldProjectFileVersion](#)

Interfaces

[ICompiler](#)

[IFileInfo](#)

[IProjectInfo](#)

[IReferenceInfo](#)

[IRemoteCompiler](#)

[IResourceInfo](#)

Enums

[CompilerOptions.OutputType](#)

[CompilerOptions.StandardModuleAddMethod](#)

[CompilerState](#)

[CompilerType](#)

[ProjectType](#)

[RemoteCompilerChannelEventType](#)

[UnitState](#)

Delegates

[ChangeCompilerStateEventDelegate](#)

[RemoteCompiler.EnvorimentIdleDelegate](#)

[RemoteCompilerChannelEventDelegate](#)

Delegate ChangeCompilerStateEventDelegate

Namespace: [PascalABCCompiler](#)

Assembly: Compiler.dll

```
public delegate void ChangeCompilerStateEventDelegate(ICompiler sender, CompilerState State,
string FileName)
```

Parameters

sender [ICompiler](#)

State [CompilerState](#)

FileName [string](#)

Constructors

ChangeCompilerStateEventDelegate(object, IntPtr)

```
public ChangeCompilerStateEventDelegate(object @object, IntPtr method)
```

Parameters

object [object](#)

method [IntPtr](#)

Methods

BeginInvoke(ICompiler, CompilerState, string, AsyncCallback, object)

```
public virtual IAsyncResult BeginInvoke(ICompiler sender, CompilerState State, string
FileName, AsyncCallback callback, object @object)
```

Parameters

sender [ICompiler](#)

State [CompilerState](#)

FileName [string](#)

callback [AsyncCallback](#)

object [object](#)

Returns

[IAsyncResult](#)

EndInvoke(IAsyncResult)

```
public virtual void EndInvoke(IAsyncResult result)
```

Parameters

result [IAsyncResult](#)

Invoke(ICompiler, CompilerState, string)

```
public virtual void Invoke(ICompiler sender, CompilerState State, string FileName)
```

Parameters

sender [ICompiler](#)

State [CompilerState](#)

FileName [string](#)

Class CompilationUnit

Namespace: [PascalABCCompiler](#)

Assembly: Compiler.dll

```
public class CompilationUnit
```

Inheritance

[object](#) ← CompilationUnit

Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.FieldSetter\(string, string, object\)](#) ,
[object.FieldGetter\(string, string, ref object\)](#) , [object.GetFieldInfo\(string, string\)](#)

Constructors

CompilationUnit()

```
public CompilationUnit()
```

Fields

CaseSensitive

```
internal bool CaseSensitive
```

Field Value

[bool](#)

Documented

```
public bool Documented
```

Field Value

[bool](#)

ErrorList

```
public List<Error> ErrorList
```

Field Value

[List](#)<Error>

State

```
public UnitState State
```

Field Value

[UnitState](#)

UnitFileName

```
public string UnitFileName
```

Field Value

[string](#)

_implementation_using_namespace_list

```
private using_namespace_list _implementation_using_namespace_list
```

Field Value

using_namespace_list

_interface_using_namespace_list

```
private using_namespace_list _interface_using_namespace_list
```

Field Value

using_namespace_list

_semanticTree

```
private unit_node _semanticTree
```

Field Value

unit_node

_syntaxTree

```
private compilation_unit _syntaxTree
```

Field Value

compilation_unit

currentUsedUnitId

поле для проверки на циклическую зависимость интерфейсов модулей

```
public string currentUsedUnitId
```

Field Value

[string](#)

possibleNamespaces

```
internal List<unit_or_namespace> possibleNamespaces
```

Field Value

[List](#) <unit_or_namespace>

syntax_error

```
public SyntaxError syntax_error
```

Field Value

SyntaxError

Properties

ImplementationUsedDirectUnits

Только "реальные" юниты (не dll и namespace)

```
public Dictionary<unit_node, CompilationUnit> ImplementationUsedDirectUnits { get; }
```

Property Value

[Dictionary](#) <unit_node, [CompilationUnit](#)>

ImplementationUsedUnits

```
public unit_node_list ImplementationUsedUnits { get; }
```

Property Value

unit_node_list

ImplementationUsingNamespaceList

```
public using_namespace_list ImplementationUsingNamespaceList { get; set; }
```

Property Value

using_namespace_list

InterfaceUsedDirectUnits

Только "реальные" юниты (не dll и namespace)

```
public Dictionary<unit_node, CompilationUnit> InterfaceUsedDirectUnits { get; }
```

Property Value

[Dictionary](#)<unit_node, [CompilationUnit](#)>

InterfaceUsedUnits

```
public unit_node_list InterfaceUsedUnits { get; }
```

Property Value

unit_node_list

InterfaceUsingNamespaceList

```
public using_namespace_list InterfaceUsingNamespaceList { get; set; }
```

Property Value

using_namespace_list

Language

```
public ILanguage Language { get; set; }
```

Property Value

ILanguage

SemanticTree

```
public unit_node SemanticTree { get; set; }
```

Property Value

unit_node

SyntaxTree

```
public compilation_unit SyntaxTree { get; set; }
```

Property Value

compilation_unit

Methods

ForEachDirectCompilationUnit(Func<CompilationUnit, string, bool>)

```
public bool ForEachDirectCompilationUnit(Func<CompilationUnit, string, bool> on_unit)
```

Parameters

on_unit [Func<CompilationUnit, string, bool>](#)

Returns

[bool](#)

Class CompilationUnitHashTable

Namespace: [PascalABCCompiler](#)

Assembly: Compiler.dll

```
public class CompilationUnitHashTable : Hashtable, IDictionary, ICollection, IEnumerable,
ISerializable, IDeserializationCallback, ICloneable
```

Inheritance

[object](#) ← [Hashtable](#) ← CompilationUnitHashTable

Implements

[IDictionary](#), [ICollection](#), [IEnumerable](#), [ISerializable](#), [IDeserializationCallback](#), [ICloneable](#)

Inherited Members

[Hashtable.HashPrime](#), [Hashtable.InitialSize](#), [Hashtable.LoadFactorName](#),
[Hashtable.VersionName](#), [Hashtable.ComparerName](#), [Hashtable.HashCodeProviderName](#),
[Hashtable.HashSizeName](#), [Hashtable.KeysName](#), [Hashtable.ValuesName](#),
[Hashtable.KeyComparerName](#), [Hashtable.buckets](#), [Hashtable.count](#), [Hashtable.occupancy](#),
[Hashtable.loadsize](#), [Hashtable.loadFactor](#), [Hashtable.version](#), [Hashtable.isWriterInProgress](#),
[Hashtable.keys](#), [Hashtable.values](#), [Hashtable.keycomparer](#), [Hashtable.syncRoot](#),
[Hashtable.InitHash\(object, int, out uint, out uint\)](#), [Hashtable.Add\(object, object\)](#), [Hashtable.Clear\(\)](#),
[Hashtable.Clone\(\)](#), [Hashtable.Contains\(object\)](#), [Hashtable.ContainsKey\(object\)](#),
[Hashtable.ContainsValue\(object\)](#), [Hashtable.CopyKeys\(Array, int\)](#),
[Hashtable.CopyEntries\(Array, int\)](#), [Hashtable.CopyTo\(Array, int\)](#), [Hashtable.ToKeyValuePairsArray\(\)](#),
[Hashtable.CopyValues\(Array, int\)](#), [Hashtable.expand\(\)](#), [Hashtable.rehash\(\)](#),
[Hashtable.UpdateVersion\(\)](#), [Hashtable.rehash\(int, bool\)](#), [Hashtable.IEnumerable.GetEnumerator\(\)](#),
[Hashtable.GetEnumerator\(\)](#), [Hashtable.GetHash\(object\)](#), [Hashtable.KeyEquals\(object, object\)](#),
[Hashtable.Insert\(object, object, bool\)](#), [Hashtable.putEntry\(Hashtable.bucket\[\], object, object, int\)](#),
[Hashtable.Remove\(object\)](#), [Hashtable.Synchronized\(Hashtable\)](#),
[Hashtable.GetObjectData\(SerializationInfo, StreamingContext\)](#), [Hashtable.OnDeserialization\(object\)](#),
[Hashtable.hcp](#), [Hashtable.comparer](#), [Hashtable.EqualityComparer](#), [Hashtable.this\[object\]](#),
[Hashtable.IsReadOnly](#), [Hashtable.IsFixedSize](#), [Hashtable.IsSynchronized](#), [Hashtable.Keys](#),
[Hashtable.Values](#), [Hashtable.SyncRoot](#), [Hashtable.Count](#), [object.ToString\(\)](#),
[object.Equals\(object\)](#), [object.Equals\(object, object\)](#), [object.ReferenceEquals\(object, object\)](#),
[object.GetHashCode\(\)](#), [object.GetType\(\)](#), [object.MemberwiseClone\(\)](#),
[object.FieldSetter\(string, string, object\)](#), [object.FieldGetter\(string, string, ref object\)](#),
[object.GetFieldInfo\(string, string\)](#)

Constructors

CompilationUnitHashTable()

```
public CompilationUnitHashTable()
```

Properties

this[string]

```
public CompilationUnit this[string key] { get; set; }
```

Parameters

key [string](#) ↗

Property Value

[CompilationUnit](#)

Class Compiler

Namespace: [PascalABCCompiler](#)

Assembly: Compiler.dll

```
public class Compiler : MarshalByRefObject, ICompiler
```

Inheritance

[object](#) ← [MarshalByRefObject](#) ← Compiler

Implements

[ICompiler](#)

Inherited Members

[MarshalByRefObject.identity](#) , [MarshalByRefObject.GetComIUnknown\(bool\)](#) ,
[MarshalByRefObject.GetComIUnknown\(MarshalByRefObject\)](#) ,
[MarshalByRefObject.IsInstanceOfType\(Type\)](#) ,
[MarshalByRefObject.InvokeMember\(string, BindingFlags, Binder, object\[\], ParameterModifier\[\], CultureInfo, string\[\]\)](#) ,
[MarshalByRefObject.MemberwiseClone\(bool\)](#) ,
[MarshalByRefObject.GetIdentity\(MarshalByRefObject, out bool\)](#) ,
[MarshalByRefObject.GetIdentity\(MarshalByRefObject\)](#) ,
[MarshalByRefObject.RaceSetServerIdentity\(ServerIdentity\)](#) ,
[MarshalByRefObject.ResetServerIdentity\(\)](#) , [MarshalByRefObject.GetLifetimeService\(\)](#) ,
[MarshalByRefObject.InitializeLifetimeService\(\)](#) , [MarshalByRefObject.CreateObjRef\(Type\)](#) ,
[MarshalByRefObject.CanCastToXmlType\(string, string\)](#) ,
[MarshalByRefObject.CanCastToXmlTypeHelper\(RuntimeType, MarshalByRefObject\)](#) ,
[MarshalByRefObject.Identity](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.FieldSetter\(string, string, object\)](#) ,
[object.FieldGetter\(string, string, ref object\)](#) , [object.GetFieldInfo\(string, string\)](#)

Constructors

Compiler()

```
public Compiler()
```

Compiler(ICompiler, SourceFilesProviderDelegate, ChangeCompilerStateEventDelegate)

```
public Compiler(ICompiler comp, SourceFilesProviderDelegate SourceFilesProvider,  
ChangeCompilerStateEventDelegate ChangeCompilerState)
```

Parameters

comp [ICompiler](#)

SourceFilesProvider SourceFilesProviderDelegate

ChangeCompilerState [ChangeCompilerStateEventDelegate](#)

Compiler(SourceFilesProviderDelegate, ChangeCompilerStateEventDelegate)

```
public Compiler(SourceFilesProviderDelegate SourceFilesProvider,  
ChangeCompilerStateEventDelegate ChangeCompilerState)
```

Parameters

SourceFilesProvider SourceFilesProviderDelegate

ChangeCompilerState [ChangeCompilerStateEventDelegate](#)

Fields

BadNodesInSyntaxTree

```
private Hashtable BadNodesInSyntaxTree
```

Field Value

[Hashtable](#)

CodeGeneratorsController

```
public Controller CodeGeneratorsController
```

Field Value

Controller

CompiledVariables

```
public List<var_definition_node> CompiledVariables
```

Field Value

[List](#) <var_definition_node>

DLLCache

```
private Dictionary<string, CompilationUnit> DLLCache
```

Field Value

[Dictionary](#) <[string](#), [CompilationUnit](#)>

PCUReadersAndWritersClosed

```
private bool PCUReadersAndWritersClosed
```

Field Value

[bool](#)

RecompileList

```
public Hashtable RecompileList
```

Field Value

[Hashtable](#)

StandardModules

```
private List<string> StandardModules
```

Field Value

[List](#) <[string](#)>

SyntaxTreeToSemanticTreeConverter

```
public SyntaxTreeToSemanticTreeConverter SyntaxTreeToSemanticTreeConverter
```

Field Value

[SyntaxTreeToSemanticTreeConverter](#)

UnitsToCompileDelayedList

список отложенной компиляции реализации (она будет откомпилирована в Compile, а не в CompileUnit)

```
private List<CompilationUnit> UnitsToCompileDelayedList
```

Field Value

[List](#) <[CompilationUnit](#)>

UnitsTopologicallySortedList

```
public List<CompilationUnit> UnitsTopologicallySortedList
```

Field Value

[List](#) <[CompilationUnit](#)>

_clear_after_compilation

```
private bool _clear_after_compilation
```

Field Value

[bool](#)

assemblyResolveScope

```
private AssemblyResolveScope assemblyResolveScope
```

Field Value

AssemblyResolveScope

beginOffset

Начало основной программы

```
public int beginOffset
```

Field Value

[int](#)

currentCompilationUnit

```
private CompilationUnit currentCompilationUnit
```

Field Value

[CompilationUnit](#)

errorsList

```
private List<Error> errorsList
```

Field Value

[List](#) <Error>

firstCompilationUnit

```
private CompilationUnit firstCompilationUnit
```

Field Value

[CompilationUnit](#)

internalDebug

```
private CompilerInternalDebug internalDebug
```

Field Value

[CompilerInternalDebug](#)

linesCompiled

```
private uint linesCompiled
```

Field Value

[uint](#)

pABCCodeHealth

```
private int pABCCodeHealth
```

Field Value

[int](#)

pcuCompilationUnits

```
private static Dictionary<string, CompilationUnit> pcuCompilationUnits
```

Field Value

[Dictionary](#)<[string](#), [CompilationUnit](#)>

project

```
private ProjectInfo project
```

Field Value

[ProjectInfo](#)

semanticTree

```
private program_node semanticTree
```

Field Value

program_node

semanticTreeConvertersController

```
private SemanticTreeConvertersController semanticTreeConvertersController
```

Field Value

[SemanticTreeConvertersController](#)

sourceFilesProvider

```
private SourceFilesProviderDelegate sourceFilesProvider
```

Field Value

SourceFilesProviderDelegate

standartAssemblyPath

```
private static string standartAssemblyPath
```

Field Value

[string](#)

standart_assembly_dict

```
public static Dictionary<string, string> standart_assembly_dict
```

Field Value

[Dictionary](#)<string, string>

state

```
private CompilerState state
```

Field Value

[CompilerState](#)

supportedProjectFiles

```
private SupportedSourceFile[] supportedProjectFiles
```

Field Value

[SupportedSourceFile](#)[]

supportedSourceFiles

```
private SupportedSourceFile[] supportedSourceFiles
```

Field Value

[SupportedSourceFile](#)[]

unitTable

```
private CompilationUnitHashTable unitTable
```

Field Value

[CompilationUnitHashTable](#)

varBeginOffset

Положение первых переменных в пространстве имен основной программы

```
public int varBeginOffset
```

Field Value

[int](#)

warnings

```
private List<CompilerWarning> warnings
```

Field Value

[List](#) <CompilerWarning>

Properties

Banner

```
public static string Banner { get; }
```

Property Value

[string](#)

BeginOffset

```
public int BeginOffset { get; }
```

Property Value

[int](#)

ClearAfterCompilation

```
public bool ClearAfterCompilation { get; set; }
```

Property Value

[bool](#)

CompilerOptions

```
public CompilerOptions CompilerOptions { get; set; }
```

Property Value

[CompilerOptions](#)

CompilerType

```
public CompilerType CompilerType { get; }
```

Property Value

[CompilerType](#)

ErrorsList

```
public List<Error> ErrorsList { get; }
```

Property Value

[List](#)<Error>

GetUnitFileNameCache

```
public Dictionary<Tuple<string, string>, string> GetUnitFileNameCache { get; }
```

Property Value

[Dictionary](#)<[Tuple](#)<string, string>, string>

InternalDebug

```
public CompilerInternalDebug InternalDebug { get; set; }
```

Property Value

[CompilerInternalDebug](#)

LanguageProvider

```
private LanguageProvider LanguageProvider { get; }
```

Property Value

LanguageProvider

LinesCompiled

```
public uint LinesCompiled { get; }
```

Property Value

[uint](#)

PABCCodeHealth

Здоровье кода на всякий случай выносим в интерфейс компилятора Реально оно будет использоваться только при запуске из под оболочки (Remote Compiler)

```
public int PABCCodeHealth { get; }
```

Property Value

[int](#)

PCUFileNamesDictionary

```
public Dictionary<Tuple<string, string>, Tuple<string, int>> PCUFileNamesDictionary { get; }
```

Property Value

[Dictionary](#)<[Tuple](#)<[string](#), [string](#)>, [Tuple](#)<[string](#), [int](#)>>

SemanticTree

```
public IProgramNode SemanticTree { get; }
```

Property Value

IProgramNode

SemanticTreeConvertersController

```
public SemanticTreeConvertersController SemanticTreeConvertersController { get; }
```

Property Value

[SemanticTreeConvertersController](#)

ShortVersion

```
public static string ShortVersion { get; }
```

Property Value

[string](#)

SourceFileNamesDictionary

```
public Dictionary<Tuple<string, string>, Tuple<string, int>> SourceFileNamesDictionary {  
    get; }
```

Property Value

[Dictionary](#)<[Tuple](#)<[string](#), [string](#)>, [Tuple](#)<[string](#), [int](#)>>

SourceFilesProvider

```
public SourceFilesProviderDelegate SourceFilesProvider { get; }
```

Property Value

SourceFilesProviderDelegate

State

```
public CompilerState State { get; }
```

Property Value

[CompilerState](#)

SupportedProjectFiles

```
public SupportedSourceFile[] SupportedProjectFiles { get; }
```

Property Value

[SupportedSourceFile\[\]](#)

SupportedSourceFiles

```
public SupportedSourceFile[] SupportedSourceFiles { get; set; }
```

Property Value

[SupportedSourceFile\[\]](#)

UnitTable

```
public CompilationUnitHashTable UnitTable { get; }
```

Property Value

[CompilationUnitHashTable](#)

VarBeginOffset

```
public int VarBeginOffset { get; }
```

Property Value

[int](#)

Version

```
public static string Version { get; }
```

Property Value

[string](#)

VersionDateTime

```
public static DateTime VersionDateTime { get; }
```

Property Value

[DateTime](#)

Warnings

```
public List<CompilerWarning> Warnings { get; }
```

Property Value

[List](#)<CompilerWarning>

Methods

AddCodeGenerationErrorToErrorList(Exception)

```
private void AddCodeGenerationErrorToErrorList(Exception err)
```

Parameters

err [Exception](#)

AddCurrentUnitAndItsReferencesToUsesLists(unit_node_list,
Dictionary<unit_node, CompilationUnit>, unit_or_namespace,
CompilationUnit, unit_node_list)

```
private void AddCurrentUnitAndItsReferencesToUsesLists(unit_node_list unitsFromUsesSection,  
Dictionary<unit_node, CompilationUnit> directUnitsFromUsesSection, unit_or_namespace  
currentUnitNode, CompilationUnit currentUnit, unit_node_list references)
```

Parameters

unitsFromUsesSection unit_node_list

directUnitsFromUsesSection [Dictionary](#)<unit_node, [CompilationUnit](#)>

currentUnitNode unit_or_namespace

currentUnit [CompilationUnit](#)

references unit_node_list

AddDeclarationsAndReferencedUnitsToNamespaces(List<unit_o
r_namespace>, string, unit_module, syntax_namespace_node)

```
private void AddDeclarationsAndReferencedUnitsToNamespaces(List<unit_or_namespace>  
namespace_modules, string file, unit_module unitModule, syntax_namespace_node namespaceNode)
```

Parameters

namespace_modules [List](#)<unit_or_namespace>

file [string](#)

`unitModule` `unit_module`

`namespaceNode` `syntax_namespace_node`

AddDocumentationToNodes(CompilationUnit, string)

```
private Dictionary<syntax_tree_node, string> AddDocumentationToNodes(CompilationUnit  
currentUnit, string text)
```

Parameters

`currentUnit` [CompilationUnit](#)

`text` [string](#)

Returns

[Dictionary](#)<syntax_tree_node, [string](#)>

AddErrorToErrorListConsideringPosition(Error)

```
private void AddErrorToErrorListConsideringPosition(Error err)
```

Parameters

`err` Error

AddInternalErrorToErrorList(CompilerInternalError)

```
private void AddInternalErrorToErrorList(CompilerInternalError internalError)
```

Parameters

`internalError` CompilerInternalError

AddNamespacesToMainDefinitions(unit_module, program_module, Dictionary<string, syntax_namespace_node>)

```
private void AddNamespacesToMainDefinitions(unit_module mainLibrary, program_module
main_program, Dictionary<string, syntax_namespace_node> namespaces)
```

Parameters

mainLibrary unit_module

main_program program_module

namespaces [Dictionary](#)<[string](#), syntax_namespace_node>

AddNamespacesToMainUsesList(unit_module, program_module, List<unit_or_namespace>)

```
private void AddNamespacesToMainUsesList(unit_module mainLibrary, program_module
main_program, List<unit_or_namespace> namespaceModules)
```

Parameters

mainLibrary unit_module

main_program program_module

namespaceModules [List](#)<unit_or_namespace>

AddNamespacesToUsingList(using_namespace_list, using_list)

```
public void AddNamespacesToUsingList(using_namespace_list using_list, using_list ul)
```

Parameters

using_list using_namespace_list

ul using_list

AddNamespacesToUsingList(using_namespace_list,
List<unit_or_namespace>, bool, Dictionary<string,
syntax_namespace_node>)

```
public void AddNamespacesToUsingList(using_namespace_list usingList,  
List<unit_or_namespace> possibleNamespaces, bool mightContainUnits, Dictionary<string,  
syntax_namespace_node> namespaces)
```

Parameters

usingList using_namespace_list

possibleNamespaces [List](#)<unit_or_namespace>

mightContainUnits [bool](#)

namespaces [Dictionary](#)<string, syntax_namespace_node>

AddReferencesToNetSystemLibraries(CompilationUnit,
List<compiler_directive>)

Добавляет ссылки на стандартные системные dll .NET - версия с директивами уровня синтаксиса

```
private void AddReferencesToNetSystemLibraries(CompilationUnit compilationUnit,  
List<compiler_directive> directives)
```

Parameters

compilationUnit [CompilationUnit](#)

directives [List](#)<compiler_directive>

AddReferencesToNetSystemLibraries(CompilationUnit,
List<compiler_directive>)

Добавляет ссылки на стандартные системные dll .NET - версия с директивами уровня семантики

```
private void AddReferencesToNetSystemLibraries(CompilationUnit compilationUnit,  
List<compiler_directive> directives)
```

Parameters

compilationUnit [CompilationUnit](#)

directives [List](#)<compiler_directive>

AddStandardUnitsToInterfaceUsesSection(CompilationUnit)

```
public void AddStandardUnitsToInterfaceUsesSection(CompilationUnit currentUnit)
```

Parameters

currentUnit [CompilationUnit](#)

AddWarnings(List<CompilerWarning>)

```
public void AddWarnings(List<CompilerWarning> WarningList)
```

Parameters

WarningList [List](#)<CompilerWarning>

AsyncClosePCUWriters()

```
private void AsyncClosePCUWriters()
```

CalculateLinesCompiled(List<Error>, compilation_unit)

```
private void CalculateLinesCompiled(List<Error> errorList, compilation_unit unitSyntaxTree)
```

Parameters

`errorList` [List](#) <Error>

`unitSyntaxTree` compilation_unit

CalculatePascalProgramHealth(compilation_unit)

```
private void CalculatePascalProgramHealth(compilation_unit unitSyntaxTree)
```

Parameters

`unitSyntaxTree` compilation_unit

ChangeCompilerStateEvent(ICompiler, CompilerState, string)

```
private void ChangeCompilerStateEvent(ICompiler sender, CompilerState State,
string FileName)
```

Parameters

`sender` [ICompiler](#)

`State` [CompilerState](#)

`FileName` [string](#)

CheckErrorsAndThrowTheFirstOne()

```
private void CheckErrorsAndThrowTheFirstOne()
```

CheckForDuplicatesInUsesSection(List<unit_or_namespace>)

Бросает ошибку если находит дубликаты в секции uses

```
private void CheckForDuplicatesInUsesSection(List<unit_or_namespace> usesList)
```

Parameters

usesList [List](#)<unit_or_namespace>

CheckForRTLErrorsAndClearAllErrorsIfFound()

```
private bool CheckForRTLErrorsAndClearAllErrorsIfFound()
```

Returns

[bool](#)

CheckPathValid(string)

```
public static bool CheckPathValid(string path)
```

Parameters

path [string](#)

Returns

[bool](#)

ClearAll(bool)

```
public void ClearAll(bool close_pcu = true)
```

Parameters

close_pcu [bool](#)

ClosePCUReadersAndWriters()

```
private void ClosePCUReadersAndWriters()
```

ClosePCUWriters()

```
private void ClosePCUWriters()
```

CombinePathsRelatively(string, string)

```
public static string CombinePathsRelatively(string path1, string path2)
```

Parameters

path1 [string](#)

path2 [string](#)

Returns

[string](#)

Compile()

```
public string Compile()
```

Returns

[string](#)

Compile(CompilerOptions)

```
public string Compile(CompilerOptions CompilerOptions)
```

Parameters

[CompilerOptions](#) [CompilerOptions](#)

Returns

[string](#)

CompileCS()

```
public string CompileCS()
```

Returns

[string](#)

CompileCurrentUnitImplementation(string, CompilationUnit, Dictionary<syntax_tree_node, string>)

```
private void CompileCurrentUnitImplementation(string UnitFileName, CompilationUnit  
currentUnit, Dictionary<syntax_tree_node, string> docs)
```

Parameters

[UnitFileName](#) [string](#)

[currentUnit](#) [CompilationUnit](#)

[docs](#) [Dictionary](#)<syntax_tree_node, [string](#)>

CompileCurrentUnitInterface(string, CompilationUnit, Dictionary<syntax_tree_node, string>)

```
private void CompileCurrentUnitInterface(string UnitFileName, CompilationUnit currentUnit,  
Dictionary<syntax_tree_node, string> docs)
```

Parameters

UnitFileName [string](#)

currentUnit [CompilationUnit](#)

docs [Dictionary](#) <syntax_tree_node, [string](#)>

CompileImplementationDependencies(string, CompilationUnit,
List<unit_or_namespace>, Dictionary<string,
syntax_namespace_node>, common_unit_node, out bool)

Компилирует модули из секции uses текущего модуля рекурсивно

```
private void CompileImplementationDependencies(string currentPath, CompilationUnit  
currentUnit, List<unit_or_namespace> implementationUsesList, Dictionary<string,  
syntax_namespace_node> namespaces, common_unit_node commonUnitNode, out  
bool shouldReturnCurrentUnit)
```

Parameters

currentPath [string](#)

currentUnit [CompilationUnit](#)

implementationUsesList [List](#) <unit_or_namespace>

namespaces [Dictionary](#) <[string](#), syntax_namespace_node>

commonUnitNode common_unit_node

shouldReturnCurrentUnit [bool](#)

CompileInterfaceDependencies(unit_node_list,
Dictionary<unit_node, CompilationUnit>, unit_or_namespace,
string, string, CompilationUnit, List<unit_or_namespace>,
unit_node_list, Dictionary<string, syntax_namespace_node>, out
bool)

Компилирует модули из секции uses интерфейса текущего модуля рекурсивно

```
private void CompileInterfaceDependencies(unit_node_list unitsFromUsesSection,
Dictionary<unit_node, CompilationUnit> directUnitsFromUsesSection, unit_or_namespace
currentUnitNode, string unitFileName, string currentPath, CompilationUnit currentUnit,
List<unit_or_namespace> interfaceUsesList, unit_node_list references, Dictionary<string,
syntax_namespace_node> namespaces, out bool shouldReturnCurrentUnit)
```

Parameters

unitsFromUsesSection unit_node_list

directUnitsFromUsesSection [Dictionary](#)<unit_node, [CompilationUnit](#)>

currentUnitNode unit_or_namespace

unitFileName [string](#)

currentPath [string](#)

currentUnit [CompilationUnit](#)

interfaceUsesList [List](#)<unit_or_namespace>

references unit_node_list

namespaces [Dictionary](#)<[string](#), syntax_namespace_node>

shouldReturnCurrentUnit [bool](#)

Exceptions

[CycleUnitReference](#)

CompileReference(unit_node_list, compiler_directive)

```
private CompilationUnit CompileReference(unit_node_list dlls, compiler_directive reference)
```

Parameters

dlls unit_node_list

reference compiler_directive

Returns

[CompilationUnit](#)

CompileUnit(unit_node_list, Dictionary<unit_node, CompilationUnit>, unit_or_namespace, string)

Компилирует основную программу и все используемые ей юниты рекурсивно

```
public CompilationUnit CompileUnit(unit_node_list unitsFromUsesSection,
Dictionary<unit_node, CompilationUnit> directUnitsFromUsesSection, unit_or_namespace
currentUnitNode, string previousPath)
```

Parameters

unitsFromUsesSection unit_node_list

Вспомогательная переменная для заполнения CompilationUnit.interfaceUsedUnits и
CompilationUnit.implementationUsedUnits (здесь могут содержаться юниты и dll)

directUnitsFromUsesSection [Dictionary](#)<unit_node, [CompilationUnit](#)>

Вспомогательная переменная для заполнения CompilationUnit.interfaceUsedDirectUnits и
CompilationUnit.implementationUsedDirectUnits

currentUnitNode unit_or_namespace

Синтаксический узел текущего модуля (или пространства имен)

previousPath string

Директория родительского модуля

Returns

[CompilationUnit](#)

Скомпилированный юнит

CompileUnitsFromDelayedList()

```
private void CompileUnitsFromDelayedList()
```

ConstructMainSemanticTree(CompilerOptions)

```
private program_node ConstructMainSemanticTree(CompilerOptions compilerOptions)
```

Parameters

compilerOptions CompilerOptions

Returns

program_node

ConstructSyntaxTree(string, CompilationUnit, string)

```
private compilation_unit ConstructSyntaxTree(string unitFileName, CompilationUnit
currentUnit, string sourceText)
```

Parameters

unitFileName [string](#)

currentUnit [CompilationUnit](#)

sourceText [string](#)

Returns

compilation_unit

ConstructSyntaxTreeAndRunSugarConversions(string, CompilationUnit, out Dictionary<syntax_tree_node, string>)

Строит синтаксическое дерево, бросает первую из найденных ошибок (если они есть) и запускает сахарные преобразования

```
private void ConstructSyntaxTreeAndRunSugarConversions(string unitFileName, CompilationUnit currentUnit, out Dictionary<syntax_tree_node, string> docs)
```

Parameters

unitFileName [string](#)

currentUnit [CompilationUnit](#)

docs [Dictionary](#)<syntax_tree_node, [string](#)>

ConvertSyntaxTree(compilation_unit, List<ISyntaxTreeConverter>)

```
private compilation_unit ConvertSyntaxTree(compilation_unit syntaxTree, List<ISyntaxTreeConverter> converters)
```

Parameters

syntaxTree compilation_unit

converters [List](#)<ISyntaxTreeConverter>

Returns

compilation_unit

CreateDependencyListsForCurrentUnit(CompilationUnit, string, out List<unit_or_namespace>, out unit_node_list, out Dictionary<string, syntax_namespace_node>)

```
private void CreateDependencyListsForCurrentUnit(CompilationUnit currentUnit, string currentDirectory, out List<unit_or_namespace> interfaceUsesList, out unit_node_list references, out Dictionary<string, syntax_namespace_node> namespaces)
```

Parameters

`currentUnit` [CompilationUnit](#)

`currentDirectory` [string](#)

`interfaceUsesList` [List](#)<unit_or_namespace>

`references` unit_node_list

`namespaces` [Dictionary](#)<[string](#), syntax_namespace_node>

CurrentUnitIsNotMainProgram()

Возвращает true, если текущий компилируемый модуль не является основной программой (program_module)

`private bool CurrentUnitIsNotMainProgram()`

Returns

[bool](#)

DebugOutputFileCreationUsingPDB()

`private void DebugOutputFileCreationUsingPDB()`

DisablePABCRTIfUsingDotnet5(List<compiler_directive>)

`private void DisablePABCRTIfUsingDotnet5(List<compiler_directive> directives)`

Parameters

`directives` [List](#)<compiler_directive>

FillNetCompilerOptionsFromCompilerDirectives(CompilerOptions, Dictionary<string, List<compiler_directive>>)

```
private void FillNetCompilerOptionsFromCompilerDirectives(CompilerOptions  
    netCompilerOptions, Dictionary<string, List<compiler_directive>> compilerDirectives)
```

Parameters

netCompilerOptions CompilerOptions

compilerDirectives [Dictionary](#)<[string](#), [List](#)<compiler_directive>>

FillNetCompilerOptionsFromProject(CompilerOptions)

```
private void FillNetCompilerOptionsFromProject(CompilerOptions netCompilerOptions)
```

Parameters

netCompilerOptions CompilerOptions

FindFileWithExtensionInDirs(string, out int, params string[])

```
private string FindFileWithExtensionInDirs(string fileName, out int foundDirIndex, params  
    string[] dirs)
```

Parameters

fileName [string](#)

foundDirIndex [int](#)

dirs [string](#)[]

Returns

[string](#)

FindPCUFileName(string, string, out int)

```
public string FindPCUFileName(string fileName, string currentPath, out int folderPriority)
```

Parameters

fileName [string](#)

currentPath [string](#)

folderPriority [int](#)

Returns

[string](#)

FindPositionForSemanticErrorInTheErrorList(Error)

```
private int FindPositionForSemanticErrorInTheErrorList(Error err)
```

Parameters

err Error

Returns

[int](#)

FindSourceFileName(string, string, out int)

```
public string FindSourceFileName(string fileName, string currentPath, out int folderPriority)
```

Parameters

fileName [string](#)

`currentPath` [string](#)

`folderPriority` [int](#)

Returns

[string](#)

FindSourceFileNameInDirs(string, out int, params string[])

```
public string FindSourceFileNameInDirs(string fileName, out int foundDirIndex, params
string[] Dirs)
```

Parameters

`fileName` [string](#)

`foundDirIndex` [int](#)

`Dirs` [string](#)[]

Returns

[string](#)

Free()

```
public void Free()
```

GenUnitDocumentation(CompilationUnit, string)

```
private Dictionary<syntax_tree_node, string> GenUnitDocumentation(CompilationUnit
currentUnit, string SourceText)
```

Parameters

`currentUnit` [CompilationUnit](#)

`SourceText` [string](#)

Returns

[Dictionary](#)<[syntax_tree_node](#), [string](#)>

GenerateILCode(program_node, CompilerOptions, List<string>)

```
private void GenerateILCode(program_node programNode, CompilerOptions compilerOptions,
List<string> resourceFiles)
```

Parameters

`programNode` program_node

`compilerOptions` CompilerOptions

`resourceFiles` [List](#)<[string](#)>

GetCompilerDirectives(List<CompilationUnit>)

Формирует словарь директив компилятора, собирая их из всех переданных модулей

```
private Dictionary<string, List<compiler_directive>>
GetCompilerDirectives(List<CompilationUnit> Units)
```

Parameters

`Units` [List](#)<[CompilationUnit](#)>

Returns

[Dictionary](#)<[string](#), [List](#)<compiler_directive>>

Exceptions

[DuplicateDirective](#)

GetDirectivesAsSemanticNodes(List<compiler_directive>, string)

преобразует в директивы семантического уровня | в syntax_tree_visitor такая же функция EVA

```
private List<compiler_directive> GetDirectivesAsSemanticNodes(List<compiler_directive>
compilerDirectives, string unitFileName)
```

Parameters

compilerDirectives [List](#)<compiler_directive>

unitFileName [string](#)

Returns

[List](#)<compiler_directive>

GetImplementationSyntaxUsingList(compilation_unit)

получение списка using - legacy code !!!

```
private using_list GetImplementationSyntaxUsingList(compilation_unit cu)
```

Parameters

cu compilation_unit

Returns

using_list

GetImplementationUsesSection(compilation_unit)

```
private List<unit_or_namespace> GetImplementationUsesSection(compilation_unit
unitSyntaxTree)
```

Parameters

`unitSyntaxTree compilation_unit`

Returns

`List<unit_or_namespace>`

GetIncludedFilesFromDirectives(CompilationUnit, List<compiler_directive>)

```
private static List<string> GetIncludedFilesFromDirectives(CompilationUnit compilationUnit,
List<compiler_directive> directives)
```

Parameters

`compilationUnit CompilationUnit`

`directives List<compiler_directive>`

Returns

`List<string>`

GetInterfaceUsesSection(compilation_unit)

Возвращает список зависимостей из интерфейсной части модуля (или основной программы)

```
public List<unit_or_namespace> GetInterfaceUsesSection(compilation_unit unitSyntaxTree)
```

Parameters

`unitSyntaxTree compilation_unit`

Returns

`List<unit_or_namespace>`

GetInterfaceUsingList(compilation_unit)

получение списка using - legacy code !!!

```
public using_list GetInterfaceUsingList(compilation_unit cu)
```

Parameters

cu compilation_unit

Returns

using_list

GetLocationFromTreenode(syntax_tree_node, string)

```
private location GetLocationFromTreenode(syntax_tree_node tn, string FileName)
```

Parameters

tn syntax_tree_node

FileName string ↗

Returns

location

GetNamespace(unit_or_namespace)

```
public using_namespace GetNamespace(unit_or_namespace _name_space)
```

Parameters

_name_space unit_or_namespace

Returns

using_namespace

GetNamespace(using_namespace_list, string,
unit_or_namespace, bool, Dictionary<string,
syntax_namespace_node>)

Формирует узел семантического дерева, соответствующий пространству имен (.NET или пользовательскому)

```
private using_namespace GetNamespace(using_namespace_list usingList, string  
fullNamespaceName, unit_or_namespace name_space, bool mightBeUnit, Dictionary<string,  
syntax_namespace_node> namespaces)
```

Parameters

usingList using_namespace_list

fullNamespaceName string ↗

name_space unit_or_namespace

mightBeUnit bool ↗

namespaces Dictionary ↗ <string ↗, syntax_namespace_node>

Returns

using_namespace

Exceptions

[UnitNotFound](#)

[NamespaceNotFound](#)

GetNamespaceSyntaxTree(string)

```
private compilation_unit GetNamespaceSyntaxTree(string fileName)
```

Parameters

fileName [string](#)

Returns

compilation_unit

GetReferenceFileName(string, SourceContext, string, bool)

```
private string GetReferenceFileName(string FileName, SourceContext sc, string curr_path,  
bool overwrite)
```

Parameters

FileName [string](#)

sc SourceContext

curr_path [string](#)

overwrite [bool](#)

Returns

[string](#)

GetReferenceFileName(string, string)

```
public static string GetReferenceFileName(string FileName, string curr_path = null)
```

Parameters

FileName [string](#)

curr_path [string](#)

Returns

[string](#)

GetReferences(CompilationUnit)

```
public unit_node_list GetReferences(CompilationUnit compilationUnit)
```

Parameters

compilationUnit [CompilationUnit](#)

Returns

unit_node_list

GetResourceFilesFromCompilerDirectives(Dictionary<string, List<compiler_directive>>)

```
private List<string> GetResourceFilesFromCompilerDirectives(Dictionary<string, List<compiler_directive>> compilerDirectives)
```

Parameters

compilerDirectives [Dictionary](#)<[string](#), [List](#)<compiler_directive>>

Returns

[List](#)<[string](#)>

GetSourceCode(string, CompilationUnit)

```
private string GetSourceCode(string UnitFileName, CompilationUnit currentUnit)
```

Parameters

UnitFileName [string](#)

`currentUnit` [CompilationUnit](#)

Returns

[string](#) ↗

GetSourceContext(compiler_directive)

```
private SourceContext GetSourceContext(compiler_directive directive)
```

Parameters

`directive` compiler_directive

Returns

SourceContext

GetSourceFileText(string)

```
public string GetSourceFileText(string FileName)
```

Parameters

`FileName` [string](#) ↗

Returns

[string](#) ↗

GetUnitFileName(unit_or_namespace, string)

```
public string GetUnitFileName(unit_or_namespace unitNode, string currentPath)
```

Parameters

`unitNode unit_or_namespace`

`currentPath string ↗`

Returns

`string ↗`

GetUnitFileName(string, string, string, SourceContext)

```
public string GetUnitFileName(string unitName, string usesPath, string currentPath,  
SourceContext sourceContext)
```

Parameters

`unitName string ↗`

`usesPath string ↗`

`currentPath string ↗`

`sourceContext SourceContext`

Returns

`string ↗`

GetUnitPath(CompilationUnit, CompilationUnit)

```
public static string GetUnitPath(CompilationUnit u1, CompilationUnit u2)
```

Parameters

`u1 CompilationUnit`

`u2 CompilationUnit`

Returns

[string](#)

HasIncludeNamespaceDirective(CompilationUnit)

```
private bool HasIncludeNamespaceDirective(CompilationUnit unit)
```

Parameters

[unit](#) [CompilationUnit](#)

Returns

[bool](#)

HasOnlySyntaxErrors(List<Error>)

```
private bool HasOnlySyntaxErrors(List<Error> errors)
```

Parameters

[errors](#) [List](#)<Error>

Returns

[bool](#)

InitializeCompilerOptionsRelatedToStandardUnits(compilation_unit)

Устанавливает значения опций DisableStandardUnits и UseDIIForSystemUnits

```
private void InitializeCompilerOptionsRelatedToStandardUnits(CompilationUnit compilation_unit, SyntaxTree unitSyntaxTree)
```

Parameters

```
unitSyntaxTree compilation_unit
```

InitializeNewUnit(string, string, ref CompilationUnit, out Dictionary<syntax_tree_node, string>)

Получение исходного кода модуля, заполнение документации, генерация синтаксического дерева, обработка синтаксических ошибок

```
private void InitializeNewUnit(string unitFileName, string UnitId, ref CompilationUnit currentUnit, out Dictionary<syntax_tree_node, string> docs)
```

Parameters

unitFileName [string](#)

UnitId [string](#)

currentUnit [CompilationUnit](#)

docs [Dictionary](#)<syntax_tree_node, [string](#)>

InitializeProjectInfoAndFillCompilerOptionsFromIt()

```
private void InitializeProjectInfoAndFillCompilerOptionsFromIt()
```

InternalParseText(ILanguage, string, string, List<Error>, List<CompilerWarning>, List<string>, bool)

```
private compilation_unit InternalParseText(ILanguage language, string fileName, string text, List<Error> errorList, List<CompilerWarning> warnings, List<string> definesList = null, bool calculateHealth = true)
```

Parameters

language [ILanguage](#)

`fileName` [string](#)

`text` [string](#)

`errorList` [List](#)<[Error](#)>

`warnings` [List](#)<[CompilerWarning](#)>

`definesList` [List](#)<[string](#)>

`calculateHealth` [bool](#)

Returns

`compilation_unit`

`IsDll(compilation_unit)`

Проверяет, является ли модуль dll по соответствующей директиве

```
public static bool IsDll(compilation_unit unitSyntaxTree)
```

Parameters

`unitSyntaxTree` `compilation_unit`

Returns

[bool](#)

`IsDll(compilation_unit, out compiler_directive)`

Проверяет, является ли модуль dll по соответствующей директиве и возвращает эту директиву выходным параметром

```
public static bool IsDll(compilation_unit unitSyntaxTree, out  
compiler_directive dllDirective)
```

Parameters

`unitSyntaxTree compilation_unit`

`dllDirective compiler_directive`

Returns

[bool](#)

IsDocumentationNeeded(compilation_unit)

`private bool IsDocumentationNeeded(compilation_unit unitSyntaxTree)`

Parameters

`unitSyntaxTree compilation_unit`

Returns

[bool](#)

IsPossibleNetNamespaceOrStandardPasFile(unit_or_namespace, bool, string)

`private bool IsPossibleNetNamespaceOrStandardPasFile(unit_or_namespace name_space, bool addToStandardModules, string currentPath)`

Parameters

`name_space unit_or_namespace`

`addToStandardModules bool`

`currentPath string`

Returns

[bool](#)

MatchSyntaxErrorsToBadNodes(CompilationUnit)

```
private void MatchSyntaxErrorsToBadNodes(CompilationUnit currentUnit)
```

Parameters

currentUnit [CompilationUnit](#)

NeedRecompiled(string, string[], PCUReader)

```
public bool NeedRecompiled(string pcu_name, string[] included, PCUReader pr)
```

Parameters

pcu_name [string](#)

included [string](#)[]

pr [PCUReader](#)

Returns

[bool](#)

ParseText(string, string, List<Error>, List<CompilerWarning>)

```
public compilation_unit ParseText(string fileName, string text, List<Error> errorList,  
List<CompilerWarning> warnings)
```

Parameters

fileName [string](#)

text [string](#)

errorList [List](#)<Error>

warnings [List](#)<CompilerWarning>

Returns

compilation_unit

PrebuildMainSemanticTreeActions(out CompilerOptions, out List<string>)

Сохраняет документацию для модулей; Выясняет тип выходного файла, целевой фреймворк, платформу; Заполняет опции .NET компиляции согласно директивам и/или информации из проекта; Находит ресурсные файлы из директив

```
private void PrebuildMainSemanticTreeActions(out CompilerOptions netCompilerOptions, out List<string> resourceFiles)
```

Parameters

netCompilerOptions CompilerOptions

resourceFiles [List](#)<[string](#)>

PreloadReference(compiler_directive)

```
private Assembly PreloadReference(compiler_directive reference)
```

Parameters

reference compiler_directive

Returns

[Assembly](#)

PrepareFinalMainFunctionForExe(program_node)

```
public void PrepareFinalMainFunctionForExe(program_node mainSemanticTree)
```

Parameters

`mainSemanticTree` `program_node`

PrepareUserNamespacesUsedInTheCurrentUnit(Compilation Unit)

```
private Dictionary<string, syntax_namespace_node>
PrepareUserNamespacesUsedInTheCurrentUnit(CompilationUnit compilationUnit)
```

Parameters

`compilationUnit` [CompilationUnit](#)

Returns

[Dictionary](#)<[string](#), syntax_namespace_node>

ReadDLL(string, SourceContext)

```
public CompilationUnit ReadDLL(string FileName, SourceContext sc = null)
```

Parameters

`FileName` [string](#)

`sc` SourceContext

Returns

[CompilationUnit](#)

ReadPCU(string)

```
public CompilationUnit ReadPCU(string FileName)
```

Parameters

[FileName](#) [string](#)

Returns

[CompilationUnit](#)

Reload()

```
public void Reload()
```

Reset()

```
private void Reset()
```

RunSemanticChecks(string, CompilationUnit)

Семантические проверки по директивам и по типу файла

```
private void RunSemanticChecks(string unitFileName, CompilationUnit currentUnit)
```

Parameters

[unitFileName](#) [string](#)

[currentUnit](#) [CompilationUnit](#)

SaveDocumentationsForUnits()

```
private void SaveDocumentationsForUnits()
```

SavePCU(CompilationUnit)

```
public void SavePCU(CompilationUnit Unit)
```

Parameters

Unit [CompilationUnit](#)

SaveUnitCheckInParsers()

Передаем парсерам возможность проверить, компилируется ли в данный момент модуль (нужно, если нет ключевого слова unit или подобного в языке)

```
private void SaveUnitCheckInParsers()
```

SemanticCheckCurrentUnitMustBeUnitModule(string, CompilationUnit, bool)

```
private void SemanticCheckCurrentUnitMustBeUnitModule(string UnitFileName, CompilationUnit currentUnit, bool isDll)
```

Parameters

UnitFileName [string](#)

currentUnit [CompilationUnit](#)

isDll [bool](#)

SemanticCheckDLLDirectiveOnlyForLibraries(compilation_unit, bool, compiler_directive)

Проверка, что директива dll только в Library - требует передачи директивы dll

```
private void SemanticCheckDLLDirectiveOnlyForLibraries(compilation_unit unitSyntaxTree, bool
```

```
isDll, compiler_directive dllDirective)
```

Parameters

`unitSyntaxTree compilation_unit`

`isDll bool`

`dllDirective compiler_directive`

SemanticCheckDisableStandardUnitsDirective InUnit(compilation_unit)

Ошибка указания директивы DisableStandardUnits в подключенному модуле

```
private void SemanticCheckDisableStandardUnitsDirectiveInUnit(compilation_unit
    unitSyntaxTree)
```

Parameters

`unitSyntaxTree compilation_unit`

SemanticCheckIsUserNamespace(compilation_unit)

```
private void SemanticCheckIsUserNamespace(compilation_unit unitSyntaxTree)
```

Parameters

`unitSyntaxTree compilation_unit`

SemanticCheckNamespacesOnlyInProjects(CompilationUnit)

```
private void SemanticCheckNamespacesOnlyInProjects(CompilationUnit currentUnit)
```

Parameters

`currentUnit` [CompilationUnit](#)

SemanticCheckNoIncludeNamespaceDirectives InUnit(CompilationUnit)

```
private void SemanticCheckNoIncludeNamespaceDirectivesInUnit(CompilationUnit currentUnit)
```

Parameters

`currentUnit` [CompilationUnit](#)

SemanticCheckNoLoopDependenciesOfInterfaces(Compilation Unit, string, unit_or_namespace, string)

```
private void SemanticCheckNoLoopDependenciesOfInterfaces(CompilationUnit currentUnit, string  
unitFileName, unit_or_namespace usedUnitNode, string currentPath)
```

Parameters

`currentUnit` [CompilationUnit](#)

`unitFileName` [string](#) ↗

`usedUnitNode` `unit_or_namespace`

`currentPath` [string](#) ↗

SemanticCheckUsesInIsNotNamespace(unit_or_namespace, CompilationUnit)

```
private void SemanticCheckUsesInIsNotNamespace(unit_or_namespace currentUnitNode,  
CompilationUnit currentUnit)
```

Parameters

```
currentUnitNode unit_or_namespace
```

```
currentUnit CompilationUnit
```

SetOutputFileTypeOption(Dictionary<string, List<compiler_directive>>)

```
private void SetOutputFileTypeOption(Dictionary<string, List<compiler_directive>>  
compilerDirectives)
```

Parameters

```
compilerDirectives Dictionary<string, List<compiler_directive>>
```

SetOutputPlatformOption(CompilerOptions, Dictionary<string, List<compiler_directive>>)

```
private void SetOutputPlatformOption(CompilerOptions netCompilerOptions, Dictionary<string,  
List<compiler_directive>> compilerDirectives)
```

Parameters

```
netCompilerOptions CompilerOptions
```

```
compilerDirectives Dictionary<string, List<compiler_directive>>
```

SetSupportedProjectFiles()

```
private void SetSupportedProjectFiles()
```

SetSupportedSourceFiles()

```
private void SetSupportedSourceFiles()
```

SetTargetTypeOption(CompilerOptions)

```
private void SetTargetTypeOption(CompilerOptions netCompilerOptions)
```

Parameters

netCompilerOptions CompilerOptions

SetUseDLLForSystemUnits(string, List<unit_or_namespace>, int)

Если в программе в секции uses есть не про-во имен и не стандартный модуль, то использование PABCRTl.dll отменяется

```
private void SetUseDLLForSystemUnits(string currentDirectory, List<unit_or_namespace>
usesList, int lastUnitIndex)
```

Parameters

currentDirectory [string](#)

usesList [List](#)<unit_or_namespace>

lastUnitIndex [int](#)

SourceFileExists(string)

```
private bool SourceFileExists(string FileName)
```

Parameters

FileName [string](#)

Returns

[bool](#)

SourceFileGetLastWriteTime(string)

```
private DateTime SourceFileGetLastWriteTime(string FileName)
```

Parameters

FileName [string](#)

Returns

[DateTime](#)

StartCompile()

```
public void StartCompile()
```

ToString()

```
public override string ToString()
```

Returns

[string](#)

TryThrowInvalidPath(string, SourceContext)

```
public static void TryThrowInvalidPath(string path, SourceContext loc)
```

Parameters

path [string](#)

loc SourceContext

UnitHasPCU(unit_node_list, Dictionary<unit_node, CompilationUnit>, unit_or_namespace, ref string, ref CompilationUnit)

```
private bool UnitHasPCU(unit_node_list unitsFromUsesSection, Dictionary<unit_node, CompilationUnit> directUnitsFromUsesSection, unit_or_namespace currentUnitNode, ref string UnitFileName, ref CompilationUnit currentUnit)
```

Parameters

unitsFromUsesSection `unit_node_list`

directUnitsFromUsesSection `Dictionary<unit_node, CompilationUnit>`

currentUnitNode `unit_or_namespace`

UnitFileName `string`

currentUnit `CompilationUnit`

Returns

`bool`

WaitCallback_ClosePCUWriters(object)

```
private void WaitCallback_ClosePCUWriters(object state)
```

Parameters

state `object`

buildImplementationUsesList(CompilationUnit)

```
private unit_node_list buildImplementationUsesList(CompilationUnit cu)
```

Parameters

cu [CompilationUnit](#)

Returns

unit_node_list

get_assembly_path(string, bool)

```
public static string get_assembly_path(string name, bool search_for_intellisense)
```

Parameters

name [string](#)

search_for_intellisense [bool](#)

Returns

[string](#)

get_standart_assembly_path(string)

```
public static string get_standart_assembly_path(string name)
```

Parameters

name [string](#)

Returns

[string](#)

pr_ChangeState(object, PCUReaderWriterState, object)

```
private void pr_ChangeState(object Sender, PCUReaderWriterState State, object obj)
```

Parameters

Sender [object](#)

State [PCUReaderWriterState](#)

obj [object](#)

semanticTreeConvertersController_ChangeState(State, ISemanticTreeConverter)

```
private void  
semanticTreeConvertersController_ChangeState(SemanticTreeConvertersController.State State,  
ISemanticTreeConverter SemanticTreeConverter)
```

Parameters

State [SemanticTreeConvertersController.State](#)

SemanticTreeConverter [ISemanticTreeConverter](#)

Events

OnChangeCompilerState

```
public event ChangeCompilerStateEventDelegate OnChangeCompilerState
```

Event Type

[ChangeCompilerStateEventDelegate](#)

Class CompilerInternalDebug

Namespace: [PascalABCCompiler](#)

Assembly: Compiler.dll

```
public class CompilerInternalDebug
```

Inheritance

[object](#) ← CompilerInternalDebug

Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.FieldSetter\(string, string, object\)](#) ,
[object.FieldGetter\(string, string, ref object\)](#) , [object.GetFieldInfo\(string, string\)](#)

Constructors

CompilerInternalDebug()

```
public CompilerInternalDebug()
```

Fields

AlwaysGenerateXMLDoc

```
public bool AlwaysGenerateXMLDoc
```

Field Value

[bool](#)

CodeGeneration

```
public bool CodeGeneration
```

Field Value

[bool](#) ↗

IncludeDebugInfoInPCU

```
public bool IncludeDebugInfoInPCU
```

Field Value

[bool](#) ↗

PCUGenerate

```
public bool PCUGenerate
```

Field Value

[bool](#) ↗

RunOnMono

```
public bool RunOnMono
```

Field Value

[bool](#) ↗

SemanticAnalysis

```
public bool SemanticAnalysis
```

Field Value

[bool](#)

SkipInternalErrorsIfSyntaxTreeIsCorrupt

```
public bool SkipInternalErrorsIfSyntaxTreeIsCorrupt
```

Field Value

[bool](#)

SkipPCUErrors

```
public bool SkipPCUErrors
```

Field Value

[bool](#)

UseStandarParserForIntellisense

```
public bool UseStandarParserForIntellisense
```

Field Value

[bool](#)

Properties

DebugVersion

```
public bool DebugVersion { get; }
```

Property Value

[bool](#)

Class CompilerOptions

Namespace: [PascalABCCompiler](#)

Assembly: Compiler.dll

Опции компиляции

```
public class CompilerOptions : MarshalByRefObject
```

Inheritance

[object](#) ← [MarshalByRefObject](#) ← CompilerOptions

Inherited Members

[MarshalByRefObject.identity](#) , [MarshalByRefObject.GetComIUnknown\(bool\)](#) ,
[MarshalByRefObject.GetComIUnknown\(MarshalByRefObject\)](#) ,
[MarshalByRefObject.IsInstanceOfType\(Type\)](#) ,
[MarshalByRefObject.InvokeMember\(string, BindingFlags, Binder, object\[\], ParameterModifier\[\], CultureInfo, string\[\]\)](#) ,
[MarshalByRefObject.MemberwiseClone\(bool\)](#) ,
[MarshalByRefObject.GetIdentity\(MarshalByRefObject, out bool\)](#) ,
[MarshalByRefObject.GetIdentity\(MarshalByRefObject\)](#) ,
[MarshalByRefObject.RaceSetServerIdentity\(ServerIdentity\)](#) ,
[MarshalByRefObject.ResetServerIdentity\(\)](#) , [MarshalByRefObject.GetLifetimeService\(\)](#) ,
[MarshalByRefObject.InitializeLifetimeService\(\)](#) , [MarshalByRefObject.CreateObjRef\(Type\)](#) ,
[MarshalByRefObject.CanCastToXmlType\(string, string\)](#) ,
[MarshalByRefObject.CanCastToXmlTypeHelper\(RuntimeType, MarshalByRefObject\)](#) ,
[MarshalByRefObject.Identity](#) , [object.ToString\(\)](#) , [object.Equals\(object\)](#) ,
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.FieldSetter\(string, string, object\)](#) ,
[object.FieldGetter\(string, string, ref object\)](#) , [object.GetFieldInfo\(string, string\)](#)

Constructors

CompilerOptions()

```
public CompilerOptions()
```

CompilerOptions(string, OutputType)

```
public CompilerOptions(string SourceFileName, CompilerOptions.OutputType OutputFileType)
```

Parameters

SourceFileName [string](#)

OutputFileType [CompilerOptions.OutputType](#)

Fields

CompiledUnitExtension

```
public string CompiledUnitExtension
```

Field Value

[string](#)

CurrentProject

```
public IProjectInfo CurrentProject
```

Field Value

[IProjectInfo](#)

Debug

```
public bool Debug
```

Field Value

[bool](#)

DisableStandardUnits

```
public bool DisableStandardUnits
```

Field Value

[bool](#)

ForDebugging

```
public bool ForDebugging
```

Field Value

[bool](#)

ForIntellisense

```
public bool ForIntellisense
```

Field Value

[bool](#)

ForceDefines

```
public List<string> ForceDefines
```

Field Value

[List](#) <[string](#)>

GenerateCode

```
public bool GenerateCode
```

Field Value

[bool](#) ↗

IgnoreRtlErrors

```
public bool IgnoreRtlErrors
```

Field Value

[bool](#) ↗

Locale

```
public string Locale
```

Field Value

[string](#) ↗

Only32Bit

```
public bool Only32Bit
```

Field Value

[bool](#) ↗

Optimise

```
public bool Optimise
```

Field Value

[bool](#)

OutputFileType

```
public CompilerOptions.OutputType OutputFileType
```

Field Value

[CompilerOptions.OutputType](#)

ParserSearchPaths

```
public string[] ParserSearchPaths
```

Field Value

[string](#)[]

ProjectCompiled

```
public bool ProjectCompiled
```

Field Value

[bool](#)

Rebuild

```
public bool Rebuild
```

Field Value

[bool](#) ↗

RunWithEnvironment

```
public bool RunWithEnvironment
```

Field Value

[bool](#) ↗

SaveDocumentation

```
public bool SaveDocumentation
```

Field Value

[bool](#) ↗

SavePCU

```
public bool SavePCU
```

Field Value

[bool](#) ↗

SavePCUInThreadPull

```
public bool SavePCUInThreadPull
```

Field Value

[bool](#)

SearchDirectories

```
public List<string> SearchDirectories
```

Field Value

[List](#) <[string](#)>

StandardDirectories

```
public Hashtable StandardDirectories
```

Field Value

[Hashtable](#)

SystemDirectory

```
public string SystemDirectory
```

Field Value

[string](#)

UnitSyntaxTree

```
public compilation_unit UnitSyntaxTree
```

Field Value

compilation_unit

outputDirectory

```
internal string outputDirectory
```

Field Value

[string](#)

outputFileName

```
private string outputFileName
```

Field Value

[string](#)

sourceFileDirectory

```
private string sourceFileDirectory
```

Field Value

[string](#)

sourceFileName

```
private string sourceFileName
```

Field Value

[string](#)

standardModules

```
private Dictionary<string, List<CompilerOptions.StandardModule>> standardModules
```

Field Value

[Dictionary](#)<[string](#), [List](#)<[CompilerOptions.StandardModule](#)>>

useDIIForSystemUnits

```
private bool useDIIForSystemUnits
```

Field Value

[bool](#)

useOutputDirectory

```
internal bool useOutputDirectory
```

Field Value

[bool](#)

Properties

OutputDirectory

```
public string OutputDirectory { get; set; }
```

Property Value

[string](#)

OutputFileName

```
public string OutputFileName { get; set; }
```

Property Value

[string](#)

SourceFileDirectory

```
public string SourceFileDirectory { get; }
```

Property Value

[string](#)

SourceFileName

```
public string SourceFileName { get; set; }
```

Property Value

[string](#)

StandardModules

Списки стандартных модулей для поддерживаемых языков (первым в списке должен быть модуль "System")

```
public Dictionary<string, List<CompilerOptions.StandardModule>> StandardModules { get;  
    set; }
```

Property Value

[Dictionary](#) <[string](#), [List](#) <[CompilerOptions.StandardModule](#)>>

UseDIIForSystemUnits

```
public bool UseDIIForSystemUnits { get; set; }
```

Property Value

[bool](#)

Methods

LoadStandardModules()

Заполняет словарь стандартных модулей для всех поддерживаемых языков

```
private void LoadStandardModules()
```

RemoveStandardModule(string, string)

```
public void RemoveStandardModule(string language, string name)
```

Parameters

language [string](#)

name [string](#)

RemoveStandardModuleAtIndex(string, int)

```
public void RemoveStandardModuleAtIndex(string language, int index)
```

Parameters

language [string](#)

index [int](#)

SetDirectories()

```
private void SetDirectories()
```

Enum CompilerOptions.OutputType

Namespace: [PascalABCCompiler](#)

Assembly: Compiler.dll

```
public enum CompilerOptions.OutputType
```

Fields

ClassLibrary = 0

ConsoleApplicaton = 1

PascalCompiledUnit = 3

SemanticTree = 4

WindowsApplication = 2

Class CompilerOptions.StandardModule

Namespace: [PascalABCCompiler](#)

Assembly: Compiler.dll

```
public class CompilerOptions.StandardModule : MarshalByRefObject
```

Inheritance

[object](#) ← [MarshalByRefObject](#) ← CompilerOptions.StandardModule

Inherited Members

[MarshalByRefObject.identity](#) , [MarshalByRefObject.GetComIUnknown\(bool\)](#) ,
[MarshalByRefObject.GetComIUnknown\(MarshalByRefObject\)](#) ,
[MarshalByRefObject.IsInstanceOfType\(Type\)](#) ,
[MarshalByRefObject.InvokeMember\(string, BindingFlags, Binder, object\[\], ParameterModifier\[\], CultureInfo, string\[\]\)](#) ,
[MarshalByRefObject.MemberwiseClone\(bool\)](#) ,
[MarshalByRefObject.GetIdentity\(MarshalByRefObject, out bool\)](#) ,
[MarshalByRefObject.GetIdentity\(MarshalByRefObject\)](#) ,
[MarshalByRefObject.RaceSetServerIdentity\(ServerIdentity\)](#) ,
[MarshalByRefObject.ResetServerIdentity\(\)](#) , [MarshalByRefObject.GetLifetimeService\(\)](#) ,
[MarshalByRefObject.InitializeLifetimeService\(\)](#) , [MarshalByRefObject.CreateObjRef\(Type\)](#) ,
[MarshalByRefObject.CanCastToXmlType\(string, string\)](#) ,
[MarshalByRefObject.CanCastToXmlTypeHelper\(RuntimeType, MarshalByRefObject\)](#) ,
[MarshalByRefObject.Identity](#) , [object.ToString\(\)](#) , [object.Equals\(object\)](#) ,
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.FieldSetter\(string, string, object\)](#) ,
[object.FieldGetter\(string, string, ref object\)](#) , [object.GetFieldInfo\(string, string\)](#)

Constructors

StandardModule(string, StandardModuleAddMethod, string)

```
public StandardModule(string name, CompilerOptions.StandardModuleAddMethod addMethod = StandardModuleAddMethod.LeftToAll, string languageToAdd = "PascalABC.NET")
```

Parameters

`name` [string](#)

`addMethod` [CompilerOptions.StandardModuleAddMethod](#)

`languageToAdd` [string](#)

Fields

addMethod

`public CompilerOptions.StandardModuleAddMethod addMethod`

Field Value

[CompilerOptions.StandardModuleAddMethod](#)

languageToAdd

`public string languageToAdd`

Field Value

[string](#)

name

`public string name`

Field Value

[string](#)

Enum CompilerOptions.StandardModuleAddMethod

Namespace: [PascalABCCompiler](#)

Assembly: Compiler.dll

```
public enum CompilerOptions.StandardModuleAddMethod
```

Fields

LeftToAll = 0

RightToMain = 1

Enum CompilerState

Namespace: [PascalABCCompiler](#)

Assembly: Compiler.dll

```
public enum CompilerState
```

Fields

BeginCompileFile = 4

BeginParsingFile = 5

CodeGeneration = 13

CompilationFinished = 14

CompilationStarting = 1

CompileImplementation = 8

CompileInterface = 7

EndCompileFile = 9

EndParsingFile = 6

PCUReadingError = 15

PCUWritingError = 16

ParserConnected = 3

ReadDLL = 10

ReadPCUFile = 11

Ready = 0

Reloading = 2

SavePCUFile = 12

SemanticTreeConversion = 18

SemanticTreeConverterConnected = 17

SyntaxTreeConversion = 19

Enum CompilerType

Namespace: [PascalABCCompiler](#)

Assembly: Compiler.dll

```
public enum CompilerType
```

Fields

Remote = 1

Standart = 0

Class ConsoleCompilerConstants

Namespace: [PascalABCCompiler](#)

Assembly: Compiler.dll

```
public class ConsoleCompilerConstants
```

Inheritance

[object](#) ← ConsoleCompilerConstants

Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.FieldSetter\(string, string, object\)](#) ,
[object.FieldGetter\(string, string, ref object\)](#) , [object.GetFieldInfo\(string, string\)](#)

Constructors

ConsoleCompilerConstants()

```
public ConsoleCompilerConstants()
```

Fields

BeginOffest

```
public const int BeginOffest = 181
```

Field Value

[int](#)

CommandCompile

```
public const int CommandCompile = 210
```

Field Value

[int ↗](#)

CommandExit

```
public const int CommandExit = 200
```

Field Value

[int ↗](#)

CommandGCollect

```
public const int CommandGCollect = 201
```

Field Value

[int ↗](#)

CommandStartNumber

```
public const int CommandStartNumber = 100
```

Field Value

[int ↗](#)

CompilerLocale

```
public const int CompilerLocale = 228
```

Field Value

[int↗](#)

CompilerOptions

```
public const int CompilerOptions = 222
```

Field Value

[int↗](#)

CompilerOptionsClearStandartModules

```
public const int CompilerOptionsClearStandartModules = 219
```

Field Value

[int↗](#)

CompilerOptionsDebug

```
public const int CompilerOptionsDebug = 212
```

Field Value

[int↗](#)

CompilerOptionsFileName

```
public const int CompilerOptionsFileName = 215
```

Field Value

[int](#)

CompilerOptionsForDebugging

```
public const int CompilerOptionsForDebugging = 223
```

Field Value

[int](#)

CompilerOptionsOutputDirectory

```
public const int CompilerOptionsOutputDirectory = 213
```

Field Value

[int](#)

CompilerOptionsOutputType

```
public const int CompilerOptionsOutputType = 211
```

Field Value

[int](#)

CompilerOptionsProjectCompiled

```
public const int CompilerOptionsProjectCompiled = 225
```

Field Value

[int ↗](#)

CompilerOptionsRebuild

```
public const int CompilerOptionsRebuild = 214
```

Field Value

[int ↗](#)

CompilerOptionsRunWithEnvironment

```
public const int CompilerOptionsRunWithEnvironment = 224
```

Field Value

[int ↗](#)

CompilerOptionsStandartModule

```
public const int CompilerOptionsStandartModule = 218
```

Field Value

[int ↗](#)

DataSeparator

```
public const string DataSeparator = "]\r\n["
```

Field Value

[string](#)

Error

```
public const int Error = 182
```

Field Value

[int](#)

FileExist

```
public const int FileExist = 184
```

Field Value

[int](#)

GetLastWriteTime

```
public const int GetLastWriteTime = 186
```

Field Value

[int](#)

IDELocale

```
public const int IDELocale = 227
```

Field Value

[int ↗](#)

InternalDebug

```
public const int InternalDebug = 221
```

Field Value

[int ↗](#)

InternalDebugSavePCU

```
public const int InternalDebugSavePCU = 216
```

Field Value

[int ↗](#)

InternalError

```
public const int InternalError = 188
```

Field Value

[int ↗](#)

LinesCompiled

```
public const int LinesCompiled = 180
```

Field Value

[int ↗](#)

MaxProcessMemoryMB

```
public const int MaxProcessMemoryMB = 500
```

Field Value

[int ↗](#)

MessageSeparator

```
public static readonly string MessageSeparator
```

Field Value

[string ↗](#)

PABCHealth

```
public const int PABCHealth = 189
```

Field Value

[int ↗](#)

SourceFileText

```
public const int SourceFileText = 185
```

Field Value

[int↗](#)

UseDIIForSystemUnits

```
public const int UseDIIForSystemUnits = 226
```

Field Value

[int↗](#)

VarBeginOffest

```
public const int VarBeginOffest = 187
```

Field Value

[int↗](#)

Warning

```
public const int Warning = 183
```

Field Value

[int↗](#)

WorkingSet

```
public const int WorkingSet = 217
```

Field Value

[int ↗](#)

Class DocXmlManager

Namespace: [PascalABCCompiler](#)

Assembly: Compiler.dll

```
public class DocXmlManager
```

Inheritance

[object](#) ← DocXmlManager

Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.FieldSetter\(string, string, object\)](#) ,
[object.FieldGetter\(string, string, ref object\)](#) , [object.GetFieldInfo\(string, string\)](#)

Constructors

DocXmlManager()

```
public DocXmlManager()
```

Fields

CU

```
private CompilationUnit cu
```

Field Value

[CompilationUnit](#)

is_assembly

```
private bool is_assembly
```

Field Value

[bool](#)

unit_name

```
private string unit_name
```

Field Value

[string](#)

xtw

```
private XmlWriter xtw
```

Field Value

[XmlWriter](#)

Methods

GetGenericFlag(common_function_node)

```
private string GetGenericFlag(common_function_node cfn)
```

Parameters

cfn common_function_node

Returns

[string](#)

GetParameters(common_function_node)

```
private string GetParameters(common_function_node cfn)
```

Parameters

cfn common_function_node

Returns

[string](#)

SaveClassConstant(class_constant_definition)

```
private void SaveClassConstant(class_constant_definition cfn)
```

Parameters

cfn class_constant_definition

SaveConstant(namespace_constant_definition)

```
private void SaveConstant(namespace_constant_definition cfn)
```

Parameters

cfn namespace_constant_definition

SaveEvent(common_event)

```
private void SaveEvent(common_event cfn)
```

Parameters

cfn common_event

SaveField(class_field)

```
private void SaveField(class_field cfn)
```

Parameters

cfn class_field

SaveFunction(common_namespace_function_node)

```
private void SaveFunction(common_namespace_function_node cfn)
```

Parameters

cfn common_namespace_function_node

SaveMembers()

```
private void SaveMembers()
```

SaveMethod(common_method_node)

```
private void SaveMethod(common_method_node cfn)
```

Parameters

cfn common_method_node

SaveProperty(common_property_node)

```
private void SaveProperty(common_property_node cfn)
```

Parameters

cfn common_property_node

SaveType(common_type_node)

```
private void SaveType(common_type_node ctn)
```

Parameters

ctn common_type_node

SaveTypeSynonim(type_synonym)

```
private void SaveTypeSynonim(type_synonym ctn)
```

Parameters

ctn type_synonym

SaveVariable(var_definition_node)

```
private void SaveVariable(var_definition_node cfn)
```

Parameters

cfn var_definition_node

SaveXml(CompilationUnit)

```
public void SaveXml(CompilationUnit cu)
```

Parameters

cu [CompilationUnit](#)

get_array_name(common_type_node)

```
private string get_array_name(common_type_node ctn)
```

Parameters

ctn common_type_node

Returns

[string](#) ↗

get_binary_file_name(common_type_node)

```
private string get_binary_file_name(common_type_node ctn)
```

Parameters

ctn common_type_node

Returns

[string](#) ↗

get_constant(constant_node)

```
private string get_constant(constant_node cn)
```

Parameters

cn constant_node

Returns

[string](#)

get_delegate_name(common_type_node)

```
private string get_delegate_name(common_type_node ctn)
```

Parameters

ctn common_type_node

Returns

[string](#)

get_diap_name(common_type_node)

```
private string get_diap_name(common_type_node ctn)
```

Parameters

ctn common_type_node

Returns

[string](#)

get_dyn_array_name(common_type_node)

```
private string get_dyn_array_name(common_type_node ctn)
```

Parameters

ctn common_type_node

Returns

[string](#)

get_name(type_node)

```
private string get_name(type_node tn)
```

Parameters

tn type_node

Returns

[string](#)

get_pointer_name(ref_type_node)

```
private string get_pointer_name(ref_type_node ctn)
```

Parameters

ctn ref_type_node

Returns

[string](#)

get_set_type_name(common_type_node)

```
private string get_set_type_name(common_type_node ctn)
```

Parameters

ctn common_type_node

Returns

[string](#)

get_shortstring_name(short_string_type_node)

```
private string get_shortstring_name(short_string_type_node ctn)
```

Parameters

ctn short_string_type_node

Returns

[string](#)

get_typed_file_name(common_type_node)

```
private string get_typed_file_name(common_type_node ctn)
```

Parameters

ctn common_type_node

Returns

[string](#)

Interface ICompiler

Namespace: [PascalABCCompiler](#)

Assembly: Compiler.dll

```
public interface ICompiler
```

Properties

BeginOffset

```
int BeginOffset { get; }
```

Property Value

[int](#)

CompilerOptions

```
CompilerOptions CompilerOptions { get; set; }
```

Property Value

[CompilerOptions](#)

CompilerType

```
CompilerType CompilerType { get; }
```

Property Value

[CompilerType](#)

ErrorsList

```
List<Error> ErrorsList { get; }
```

Property Value

[List](#) <Error>

InternalDebug

```
CompilerInternalDebug InternalDebug { get; set; }
```

Property Value

[CompilerInternalDebug](#)

LinesCompiled

```
uint LinesCompiled { get; }
```

Property Value

[uint](#)

PABCCodeHealth

```
int PABCCodeHealth { get; }
```

Property Value

[int](#)

SemanticTree

```
IProgramNode SemanticTree { get; }
```

Property Value

IProgramNode

SemanticTreeConvertersController

```
SemanticTreeConvertersController SemanticTreeConvertersController { get; }
```

Property Value

[SemanticTreeConvertersController](#)

SourceFilesProvider

```
SourceFilesProviderDelegate SourceFilesProvider { get; }
```

Property Value

SourceFilesProviderDelegate

State

```
CompilerState State { get; }
```

Property Value

[CompilerState](#)

SupportedProjectFiles

```
SupportedSourceFile[] SupportedProjectFiles { get; }
```

Property Value

[SupportedSourceFile\[\]](#)

SupportedSourceFiles

```
SupportedSourceFile[] SupportedSourceFiles { get; }
```

Property Value

[SupportedSourceFile\[\]](#)

UnitTable

```
CompilationUnitHashTable UnitTable { get; }
```

Property Value

[CompilationUnitHashTable](#)

VarBeginOffset

```
int VarBeginOffset { get; }
```

Property Value

[int↗](#)

Warnings

```
List<CompilerWarning> Warnings { get; }
```

Property Value

[List](#)<CompilerWarning>

Methods

AddWarnings(List<CompilerWarning>)

```
void AddWarnings(List<CompilerWarning> WarningList)
```

Parameters

WarningList [List](#)<CompilerWarning>

Compile()

```
string Compile()
```

Returns

[string](#)

Free()

```
void Free()
```

GetSourceFileText(string)

```
string GetSourceFileText(string FileName)
```

Parameters

FileName [string](#)

Returns

[string](#)

ParseText(string, string, List<Error>, List<CompilerWarning>)

```
compilation_unit ParseText(string FileName, string Text, List<Error> ErrorList,  
List<CompilerWarning> Warnings)
```

Parameters

FileName [string](#)

Text [string](#)

ErrorList [List](#)<Error>

Warnings [List](#)<CompilerWarning>

Returns

compilation_unit

Reload()

```
void Reload()
```

StartCompile()

```
void StartCompile()
```

Events

OnChangeCompilerState

`event ChangeCompilerStateEventDelegate OnChangeCompilerState`

Event Type

[ChangeCompilerStateEventDelegate](#)

Interface IFileInfo

Namespace: [PascalABCCompiler](#)

Assembly: Compiler.dll

```
public interface IFileInfo
```

Properties

Name

```
string Name { get; set; }
```

Property Value

[string](#) ↗

Path

```
string Path { get; }
```

Property Value

[string](#) ↗

Interface IProjectInfo

Namespace: [PascalABCCompiler](#)

Assembly: Compiler.dll

```
public interface IProjectInfo
```

Properties

AppIcon

```
string AppIcon { get; set; }
```

Property Value

[string](#)

BuildVersion

```
int BuildVersion { get; set; }
```

Property Value

[int](#)

CommandLineArguments

```
string CommandLineArguments { get; set; }
```

Property Value

[string](#)

Company

```
string Company { get; set; }
```

Property Value

[string](#) ↗

Copyright

```
string Copyright { get; set; }
```

Property Value

[string](#) ↗

DeleteEXE

```
bool DeleteEXE { get; set; }
```

Property Value

[bool](#) ↗

DeletePDB

```
bool DeletePDB { get; set; }
```

Property Value

[bool](#) ↗

Description

```
string Description { get; set; }
```

Property Value

[string](#)

GenerateXMLDoc

```
bool GenerateXMLDoc { get; set; }
```

Property Value

[bool](#)

IncludeDebugInfo

```
bool IncludeDebugInfo { get; set; }
```

Property Value

[bool](#)

MainFile

```
string MainFile { get; set; }
```

Property Value

[string](#)

MajorVersion

```
int MajorVersion { get; set; }
```

Property Value

[int](#)

MinorVersion

```
int MinorVersion { get; set; }
```

Property Value

[int](#)

Name

```
string Name { get; }
```

Property Value

[string](#)

OutputDirectory

```
string OutputDirectory { get; set; }
```

Property Value

[string](#)

OutputFileName

```
string OutputFileName { get; }
```

Property Value

[string](#)

Path

```
string Path { get; }
```

Property Value

[string](#)

Product

```
string Product { get; set; }
```

Property Value

[string](#)

ProjectDirectory

```
string ProjectDirectory { get; }
```

Property Value

[string](#)

ProjectType

```
ProjectType ProjectType { get; }
```

Property Value

[ProjectType](#)

References

```
IReferenceInfo[] References { get; }
```

Property Value

[IReferenceInfo\[\]](#)

Resources

```
IResourceInfo[] Resources { get; }
```

Property Value

[IResourceInfo\[\]](#)

RevisionVersion

```
int RevisionVersion { get; set; }
```

Property Value

[int](#)

SourceFiles

```
IFileInfo[] SourceFiles { get; }
```

Property Value

[IFileInfo\[\]](#)

Title

```
string Title { get; set; }
```

Property Value

[string](#)

Trademark

```
string Trademark { get; set; }
```

Property Value

[string](#)

Methods

ContainsSourceFile(string)

```
bool ContainsSourceFile(string FileName)
```

Parameters

FileName [string](#)

Returns

bool ↗

Interface IReferenceInfo

Namespace: [PascalABCCompiler](#)

Assembly: Compiler.dll

```
public interface IReferenceInfo
```

Properties

AssemblyName

```
string AssemblyName { get; }
```

Property Value

[string](#) ↗

FullAssemblyName

```
string FullAssemblyName { get; }
```

Property Value

[string](#) ↗

Interface IRemoteCompiler

Namespace: [PascalABCCompiler](#)

Assembly: Compiler.dll

```
public interface IRemoteCompiler
```

Properties

MaxProcessMemoryMB

```
int MaxProcessMemoryMB { get; set; }
```

Property Value

[int](#)

Events

RemoteCompilerChannelAction

```
event RemoteCompilerChannelEventDelegate RemoteCompilerChannelAction
```

Event Type

[RemoteCompilerChannelEventDelegate](#)

Interface IResourceInfo

Namespace: [PascalABCCompiler](#)

Assembly: Compiler.dll

```
public interface IResourceInfo
```

Properties

Name

```
string Name { get; }
```

Property Value

[string](#) ↗

Class ProjectInfo

Namespace: [PascalABCCompiler](#)

Assembly: Compiler.dll

```
public class ProjectInfo : IProjectInfo
```

Inheritance

[object](#) ← ProjectInfo

Implements

[IProjectInfo](#)

Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.FieldSetter\(string, string, object\)](#) ,
[object.FieldGetter\(string, string, ref object\)](#) , [object.GetFieldInfo\(string, string\)](#)

Constructors

ProjectInfo()

```
public ProjectInfo()
```

Fields

ProjectFileVersion

```
private static int ProjectFileVersion
```

Field Value

[int](#)

_app_icon

```
private string _app_icon
```

Field Value

[string](#) ↗

_build_version

```
private int _build_version
```

Field Value

[int](#) ↗

_command_line_arguments

```
private string _command_line_arguments
```

Field Value

[string](#) ↗

_company

```
private string _company
```

Field Value

[string](#) ↗

_copyright

```
private string _copyright
```

Field Value

[string](#) ↗

_delete_exe

```
private bool _delete_exe
```

Field Value

[bool](#) ↗

_delete_pdb

```
private bool _delete_pdb
```

Field Value

[bool](#) ↗

_description

```
private string _description
```

Field Value

[string](#) ↗

_generate_xml_doc

```
private bool _generate_xml_doc
```

Field Value

[bool](#)

_include_debug_info

```
private bool _include_debug_info
```

Field Value

[bool](#)

_main_file

```
private string _main_file
```

Field Value

[string](#)

_major_version

```
private int _major_version
```

Field Value

[int](#)

_minor_version

```
private int _minor_version
```

Field Value

[int](#)

_name

```
private string _name
```

Field Value

[string](#)

_output_directory

```
private string _output_directory
```

Field Value

[string](#)

_output_file_name

```
private string _output_file_name
```

Field Value

[string](#)

_path

```
private string _path
```

Field Value

[string](#)

_product

```
private string _product
```

Field Value

[string](#)

_project_type

```
private ProjectType _project_type
```

Field Value

[ProjectType](#)

_references

```
private List<ReferenceInfo> _references
```

Field Value

[List](#) <[ReferenceInfo](#)>

_resources

```
private List<ResourceInfo> _resources
```

Field Value

[List](#) <[ResourceInfo](#)>

_revision_version

```
private int _revision_version
```

Field Value

[int](#)

_source_files

```
private List<SourceCodeFileInfo> _source_files
```

Field Value

[List](#) <[SourceCodeFileInfo](#)>

_title

```
private string _title
```

Field Value

[string](#)

_trademark

```
private string _trademark
```

Field Value

[string](#) ↗

Properties

AppIcon

```
public string AppIcon { get; set; }
```

Property Value

[string](#) ↗

BuildVersion

```
public int BuildVersion { get; set; }
```

Property Value

[int](#) ↗

CommandLineArguments

```
public string CommandLineArguments { get; set; }
```

Property Value

[string](#) ↗

Company

```
public string Company { get; set; }
```

Property Value

[string](#) ↗

Copyright

```
public string Copyright { get; set; }
```

Property Value

[string](#) ↗

DeleteEXE

```
public bool DeleteEXE { get; set; }
```

Property Value

[bool](#) ↗

DeletePDB

```
public bool DeletePDB { get; set; }
```

Property Value

[bool](#) ↗

Description

```
public string Description { get; set; }
```

Property Value

[string](#)

GenerateXMLDoc

```
public bool GenerateXMLDoc { get; set; }
```

Property Value

[bool](#)

IncludeDebugInfo

```
public bool IncludeDebugInfo { get; set; }
```

Property Value

[bool](#)

MainFile

```
public string MainFile { get; set; }
```

Property Value

[string](#)

MajorVersion

```
public int MajorVersion { get; set; }
```

Property Value

[int ↗](#)

MinorVersion

```
public int MinorVersion { get; set; }
```

Property Value

[int ↗](#)

Name

```
public string Name { get; }
```

Property Value

[string ↗](#)

OutputDirectory

```
public string OutputDirectory { get; set; }
```

Property Value

[string ↗](#)

OutputFileName

```
public string OutputFileName { get; }
```

Property Value

[string](#) ↗

Path

```
public string Path { get; }
```

Property Value

[string](#) ↗

Product

```
public string Product { get; set; }
```

Property Value

[string](#) ↗

ProjectDirectory

```
public string ProjectDirectory { get; }
```

Property Value

[string](#) ↗

ProjectType

```
public ProjectType ProjectType { get; }
```

Property Value

[ProjectType](#)

References

```
public IReferenceInfo[] References { get; }
```

Property Value

[IReferenceInfo\[\]](#)

Resources

```
public IResourceInfo[] Resources { get; }
```

Property Value

[IResourceInfo\[\]](#)

RevisionVersion

```
public int RevisionVersion { get; set; }
```

Property Value

[int↗](#)

SourceFiles

```
public IFileInfo[] SourceFiles { get; }
```

Property Value

[IFileInfo\[\]](#)

Title

```
public string Title { get; set; }
```

Property Value

[string](#)

Trademark

```
public string Trademark { get; set; }
```

Property Value

[string](#)

app_icon

```
public string app_icon { get; set; }
```

Property Value

[string](#)

build_version

```
public int build_version { get; set; }
```

Property Value

[int ↗](#)

command_line_arguments

```
public string command_line_arguments { get; set; }
```

Property Value

[string ↗](#)

company

```
public string company { get; set; }
```

Property Value

[string ↗](#)

copyright

```
public string copyright { get; set; }
```

Property Value

[string ↗](#)

delete_exe

```
public bool delete_exe { get; set; }
```

Property Value

[bool](#)

delete_pdb

```
public bool delete_pdb { get; set; }
```

Property Value

[bool](#)

description

```
public string description { get; set; }
```

Property Value

[string](#)

generate_xml_doc

```
public bool generate_xml_doc { get; set; }
```

Property Value

[bool](#)

include_debug_info

```
public bool include_debug_info { get; set; }
```

Property Value

[bool](#)

main_file

```
public string main_file { get; set; }
```

Property Value

[string](#)

major_version

```
public int major_version { get; set; }
```

Property Value

[int](#)

minor_version

```
public int minor_version { get; set; }
```

Property Value

[int](#)

name

```
public string name { get; set; }
```

Property Value

[string](#) ↗

output_directory

```
public string output_directory { get; set; }
```

Property Value

[string](#) ↗

output_file_name

```
public string output_file_name { get; set; }
```

Property Value

[string](#) ↗

path

```
public string path { get; set; }
```

Property Value

[string](#) ↗

product

```
public string product { get; set; }
```

Property Value

[string](#)

project_type

```
public ProjectType project_type { get; set; }
```

Property Value

[ProjectType](#)

references

```
public List<ReferenceInfo> references { get; set; }
```

Property Value

[List](#) <[ReferenceInfo](#)>

resources

```
public List<ResourceInfo> resources { get; set; }
```

Property Value

[List](#) <[ResourceInfo](#)>

revision_version

```
public int revision_version { get; set; }
```

Property Value

[int](#)

source_files

```
public List<SourceCodeFileInfo> source_files { get; set; }
```

Property Value

[List](#)<[SourceCodeFileInfo](#)>

title

```
public string title { get; set; }
```

Property Value

[string](#)

trademark

```
public string trademark { get; set; }
```

Property Value

[string](#)

Methods

ContainsSourceFile(string)

```
public bool ContainsSourceFile(string FileName)
```

Parameters

FileName [string](#)

Returns

[bool](#)

ExcludeFile(IFileInfo)

```
public void ExcludeFile(IFileInfo fi)
```

Parameters

fi [IFileInfo](#)

Load(string)

```
public void Load(string FileName)
```

Parameters

FileName [string](#)

RemoveReference(IReferenceInfo)

```
public void RemoveReference(IReferenceInfo ri)
```

Parameters

ri [IReferenceInfo](#)

Save()

```
public void Save()
```

Enum ProjectType

Namespace: [PascalABCCompiler](#)

Assembly: Compiler.dll

```
public enum ProjectType
```

Fields

ConsoleApp = 0

Library = 1

WindowsApp = 2

Class ReferenceInfo

Namespace: [PascalABCCompiler](#)

Assembly: Compiler.dll

```
public class ReferenceInfo : IReferenceInfo
```

Inheritance

[object](#) ← ReferenceInfo

Implements

[IReferenceInfo](#)

Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.FieldSetter\(string, string, object\)](#) ,
[object.FieldGetter\(string, string, ref object\)](#) , [object.GetFieldInfo\(string, string\)](#)

Constructors

ReferenceInfo()

```
public ReferenceInfo()
```

ReferenceInfo(string, string)

```
public ReferenceInfo(string assembly_name, string full_assembly_name)
```

Parameters

assembly_name [string](#)

full_assembly_name [string](#)

Fields

_assembly_name

```
private string _assembly_name
```

Field Value

[string](#)

_full_assembly_name

```
private string _full_assembly_name
```

Field Value

[string](#)

Properties

AssemblyName

```
public string AssemblyName { get; }
```

Property Value

[string](#)

FullAssemblyName

```
public string FullAssemblyName { get; }
```

Property Value

[string](#) ↗

assembly_name

```
public string assembly_name { get; set; }
```

Property Value

[string](#) ↗

full_assembly_name

```
public string full_assembly_name { get; set; }
```

Property Value

[string](#) ↗

Class RemoteCompiler

Namespace: [PascalABCCompiler](#)

Assembly: Compiler.dll

```
public class RemoteCompiler : ICompiler
```

Inheritance

[object](#) ← RemoteCompiler

Implements

[ICompiler](#)

Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.FieldSetter\(string, string, object\)](#) ,
[object.FieldGetter\(string, string, ref object\)](#) , [object.GetFieldInfo\(string, string\)](#)

Constructors

RemoteCompiler(int, ChangeCompilerStateEventDelegate, SourceFilesProviderDelegate)

```
public RemoteCompiler(int maxProcessMemoryMB, ChangeCompilerStateEventDelegate  
ChangeCompilerState, SourceFilesProviderDelegate sourceFilesProvider)
```

Parameters

maxProcessMemoryMB [int](#)

ChangeCompilerState [ChangeCompilerStateEventDelegate](#)

sourceFilesProvider SourceFilesProviderDelegate

Fields

beginOffset

```
private int beginOffset
```

Field Value

[int](#)

compilationSatarted

```
private bool compilationSatarted
```

Field Value

[bool](#)

compilerOptions

```
private CompilerOptions compilerOptions
```

Field Value

[CompilerOptions](#)

compilerReloading

```
private bool compilerReloading
```

Field Value

[bool](#)

compilerState

```
public volatile CompilerState compilerState
```

Field Value

[CompilerState](#)

errorsList

```
private List<Error> errorsList
```

Field Value

[List](#)<Error>

inputEncoding

```
private Encoding inputEncoding
```

Field Value

[Encoding](#)

inputId

```
private string inputId
```

Field Value

[string](#)

internalDebug

```
private CompilerInternalDebug internalDebug
```

Field Value

[CompilerInternalDebug](#)

linesCompiled

```
private uint linesCompiled
```

Field Value

[uint](#)

maxProcessMemoryMB

```
private int maxProcessMemoryMB
```

Field Value

[int](#)

pABCCodeHealth

```
private int pABCCodeHealth
```

Field Value

[int](#)

pabcnetcFileName

```
private string pabcnetcFileName
```

Field Value

[string](#)

pabcnetcProcess

```
private Process pabcnetcProcess
```

Field Value

[Process](#)

pabcnetcStreamReader

```
private EventedStreamReaderList pabcnetcStreamReader
```

Field Value

EventedStreamReaderList

remoteCompilerWorkingSet

```
private long remoteCompilerWorkingSet
```

Field Value

[long](#)

sendCommandStartNumber

```
public const int sendCommandStartNumber = 100
```

Field Value

[int](#)

sourceFilesProvider

```
private SourceFilesProviderDelegate sourceFilesProvider
```

Field Value

SourceFilesProviderDelegate

varBeginOffset

```
private int varBeginOffset
```

Field Value

[int](#)

warnings

```
public List<CompilerWarning> warnings
```

Field Value

[List](#) <CompilerWarning>

Properties

BeginOffset

```
public int BeginOffset { get; }
```

Property Value

[int](#)

CompilerOptions

```
public CompilerOptions CompilerOptions { get; set; }
```

Property Value

[CompilerOptions](#)

CompilerType

```
public CompilerType CompilerType { get; }
```

Property Value

[CompilerType](#)

ErrorsList

```
public List<Error> ErrorsList { get; }
```

Property Value

[List](#) <Error>

InternalDebug

```
public CompilerInternalDebug InternalDebug { get; set; }
```

Property Value

[CompilerInternalDebug](#)

LinesCompiled

```
public uint LinesCompiled { get; }
```

Property Value

[uint](#)

MaxProcessMemoryMB

```
public int MaxProcessMemoryMB { get; set; }
```

Property Value

[int](#)

PABCCodeHealth

Здоровье кода на всякий случай выносим в интерфейс компилятора Реально оно будет использоваться только при запуске из под оболочки (Remote Compiler)

```
public int PABCCodeHealth { get; }
```

Property Value

[int](#)

RemoteCompilerWorkingSet

```
public long RemoteCompilerWorkingSet { get; }
```

Property Value

[long](#)

SemanticTree

```
public IProgramNode SemanticTree { get; }
```

Property Value

IProgramNode

SemanticTreeConvertersController

```
public SemanticTreeConvertersController SemanticTreeConvertersController { get; }
```

Property Value

[SemanticTreeConvertersController](#)

SourceFilesProvider

```
public SourceFilesProviderDelegate SourceFilesProvider { get; set; }
```

Property Value

SourceFilesProviderDelegate

State

```
public CompilerState State { get; }
```

Property Value

[CompilerState](#)

SupportedProjectFiles

```
public SupportedSourceFile[] SupportedProjectFiles { get; }
```

Property Value

[SupportedSourceFile\[\]](#)

SupportedSourceFiles

```
public SupportedSourceFile[] SupportedSourceFiles { get; }
```

Property Value

[SupportedSourceFile\[\]](#)

UnitTable

```
public CompilationUnitHashTable UnitTable { get; }
```

Property Value

[CompilationUnitHashTable](#)

VarBeginOffset

```
public int VarBeginOffset { get; }
```

Property Value

[int](#)

Warnings

```
public List<CompilerWarning> Warnings { get; }
```

Property Value

[List](#)<CompilerWarning>

Methods

AddWarnings(List<CompilerWarning>)

```
public void AddWarnings(List<CompilerWarning> WarningList)
```

Parameters

WarningList [List](#)<CompilerWarning>

ChangeCompilerState(CompilerState, string)

```
private void ChangeCompilerState(CompilerState State, string FileName)
```

Parameters

State [CompilerState](#)

FileName [string](#)

CheckProcessMemory()

```
private bool CheckProcessMemory()
```

Returns

[bool](#)

Compile()

```
public string Compile()
```

Returns

[string](#)

~RemoteCompiler()

```
protected ~RemoteCompiler()
```

Free()

```
public void Free()
```

GetSourceFileText(string)

```
public string GetSourceFileText(string FileName)
```

Parameters

[FileName](#) [string](#)

Returns

[string](#)

IsUnix()

```
public static bool IsUnix()
```

Returns

[bool](#)

ParseText(string, string, List<Error>, List<CompilerWarning>)

```
public compilation_unit ParseText(string FileName, string Text, List<Error> ErrorList,  
List<CompilerWarning> Warnings)
```

Parameters

FileName [string](#)

Text [string](#)

ErrorList [List](#)<Error>

Warnings [List](#)<CompilerWarning>

Returns

compilation_unit

Reload()

```
public void Reload()
```

StartCompile()

```
public void StartCompile()
```

pabcnetcProcess_Exited(object, EventArgs)

```
private void pabcnetcProcess_Exited(object sender, EventArgs e)
```

Parameters

sender [object](#)

e [EventArgs](#)

sendCommand(int)

```
private void sendCommand(int command)
```

Parameters

command [int](#)

sendCommand(int, object)

```
private void sendCommand(int command, object arg)
```

Parameters

command [int](#)

arg [object](#)

sendCommand(int, params object[])

```
private void sendCommand(int command, params object[] args)
```

Parameters

command [int](#)

args [object](#)[]

sendCompilerOptions()

```
private void sendCompilerOptions()
```

sendObject(int, object)

```
private void sendObject(int id, object o)
```

Parameters

id [int](#)

o [object](#)

sendObjectAsByteArray(int, string)

```
private void sendObjectAsByteArray(int command, string obj)
```

Parameters

command [int](#)

obj [string](#)

stopCompiler()

```
private void stopCompiler()

stringReceived(string, string)

public void stringReceived(string id, string line)
```

Parameters

`id` [string](#)

`line` [string](#)

waitCompilerReloading()

```
private void waitCompilerReloading()
```

Events

EnvorimentIdle

```
public event RemoteCompiler.EnvorimentIdleDelegate EnvorimentIdle
```

Event Type

[RemoteCompiler.EnvorimentIdleDelegate](#)

OnChangeCompilerState

```
public event ChangeCompilerStateEventDelegate OnChangeCompilerState
```

Event Type

[ChangeCompilerStateEventDelegate](#)

Delegate RemoteCompiler.EnvorimentIdleDelegate

Namespace: [PascalABCCompiler](#)

Assembly: Compiler.dll

```
public delegate void RemoteCompiler.EnvorimentIdleDelegate()
```

Constructors

EnvorimentIdleDelegate(object, IntPtr)

```
public EnvorimentIdleDelegate(object @object, IntPtr method)
```

Parameters

object [object](#)

method [IntPtr](#)

Methods

BeginInvoke(AsyncCallback, object)

```
public virtual IAsyncResult BeginInvoke(AsyncCallback callback, object @object)
```

Parameters

callback [AsyncCallback](#)

object [object](#)

Returns

[IAsyncResult](#)

EndInvoke(IAsyncResult)

```
public virtual void EndInvoke(IAsyncResult result)
```

Parameters

result [IAsyncResult](#)

Invoke()

```
public virtual void Invoke()
```

Delegate RemoteCompilerChannelEventDelegate

Namespace: [PascalABCCompiler](#)

Assembly: Compiler.dll

```
public delegate void RemoteCompilerChannelEventDelegate(RemoteCompilerChannelEventType  
eventType, string text)
```

Parameters

eventType [RemoteCompilerChannelEventType](#)

text [string](#)

Constructors

RemoteCompilerChannelEventDelegate(object, IntPtr)

```
public RemoteCompilerChannelEventDelegate(object @object, IntPtr method)
```

Parameters

object [object](#)

method [IntPtr](#)

Methods

BeginInvoke(RemoteCompilerChannelEventType, string, AsyncCallback, object)

```
public virtual IAsyncResult BeginInvoke(RemoteCompilerChannelEventType eventType, string  
text, AsyncCallback callback, object @object)
```

Parameters

eventType [RemoteCompilerChannelEventType](#)

text [string](#)

callback [AsyncCallback](#)

object [object](#)

Returns

[IAsyncResult](#)

EndInvoke(IAsyncResult)

`public virtual void EndInvoke(IAsyncResult result)`

Parameters

result [IAsyncResult](#)

Invoke(RemoteCompilerChannelEventType, string)

`public virtual void Invoke(RemoteCompilerChannelEventType eventType, string text)`

Parameters

eventType [RemoteCompilerChannelEventType](#)

text [string](#)

Enum RemoteCompilerChannelEventType

Namespace: [PascalABCCompiler](#)

Assembly: Compiler.dll

```
public enum RemoteCompilerChannelEventType
```

Fields

Receive = 1

Send = 0

Class RemoteCompilerError

Namespace: [PascalABCCompiler](#)

Assembly: Compiler.dll

```
public class RemoteCompilerError : CompilationErrorException, ISerializable, _Exception
```

Inheritance

```
object ↗ ← Exception ↗ ← Error ← LocatedError ← SemanticError ← CompilationError ←  
CompilationErrorException ← RemoteCompilerError
```

Implements

```
ISerializable ↗ , Exception ↗
```

Inherited Members

```
CompilationErrorException._loc , CompilationErrorException.loc ,  
CompilationErrorException.Location , CompilationErrorException.loc_to_string(ILocation) ,  
CompilationErrorException.get_location(semantic_node) , CompilationErrorException.Message ,  
SemanticErrorException.SourceLocation , LocatedErrorException.source_context , LocatedErrorException.sourceLocation ,  
LocatedErrorException.fileName , LocatedErrorException.SourceContext , LocatedErrorException.FileName , Error.MustThrow ,  
Exception.s\_EDILock ↗ , Exception.className ↗ , Exception.exceptionMethod ↗ ,  
Exception.exceptionMethodString ↗ , Exception.message ↗ , Exception.data ↗ ,  
Exception.innerException ↗ , Exception.helpURL ↗ , Exception.stackTrace ↗ ,  
Exception.watsonBuckets ↗ , Exception.stackTraceString ↗ , Exception.remoteStackTraceString ↗ ,  
Exception.remoteStackIndex ↗ , Exception.dynamicMethods ↗ , Exception.HResult ↗ ,  
Exception.source ↗ , Exception.xptrs ↗ , Exception.xcode ↗ , Exception.ipForWatsonBuckets ↗ ,  
Exception.safeSerializationManager ↗ , Exception.COMPlusExceptionCode ↗ , Exception.Init\(\) ↗ ,  
Exception.IsImmutableAgileException(Exception) ↗ ,  
Exception.AddExceptionDataForRestrictedErrorInfo(string, string, string, object, bool) ↗ ,  
Exception.TryGetRestrictedLanguageErrorObject(out object) ↗ , Exception.GetClassName\(\) ↗ ,  
Exception.GetBaseException\(\) ↗ , Exception.GetMethodFromStackTrace(object) ↗ ,  
Exception.GetExceptionMethodFromStackTrace() ↗ , Exception.GetTargetSiteInternal() ↗ ,  
Exception.GetStackTrace(bool) ↗ , Exception.SetErrorCode(int) ↗ , Exception.ToString(bool, bool) ↗ ,  
Exception.GetExceptionMethodString() ↗ , Exception.GetExceptionMethodFromString() ↗ ,  
Exception.GetObjectData(SerializationInfo, StreamingContext) ↗ , Exception.PrepForRemoting() ↗ ,  
Exception.OnDeserialized(StreamingContext) ↗ , Exception.InternalPreserveStackTrace() ↗ ,  
Exception.PrepareForForeignExceptionRaise() ↗ ,  
Exception.GetStackTracesDeepCopy(Exception, out object, out object) ↗ ,  
Exception.SaveStackTracesFromDeepCopy(Exception, object, object) ↗ ,
```

[Exception.CopyStackTrace\(object\)](#) , [Exception.CopyDynamicMethods\(object\)](#) ,
[Exception.StripFileInfo\(string, bool\)](#) , [Exception.DeepCopyStackTrace\(object\)](#) ,
[Exception.DeepCopyDynamicMethods\(object\)](#) ,
[Exception.GetStackTracesDeepCopy\(out object, out object\)](#) ,
[Exception.RestoreExceptionDispatchInfo\(ExceptionDispatchInfo\)](#) , [Exception.InternalToString\(\)](#) ,
[Exception.GetType\(\)](#) , [Exception.nlsTransient\(int\)](#) ,
[Exception.GetMessageFromNativeResources\(Exception.ExceptionMessageKind\)](#) ,
[Exception.GetMessageFromNativeResources\(Exception.ExceptionMessageKind, StringHandleOnStack\)](#) ,
[Exception.Data](#) , [Exception.InnerException](#) , [Exception.TargetSite](#) , [Exception.StackTrace](#) ,
[Exception.HelpLink](#) , [Exception.Source](#) , [Exception.IPForWatsonBuckets](#) ,
[Exception.WatsonBuckets](#) , [Exception.RemoteStackTrace](#) , [Exception.HResult](#) ,
[Exception.IsTransient](#) , [Exception.SerializeObjectState](#) , [object.Equals\(object\)](#) ,
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.FieldSetter\(string, string, object\)](#) ,
[object.FieldGetter\(string, string, ref object\)](#) , [object.GetFieldInfo\(string, string\)](#)

Constructors

RemoteCompilerError(string, location)

```
public RemoteCompilerError(string msg, location loc)
```

Parameters

msg [string](#)

loc location

Fields

msg

```
private string msg
```

Field Value

[string](#)

Methods

ToString()

```
public override string ToString()
```

Returns

string ↗

Class RemoteCompilerInternalError

Namespace: [PascalABCCompiler](#)

Assembly: Compiler.dll

```
public class RemoteCompilerInternalError : CompilerInternalError, ISerializable, _Exception
```

Inheritance

[object](#) ← [Exception](#) ← Error ← CompilerInternalError ← RemoteCompilerInternalError

Implements

[ISerializable](#), [Exception](#)

Inherited Members

CompilerInternalError.exception , CompilerInternalError.Module , CompilerInternalError.ToString() ,
Error.MustThrow , [Exception.s_EDILock](#) , [Exception.className](#) , [Exception.exceptionMethod](#) ,
[Exception.exceptionMethodString](#) , [Exception.message](#) , [Exception.data](#) ,
[Exception.innerException](#) , [Exception.helpURL](#) , [Exception.stackTrace](#) ,
[Exception.watsonBuckets](#) , [Exception.stackTraceString](#) , [Exception.remoteStackTraceString](#) ,
[Exception.remoteStackIndex](#) , [Exception.dynamicMethods](#) , [Exception.HResult](#) ,
[Exception.source](#) , [Exception.xptrs](#) , [Exception.xcode](#) , [Exception.ipForWatsonBuckets](#) ,
[Exception.safeSerializationManager](#) , [Exception.COMPlusExceptionCode](#) , [Exception.Init\(\)](#) ,
[Exception.IsImmutableAgileException](#)([Exception](#)) ,
[Exception.AddExceptionDataForRestrictedErrorInfo](#)([string](#), [string](#), [string](#), [object](#), [bool](#)) ,
[Exception.TryGetRestrictedLanguageErrorObject](#)([out object](#)) , [Exception.GetClassName\(\)](#) ,
[Exception.GetBaseException\(\)](#) , [Exception.GetMethodFromStackTrace](#)([object](#)) ,
[Exception.GetExceptionMethodFromStackTrace\(\)](#) , [Exception.GetTargetSiteInternal\(\)](#) ,
[Exception.GetStackTrace](#)([bool](#)) , [Exception.SetErrorCode](#)([int](#)) , [Exception.ToString](#)([bool](#), [bool](#)) ,
[Exception.GetExceptionMethodString\(\)](#) , [Exception.GetExceptionMethodFromString\(\)](#) ,
[Exception.GetObjectData](#)([SerializationInfo](#), [StreamingContext](#)) , [Exception.PrepForRemoting\(\)](#) ,
[Exception.OnDeserialized](#)([StreamingContext](#)) , [Exception.InternalPreserveStackTrace\(\)](#) ,
[Exception.PrepareForForeignExceptionRaise\(\)](#) ,
[Exception.GetStackTracesDeepCopy](#)([Exception](#), [out object](#), [out object](#)) ,
[Exception.SaveStackTracesFromDeepCopy](#)([Exception](#), [object](#), [object](#)) ,
[Exception.CopyStackTrace](#)([object](#)) , [Exception.CopyDynamicMethods](#)([object](#)) ,
[Exception.StripFileInfo](#)([string](#), [bool](#)) , [Exception.DeepCopyStackTrace](#)([object](#)) ,
[Exception.DeepCopyDynamicMethods](#)([object](#)) ,
[Exception.GetStackTracesDeepCopy](#)([out object](#), [out object](#)) ,
[Exception.RestoreExceptionDispatchInfo](#)([ExceptionDispatchInfo](#)) , [Exception.InternalToString\(\)](#) ,

[Exception.GetType\(\)](#) , [Exception.nlsTransient\(int\)](#) ,
[Exception.GetMessageFromNativeResources\(Exception.ExceptionMessageKind\)](#) ,
[Exception.GetMessageFromNativeResources\(Exception.ExceptionMessageKind, StringHandleOnStack\)](#) ,
[Exception.Message](#) , [Exception.Data](#) , [Exception.InnerException](#) , [Exception.TargetSite](#) ,
[Exception.StackTrace](#) , [Exception.HelpLink](#) , [Exception.Source](#) , [Exception.IPForWatsonBuckets](#) ,
[Exception.WatsonBuckets](#) , [Exception.RemoteStackTrace](#) , [Exception.HResult](#) ,
[Exception.IsTransient](#) , [Exception.SerializeObjectState](#) , [object.Equals\(object\)](#) ,
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.FieldSetter\(string, string, object\)](#) ,
[object.FieldGetter\(string, string, ref object\)](#) , [object.GetFieldInfo\(string, string\)](#)

Constructors

RemoteCompilerInternalError(string)

```
public RemoteCompilerInternalError(string msg)
```

Parameters

msg [string](#)

Class RemoteCompilerWarning

Namespace: [PascalABCCompiler](#)

Assembly: Compiler.dll

```
public class RemoteCompilerWarning : CompilerWarning, ISerializable, _Exception
```

Inheritance

[object](#) ← [Exception](#) ← Error ← LocatedError ← CompilerWarning ← RemoteCompilerWarning

Implements

[ISerializable](#), [Exception](#)

Inherited Members

CompilerWarning.Message , LocatedError.source_context , LocatedError.sourceLocation , LocatedError.fileName , LocatedError.SourceContext , LocatedError.FileName , Error.MustThrow , [Exception.s_EDILock](#) , [Exception.className](#) , [Exception.exceptionMethod](#) , [Exception.exceptionMethodString](#) , [Exception.message](#) , [Exception.data](#) , [Exception.innerException](#) , [Exception.helpURL](#) , [Exception.stackTrace](#) , [Exception.watsonBuckets](#) , [Exception.stackTraceString](#) , [Exception.remoteStackTraceString](#) , [Exception.remoteStackIndex](#) , [Exception.dynamicMethods](#) , [Exception.HResult](#) , [Exception.source](#) , [Exception.xptrs](#) , [Exception.xcode](#) , [Exception.ipForWatsonBuckets](#) , [Exception.safeSerializationManager](#) , [Exception.COMPlusExceptionCode](#) , [Exception.Init\(\)](#) , [Exception.IsImmutableAgileException](#)([Exception](#)) , [Exception.AddExceptionDataForRestrictedErrorInfo](#)([string](#), [string](#), [string](#), [object](#), [bool](#)) , [Exception.TryGetRestrictedLanguageErrorObject](#)([out object](#)) , [Exception.GetClassName\(\)](#) , [Exception.GetBaseException\(\)](#) , [ExceptionGetMethodFromStackTrace](#)([object](#)) , [Exception.GetExceptionMethodFromStackTrace](#)() , [Exception.GetTargetSiteInternal](#)() , [Exception.GetStackTrace](#)([bool](#)) , [Exception.SetErrorCode](#)([int](#)) , [Exception.ToString](#)([bool](#), [bool](#)) , [Exception.GetExceptionMethodString](#)() , [Exception.GetExceptionMethodFromString](#)() , [Exception.GetObjectData](#)([SerializationInfo](#), [StreamingContext](#)) , [Exception.PrepForRemoting](#)() , [Exception.OnDeserialized](#)([StreamingContext](#)) , [Exception.InternalPreserveStackTrace](#)() , [Exception.PrepareForForeignExceptionRaise](#)() , [Exception.GetStackTracesDeepCopy](#)([Exception](#), [out object](#), [out object](#)) , [Exception.SaveStackTracesFromDeepCopy](#)([Exception](#), [object](#), [object](#)) , [Exception.CopyStackTrace](#)([object](#)) , [Exception.CopyDynamicMethods](#)([object](#)) , [Exception.StripFileInfo](#)([string](#), [bool](#)) , [Exception.DeepCopyStackTrace](#)([object](#)) , [Exception.DeepCopyDynamicMethods](#)([object](#)) , [Exception.GetStackTracesDeepCopy](#)([out object](#), [out object](#)) ,

[Exception.RestoreExceptionDispatchInfo\(ExceptionDispatchInfo\)](#) , [Exception.InternalToString\(\)](#) ,
[Exception.GetType\(\)](#) , [Exception.nIsTransient\(int\)](#) ,
[Exception.GetMessageFromNativeResources\(Exception.ExceptionMessageKind\)](#) ,
[Exception.GetMessageFromNativeResources\(Exception.ExceptionMessageKind, StringHandleOnStack\)](#) ,
[Exception.Data](#) , [Exception.InnerException](#) , [Exception.TargetSite](#) , [Exception.StackTrace](#) ,
[Exception.HelpLink](#) , [Exception.Source](#) , [Exception.IPForWatsonBuckets](#) ,
[Exception.WatsonBuckets](#) , [Exception.RemoteStackTrace](#) , [Exception.HResult](#) ,
[Exception.IsTransient](#) , [Exception.SerializeObjectState](#) , [object.Equals\(object\)](#) ,
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.FieldSetter\(string, string, object\)](#) ,
[object.FieldGetter\(string, string, ref object\)](#) , [object.GetFieldInfo\(string, string\)](#)

Constructors

RemoteCompilerWarning(string, location)

```
public RemoteCompilerWarning(string msg, location loc)
```

Parameters

msg [string](#)

loc location

Fields

loc

```
private location loc
```

Field Value

location

msg

```
private string msg
```

Field Value

[string](#) ↗

Properties

SourceLocation

```
public override SourceLocation SourceLocation { get; }
```

Property Value

SourceLocation

Methods

ToString()

```
public override string ToString()
```

Returns

[string](#) ↗

Class ResourceInfo

Namespace: [PascalABCCompiler](#)

Assembly: Compiler.dll

```
public class ResourceInfo : IResourceInfo
```

Inheritance

[object](#) ← ResourceInfo

Implements

[IResourceInfo](#)

Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.FieldSetter\(string, string, object\)](#) ,
[object.FieldGetter\(string, string, ref object\)](#) , [object.GetFieldInfo\(string, string\)](#)

Constructors

ResourceInfo()

```
public ResourceInfo()
```

Fields

_name

```
private string _name
```

Field Value

[string](#)

Properties

Name

```
public string Name { get; }
```

Property Value

[string ↗](#)

name

```
public string name { get; set; }
```

Property Value

[string ↗](#)

Class SourceCodeFileInfo

Namespace: [PascalABCCompiler](#)

Assembly: Compiler.dll

```
public class SourceCodeFileInfo : IFileInfo
```

Inheritance

[object](#) ← SourceCodeFileInfo

Implements

[IFileInfo](#)

Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.FieldSetter\(string, string, object\)](#) ,
[object.FieldGetter\(string, string, ref object\)](#) , [object.GetFieldInfo\(string, string\)](#)

Constructors

SourceCodeFileInfo()

```
public SourceCodeFileInfo()
```

SourceCodeFileInfo(string, string)

```
public SourceCodeFileInfo(string name, string path)
```

Parameters

name [string](#)

path [string](#)

Fields

_name

```
private string _name
```

Field Value

[string](#) ↗

_path

```
private string _path
```

Field Value

[string](#) ↗

Properties

Name

```
public string Name { get; set; }
```

Property Value

[string](#) ↗

Path

```
public string Path { get; }
```

Property Value

[string](#) ↗

name

```
public string name { get; set; }
```

Property Value

[string](#) ↗

path

```
public string path { get; set; }
```

Property Value

[string](#) ↗

Class SupportedSourceFile

Namespace: [PascalABCCompiler](#)

Assembly: Compiler.dll

```
public class SupportedSourceFile
```

Inheritance

[object](#) ← SupportedSourceFile

Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) ,
[object.GetHashCode\(\)](#) , [object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) ,
[object.FieldSetter\(string, string, object\)](#) , [object.FieldGetter\(string, string, ref object\)](#) ,
[object.GetFieldInfo\(string, string\)](#)

Constructors

SupportedSourceFile(string[], string)

```
public SupportedSourceFile(string[] extensions, string lname)
```

Parameters

extensions [string](#)[]

lname [string](#)

Fields

extensions

```
private readonly string[] extensions
```

Field Value

[string](#)[]

languageName

`private readonly string languageName`

Field Value

[string](#)[]

Properties

Extensions

`public string[] Extensions { get; }`

Property Value

[string](#)[]

LanguageName

`public string LanguageName { get; }`

Property Value

[string](#)[]

Methods

Make(ILanguage)

`public static SupportedSourceFile Make(ILanguage language)`

Parameters

language [ILanguage](#)

Returns

[SupportedSourceFile](#)

ToString()

```
public override string ToString()
```

Returns

[string](#) ↗

Class TooOldProjectFileVersion

Namespace: [PascalABCCompiler](#)

Assembly: Compiler.dll

```
public class TooOldProjectFileVersion : Exception, ISerializable, _Exception
```

Inheritance

[object](#) ← [Exception](#) ← TooOldProjectFileVersion

Implements

[ISerializable](#), [Exception](#)

Inherited Members

[Exception.s_EDILock](#), [Exception.className](#), [Exception.exceptionMethod](#),
[Exception.exceptionMethodString](#), [Exception.message](#), [Exception.data](#),
[Exception.innerException](#), [Exception.helpURL](#), [Exception.stackTrace](#),
[Exception.watsonBuckets](#), [Exception.stackTraceString](#), [Exception.remoteStackTraceString](#),
[Exception.remoteStackIndex](#), [Exception.dynamicMethods](#), [Exception.HResult](#),
[Exception.source](#), [Exception.xptrs](#), [Exception.xcode](#), [Exception.ipForWatsonBuckets](#),
[Exception.safeSerializationManager](#), [Exception.COMPlusExceptionCode](#), [Exception.Init\(\)](#),
[Exception.IsImmutableAgileException\(Exception\)](#),
[Exception.AddExceptionDataForRestrictedErrorInfo\(string, string, string, object, bool\)](#),
[Exception.TryGetRestrictedLanguageErrorObject\(out object\)](#), [Exception.GetClassName\(\)](#),
[Exception.GetBaseException\(\)](#), [ExceptionGetMethodFromStackTrace\(object\)](#),
[Exception.GetExceptionMethodFromStackTrace\(\)](#), [Exception.GetTargetSiteInternal\(\)](#),
[Exception.GetStackTrace\(bool\)](#), [Exception.SetErrorCode\(int\)](#), [Exception.ToString\(\)](#),
[Exception.ToString\(bool, bool\)](#), [Exception.GetExceptionMethodString\(\)](#),
[Exception.GetExceptionMethodFromString\(\)](#),
[Exception.GetObjectData\(SerializationInfo, StreamingContext\)](#), [Exception.PrepForRemoting\(\)](#),
[Exception.OnDeserialized\(StreamingContext\)](#), [Exception.InternalPreserveStackTrace\(\)](#),
[Exception.PrepareForForeignExceptionRaise\(\)](#),
[Exception.GetStackTracesDeepCopy\(Exception, out object, out object\)](#),
[Exception.SaveStackTracesFromDeepCopy\(Exception, object, object\)](#),
[Exception.CopyStackTrace\(object\)](#), [Exception.CopyDynamicMethods\(object\)](#),
[Exception.StripFileInfo\(string, bool\)](#), [Exception.DeepCopyStackTrace\(object\)](#),
[Exception.DeepCopyDynamicMethods\(object\)](#),
[Exception.GetStackTracesDeepCopy\(out object, out object\)](#),
[Exception.RestoreExceptionDispatchInfo\(ExceptionDispatchInfo\)](#), [Exception.InternalToString\(\)](#),

[Exception.GetType\(\)](#) , [Exception.nlsTransient\(int\)](#) ,
[Exception.GetMessageFromNativeResources\(Exception.ExceptionMessageKind\)](#) ,
[Exception.GetMessageFromNativeResources\(Exception.ExceptionMessageKind, StringHandleOnStack\)](#) ,
[Exception.Message](#) , [Exception.Data](#) , [Exception.InnerException](#) , [Exception.TargetSite](#) ,
[Exception.StackTrace](#) , [Exception.HelpLink](#) , [Exception.Source](#) , [Exception.IPForWatsonBuckets](#) ,
[Exception.WatsonBuckets](#) , [Exception.RemoteStackTrace](#) , [Exception.HResult](#) ,
[Exception.IsTransient](#) , [Exception.SerializeObjectState](#) , [object.Equals\(object\)](#) ,
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.FieldSetter\(string, string, object\)](#) ,
[object.FieldGetter\(string, string, ref object\)](#) , [object.GetFieldInfo\(string, string\)](#)

Constructors

TooOldProjectFileVersion()

```
public TooOldProjectFileVersion()
```

Enum UnitState

Namespace: [PascalABCCompiler](#)

Assembly: Compiler.dll

```
public enum UnitState
```

Fields

BeginCompilation = 0

Compiled = 2

InterfaceCompiled = 1

Namespace PascalABCCompiler.Errors

Classes

[AppTypeDIIIsNotAllowedForLibraries](#)

Бросается при обнаружении директивы {\$apptype dll} не в библиотеке

[AssemblyNotFound](#)

Бросается, если файл сборки не найден

[AssemblyReadingError](#)

Бросается при невозможности чтения сборки

[CompilerThrownError](#)

Базовый класс для ошибок, бросаемых компилятором

[CycleUnitReference](#)

Бросается в случае обнаружения циклической зависимости модулей

[DisableStandardUnitsDirectiveDisallowedInUsedUnits](#)

Бросается при обнаружении директивы {\$DisableStandardUnits} в подключенном модуле

[DuplicateDirective](#)

Бросается при нахождении дубликатов директив, не поддерживающих многократное использование в рамках некоторого контекста

[DuplicateUsesUnit](#)

Бросается при нахождении дубликатов в секции uses

[FileNotFoundException](#)

Бросается при отсутствии некоторого файла по некоторому пути

[IncludeNamespacelnUnitError](#)

Бросается при подключении явного пространства имен в модуле

[InvalidAssemblyPathError](#)

Бросается при попытке обработки неправильного пути к сборке

[InvalidPathError](#)

Бросается при попытке обработки неправильного пути к файлу

[MainResourceNotAllowed](#)

Бросается в случае некорректного использования директивы {\$mainresource ...}

[NamespaceCannotHaveInSection](#)

Бросается в случае использования uses in в пространстве имен

[NamespaceModuleExpected](#)

Бросается, если встреченено не явное пространство имен

[NamespacesCanBeCompiledOnlyInProjects](#)

Legacy, бросается, если unitModule.unitName.HeaderKeyword ==
SyntaxTree.UnitHeaderKeyword.Namespace

[ProgramModuleExpected](#)

Бросается, если встречена не основная программа (там где она должна быть)

[ReadPCUError](#)

Бросается в случае невозможности чтения рсц

[ResourceFileNotFound](#)

Бросается при неудаче в нахождении файла ресурсов

[SourceFileNotFoundException](#)

Бросается при невозможности найти исходник по некоторому пути

[UnauthorizedAccessToFile](#)

Бросается при попытке файловой операции с недостаточными правами

[UnitModuleExpected](#)

Бросается, если встреченен не модуль (там где он должна быть)

[UnitModuleExpectedLibraryFound](#)

Бросается при компиляции библиотеки не первой

[UnitNotFoundException](#)

Бросается, если модуль (unit) не найден по некоторому пути

[UnsupportedOutputFileType](#)

Бросается, если пользователь указывает неподдерживаемый тип выходного файла (например, в директиве {\$apptype ...})

[UnsupportedTargetFramework](#)

Бросается, если пользователь указывает неподдерживаемый целевой framework

[UnsupportedTargetPlatform](#)

Бросается, если пользователь указывает неподдерживаемую целевую платформу

[UsesInWrongName](#)

Бросается при некорректном пути в uses in

Class AppTypeDllIsAllowedOnlyForLibraries

Namespace: [PascalABCCompiler.Errors](#)

Assembly: Compiler.dll

Бросается при обнаружении директивы {\$apptype dll} не в библиотеке

```
public class AppTypeDllIsAllowedOnlyForLibraries : CompilerThrownError,  
ISerializable, _Exception
```

Inheritance

[object](#) ← [Exception](#) ← Error ← LocatedError ← [CompilerThrownError](#) ←
AppTypeDllIsAllowedOnlyForLibraries

Implements

[ISerializable](#), [Exception](#)

Inherited Members

[CompilerThrownError.ToString\(\)](#), LocatedError.source_context, LocatedError.sourceLocation, LocatedError.fileName, LocatedError.SourceLocation, LocatedError.SourceContext, LocatedError.FileName, Error.MustThrow, [Exception.s_EDILock](#), [Exception.className](#), [Exception.exceptionMethod](#), [Exception.exceptionMethodString](#), [Exception.message](#), [Exception.data](#), [Exception.innerException](#), [Exception.helpURL](#), [Exception.stackTrace](#), [Exception.watsonBuckets](#), [Exception.stackTraceString](#), [Exception.remoteStackTraceString](#), [Exception.remoteStackIndex](#), [Exception.dynamicMethods](#), [Exception.HResult](#), [Exception.source](#), [Exception.xptrs](#), [Exception.xcode](#), [Exception.ipForWatsonBuckets](#), [Exception.safeSerializationManager](#), [Exception.COMPlusExceptionCode](#), [Exception.Init\(\)](#), [Exception.IsImmutableAgileException](#)([Exception](#)), [Exception.AddExceptionDataForRestrictedErrorInfo](#)(string, string, string, object, bool), [Exception.TryGetRestrictedLanguageErrorObject](#)(out object), [Exception.GetClassName\(\)](#), [Exception.GetBaseException\(\)](#), [ExceptionGetMethodFromStackTrace](#)(object), [Exception.GetExceptionMethodFromStackTrace](#)(), [Exception.GetTargetSiteInternal](#)(), [Exception.GetStackTrace](#)(bool), [Exception.SetErrorCode](#)(int), [Exception.ToString](#)(bool, bool), [Exception.GetExceptionMethodString](#)(), [Exception.GetExceptionMethodFromString](#)(), [Exception.GetObjectData](#)(SerializationInfo, StreamingContext), [Exception.PrepForRemoting](#)(), [Exception.OnDeserialized](#)(StreamingContext), [Exception.InternalPreserveStackTrace](#)(), [Exception.PrepareForForeignExceptionRaise](#)(), [Exception.GetStackTracesDeepCopy](#)([Exception](#), out object, out object), [Exception.SaveStackTracesFromDeepCopy](#)([Exception](#), object, object), [Exception.CopyStackTrace](#)(object), [Exception.CopyDynamicMethods](#)(object),

[Exception.StripFileInfo\(string, bool\)](#) , [Exception.DeepCopyStackTrace\(object\)](#) ,
[Exception.DeepCopyDynamicMethods\(object\)](#) ,
[Exception.GetStackTracesDeepCopy\(out object, out object\)](#) ,
[Exception.RestoreExceptionDispatchInfo\(ExceptionDispatchInfo\)](#) , [Exception.InternalToString\(\)](#) ,
[Exception.GetType\(\)](#) , [Exception.nlsTransient\(int\)](#) ,
[Exception.GetMessageFromNativeResources\(Exception.ExceptionMessageKind\)](#) ,
[Exception.GetMessageFromNativeResources\(Exception.ExceptionMessageKind, StringHandleOnStack\)](#) ,
[Exception.Message](#) , [Exception.Data](#) , [Exception.InnerException](#) , [Exception.TargetSite](#) ,
[Exception.StackTrace](#) , [Exception.HelpLink](#) , [Exception.Source](#) , [Exception.IPForWatsonBuckets](#) ,
[Exception.WatsonBuckets](#) , [Exception.RemoteStackTrace](#) , [Exception.HResult](#) ,
[Exception.IsTransient](#) , [Exception.SerializeObjectState](#) , [object.Equals\(object\)](#) ,
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.FieldSetter\(string, string, object\)](#) ,
[object.FieldGetter\(string, string, ref object\)](#) , [object.GetFieldInfo\(string, string\)](#)

Constructors

AppTypeDllIsAllowedOnlyForLibraries(string, SourceContext)

```
public AppTypeDllIsAllowedOnlyForLibraries(string FileName, SourceContext sc)
```

Parameters

FileName [string](#)

sc SourceContext

Class AssemblyNotFound

Namespace: [PascalABCCompiler.Errors](#)

Assembly: Compiler.dll

Бросается, если файл сборки не найден

```
public class AssemblyNotFound : CompilerThrownError, ISerializable, _Exception
```

Inheritance

[object](#) ← [Exception](#) ← Error ← LocatedError ← [CompilerThrownError](#) ← AssemblyNotFound

Implements

[ISerializable](#), [Exception](#)

Inherited Members

[CompilerThrownError.ToString\(\)](#), LocatedError.source_context, LocatedError.sourceLocation, LocatedError.fileName, LocatedError.SourceLocation, LocatedError.SourceContext, LocatedError.FileName, Error.MustThrow, [Exception.s_EDILock](#), [Exception.className](#), [Exception.exceptionMethod](#), [Exception.exceptionMethodString](#), [Exception.message](#), [Exception.data](#), [Exception.innerException](#), [Exception.helpURL](#), [Exception.stackTrace](#), [Exception.watsonBuckets](#), [Exception.stackTraceString](#), [Exception.remoteStackTraceString](#), [Exception.remoteStackIndex](#), [Exception.dynamicMethods](#), [Exception.HResult](#), [Exception.source](#), [Exception.xptrs](#), [Exception.xcode](#), [Exception.ipForWatsonBuckets](#), [Exception.safeSerializationManager](#), [Exception.COMPlusExceptionCode](#), [Exception.Init\(\)](#), [Exception.IsImmutableAgileException\(Exception\)](#), [Exception.AddExceptionDataForRestrictedErrorInfo\(string, string, string, object, bool\)](#), [Exception.TryGetRestrictedLanguageErrorObject\(out object\)](#), [Exception.GetClassName\(\)](#), [Exception.GetBaseException\(\)](#), [Exception.GetMethodFromStackTrace\(object\)](#), [Exception.GetExceptionMethodFromStackTrace\(\)](#), [Exception.GetTargetSiteInternal\(\)](#), [Exception.GetStackTrace\(bool\)](#), [Exception.SetErrorCode\(int\)](#), [Exception.ToString\(bool, bool\)](#), [Exception.GetExceptionMethodString\(\)](#), [Exception.GetExceptionMethodFromString\(\)](#), [Exception.GetObjectData\(SerializationInfo, StreamingContext\)](#), [Exception.PrepForRemoting\(\)](#), [Exception.OnDeserialized\(StreamingContext\)](#), [Exception.InternalPreserveStackTrace\(\)](#), [Exception.PrepareForForeignExceptionRaise\(\)](#), [Exception.GetStackTracesDeepCopy\(Exception, out object, out object\)](#), [Exception.SaveStackTracesFromDeepCopy\(Exception, object, object\)](#), [Exception.CopyStackTrace\(object\)](#), [Exception.CopyDynamicMethods\(object\)](#), [Exception.StripFileInfo\(string, bool\)](#), [Exception.DeepCopyStackTrace\(object\)](#),

[Exception.DeepCopyDynamicMethods\(object\)](#) ,
[Exception.GetStackTracesDeepCopy\(out object, out object\)](#) ,
[Exception.RestoreExceptionDispatchInfo\(ExceptionDispatchInfo\)](#) , [Exception.InternalToString\(\)](#) ,
[Exception.GetType\(\)](#) , [Exception.nIsTransient\(int\)](#) ,
[Exception.GetMessageFromNativeResources\(Exception.ExceptionMessageKind\)](#) ,
[Exception.GetMessageFromNativeResources\(Exception.ExceptionMessageKind, StringHandleOnStack\)](#) ,
[Exception.Message](#) , [Exception.Data](#) , [Exception.InnerException](#) , [Exception.TargetSite](#) ,
[Exception.StackTrace](#) , [Exception.HelpLink](#) , [Exception.Source](#) , [Exception.IPForWatsonBuckets](#) ,
[Exception.WatsonBuckets](#) , [Exception.RemoteStackTrace](#) , [Exception.HResult](#) ,
[Exception.IsTransient](#) , [Exception.SerializeObjectState](#) , [object.Equals\(object\)](#) ,
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.FieldSetter\(string, string, object\)](#) ,
[object.FieldGetter\(string, string, ref object\)](#) , [object.GetFieldInfo\(string, string\)](#)

Constructors

AssemblyNotFound(string, string, SourceContext)

```
public AssemblyNotFound(string FileName, string AssemblyFileName, SourceContext sc)
```

Parameters

FileName [string](#)

AssemblyFileName [string](#)

sc SourceContext

Fields

AssemblyFileName

```
public string AssemblyFileName
```

Field Value

[string](#)

Class AssemblyReadingError

Namespace: [PascalABCCompiler.Errors](#)

Assembly: Compiler.dll

Бросается при невозможности чтения сборки

```
public class AssemblyReadingError : CompilerThrownError, ISerializable, _Exception
```

Inheritance

[object](#) ← [Exception](#) ← Error ← LocatedError ← [CompilerThrownError](#) ← AssemblyReadingError

Implements

[ISerializable](#), [Exception](#)

Inherited Members

[CompilerThrownError.ToString\(\)](#), LocatedError.source_context, LocatedError.sourceLocation, LocatedError.fileName, LocatedError.SourceLocation, LocatedError.SourceContext, LocatedError.FileName, Error.MustThrow, [Exception.s_EDILock](#), [Exception.className](#), [Exception.exceptionMethod](#), [Exception.exceptionMethodString](#), [Exception.message](#), [Exception.data](#), [Exception.innerException](#), [Exception.helpURL](#), [Exception.stackTrace](#), [Exception.watsonBuckets](#), [Exception.stackTraceString](#), [Exception.remoteStackTraceString](#), [Exception.remoteStackIndex](#), [Exception.dynamicMethods](#), [Exception.HResult](#), [Exception.source](#), [Exception.xptrs](#), [Exception.xcode](#), [Exception.ipForWatsonBuckets](#), [Exception.safeSerializationManager](#), [Exception.COMPlusExceptionCode](#), [Exception.Init\(\)](#), [Exception.IsImmutableAgileException\(Exception\)](#), [Exception.AddExceptionDataForRestrictedErrorInfo\(string, string, string, object, bool\)](#), [Exception.TryGetRestrictedLanguageErrorObject\(out object\)](#), [Exception.GetClassName\(\)](#), [Exception.GetBaseException\(\)](#), [Exception.GetMethodFromStackTrace\(object\)](#), [Exception.GetExceptionMethodFromStackTrace\(\)](#), [Exception.GetTargetSiteInternal\(\)](#), [Exception.GetStackTrace\(bool\)](#), [Exception.SetErrorCode\(int\)](#), [Exception.ToString\(bool, bool\)](#), [Exception.GetExceptionMethodString\(\)](#), [Exception.GetExceptionMethodFromString\(\)](#), [Exception.GetObjectData\(SerializationInfo, StreamingContext\)](#), [Exception.PrepForRemoting\(\)](#), [Exception.OnDeserialized\(StreamingContext\)](#), [Exception.InternalPreserveStackTrace\(\)](#), [Exception.PrepareForForeignExceptionRaise\(\)](#), [Exception.GetStackTracesDeepCopy\(Exception, out object, out object\)](#), [Exception.SaveStackTracesFromDeepCopy\(Exception, object, object\)](#), [Exception.CopyStackTrace\(object\)](#), [Exception.CopyDynamicMethods\(object\)](#), [Exception.StripFileInfo\(string, bool\)](#), [Exception.DeepCopyStackTrace\(object\)](#),

[Exception.DeepCopyDynamicMethods\(object\)](#) ,
[Exception.GetStackTracesDeepCopy\(out object, out object\)](#) ,
[Exception.RestoreExceptionDispatchInfo\(ExceptionDispatchInfo\)](#) , [Exception.InternalToString\(\)](#) ,
[Exception.GetType\(\)](#) , [Exception.nIsTransient\(int\)](#) ,
[Exception.GetMessageFromNativeResources\(Exception.ExceptionMessageKind\)](#) ,
[Exception.GetMessageFromNativeResources\(Exception.ExceptionMessageKind, StringHandleOnStack\)](#) ,
[Exception.Message](#) , [Exception.Data](#) , [Exception.InnerException](#) , [Exception.TargetSite](#) ,
[Exception.StackTrace](#) , [Exception.HelpLink](#) , [Exception.Source](#) , [Exception.IPForWatsonBuckets](#) ,
[Exception.WatsonBuckets](#) , [Exception.RemoteStackTrace](#) , [Exception.HResult](#) ,
[Exception.IsTransient](#) , [Exception.SerializeObjectState](#) , [object.Equals\(object\)](#) ,
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.FieldSetter\(string, string, object\)](#) ,
[object.FieldGetter\(string, string, ref object\)](#) , [object.GetFieldInfo\(string, string\)](#)

Constructors

AssemblyReadingError(string, string, SourceContext)

```
public AssemblyReadingError(string FileName, string AssemblyFileName, SourceContext sc)
```

Parameters

FileName [string](#)

AssemblyFileName [string](#)

sc SourceContext

Fields

AssemblyFileName

```
public string AssemblyFileName
```

Field Value

[string](#)

Class CompilerThrownError

Namespace: [PascalABCCompiler.Errors](#)

Assembly: Compiler.dll

Базовый класс для ошибок, бросаемых компилятором

```
public class CompilerThrownError : LocatedError, ISerializable, _Exception
```

Inheritance

[object](#) ← [Exception](#) ← Error ← LocatedError ← CompilerThrownError

Implements

[ISerializable](#), [Exception](#)

Derived

[AppTypeDlIsAllowedOnlyForLibraries](#), [AssemblyNotFound](#), [AssemblyReadingError](#), [CycleUnitReference](#),
[DisableStandardUnitsDirectiveDisallowedInUsedUnits](#), [DuplicateDirective](#), [DuplicateUsesUnit](#),
[FileNotFoundException](#), [IncludeNamespacelnUnitError](#), [InvalidAssemblyPathError](#), [InvalidPathError](#),
[MainResourceNotAllowed](#), [NamespaceCannotHaveInSection](#), [NamespaceModuleExpected](#),
[NamespacesCanBeCompiledOnlyInProjects](#), [ProgramModuleExpected](#), [ReadPCUError](#),
[ResourceFileNotFound](#), [SourceFileNotFound](#), [UnauthorizedAccessToFile](#), [UnitModuleExpected](#),
[UnitModuleExpectedLibraryFound](#), [UnitNotFound](#), [UnsupportedOutputFileType](#),
[UnsupportedTargetFramework](#), [UnsupportedTargetPlatform](#), [UsesInWrongName](#)

Inherited Members

LocatedError.source_context, LocatedError.sourceLocation, LocatedError.fileName,
LocatedError.SourceLocation, LocatedError.SourceContext, LocatedError.FileName, Error.MustThrow,
[Exception.s_EDILock](#), [Exception.className](#), [Exception.exceptionMethod](#),
[Exception.exceptionMethodString](#), [Exception.message](#), [Exception.data](#),
[Exception.innerException](#), [Exception.helpURL](#), [Exception.stackTrace](#),
[Exception.watsonBuckets](#), [Exception.stackTraceString](#), [Exception.remoteStackTraceString](#),
[Exception.remoteStackIndex](#), [Exception.dynamicMethods](#), [Exception.HResult](#),
[Exception.source](#), [Exception.xptrs](#), [Exception.xcode](#), [Exception.ipForWatsonBuckets](#),
[Exception.safeSerializationManager](#), [Exception.COMPlusExceptionCode](#), [Exception.Init\(\)](#),
[Exception.IsImmutableAgileException](#)([Exception](#)),
[Exception.AddExceptionDataForRestrictedErrorInfo](#)([string](#), [string](#), [string](#), [object](#), [bool](#)),
[Exception.TryGetRestrictedLanguageErrorObject](#)([out object](#)), [Exception.GetClassName\(\)](#),
[Exception.GetBaseException\(\)](#), [Exception.GetMethodFromStackTrace](#)([object](#)),
[Exception.GetExceptionMethodFromStackTrace\(\)](#), [Exception.GetTargetSiteInternal\(\)](#),

[Exception.GetStackTrace\(bool\)](#) , [Exception.SetErrorCode\(int\)](#) , [Exception.ToString\(bool, bool\)](#) ,
[Exception.GetExceptionMethodString\(\)](#) , [Exception.GetExceptionMethodFromString\(\)](#) ,
[Exception.GetObjectData\(SerializationInfo, StreamingContext\)](#) , [Exception.PrepForRemoting\(\)](#) ,
[Exception.OnDeserialized\(StreamingContext\)](#) , [Exception.InternalPreserveStackTrace\(\)](#) ,
[Exception.PrepareForForeignExceptionRaise\(\)](#) ,
[Exception.GetStackTracesDeepCopy\(Exception, out object, out object\)](#) ,
[Exception.SaveStackTracesFromDeepCopy\(Exception, object, object\)](#) ,
[Exception.CopyStackTrace\(object\)](#) , [Exception.CopyDynamicMethods\(object\)](#) ,
[Exception.StripFileInfo\(string, bool\)](#) , [Exception.DeepCopyStackTrace\(object\)](#) ,
[Exception.DeepCopyDynamicMethods\(object\)](#) ,
[Exception.GetStackTracesDeepCopy\(out object, out object\)](#) ,
[Exception.RestoreExceptionDispatchInfo\(ExceptionDispatchInfo\)](#) , [Exception.InternalToString\(\)](#) ,
[Exception.GetType\(\)](#) , [Exception.nIsTransient\(int\)](#) ,
[Exception.GetMessageFromNativeResources\(Exception.ExceptionMessageKind\)](#) ,
[Exception.GetMessageFromNativeResources\(Exception.ExceptionMessageKind, StringHandleOnStack\)](#) ,
[Exception.Message](#) , [Exception.Data](#) , [Exception.InnerException](#) , [Exception.TargetSite](#) ,
[Exception.StackTrace](#) , [Exception.HelpLink](#) , [Exception.Source](#) , [Exception.IPForWatsonBuckets](#) ,
[Exception.WatsonBuckets](#) , [Exception.RemoteStackTrace](#) , [Exception.HResult](#) ,
[Exception.IsTransient](#) , [Exception.SerializeObjectState](#) , [object.Equals\(object\)](#) ,
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.FieldSetter\(string, string, object\)](#) ,
[object.FieldGetter\(string, string, ref object\)](#) , [object.GetFieldInfo\(string, string\)](#)

Constructors

CompilerThrownError(string)

```
public CompilerThrownError(string message)
```

Parameters

message [string](#)

CompilerThrownError(string, string)

```
public CompilerThrownError(string message, string FileName)
```

Parameters

message [string](#)

FileName [string](#)

Methods

ToString()

```
public override string ToString()
```

Returns

[string](#)

Class CycleUnitReference

Namespace: [PascalABCCompiler.Errors](#)

Assembly: Compiler.dll

Бросается в случае обнаружения циклической зависимости модулей

```
public class CycleUnitReference : CompilerThrownError, ISerializable, _Exception
```

Inheritance

[object](#) ← [Exception](#) ← Error ← LocatedError ← [CompilerThrownError](#) ← CycleUnitReference

Implements

[ISerializable](#), [Exception](#)

Inherited Members

[CompilerThrownError.ToString\(\)](#), LocatedError.source_context, LocatedError.sourceLocation, LocatedError.fileName, LocatedError.SourceLocation, LocatedError.SourceContext, LocatedError.FileName, Error.MustThrow, [Exception.s_EDILock](#), [Exception.className](#), [Exception.exceptionMethod](#), [Exception.exceptionMethodString](#), [Exception.message](#), [Exception.data](#), [Exception.innerException](#), [Exception.helpURL](#), [Exception.stackTrace](#), [Exception.watsonBuckets](#), [Exception.stackTraceString](#), [Exception.remoteStackTraceString](#), [Exception.remoteStackIndex](#), [Exception.dynamicMethods](#), [Exception.HResult](#), [Exception.source](#), [Exception.xptrs](#), [Exception.xcode](#), [Exception.ipForWatsonBuckets](#), [Exception.safeSerializationManager](#), [Exception.COMPlusExceptionCode](#), [Exception.Init\(\)](#), [Exception.IsImmutableAgileException\(Exception\)](#), [Exception.AddExceptionDataForRestrictedErrorInfo\(string, string, string, object, bool\)](#), [Exception.TryGetRestrictedLanguageErrorObject\(out object\)](#), [Exception.GetClassName\(\)](#), [Exception.GetBaseException\(\)](#), [Exception.GetMethodFromStackTrace\(object\)](#), [Exception.GetExceptionMethodFromStackTrace\(\)](#), [Exception.GetTargetSiteInternal\(\)](#), [Exception.GetStackTrace\(bool\)](#), [Exception.SetErrorCode\(int\)](#), [Exception.ToString\(bool, bool\)](#), [Exception.GetExceptionMethodString\(\)](#), [Exception.GetExceptionMethodFromString\(\)](#), [Exception.GetObjectData\(SerializationInfo, StreamingContext\)](#), [Exception.PrepForRemoting\(\)](#), [Exception.OnDeserialized\(StreamingContext\)](#), [Exception.InternalPreserveStackTrace\(\)](#), [Exception.PrepareForForeignExceptionRaise\(\)](#), [Exception.GetStackTracesDeepCopy\(Exception, out object, out object\)](#), [Exception.SaveStackTracesFromDeepCopy\(Exception, object, object\)](#), [Exception.CopyStackTrace\(object\)](#), [Exception.CopyDynamicMethods\(object\)](#), [Exception.StripFileInfo\(string, bool\)](#), [Exception.DeepCopyStackTrace\(object\)](#),

[Exception.DeepCopyDynamicMethods\(object\)](#) ,
[Exception.GetStackTracesDeepCopy\(out object, out object\)](#) ,
[Exception.RestoreExceptionDispatchInfo\(ExceptionDispatchInfo\)](#) , [Exception.InternalToString\(\)](#) ,
[Exception.GetType\(\)](#) , [Exception.nIsTransient\(int\)](#) ,
[Exception.GetMessageFromNativeResources\(Exception.ExceptionMessageKind\)](#) ,
[Exception.GetMessageFromNativeResources\(Exception.ExceptionMessageKind, StringHandleOnStack\)](#) ,
[Exception.Message](#) , [Exception.Data](#) , [Exception.InnerException](#) , [Exception.TargetSite](#) ,
[Exception.StackTrace](#) , [Exception.HelpLink](#) , [Exception.Source](#) , [Exception.IPForWatsonBuckets](#) ,
[Exception.WatsonBuckets](#) , [Exception.RemoteStackTrace](#) , [Exception.HResult](#) ,
[Exception.IsTransient](#) , [Exception.SerializeObjectState](#) , [object.Equals\(object\)](#) ,
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.FieldSetter\(string, string, object\)](#) ,
[object.FieldGetter\(string, string, ref object\)](#) , [object.GetFieldInfo\(string, string\)](#)

Constructors

CycleUnitReference(string, unit_or_namespace)

```
public CycleUnitReference(string FileName, unit_or_namespace SyntaxUsesUnit)
```

Parameters

FileName [string](#)

SyntaxUsesUnit [unit_or_namespace](#)

Class DisableStandardUnitsDirectiveDisallowedInUsedUnits

Namespace: [PascalABCCompiler.Errors](#)

Assembly: Compiler.dll

Бросается при обнаружении директивы {\$DisableStandardUnits} в подключенном модуле

```
public class DisableStandardUnitsDirectiveDisallowedInUsedUnits : CompilerThrownError,  
ISerializable, _Exception
```

Inheritance

[Object](#) ← [Exception](#) ← Error ← LocatedError ← [CompilerThrownError](#) ←

DisableStandardUnitsDirectiveDisallowedInUsedUnits

Implements

[ISerializable](#), [Exception](#)

Inherited Members

[CompilerThrownError.ToString\(\)](#), LocatedError.source_context, LocatedError.sourceLocation, LocatedError.fileName, LocatedError.SourceLocation, LocatedError.SourceContext, LocatedError.FileName, Error.MustThrow, [Exception.s_EDILock](#), [Exception.className](#), [Exception.exceptionMethod](#), [Exception.exceptionMethodString](#), [Exception.message](#), [Exception.data](#), [Exception.innerException](#), [Exception.helpURL](#), [Exception.stackTrace](#), [Exception.watsonBuckets](#), [Exception.stackTraceString](#), [Exception.remoteStackTraceString](#), [Exception.remoteStackIndex](#), [Exception.dynamicMethods](#), [Exception.HResult](#), [Exception.source](#), [Exception.xptrs](#), [Exception.xcode](#), [Exception.ipForWatsonBuckets](#), [Exception.safeSerializationManager](#), [Exception.COMPlusExceptionCode](#), [Exception.Init\(\)](#), [Exception.IsImmutableAgileException](#)([Exception](#)), [Exception.AddExceptionDataForRestrictedErrorInfo](#)(string, string, string, object, bool), [Exception.TryGetRestrictedLanguageErrorObject](#)(out object), [Exception.GetClassName\(\)](#), [Exception.GetBaseException\(\)](#), [ExceptionGetMethodFromStackTrace](#)(object), [Exception.GetExceptionMethodFromStackTrace](#)(), [Exception.GetTargetSiteInternal](#)(), [Exception.GetStackTrace](#)(bool), [Exception.SetErrorCode](#)(int), [Exception.ToString](#)(bool, bool), [Exception.GetExceptionMethodString](#)(), [Exception.GetExceptionMethodFromString](#)(), [Exception.GetObjectData](#)(SerializationInfo, StreamingContext), [Exception.PrepForRemoting](#)(), [Exception.OnDeserialized](#)(StreamingContext), [Exception.InternalPreserveStackTrace](#)(), [Exception.PrepareForForeignExceptionRaise](#)(), [Exception.GetStackTracesDeepCopy](#)([Exception](#), out object, out object),

[Exception.SaveStackTracesFromDeepCopy\(Exception, object, object\)](#) ,
[Exception.CopyStackTrace\(object\)](#) , [Exception.CopyDynamicMethods\(object\)](#) ,
[Exception.StripFileInfo\(string, bool\)](#) , [Exception.DeepCopyStackTrace\(object\)](#) ,
[Exception.DeepCopyDynamicMethods\(object\)](#) ,
[Exception.GetStackTracesDeepCopy\(out object, out object\)](#) ,
[Exception.RestoreExceptionDispatchInfo\(ExceptionDispatchInfo\)](#) , [Exception.InternalToString\(\)](#) ,
[Exception.GetType\(\)](#) , [Exception.nIsTransient\(int\)](#) ,
[Exception.GetMessageFromNativeResources\(Exception.ExceptionMessageKind\)](#) ,
[Exception.GetMessageFromNativeResources\(Exception.ExceptionMessageKind, StringHandleOnStack\)](#) ,
[Exception.Message](#) , [Exception.Data](#) , [Exception.InnerException](#) , [Exception.TargetSite](#) ,
[Exception.StackTrace](#) , [Exception.HelpLink](#) , [Exception.Source](#) , [Exception.IPForWatsonBuckets](#) ,
[Exception.WatsonBuckets](#) , [Exception.RemoteStackTrace](#) , [Exception.HResult](#) ,
[Exception.IsTransient](#) , [Exception.SerializeObjectState](#) , [object.Equals\(object\)](#) ,
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.FieldSetter\(string, string, object\)](#) ,
[object.FieldGetter\(string, string, ref object\)](#) , [object.GetFieldInfo\(string, string\)](#)

Constructors

DisableStandardUnitsDirectiveDisallowedInUsedUnits(string, SourceContext)

```
public DisableStandardUnitsDirectiveDisallowedInUsedUnits(string FileName, SourceContext sc)
```

Parameters

FileName [string](#)

sc [SourceContext](#)

Class DuplicateDirective

Namespace: [PascalABCCompiler.Errors](#)

Assembly: Compiler.dll

Бросается при нахождении дубликатов директив, не поддерживающих многократное использование в рамках некоторого контекста

```
public class DuplicateDirective : CompilerThrownError, ISerializable, _Exception
```

Inheritance

[object](#) ← [Exception](#) ← Error ← LocatedError ← [CompilerThrownError](#) ← DuplicateDirective

Implements

[ISerializable](#), [Exception](#)

Inherited Members

[CompilerThrownError.ToString\(\)](#), LocatedError.source_context, LocatedError.sourceLocation, LocatedError.fileName, LocatedError.SourceLocation, LocatedError.SourceContext, LocatedError.FileName, Error.MustThrow, [Exception.s_EDILock](#), [Exception.className](#), [Exception.exceptionMethod](#), [Exception.exceptionMethodString](#), [Exception.message](#), [Exception.data](#), [Exception.innerException](#), [Exception.helpURL](#), [Exception.stackTrace](#), [Exception.watsonBuckets](#), [Exception.stackTraceString](#), [Exception.remoteStackTraceString](#), [Exception.remoteStackIndex](#), [Exception.dynamicMethods](#), [Exception.HResult](#), [Exception.source](#), [Exception.xptrs](#), [Exception.xcode](#), [Exception.ipForWatsonBuckets](#), [Exception.safeSerializationManager](#), [Exception.COMPlusExceptionCode](#), [Exception.Init\(\)](#), [Exception.IsImmutableAgileException\(Exception\)](#), [Exception.AddExceptionDataForRestrictedErrorInfo\(string, string, string, object, bool\)](#), [Exception.TryGetRestrictedLanguageErrorObject\(out object\)](#), [Exception.GetClassName\(\)](#), [Exception.GetBaseException\(\)](#), [ExceptionGetMethodFromStackTrace\(object\)](#), [Exception.GetExceptionMethodFromStackTrace\(\)](#), [Exception.GetTargetSiteInternal\(\)](#), [Exception.GetStackTrace\(bool\)](#), [Exception.SetErrorCode\(int\)](#), [Exception.ToString\(bool, bool\)](#), [Exception.GetExceptionMethodString\(\)](#), [Exception.GetExceptionMethodFromString\(\)](#), [Exception.GetObjectData\(SerializationInfo, StreamingContext\)](#), [Exception.PrepForRemoting\(\)](#), [Exception.OnDeserialized\(StreamingContext\)](#), [Exception.InternalPreserveStackTrace\(\)](#), [Exception.PrepareForForeignExceptionRaise\(\)](#), [Exception.GetStackTracesDeepCopy\(Exception, out object, out object\)](#), [Exception.SaveStackTracesFromDeepCopy\(Exception, object, object\)](#), [Exception.CopyStackTrace\(object\)](#), [Exception.CopyDynamicMethods\(object\)](#),

[Exception.StripFileInfo\(string, bool\)](#) , [Exception.DeepCopyStackTrace\(object\)](#) ,
[Exception.DeepCopyDynamicMethods\(object\)](#) ,
[Exception.GetStackTracesDeepCopy\(out object, out object\)](#) ,
[Exception.RestoreExceptionDispatchInfo\(ExceptionDispatchInfo\)](#) , [Exception.InternalToString\(\)](#) ,
[Exception.GetType\(\)](#) , [Exception.nlsTransient\(int\)](#) ,
[Exception.GetMessageFromNativeResources\(Exception.ExceptionMessageKind\)](#) ,
[Exception.GetMessageFromNativeResources\(Exception.ExceptionMessageKind, StringHandleOnStack\)](#) ,
[Exception.Message](#) , [Exception.Data](#) , [Exception.InnerException](#) , [Exception.TargetSite](#) ,
[Exception.StackTrace](#) , [Exception.HelpLink](#) , [Exception.Source](#) , [Exception.IPForWatsonBuckets](#) ,
[Exception.WatsonBuckets](#) , [Exception.RemoteStackTrace](#) , [Exception.HResult](#) ,
[Exception.IsTransient](#) , [Exception.SerializeObjectState](#) , [object.Equals\(object\)](#) ,
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.FieldSetter\(string, string, object\)](#) ,
[object.FieldGetter\(string, string, ref object\)](#) , [object.GetFieldInfo\(string, string\)](#)

Constructors

DuplicateDirective(string, string, SourceContext)

```
public DuplicateDirective(string FileName, string DirectiveName, SourceContext sc)
```

Parameters

FileName [string](#)

DirectiveName [string](#)

sc SourceContext

Fields

DirectiveName

```
public string DirectiveName
```

Field Value

[string](#) ↗

Class DuplicateUsesUnit

Namespace: [PascalABCCompiler.Errors](#)

Assembly: Compiler.dll

Бросается при нахождении дубликатов в секции uses

```
public class DuplicateUsesUnit : CompilerThrownError, ISerializable, _Exception
```

Inheritance

[object](#) ← [Exception](#) ← Error ← LocatedError ← [CompilerThrownError](#) ← DuplicateUsesUnit

Implements

[ISerializable](#), [Exception](#)

Inherited Members

[CompilerThrownError.ToString\(\)](#), LocatedError.source_context, LocatedError.sourceLocation, LocatedError.fileName, LocatedError.SourceLocation, LocatedError.SourceContext, LocatedError.FileName, Error.MustThrow, [Exception.s_EDILock](#), [Exception.className](#), [Exception.exceptionMethod](#), [Exception.exceptionMethodString](#), [Exception.message](#), [Exception.data](#), [Exception.innerException](#), [Exception.helpURL](#), [Exception.stackTrace](#), [Exception.watsonBuckets](#), [Exception.stackTraceString](#), [Exception.remoteStackTraceString](#), [Exception.remoteStackIndex](#), [Exception.dynamicMethods](#), [Exception.HResult](#), [Exception.source](#), [Exception.xptrs](#), [Exception.xcode](#), [Exception.ipForWatsonBuckets](#), [Exception.safeSerializationManager](#), [Exception.COMPlusExceptionCode](#), [Exception.Init\(\)](#), [Exception.IsImmutableAgileException\(Exception\)](#), [Exception.AddExceptionDataForRestrictedErrorInfo\(string, string, string, object, bool\)](#), [Exception.TryGetRestrictedLanguageErrorObject\(out object\)](#), [Exception.GetClassName\(\)](#), [Exception.GetBaseException\(\)](#), [Exception.GetMethodFromStackTrace\(object\)](#), [Exception.GetExceptionMethodFromStackTrace\(\)](#), [Exception.GetTargetSiteInternal\(\)](#), [Exception.GetStackTrace\(bool\)](#), [Exception.SetErrorCode\(int\)](#), [Exception.ToString\(bool, bool\)](#), [Exception.GetExceptionMethodString\(\)](#), [Exception.GetExceptionMethodFromString\(\)](#), [Exception.GetObjectData\(SerializationInfo, StreamingContext\)](#), [Exception.PrepForRemoting\(\)](#), [Exception.OnDeserialized\(StreamingContext\)](#), [Exception.InternalPreserveStackTrace\(\)](#), [Exception.PrepareForForeignExceptionRaise\(\)](#), [Exception.GetStackTracesDeepCopy\(Exception, out object, out object\)](#), [Exception.SaveStackTracesFromDeepCopy\(Exception, object, object\)](#), [Exception.CopyStackTrace\(object\)](#), [Exception.CopyDynamicMethods\(object\)](#), [Exception.StripFileInfo\(string, bool\)](#), [Exception.DeepCopyStackTrace\(object\)](#),

[Exception.DeepCopyDynamicMethods\(object\)](#) ,
[Exception.GetStackTracesDeepCopy\(out object, out object\)](#) ,
[Exception.RestoreExceptionDispatchInfo\(ExceptionDispatchInfo\)](#) , [Exception.InternalToString\(\)](#) ,
[Exception.GetType\(\)](#) , [Exception.nIsTransient\(int\)](#) ,
[Exception.GetMessageFromNativeResources\(Exception.ExceptionMessageKind\)](#) ,
[Exception.GetMessageFromNativeResources\(Exception.ExceptionMessageKind, StringHandleOnStack\)](#) ,
[Exception.Message](#) , [Exception.Data](#) , [Exception.InnerException](#) , [Exception.TargetSite](#) ,
[Exception.StackTrace](#) , [Exception.HelpLink](#) , [Exception.Source](#) , [Exception.IPForWatsonBuckets](#) ,
[Exception.WatsonBuckets](#) , [Exception.RemoteStackTrace](#) , [Exception.HResult](#) ,
[Exception.IsTransient](#) , [Exception.SerializeObjectState](#) , [object.Equals\(object\)](#) ,
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.FieldSetter\(string, string, object\)](#) ,
[object.FieldGetter\(string, string, ref object\)](#) , [object.GetFieldInfo\(string, string\)](#)

Constructors

DuplicateUsesUnit(string, string, SourceContext)

```
public DuplicateUsesUnit(string FileName, string UnitName, SourceContext sc)
```

Parameters

FileName [string](#)

UnitName [string](#)

sc SourceContext

Fields

UnitName

```
public string UnitName
```

Field Value

[string](#)

Class FileNotFoundException

Namespace: [PascalABCCompiler.Errors](#)

Assembly: Compiler.dll

Бросается при отсутствии некоторого файла по некоторому пути

```
public class FileNotFoundException : CompilerThrownError, ISerializable, _Exception
```

Inheritance

[object](#) ← [Exception](#) ← Error ← LocatedError ← [CompilerThrownError](#) ← FileNotFoundException

Implements

[ISerializable](#), [Exception](#)

Inherited Members

[CompilerThrownError.ToString\(\)](#), LocatedError.source_context, LocatedError.sourceLocation, LocatedError.fileName, LocatedError.SourceLocation, LocatedError.SourceContext, LocatedError.FileName, Error.MustThrow, [Exception.s_EDILock](#), [Exception.className](#), [Exception.exceptionMethod](#), [Exception.exceptionMethodString](#), [Exception.message](#), [Exception.data](#), [Exception.innerException](#), [Exception.helpURL](#), [Exception.stackTrace](#), [Exception.watsonBuckets](#), [Exception.stackTraceString](#), [Exception.remoteStackTraceString](#), [Exception.remoteStackIndex](#), [Exception.dynamicMethods](#), [Exception.HResult](#), [Exception.source](#), [Exception.xptrs](#), [Exception.xcode](#), [Exception.ipForWatsonBuckets](#), [Exception.safeSerializationManager](#), [Exception.COMPlusExceptionCode](#), [Exception.Init\(\)](#), [Exception.IsImmutableAgileException\(Exception\)](#), [Exception.AddExceptionDataForRestrictedErrorInfo\(string, string, string, object, bool\)](#), [Exception.TryGetRestrictedLanguageErrorObject\(out object\)](#), [Exception.GetClassName\(\)](#), [Exception.GetBaseException\(\)](#), [Exception.GetMethodFromStackTrace\(object\)](#), [Exception.GetExceptionMethodFromStackTrace\(\)](#), [Exception.GetTargetSiteInternal\(\)](#), [Exception.GetStackTrace\(bool\)](#), [Exception.SetErrorCode\(int\)](#), [Exception.ToString\(bool, bool\)](#), [Exception.GetExceptionMethodString\(\)](#), [Exception.GetExceptionMethodFromString\(\)](#), [Exception.GetObjectData\(SerializationInfo, StreamingContext\)](#), [Exception.PrepForRemoting\(\)](#), [Exception.OnDeserialized\(StreamingContext\)](#), [Exception.InternalPreserveStackTrace\(\)](#), [Exception.PrepareForForeignExceptionRaise\(\)](#), [Exception.GetStackTracesDeepCopy\(Exception, out object, out object\)](#), [Exception.SaveStackTracesFromDeepCopy\(Exception, object, object\)](#), [Exception.CopyStackTrace\(object\)](#), [Exception.CopyDynamicMethods\(object\)](#), [Exception.StripFileInfo\(string, bool\)](#), [Exception.DeepCopyStackTrace\(object\)](#),

[Exception.DeepCopyDynamicMethods\(object\)](#) ,
[Exception.GetStackTracesDeepCopy\(out object, out object\)](#) ,
[Exception.RestoreExceptionDispatchInfo\(ExceptionDispatchInfo\)](#) , [Exception.InternalToString\(\)](#) ,
[Exception.GetType\(\)](#) , [Exception.nIsTransient\(int\)](#) ,
[Exception.GetMessageFromNativeResources\(Exception.ExceptionMessageKind\)](#) ,
[Exception.GetMessageFromNativeResources\(Exception.ExceptionMessageKind, StringHandleOnStack\)](#) ,
[Exception.Message](#) , [Exception.Data](#) , [Exception.InnerException](#) , [Exception.TargetSite](#) ,
[Exception.StackTrace](#) , [Exception.HelpLink](#) , [Exception.Source](#) , [Exception.IPForWatsonBuckets](#) ,
[Exception.WatsonBuckets](#) , [Exception.RemoteStackTrace](#) , [Exception.HResult](#) ,
[Exception.IsTransient](#) , [Exception.SerializeObjectState](#) , [object.Equals\(object\)](#) ,
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.FieldSetter\(string, string, object\)](#) ,
[object.FieldGetter\(string, string, ref object\)](#) , [object.GetFieldInfo\(string, string\)](#)

Constructors

FileNotFoundException(string, SourceContext)

```
public FileNotFoundException(string fileName, SourceContext sc)
```

Parameters

fileName [string](#)

sc [SourceContext](#)

Class IncludeNamespaceInUnitError

Namespace: [PascalABCCompiler.Errors](#)

Assembly: Compiler.dll

Бросается при подключении явного пространства имен в модуле

```
public class IncludeNamespaceInUnitError : CompilerThrownError, ISerializable, _Exception
```

Inheritance

[object](#) ← [Exception](#) ← Error ← LocatedError ← [CompilerThrownError](#) ← IncludeNamespaceInUnitError

Implements

[ISerializable](#), [Exception](#)

Inherited Members

[CompilerThrownError.ToString\(\)](#), LocatedError.source_context, LocatedError.sourceLocation, LocatedError.fileName, LocatedError.SourceLocation, LocatedError.SourceContext, LocatedError.FileName, Error.MustThrow, [Exception.s_EDILock](#), [Exception.className](#), [Exception.exceptionMethod](#), [Exception.exceptionMethodString](#), [Exception.message](#), [Exception.data](#), [Exception.innerException](#), [Exception.helpURL](#), [Exception.stackTrace](#), [Exception.watsonBuckets](#), [Exception.stackTraceString](#), [Exception.remoteStackTraceString](#), [Exception.remoteStackIndex](#), [Exception.dynamicMethods](#), [Exception.HResult](#), [Exception.source](#), [Exception.xptrs](#), [Exception.xcode](#), [Exception.ipForWatsonBuckets](#), [Exception.safeSerializationManager](#), [Exception.COMPlusExceptionCode](#), [Exception.Init\(\)](#), [Exception.IsImmutableAgileException\(Exception\)](#), [Exception.AddExceptionDataForRestrictedErrorInfo\(string, string, string, object, bool\)](#), [Exception.TryGetRestrictedLanguageErrorObject\(out object\)](#), [Exception.GetClassName\(\)](#), [Exception.GetBaseException\(\)](#), [Exception.GetMethodFromStackTrace\(object\)](#), [Exception.GetExceptionMethodFromStackTrace\(\)](#), [Exception.GetTargetSiteInternal\(\)](#), [Exception.GetStackTrace\(bool\)](#), [Exception.SetErrorCode\(int\)](#), [Exception.ToString\(bool, bool\)](#), [Exception.GetExceptionMethodString\(\)](#), [Exception.GetExceptionMethodFromString\(\)](#), [Exception.GetObjectData\(SerializationInfo, StreamingContext\)](#), [Exception.PrepForRemoting\(\)](#), [Exception.OnDeserialized\(StreamingContext\)](#), [Exception.InternalPreserveStackTrace\(\)](#), [Exception.PrepareForForeignExceptionRaise\(\)](#), [Exception.GetStackTracesDeepCopy\(Exception, out object, out object\)](#), [Exception.SaveStackTracesFromDeepCopy\(Exception, object, object\)](#), [Exception.CopyStackTrace\(object\)](#), [Exception.CopyDynamicMethods\(object\)](#), [Exception.StripFileInfo\(string, bool\)](#), [Exception.DeepCopyStackTrace\(object\)](#),

[Exception.DeepCopyDynamicMethods\(object\)](#) ,
[Exception.GetStackTracesDeepCopy\(out object, out object\)](#) ,
[Exception.RestoreExceptionDispatchInfo\(ExceptionDispatchInfo\)](#) , [Exception.InternalToString\(\)](#) ,
[Exception.GetType\(\)](#) , [Exception.nIsTransient\(int\)](#) ,
[Exception.GetMessageFromNativeResources\(Exception.ExceptionMessageKind\)](#) ,
[Exception.GetMessageFromNativeResources\(Exception.ExceptionMessageKind, StringHandleOnStack\)](#) ,
[Exception.Message](#) , [Exception.Data](#) , [Exception.InnerException](#) , [Exception.TargetSite](#) ,
[Exception.StackTrace](#) , [Exception.HelpLink](#) , [Exception.Source](#) , [Exception.IPForWatsonBuckets](#) ,
[Exception.WatsonBuckets](#) , [Exception.RemoteStackTrace](#) , [Exception.HResult](#) ,
[Exception.IsTransient](#) , [Exception.SerializeObjectState](#) , [object.Equals\(object\)](#) ,
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.FieldSetter\(string, string, object\)](#) ,
[object.FieldGetter\(string, string, ref object\)](#) , [object.GetFieldInfo\(string, string\)](#)

Constructors

IncludeNamespaceInUnitError(string, SourceContext)

```
public IncludeNamespaceInUnitError(string FileName, SourceContext sc)
```

Parameters

FileName [string](#)

sc SourceContext

Class InvalidAssemblyPathError

Namespace: [PascalABCCompiler.Errors](#)

Assembly: Compiler.dll

Бросается при попытке обработки неправильного пути к сборке

```
public class InvalidAssemblyPathError : CompilerThrownError, ISerializable, _Exception
```

Inheritance

[object](#) ← [Exception](#) ← Error ← LocatedError ← [CompilerThrownError](#) ← InvalidAssemblyPathError

Implements

[ISerializable](#), [Exception](#)

Inherited Members

[CompilerThrownError.ToString\(\)](#), LocatedError.source_context, LocatedError.sourceLocation, LocatedError.fileName, LocatedError.SourceLocation, LocatedError.SourceContext, LocatedError.FileName, Error.MustThrow, [Exception.s_EDILock](#), [Exception.className](#), [Exception.exceptionMethod](#), [Exception.exceptionMethodString](#), [Exception.message](#), [Exception.data](#), [Exception.innerException](#), [Exception.helpURL](#), [Exception.stackTrace](#), [Exception.watsonBuckets](#), [Exception.stackTraceString](#), [Exception.remoteStackTraceString](#), [Exception.remoteStackIndex](#), [Exception.dynamicMethods](#), [Exception.HResult](#), [Exception.source](#), [Exception.xptrs](#), [Exception.xcode](#), [Exception.ipForWatsonBuckets](#), [Exception.safeSerializationManager](#), [Exception.COMPlusExceptionCode](#), [Exception.Init\(\)](#), [Exception.IsImmutableAgileException\(Exception\)](#), [Exception.AddExceptionDataForRestrictedErrorInfo\(string, string, string, object, bool\)](#), [Exception.TryGetRestrictedLanguageErrorObject\(out object\)](#), [Exception.GetClassName\(\)](#), [Exception.GetBaseException\(\)](#), [Exception.GetMethodFromStackTrace\(object\)](#), [Exception.GetExceptionMethodFromStackTrace\(\)](#), [Exception.GetTargetSiteInternal\(\)](#), [Exception.GetStackTrace\(bool\)](#), [Exception.SetErrorCode\(int\)](#), [Exception.ToString\(bool, bool\)](#), [Exception.GetExceptionMethodString\(\)](#), [Exception.GetExceptionMethodFromString\(\)](#), [Exception.GetObjectData\(SerializationInfo, StreamingContext\)](#), [Exception.PrepForRemoting\(\)](#), [Exception.OnDeserialized\(StreamingContext\)](#), [Exception.InternalPreserveStackTrace\(\)](#), [Exception.PrepareForForeignExceptionRaise\(\)](#), [Exception.GetStackTracesDeepCopy\(Exception, out object, out object\)](#), [Exception.SaveStackTracesFromDeepCopy\(Exception, object, object\)](#), [Exception.CopyStackTrace\(object\)](#), [Exception.CopyDynamicMethods\(object\)](#), [Exception.StripFileInfo\(string, bool\)](#), [Exception.DeepCopyStackTrace\(object\)](#),

[Exception.DeepCopyDynamicMethods\(object\)](#) ,
[Exception.GetStackTracesDeepCopy\(out object, out object\)](#) ,
[Exception.RestoreExceptionDispatchInfo\(ExceptionDispatchInfo\)](#) , [Exception.InternalToString\(\)](#) ,
[Exception.GetType\(\)](#) , [Exception.nIsTransient\(int\)](#) ,
[Exception.GetMessageFromNativeResources\(Exception.ExceptionMessageKind\)](#) ,
[Exception.GetMessageFromNativeResources\(Exception.ExceptionMessageKind, StringHandleOnStack\)](#) ,
[Exception.Message](#) , [Exception.Data](#) , [Exception.InnerException](#) , [Exception.TargetSite](#) ,
[Exception.StackTrace](#) , [Exception.HelpLink](#) , [Exception.Source](#) , [Exception.IPForWatsonBuckets](#) ,
[Exception.WatsonBuckets](#) , [Exception.RemoteStackTrace](#) , [Exception.HResult](#) ,
[Exception.IsTransient](#) , [Exception.SerializeObjectState](#) , [object.Equals\(object\)](#) ,
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.FieldSetter\(string, string, object\)](#) ,
[object.FieldGetter\(string, string, ref object\)](#) , [object.GetFieldInfo\(string, string\)](#)

Constructors

InvalidAssemblyPathError(string, SourceContext)

```
public InvalidAssemblyPathError(string FileName, SourceContext sc)
```

Parameters

FileName [string](#)

sc SourceContext

Class InvalidPathError

Namespace: [PascalABCCompiler.Errors](#)

Assembly: Compiler.dll

Бросается при попытке обработки неправильного пути к файлу

```
public class InvalidPathError : CompilerThrownError, ISerializable, _Exception
```

Inheritance

[object](#) ← [Exception](#) ← Error ← LocatedError ← [CompilerThrownError](#) ← InvalidPathError

Implements

[ISerializable](#), [Exception](#)

Inherited Members

[CompilerThrownError.ToString\(\)](#), LocatedError.source_context, LocatedError.sourceLocation, LocatedError.fileName, LocatedError.SourceLocation, LocatedError.SourceContext, LocatedError.FileName, Error.MustThrow, [Exception.s_EDILock](#), [Exception.className](#), [Exception.exceptionMethod](#), [Exception.exceptionMethodString](#), [Exception.message](#), [Exception.data](#), [Exception.innerException](#), [Exception.helpURL](#), [Exception.stackTrace](#), [Exception.watsonBuckets](#), [Exception.stackTraceString](#), [Exception.remoteStackTraceString](#), [Exception.remoteStackIndex](#), [Exception.dynamicMethods](#), [Exception.HResult](#), [Exception.source](#), [Exception.xptrs](#), [Exception.xcode](#), [Exception.ipForWatsonBuckets](#), [Exception.safeSerializationManager](#), [Exception.COMPlusExceptionCode](#), [Exception.Init\(\)](#), [Exception.IsImmutableAgileException\(Exception\)](#), [Exception.AddExceptionDataForRestrictedErrorInfo\(string, string, string, object, bool\)](#), [Exception.TryGetRestrictedLanguageErrorObject\(out object\)](#), [Exception.GetClassName\(\)](#), [Exception.GetBaseException\(\)](#), [Exception.GetMethodFromStackTrace\(object\)](#), [Exception.GetExceptionMethodFromStackTrace\(\)](#), [Exception.GetTargetSiteInternal\(\)](#), [Exception.GetStackTrace\(bool\)](#), [Exception.SetErrorCode\(int\)](#), [Exception.ToString\(bool, bool\)](#), [Exception.GetExceptionMethodString\(\)](#), [Exception.GetExceptionMethodFromString\(\)](#), [Exception.GetObjectData\(SerializationInfo, StreamingContext\)](#), [Exception.PrepForRemoting\(\)](#), [Exception.OnDeserialized\(StreamingContext\)](#), [Exception.InternalPreserveStackTrace\(\)](#), [Exception.PrepareForForeignExceptionRaise\(\)](#), [Exception.GetStackTracesDeepCopy\(Exception, out object, out object\)](#), [Exception.SaveStackTracesFromDeepCopy\(Exception, object, object\)](#), [Exception.CopyStackTrace\(object\)](#), [Exception.CopyDynamicMethods\(object\)](#), [Exception.StripFileInfo\(string, bool\)](#), [Exception.DeepCopyStackTrace\(object\)](#),

[Exception.DeepCopyDynamicMethods\(object\)](#) ,
[Exception.GetStackTracesDeepCopy\(out object, out object\)](#) ,
[Exception.RestoreExceptionDispatchInfo\(ExceptionDispatchInfo\)](#) , [Exception.InternalToString\(\)](#) ,
[Exception.GetType\(\)](#) , [Exception.nIsTransient\(int\)](#) ,
[Exception.GetMessageFromNativeResources\(Exception.ExceptionMessageKind\)](#) ,
[Exception.GetMessageFromNativeResources\(Exception.ExceptionMessageKind, StringHandleOnStack\)](#) ,
[Exception.Message](#) , [Exception.Data](#) , [Exception.InnerException](#) , [Exception.TargetSite](#) ,
[Exception.StackTrace](#) , [Exception.HelpLink](#) , [Exception.Source](#) , [Exception.IPForWatsonBuckets](#) ,
[Exception.WatsonBuckets](#) , [Exception.RemoteStackTrace](#) , [Exception.HResult](#) ,
[Exception.IsTransient](#) , [Exception.SerializeObjectState](#) , [object.Equals\(object\)](#) ,
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.FieldSetter\(string, string, object\)](#) ,
[object.FieldGetter\(string, string, ref object\)](#) , [object.GetFieldInfo\(string, string\)](#)

Constructors

InvalidPathError(SourceContext)

```
public InvalidPathError(SourceContext sc)
```

Parameters

sc SourceContext

Class MainResourceNotAllowed

Namespace: [PascalABCCompiler.Errors](#)

Assembly: Compiler.dll

Бросается в случае некорректного использования директивы {\$mainresource ...}

```
public class MainResourceNotAllowed : CompilerThrownError, ISerializable, _Exception
```

Inheritance

[object](#) ← [Exception](#) ← Error ← LocatedError ← [CompilerThrownError](#) ← MainResourceNotAllowed

Implements

[ISerializable](#), [Exception](#)

Inherited Members

[CompilerThrownError.ToString\(\)](#), LocatedError.source_context, LocatedError.sourceLocation, LocatedError.fileName, LocatedError.SourceLocation, LocatedError.SourceContext, LocatedError.FileName, Error.MustThrow, [Exception.s_EDILock](#), [Exception.className](#), [Exception.exceptionMethod](#), [Exception.exceptionMethodString](#), [Exception.message](#), [Exception.data](#), [Exception.innerException](#), [Exception.helpURL](#), [Exception.stackTrace](#), [Exception.watsonBuckets](#), [Exception.stackTraceString](#), [Exception.remoteStackTraceString](#), [Exception.remoteStackIndex](#), [Exception.dynamicMethods](#), [Exception.HResult](#), [Exception.source](#), [Exception.xptrs](#), [Exception.xcode](#), [Exception.ipForWatsonBuckets](#), [Exception.safeSerializationManager](#), [Exception.COMPlusExceptionCode](#), [Exception.Init\(\)](#), [Exception.IsImmutableAgileException\(Exception\)](#), [Exception.AddExceptionDataForRestrictedErrorInfo\(string, string, string, object, bool\)](#), [Exception.TryGetRestrictedLanguageErrorObject\(out object\)](#), [Exception.GetClassName\(\)](#), [Exception.GetBaseException\(\)](#), [Exception.GetMethodFromStackTrace\(object\)](#), [Exception.GetExceptionMethodFromStackTrace\(\)](#), [Exception.GetTargetSiteInternal\(\)](#), [Exception.GetStackTrace\(bool\)](#), [Exception.SetErrorCode\(int\)](#), [Exception.ToString\(bool, bool\)](#), [Exception.GetExceptionMethodString\(\)](#), [Exception.GetExceptionMethodFromString\(\)](#), [Exception.GetObjectData\(SerializationInfo, StreamingContext\)](#), [Exception.PrepForRemoting\(\)](#), [Exception.OnDeserialized\(StreamingContext\)](#), [Exception.InternalPreserveStackTrace\(\)](#), [Exception.PrepareForForeignExceptionRaise\(\)](#), [Exception.GetStackTracesDeepCopy\(Exception, out object, out object\)](#), [Exception.SaveStackTracesFromDeepCopy\(Exception, object, object\)](#), [Exception.CopyStackTrace\(object\)](#), [Exception.CopyDynamicMethods\(object\)](#), [Exception.StripFileInfo\(string, bool\)](#), [Exception.DeepCopyStackTrace\(object\)](#),

[Exception.DeepCopyDynamicMethods\(object\)](#) ,
[Exception.GetStackTracesDeepCopy\(out object, out object\)](#) ,
[Exception.RestoreExceptionDispatchInfo\(ExceptionDispatchInfo\)](#) , [Exception.InternalToString\(\)](#) ,
[Exception.GetType\(\)](#) , [Exception.nIsTransient\(int\)](#) ,
[Exception.GetMessageFromNativeResources\(Exception.ExceptionMessageKind\)](#) ,
[Exception.GetMessageFromNativeResources\(Exception.ExceptionMessageKind, StringHandleOnStack\)](#) ,
[Exception.Message](#) , [Exception.Data](#) , [Exception.InnerException](#) , [Exception.TargetSite](#) ,
[Exception.StackTrace](#) , [Exception.HelpLink](#) , [Exception.Source](#) , [Exception.IPForWatsonBuckets](#) ,
[Exception.WatsonBuckets](#) , [Exception.RemoteStackTrace](#) , [Exception.HResult](#) ,
[Exception.IsTransient](#) , [Exception.SerializeObjectState](#) , [object.Equals\(object\)](#) ,
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.FieldSetter\(string, string, object\)](#) ,
[object.FieldGetter\(string, string, ref object\)](#) , [object.GetFieldInfo\(string, string\)](#)

Constructors

MainResourceNotAllowed(string, SourceContext)

```
public MainResourceNotAllowed(string fileName, SourceContext sc)
```

Parameters

fileName [string](#)

sc [SourceContext](#)

Class NamespaceCannotHaveInSection

Namespace: [PascalABCCompiler.Errors](#)

Assembly: Compiler.dll

Бросается в случае использования uses in в пространстве имен

```
public class NamespaceCannotHaveInSection : CompilerThrownError, ISerializable, _Exception
```

Inheritance

[object](#) ← [Exception](#) ← Error ← LocatedError ← [CompilerThrownError](#) ← NamespaceCannotHaveInSection

Implements

[ISerializable](#), [_Exception](#)

Inherited Members

[CompilerThrownError.ToString\(\)](#), LocatedError.source_context, LocatedError.sourceLocation, LocatedError.fileName, LocatedError.SourceLocation, LocatedError.SourceContext, LocatedError.FileName, Error.MustThrow, [Exception.s_EDILock](#), [Exception.className](#), [Exception.exceptionMethod](#), [Exception.exceptionMethodString](#), [Exception.message](#), [Exception.data](#), [Exception.innerException](#), [Exception.helpURL](#), [Exception.stackTrace](#), [Exception.watsonBuckets](#), [Exception.stackTraceString](#), [Exception.remoteStackTraceString](#), [Exception.remoteStackIndex](#), [Exception.dynamicMethods](#), [Exception.HResult](#), [Exception.source](#), [Exception.xptrs](#), [Exception.xcode](#), [Exception.ipForWatsonBuckets](#), [Exception.safeSerializationManager](#), [Exception.COMPlusExceptionCode](#), [Exception.Init\(\)](#), [Exception.IsImmutableAgileException\(Exception\)](#), [Exception.AddExceptionDataForRestrictedErrorInfo\(string, string, string, object, bool\)](#), [Exception.TryGetRestrictedLanguageErrorObject\(out object\)](#), [Exception.GetClassName\(\)](#), [Exception.GetBaseException\(\)](#), [ExceptionGetMethodFromStackTrace\(object\)](#), [Exception.GetExceptionMethodFromStackTrace\(\)](#), [Exception.GetTargetSiteInternal\(\)](#), [Exception.GetStackTrace\(bool\)](#), [Exception.SetErrorCode\(int\)](#), [Exception.ToString\(bool, bool\)](#), [Exception.GetExceptionMethodString\(\)](#), [Exception.GetExceptionMethodFromString\(\)](#), [Exception.GetObjectData\(SerializationInfo, StreamingContext\)](#), [Exception.PrepForRemoting\(\)](#), [Exception.OnDeserialized\(StreamingContext\)](#), [Exception.InternalPreserveStackTrace\(\)](#), [Exception.PrepareForForeignExceptionRaise\(\)](#), [Exception.GetStackTracesDeepCopy\(Exception, out object, out object\)](#), [Exception.SaveStackTracesFromDeepCopy\(Exception, object, object\)](#), [Exception.CopyStackTrace\(object\)](#), [Exception.CopyDynamicMethods\(object\)](#),

[Exception.StripFileInfo\(string, bool\)](#) , [Exception.DeepCopyStackTrace\(object\)](#) ,
[Exception.DeepCopyDynamicMethods\(object\)](#) ,
[Exception.GetStackTracesDeepCopy\(out object, out object\)](#) ,
[Exception.RestoreExceptionDispatchInfo\(ExceptionDispatchInfo\)](#) , [Exception.InternalToString\(\)](#) ,
[Exception.GetType\(\)](#) , [Exception.nlsTransient\(int\)](#) ,
[Exception.GetMessageFromNativeResources\(Exception.ExceptionMessageKind\)](#) ,
[Exception.GetMessageFromNativeResources\(Exception.ExceptionMessageKind, StringHandleOnStack\)](#) ,
[Exception.Message](#) , [Exception.Data](#) , [Exception.InnerException](#) , [Exception.TargetSite](#) ,
[Exception.StackTrace](#) , [Exception.HelpLink](#) , [Exception.Source](#) , [Exception.IPForWatsonBuckets](#) ,
[Exception.WatsonBuckets](#) , [Exception.RemoteStackTrace](#) , [Exception.HResult](#) ,
[Exception.IsTransient](#) , [Exception.SerializeObjectState](#) , [object.Equals\(object\)](#) ,
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.FieldSetter\(string, string, object\)](#) ,
[object.FieldGetter\(string, string, ref object\)](#) , [object.GetFieldInfo\(string, string\)](#)

Constructors

NamespaceCannotHaveInSection(SourceContext)

```
public NamespaceCannotHaveInSection(SourceContext sc)
```

Parameters

sc SourceContext

Class NamespaceModuleExpected

Namespace: [PascalABCCompiler.Errors](#)

Assembly: Compiler.dll

Бросается, если встреченено не явное пространство имен

```
public class NamespaceModuleExpected : CompilerThrownError, ISerializable, _Exception
```

Inheritance

[object](#) ↗ ← [Exception](#) ↗ ← Error ← LocatedError ← [CompilerThrownError](#) ← NamespaceModuleExpected

Implements

[ISerializable](#) ↗ , [Exception](#) ↗

Inherited Members

[CompilerThrownError.ToString\(\)](#) , LocatedError.source_context , LocatedError.sourceLocation , LocatedError.fileName , LocatedError.SourceLocation , LocatedError.SourceContext , LocatedError.FileName , Error.MustThrow , [Exception.s_EDILock](#) ↗ , [Exception.className](#) ↗ , [Exception.exceptionMethod](#) ↗ , [Exception.exceptionMethodString](#) ↗ , [Exception.message](#) ↗ , [Exception.data](#) ↗ , [Exception.innerException](#) ↗ , [Exception.helpURL](#) ↗ , [Exception.stackTrace](#) ↗ , [Exception.watsonBuckets](#) ↗ , [Exception.stackTraceString](#) ↗ , [Exception.remoteStackTraceString](#) ↗ , [Exception.remoteStackIndex](#) ↗ , [Exception.dynamicMethods](#) ↗ , [Exception.HResult](#) ↗ , [Exception.source](#) ↗ , [Exception.xptrs](#) ↗ , [Exception.xcode](#) ↗ , [Exception.ipForWatsonBuckets](#) ↗ , [Exception.safeSerializationManager](#) ↗ , [Exception.COMPlusExceptionCode](#) ↗ , [Exception.Init\(\)](#) ↗ , [Exception.IsImmutableAgileException](#)([Exception](#)) ↗ , [Exception.AddExceptionDataForRestrictedErrorInfo](#)(string, string, string, object, bool) ↗ , [Exception.TryGetRestrictedLanguageErrorObject](#)(out object) ↗ , [Exception.GetClassName\(\)](#) ↗ , [Exception.GetBaseException\(\)](#) ↗ , [Exception.GetMethodFromStackTrace](#)(object) ↗ , [Exception.GetExceptionMethodFromStackTrace](#)() ↗ , [Exception.GetTargetSiteInternal](#)() ↗ , [Exception.GetStackTrace](#)(bool) ↗ , [Exception.SetErrorCode](#)(int) ↗ , [Exception.ToString](#)(bool, bool) ↗ , [Exception.GetExceptionMethodString](#)() ↗ , [Exception.GetExceptionMethodFromString](#)() ↗ , [Exception.GetObjectData](#)([SerializationInfo](#), [StreamingContext](#)) ↗ , [Exception.PrepForRemoting](#)() ↗ , [Exception.OnDeserialized](#)([StreamingContext](#)) ↗ , [Exception.InternalPreserveStackTrace](#)() ↗ , [Exception.PrepareForForeignExceptionRaise](#)() ↗ , [Exception.GetStackTracesDeepCopy](#)([Exception](#), out object, out object) ↗ , [Exception.SaveStackTracesFromDeepCopy](#)([Exception](#), object, object) ↗ , [Exception.CopyStackTrace](#)(object) ↗ , [Exception.CopyDynamicMethods](#)(object) ↗ , [Exception.StripFileInfo](#)(string, bool) ↗ , [Exception.DeepCopyStackTrace](#)(object) ↗ ,

[Exception.DeepCopyDynamicMethods\(object\)](#) ,
[Exception.GetStackTracesDeepCopy\(out object, out object\)](#) ,
[Exception.RestoreExceptionDispatchInfo\(ExceptionDispatchInfo\)](#) , [Exception.InternalToString\(\)](#) ,
[Exception.GetType\(\)](#) , [Exception.nIsTransient\(int\)](#) ,
[Exception.GetMessageFromNativeResources\(Exception.ExceptionMessageKind\)](#) ,
[Exception.GetMessageFromNativeResources\(Exception.ExceptionMessageKind, StringHandleOnStack\)](#) ,
[Exception.Message](#) , [Exception.Data](#) , [Exception.InnerException](#) , [Exception.TargetSite](#) ,
[Exception.StackTrace](#) , [Exception.HelpLink](#) , [Exception.Source](#) , [Exception.IPForWatsonBuckets](#) ,
[Exception.WatsonBuckets](#) , [Exception.RemoteStackTrace](#) , [Exception.HResult](#) ,
[Exception.IsTransient](#) , [Exception.SerializeObjectState](#) , [object.Equals\(object\)](#) ,
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.FieldSetter\(string, string, object\)](#) ,
[object.FieldGetter\(string, string, ref object\)](#) , [object.GetFieldInfo\(string, string\)](#)

Constructors

NamespaceModuleExpected(SourceContext)

```
public NamespaceModuleExpected(SourceContext sc)
```

Parameters

sc SourceContext

Class NamespacesCanBeCompiledOnlyInProjects

Namespace: [PascalABCCompiler.Errors](#)

Assembly: Compiler.dll

Legacy, бросается, если unitModule.unitName.HeaderKeyword ==
SyntaxTree.UnitHeaderKeyword.Namespace

```
public class NamespacesCanBeCompiledOnlyInProjects : CompilerThrownError,  
ISerializable, _Exception
```

Inheritance

[object](#) ← [Exception](#) ← Error ← LocatedError ← [CompilerThrownError](#) ←
NamespacesCanBeCompiledOnlyInProjects

Implements

[ISerializable](#), [_Exception](#)

Inherited Members

[CompilerThrownError.ToString\(\)](#), LocatedError.source_context, LocatedError.sourceLocation,
LocatedError.fileName, LocatedError.SourceLocation, LocatedError.SourceContext,
LocatedError.FileName, Error.MustThrow, [Exception.s_EDILock](#), [Exception.className](#),
[Exception.exceptionMethod](#), [Exception.exceptionMethodString](#), [Exception.message](#),
[Exception.data](#), [Exception.innerException](#), [Exception.helpURL](#), [Exception.stackTrace](#),
[Exception.watsonBuckets](#), [Exception.stackTraceString](#), [Exception.remoteStackTraceString](#),
[Exception.remoteStackIndex](#), [Exception.dynamicMethods](#), [Exception.HResult](#),
[Exception.source](#), [Exception.xptrs](#), [Exception.xcode](#), [Exception.ipForWatsonBuckets](#),
[Exception.safeSerializationManager](#), [Exception.COMPlusExceptionCode](#), [Exception.Init\(\)](#),
[Exception.IsImmutableAgileException\(Exception\)](#),
[Exception.AddExceptionDataForRestrictedErrorInfo\(string, string, string, object, bool\)](#),
[Exception.TryGetRestrictedLanguageErrorObject\(out object\)](#), [Exception.GetClassName\(\)](#),
[Exception.GetBaseException\(\)](#), [Exception.GetMethodFromStackTrace\(object\)](#),
[Exception.GetExceptionMethodFromStackTrace\(\)](#), [Exception.GetTargetSiteInternal\(\)](#),
[Exception.GetStackTrace\(bool\)](#), [Exception.SetErrorCode\(int\)](#), [Exception.ToString\(bool, bool\)](#),
[Exception.GetExceptionMethodString\(\)](#), [Exception.GetExceptionMethodFromString\(\)](#),
[Exception.GetObjectData\(SerializationInfo, StreamingContext\)](#), [Exception.PrepForRemoting\(\)](#),
[Exception.OnDeserialized\(StreamingContext\)](#), [Exception.InternalPreserveStackTrace\(\)](#),
[Exception.PrepareForForeignExceptionRaise\(\)](#),

[Exception.GetStackTracesDeepCopy\(Exception, out object, out object\)](#) ,
[Exception.SaveStackTracesFromDeepCopy\(Exception, object, object\)](#) ,
[Exception.CopyStackTrace\(object\)](#) , [Exception.CopyDynamicMethods\(object\)](#) ,
[Exception.StripFileInfo\(string, bool\)](#) , [Exception.DeepCopyStackTrace\(object\)](#) ,
[Exception.DeepCopyDynamicMethods\(object\)](#) ,
[Exception.GetStackTracesDeepCopy\(out object, out object\)](#) ,
[Exception.RestoreExceptionDispatchInfo\(ExceptionDispatchInfo\)](#) , [Exception.InternalToString\(\)](#) ,
[Exception.GetType\(\)](#) , [Exception.nIsTransient\(int\)](#) ,
[Exception.GetMessageFromNativeResources\(Exception.ExceptionMessageKind\)](#) ,
[Exception.GetMessageFromNativeResources\(Exception.ExceptionMessageKind, StringHandleOnStack\)](#) ,
[Exception.Message](#) , [Exception.Data](#) , [Exception.InnerException](#) , [Exception.TargetSite](#) ,
[Exception.StackTrace](#) , [Exception.HelpLink](#) , [Exception.Source](#) , [Exception.IPForWatsonBuckets](#) ,
[Exception.WatsonBuckets](#) , [Exception.RemoteStackTrace](#) , [Exception.HResult](#) ,
[Exception.IsTransient](#) , [Exception.SerializeObjectState](#) , [object.Equals\(object\)](#) ,
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.FieldSetter\(string, string, object\)](#) ,
[object.FieldGetter\(string, string, ref object\)](#) , [object.GetFieldInfo\(string, string\)](#)

Constructors

NamespacesCanBeCompiledOnlyInProjects(SourceContext)

```
public NamespacesCanBeCompiledOnlyInProjects(SourceContext sc)
```

Parameters

sc SourceContext

Class ProgramModuleExpected

Namespace: [PascalABCCompiler.Errors](#)

Assembly: Compiler.dll

Бросается, если встретчена не основная программа (там где она должна быть)

```
public class ProgramModuleExpected : CompilerThrownError, ISerializable, _Exception
```

Inheritance

[object](#) ← [Exception](#) ← Error ← LocatedError ← [CompilerThrownError](#) ← ProgramModuleExpected

Implements

[ISerializable](#), [Exception](#)

Inherited Members

[CompilerThrownError.ToString\(\)](#), LocatedError.source_context, LocatedError.sourceLocation, LocatedError.fileName, LocatedError.SourceLocation, LocatedError.SourceContext, LocatedError.FileName, Error.MustThrow, [Exception.s_EDILock](#), [Exception.className](#), [Exception.exceptionMethod](#), [Exception.exceptionMethodString](#), [Exception.message](#), [Exception.data](#), [Exception.innerException](#), [Exception.helpURL](#), [Exception.stackTrace](#), [Exception.watsonBuckets](#), [Exception.stackTraceString](#), [Exception.remoteStackTraceString](#), [Exception.remoteStackIndex](#), [Exception.dynamicMethods](#), [Exception.HResult](#), [Exception.source](#), [Exception.xptrs](#), [Exception.xcode](#), [Exception.ipForWatsonBuckets](#), [Exception.safeSerializationManager](#), [Exception.COMPlusExceptionCode](#), [Exception.Init\(\)](#), [Exception.IsImmutableAgileException\(Exception\)](#), [Exception.AddExceptionDataForRestrictedErrorInfo\(string, string, string, object, bool\)](#), [Exception.TryGetRestrictedLanguageErrorObject\(out object\)](#), [Exception.GetClassName\(\)](#), [Exception.GetBaseException\(\)](#), [Exception.GetMethodFromStackTrace\(object\)](#), [Exception.GetExceptionMethodFromStackTrace\(\)](#), [Exception.GetTargetSiteInternal\(\)](#), [Exception.GetStackTrace\(bool\)](#), [Exception.SetErrorCode\(int\)](#), [Exception.ToString\(bool, bool\)](#), [Exception.GetExceptionMethodString\(\)](#), [Exception.GetExceptionMethodFromString\(\)](#), [Exception.GetObjectData\(SerializationInfo, StreamingContext\)](#), [Exception.PrepForRemoting\(\)](#), [Exception.OnDeserialized\(StreamingContext\)](#), [Exception.InternalPreserveStackTrace\(\)](#), [Exception.PrepareForForeignExceptionRaise\(\)](#), [Exception.GetStackTracesDeepCopy\(Exception, out object, out object\)](#), [Exception.SaveStackTracesFromDeepCopy\(Exception, object, object\)](#), [Exception.CopyStackTrace\(object\)](#), [Exception.CopyDynamicMethods\(object\)](#), [Exception.StripFileInfo\(string, bool\)](#), [Exception.DeepCopyStackTrace\(object\)](#),

[Exception.DeepCopyDynamicMethods\(object\)](#) ,
[Exception.GetStackTracesDeepCopy\(out object, out object\)](#) ,
[Exception.RestoreExceptionDispatchInfo\(ExceptionDispatchInfo\)](#) , [Exception.InternalToString\(\)](#) ,
[Exception.GetType\(\)](#) , [Exception.nIsTransient\(int\)](#) ,
[Exception.GetMessageFromNativeResources\(Exception.ExceptionMessageKind\)](#) ,
[Exception.GetMessageFromNativeResources\(Exception.ExceptionMessageKind, StringHandleOnStack\)](#) ,
[Exception.Message](#) , [Exception.Data](#) , [Exception.InnerException](#) , [Exception.TargetSite](#) ,
[Exception.StackTrace](#) , [Exception.HelpLink](#) , [Exception.Source](#) , [Exception.IPForWatsonBuckets](#) ,
[Exception.WatsonBuckets](#) , [Exception.RemoteStackTrace](#) , [Exception.HResult](#) ,
[Exception.IsTransient](#) , [Exception.SerializeObjectState](#) , [object.Equals\(object\)](#) ,
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.FieldSetter\(string, string, object\)](#) ,
[object.FieldGetter\(string, string, ref object\)](#) , [object.GetFieldInfo\(string, string\)](#)

Constructors

ProgramModuleExpected(string, SourceContext)

```
public ProgramModuleExpected(string FileName, SourceContext sc)
```

Parameters

FileName [string](#)

sc [SourceContext](#)

Class ReadPCUError

Namespace: [PascalABCCompiler.Errors](#)

Assembly: Compiler.dll

Бросается в случае невозможности чтения рсн

```
public class ReadPCUError : CompilerThrownError, ISerializable, _Exception
```

Inheritance

[object](#) ← [Exception](#) ← Error ← LocatedError ← [CompilerThrownError](#) ← ReadPCUError

Implements

[ISerializable](#), [Exception](#)

Inherited Members

[CompilerThrownError.ToString\(\)](#), LocatedError.source_context, LocatedError.sourceLocation, LocatedError.fileName, LocatedError.SourceLocation, LocatedError.SourceContext, LocatedError.FileName, Error.MustThrow, [Exception.s_EDILock](#), [Exception.className](#), [Exception.exceptionMethod](#), [Exception.exceptionMethodString](#), [Exception.message](#), [Exception.data](#), [Exception.innerException](#), [Exception.helpURL](#), [Exception.stackTrace](#), [Exception.watsonBuckets](#), [Exception.stackTraceString](#), [Exception.remoteStackTraceString](#), [Exception.remoteStackIndex](#), [Exception.dynamicMethods](#), [Exception.HResult](#), [Exception.source](#), [Exception.xptrs](#), [Exception.xcode](#), [Exception.ipForWatsonBuckets](#), [Exception.safeSerializationManager](#), [Exception.COMPlusExceptionCode](#), [Exception.Init\(\)](#), [Exception.IsImmutableAgileException\(Exception\)](#), [Exception.AddExceptionDataForRestrictedErrorInfo\(string, string, string, object, bool\)](#), [Exception.TryGetRestrictedLanguageErrorObject\(out object\)](#), [Exception.GetClassName\(\)](#), [Exception.GetBaseException\(\)](#), [Exception.GetMethodFromStackTrace\(object\)](#), [Exception.GetExceptionMethodFromStackTrace\(\)](#), [Exception.GetTargetSiteInternal\(\)](#), [Exception.GetStackTrace\(bool\)](#), [Exception.SetErrorCode\(int\)](#), [Exception.ToString\(bool, bool\)](#), [Exception.GetExceptionMethodString\(\)](#), [Exception.GetExceptionMethodFromString\(\)](#), [Exception.GetObjectData\(SerializationInfo, StreamingContext\)](#), [Exception.PrepForRemoting\(\)](#), [Exception.OnDeserialized\(StreamingContext\)](#), [Exception.InternalPreserveStackTrace\(\)](#), [Exception.PrepareForForeignExceptionRaise\(\)](#), [Exception.GetStackTracesDeepCopy\(Exception, out object, out object\)](#), [Exception.SaveStackTracesFromDeepCopy\(Exception, object, object\)](#), [Exception.CopyStackTrace\(object\)](#), [Exception.CopyDynamicMethods\(object\)](#), [Exception.StripFileInfo\(string, bool\)](#), [Exception.DeepCopyStackTrace\(object\)](#),

[Exception.DeepCopyDynamicMethods\(object\)](#) ,
[Exception.GetStackTracesDeepCopy\(out object, out object\)](#) ,
[Exception.RestoreExceptionDispatchInfo\(ExceptionDispatchInfo\)](#) , [Exception.InternalToString\(\)](#) ,
[Exception.GetType\(\)](#) , [Exception.nIsTransient\(int\)](#) ,
[Exception.GetMessageFromNativeResources\(Exception.ExceptionMessageKind\)](#) ,
[Exception.GetMessageFromNativeResources\(Exception.ExceptionMessageKind, StringHandleOnStack\)](#) ,
[Exception.Message](#) , [Exception.Data](#) , [Exception.InnerException](#) , [Exception.TargetSite](#) ,
[Exception.StackTrace](#) , [Exception.HelpLink](#) , [Exception.Source](#) , [Exception.IPForWatsonBuckets](#) ,
[Exception.WatsonBuckets](#) , [Exception.RemoteStackTrace](#) , [Exception.HResult](#) ,
[Exception.IsTransient](#) , [Exception.SerializeObjectState](#) , [object.Equals\(object\)](#) ,
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.FieldSetter\(string, string, object\)](#) ,
[object.FieldGetter\(string, string, ref object\)](#) , [object.GetFieldInfo\(string, string\)](#)

Constructors

ReadPCUError(string)

```
public ReadPCUError(string FileName)
```

Parameters

FileName [string](#)

Class ResourceFileNotFoundException

Namespace: [PascalABCCompiler.Errors](#)

Assembly: Compiler.dll

Бросается при неудаче в нахождении файла ресурсов

```
public class ResourceFileNotFoundException : CompilerThrownError, ISerializable, _Exception
```

Inheritance

[object](#) ↗ ← [Exception](#) ↗ ← Error ← LocatedError ← [CompilerThrownError](#) ← ResourceFileNotFoundException

Implements

[ISerializable](#) ↗ , [Exception](#) ↗

Inherited Members

[CompilerThrownError.ToString\(\)](#) , LocatedError.source_context , LocatedError.sourceLocation , LocatedError.fileName , LocatedError.SourceLocation , LocatedError.SourceContext , LocatedError.FileName , Error.MustThrow , [Exception.s_EDILock](#) ↗ , [Exception.className](#) ↗ , [Exception.exceptionMethod](#) ↗ , [Exception.exceptionMethodString](#) ↗ , [Exception.message](#) ↗ , [Exception.data](#) ↗ , [Exception.innerException](#) ↗ , [Exception.helpURL](#) ↗ , [Exception.stackTrace](#) ↗ , [Exception.watsonBuckets](#) ↗ , [Exception.stackTraceString](#) ↗ , [Exception.remoteStackTraceString](#) ↗ , [Exception.remoteStackIndex](#) ↗ , [Exception.dynamicMethods](#) ↗ , [Exception.HResult](#) ↗ , [Exception.source](#) ↗ , [Exception.xptrs](#) ↗ , [Exception.xcode](#) ↗ , [Exception.ipForWatsonBuckets](#) ↗ , [Exception.safeSerializationManager](#) ↗ , [Exception.COMPlusExceptionCode](#) ↗ , [Exception.Init\(\)](#) ↗ , [Exception.IsImmutableAgileException](#)([Exception](#)) ↗ , [Exception.AddExceptionDataForRestrictedErrorInfo\(string, string, string, object, bool\)](#) ↗ , [Exception.TryGetRestrictedLanguageErrorObject\(out object\)](#) ↗ , [Exception.GetClassName\(\)](#) ↗ , [Exception.GetBaseException\(\)](#) ↗ , [Exception.GetMethodFromStackTrace\(object\)](#) ↗ , [Exception.GetExceptionMethodFromStackTrace\(\)](#) ↗ , [Exception.GetTargetSiteInternal\(\)](#) ↗ , [Exception.GetStackTrace\(bool\)](#) ↗ , [Exception.SetErrorCode\(int\)](#) ↗ , [Exception.ToString\(bool, bool\)](#) ↗ , [Exception.GetExceptionMethodString\(\)](#) ↗ , [Exception.GetExceptionMethodFromString\(\)](#) ↗ , [Exception.GetObjectData\(SerializationInfo, StreamingContext\)](#) ↗ , [Exception.PrepForRemoting\(\)](#) ↗ , [Exception.OnDeserialized\(StreamingContext\)](#) ↗ , [Exception.InternalPreserveStackTrace\(\)](#) ↗ , [Exception.PrepareForForeignExceptionRaise\(\)](#) ↗ , [Exception.GetStackTracesDeepCopy\(Exception, out object, out object\)](#) ↗ , [Exception.SaveStackTracesFromDeepCopy\(Exception, object, object\)](#) ↗ , [Exception.CopyStackTrace\(object\)](#) ↗ , [Exception.CopyDynamicMethods\(object\)](#) ↗ , [Exception.StripFileInfo\(string, bool\)](#) ↗ , [Exception.DeepCopyStackTrace\(object\)](#) ↗ ,

[Exception.DeepCopyDynamicMethods\(object\)](#) ,
[Exception.GetStackTracesDeepCopy\(out object, out object\)](#) ,
[Exception.RestoreExceptionDispatchInfo\(ExceptionDispatchInfo\)](#) , [Exception.InternalToString\(\)](#) ,
[Exception.GetType\(\)](#) , [Exception.nIsTransient\(int\)](#) ,
[Exception.GetMessageFromNativeResources\(Exception.ExceptionMessageKind\)](#) ,
[Exception.GetMessageFromNativeResources\(Exception.ExceptionMessageKind, StringHandleOnStack\)](#) ,
[Exception.Message](#) , [Exception.Data](#) , [Exception.InnerException](#) , [Exception.TargetSite](#) ,
[Exception.StackTrace](#) , [Exception.HelpLink](#) , [Exception.Source](#) , [Exception.IPForWatsonBuckets](#) ,
[Exception.WatsonBuckets](#) , [Exception.RemoteStackTrace](#) , [Exception.HResult](#) ,
[Exception.IsTransient](#) , [Exception.SerializeObjectState](#) , [object.Equals\(object\)](#) ,
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.FieldSetter\(string, string, object\)](#) ,
[object.FieldGetter\(string, string, ref object\)](#) , [object.GetFieldInfo\(string, string\)](#)

Constructors

ResourceFileNotFoundException(string, string, SourceContext)

```
public ResourceFileNotFoundException(string fileName, string ResFileName, SourceContext sc)
```

Parameters

fileName [string](#)

ResFileName [string](#)

sc [SourceContext](#)

Class SourceFileNotFoundException

Namespace: [PascalABCCompiler.Errors](#)

Assembly: Compiler.dll

Бросается при невозможности найти исходник по некоторому пути

```
public class SourceFileNotFoundException : CompilerThrownError, ISerializable, _Exception
```

Inheritance

[object](#) ← [Exception](#) ← Error ← LocatedError ← [CompilerThrownError](#) ← SourceFileNotFoundException

Implements

[ISerializable](#), [Exception](#)

Inherited Members

[CompilerThrownError.ToString\(\)](#), LocatedError.source_context, LocatedError.sourceLocation, LocatedError.fileName, LocatedError.SourceLocation, LocatedError.SourceContext, LocatedError.FileName, Error.MustThrow, [Exception.s_EDILock](#), [Exception.className](#), [Exception.exceptionMethod](#), [Exception.exceptionMethodString](#), [Exception.message](#), [Exception.data](#), [Exception.innerException](#), [Exception.helpURL](#), [Exception.stackTrace](#), [Exception.watsonBuckets](#), [Exception.stackTraceString](#), [Exception.remoteStackTraceString](#), [Exception.remoteStackIndex](#), [Exception.dynamicMethods](#), [Exception.HResult](#), [Exception.source](#), [Exception.xptrs](#), [Exception.xcode](#), [Exception.ipForWatsonBuckets](#), [Exception.safeSerializationManager](#), [Exception.COMPlusExceptionCode](#), [Exception.Init\(\)](#), [Exception.IsImmutableAgileException\(Exception\)](#), [Exception.AddExceptionDataForRestrictedErrorInfo\(string, string, string, object, bool\)](#), [Exception.TryGetRestrictedLanguageErrorObject\(out object\)](#), [Exception.GetClassName\(\)](#), [Exception.GetBaseException\(\)](#), [Exception.GetMethodFromStackTrace\(object\)](#), [Exception.GetExceptionMethodFromStackTrace\(\)](#), [Exception.GetTargetSiteInternal\(\)](#), [Exception.GetStackTrace\(bool\)](#), [Exception.SetErrorCode\(int\)](#), [Exception.ToString\(bool, bool\)](#), [Exception.GetExceptionMethodString\(\)](#), [Exception.GetExceptionMethodFromString\(\)](#), [Exception.GetObjectData\(SerializationInfo, StreamingContext\)](#), [Exception.PrepForRemoting\(\)](#), [Exception.OnDeserialized\(StreamingContext\)](#), [Exception.InternalPreserveStackTrace\(\)](#), [Exception.PrepareForForeignExceptionRaise\(\)](#), [Exception.GetStackTracesDeepCopy\(Exception, out object, out object\)](#), [Exception.SaveStackTracesFromDeepCopy\(Exception, object, object\)](#), [Exception.CopyStackTrace\(object\)](#), [Exception.CopyDynamicMethods\(object\)](#), [Exception.StripFileInfo\(string, bool\)](#), [Exception.DeepCopyStackTrace\(object\)](#),

[Exception.DeepCopyDynamicMethods\(object\)](#) ,
[Exception.GetStackTracesDeepCopy\(out object, out object\)](#) ,
[Exception.RestoreExceptionDispatchInfo\(ExceptionDispatchInfo\)](#) , [Exception.InternalToString\(\)](#) ,
[Exception.GetType\(\)](#) , [Exception.nIsTransient\(int\)](#) ,
[Exception.GetMessageFromNativeResources\(Exception.ExceptionMessageKind\)](#) ,
[Exception.GetMessageFromNativeResources\(Exception.ExceptionMessageKind, StringHandleOnStack\)](#) ,
[Exception.Message](#) , [Exception.Data](#) , [Exception.InnerException](#) , [Exception.TargetSite](#) ,
[Exception.StackTrace](#) , [Exception.HelpLink](#) , [Exception.Source](#) , [Exception.IPForWatsonBuckets](#) ,
[Exception.WatsonBuckets](#) , [Exception.RemoteStackTrace](#) , [Exception.HResult](#) ,
[Exception.IsTransient](#) , [Exception.SerializeObjectState](#) , [object.Equals\(object\)](#) ,
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.FieldSetter\(string, string, object\)](#) ,
[object.FieldGetter\(string, string, ref object\)](#) , [object.GetFieldInfo\(string, string\)](#)

Constructors

SourceFileNotFoundException(string)

```
public SourceFileNotFoundException(string FileName)
```

Parameters

FileName [string](#)

Class UnauthorizedAccessToFile

Namespace: [PascalABCCompiler.Errors](#)

Assembly: Compiler.dll

Бросается при попытке файловой операции с недостаточными правами

```
public class UnauthorizedAccessToFile : CompilerThrownError, ISerializable, _Exception
```

Inheritance

[object](#) ← [Exception](#) ← Error ← LocatedError ← [CompilerThrownError](#) ← UnauthorizedAccessToFile

Implements

[ISerializable](#), [Exception](#)

Inherited Members

[CompilerThrownError.ToString\(\)](#), LocatedError.source_context, LocatedError.sourceLocation, LocatedError.fileName, LocatedError.SourceLocation, LocatedError.SourceContext, LocatedError.FileName, Error.MustThrow, [Exception.s_EDILock](#), [Exception.className](#), [Exception.exceptionMethod](#), [Exception.exceptionMethodString](#), [Exception.message](#), [Exception.data](#), [Exception.innerException](#), [Exception.helpURL](#), [Exception.stackTrace](#), [Exception.watsonBuckets](#), [Exception.stackTraceString](#), [Exception.remoteStackTraceString](#), [Exception.remoteStackIndex](#), [Exception.dynamicMethods](#), [Exception.HResult](#), [Exception.source](#), [Exception.xptrs](#), [Exception.xcode](#), [Exception.ipForWatsonBuckets](#), [Exception.safeSerializationManager](#), [Exception.COMPlusExceptionCode](#), [Exception.Init\(\)](#), [Exception.IsImmutableAgileException\(Exception\)](#), [Exception.AddExceptionDataForRestrictedErrorInfo\(string, string, string, object, bool\)](#), [Exception.TryGetRestrictedLanguageErrorObject\(out object\)](#), [Exception.GetClassName\(\)](#), [Exception.GetBaseException\(\)](#), [Exception.GetMethodFromStackTrace\(object\)](#), [Exception.GetExceptionMethodFromStackTrace\(\)](#), [Exception.GetTargetSiteInternal\(\)](#), [Exception.GetStackTrace\(bool\)](#), [Exception.SetErrorCode\(int\)](#), [Exception.ToString\(bool, bool\)](#), [Exception.GetExceptionMethodString\(\)](#), [Exception.GetExceptionMethodFromString\(\)](#), [Exception.GetObjectData\(SerializationInfo, StreamingContext\)](#), [Exception.PrepForRemoting\(\)](#), [Exception.OnDeserialized\(StreamingContext\)](#), [Exception.InternalPreserveStackTrace\(\)](#), [Exception.PrepareForForeignExceptionRaise\(\)](#), [Exception.GetStackTracesDeepCopy\(Exception, out object, out object\)](#), [Exception.SaveStackTracesFromDeepCopy\(Exception, object, object\)](#), [Exception.CopyStackTrace\(object\)](#), [Exception.CopyDynamicMethods\(object\)](#), [Exception.StripFileInfo\(string, bool\)](#), [Exception.DeepCopyStackTrace\(object\)](#),

[Exception.DeepCopyDynamicMethods\(object\)](#) ,
[Exception.GetStackTracesDeepCopy\(out object, out object\)](#) ,
[Exception.RestoreExceptionDispatchInfo\(ExceptionDispatchInfo\)](#) , [Exception.InternalToString\(\)](#) ,
[Exception.GetType\(\)](#) , [Exception.nIsTransient\(int\)](#) ,
[Exception.GetMessageFromNativeResources\(Exception.ExceptionMessageKind\)](#) ,
[Exception.GetMessageFromNativeResources\(Exception.ExceptionMessageKind, StringHandleOnStack\)](#) ,
[Exception.Message](#) , [Exception.Data](#) , [Exception.InnerException](#) , [Exception.TargetSite](#) ,
[Exception.StackTrace](#) , [Exception.HelpLink](#) , [Exception.Source](#) , [Exception.IPForWatsonBuckets](#) ,
[Exception.WatsonBuckets](#) , [Exception.RemoteStackTrace](#) , [Exception.HResult](#) ,
[Exception.IsTransient](#) , [Exception.SerializeObjectState](#) , [object.Equals\(object\)](#) ,
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.FieldSetter\(string, string, object\)](#) ,
[object.FieldGetter\(string, string, ref object\)](#) , [object.GetFieldInfo\(string, string\)](#)

Constructors

UnauthorizedAccessToFile(string)

```
public UnauthorizedAccessToFile(string FileName)
```

Parameters

FileName [string](#)

Class UnitModuleExpected

Namespace: [PascalABCCompiler.Errors](#)

Assembly: Compiler.dll

Бросается, если встречен не модуль (там где он должна быть)

```
public class UnitModuleExpected : CompilerThrownError, ISerializable, _Exception
```

Inheritance

[object](#) ← [Exception](#) ← Error ← LocatedError ← [CompilerThrownError](#) ← UnitModuleExpected

Implements

[ISerializable](#), [Exception](#)

Inherited Members

[CompilerThrownError.ToString\(\)](#), LocatedError.source_context, LocatedError.sourceLocation, LocatedError.fileName, LocatedError.SourceLocation, LocatedError.SourceContext, LocatedError.FileName, Error.MustThrow, [Exception.s_EDILock](#), [Exception.className](#), [Exception.exceptionMethod](#), [Exception.exceptionMethodString](#), [Exception.message](#), [Exception.data](#), [Exception.innerException](#), [Exception.helpURL](#), [Exception.stackTrace](#), [Exception.watsonBuckets](#), [Exception.stackTraceString](#), [Exception.remoteStackTraceString](#), [Exception.remoteStackIndex](#), [Exception.dynamicMethods](#), [Exception.HResult](#), [Exception.source](#), [Exception.xptrs](#), [Exception.xcode](#), [Exception.ipForWatsonBuckets](#), [Exception.safeSerializationManager](#), [Exception.COMPlusExceptionCode](#), [Exception.Init\(\)](#), [Exception.IsImmutableAgileException\(Exception\)](#), [Exception.AddExceptionDataForRestrictedErrorInfo\(string, string, string, object, bool\)](#), [Exception.TryGetRestrictedLanguageErrorObject\(out object\)](#), [Exception.GetClassName\(\)](#), [Exception.GetBaseException\(\)](#), [Exception.GetMethodFromStackTrace\(object\)](#), [Exception.GetExceptionMethodFromStackTrace\(\)](#), [Exception.GetTargetSiteInternal\(\)](#), [Exception.GetStackTrace\(bool\)](#), [Exception.SetErrorCode\(int\)](#), [Exception.ToString\(bool, bool\)](#), [Exception.GetExceptionMethodString\(\)](#), [Exception.GetExceptionMethodFromString\(\)](#), [Exception.GetObjectData\(SerializationInfo, StreamingContext\)](#), [Exception.PrepForRemoting\(\)](#), [Exception.OnDeserialized\(StreamingContext\)](#), [Exception.InternalPreserveStackTrace\(\)](#), [Exception.PrepareForForeignExceptionRaise\(\)](#), [Exception.GetStackTracesDeepCopy\(Exception, out object, out object\)](#), [Exception.SaveStackTracesFromDeepCopy\(Exception, object, object\)](#), [Exception.CopyStackTrace\(object\)](#), [Exception.CopyDynamicMethods\(object\)](#), [Exception.StripFileInfo\(string, bool\)](#), [Exception.DeepCopyStackTrace\(object\)](#),

[Exception.DeepCopyDynamicMethods\(object\)](#) ,
[Exception.GetStackTracesDeepCopy\(out object, out object\)](#) ,
[Exception.RestoreExceptionDispatchInfo\(ExceptionDispatchInfo\)](#) , [Exception.InternalToString\(\)](#) ,
[Exception.GetType\(\)](#) , [Exception.nIsTransient\(int\)](#) ,
[Exception.GetMessageFromNativeResources\(Exception.ExceptionMessageKind\)](#) ,
[Exception.GetMessageFromNativeResources\(Exception.ExceptionMessageKind, StringHandleOnStack\)](#) ,
[Exception.Message](#) , [Exception.Data](#) , [Exception.InnerException](#) , [Exception.TargetSite](#) ,
[Exception.StackTrace](#) , [Exception.HelpLink](#) , [Exception.Source](#) , [Exception.IPForWatsonBuckets](#) ,
[Exception.WatsonBuckets](#) , [Exception.RemoteStackTrace](#) , [Exception.HResult](#) ,
[Exception.IsTransient](#) , [Exception.SerializeObjectState](#) , [object.Equals\(object\)](#) ,
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.FieldSetter\(string, string, object\)](#) ,
[object.FieldGetter\(string, string, ref object\)](#) , [object.GetFieldInfo\(string, string\)](#)

Constructors

UnitModuleExpected(string, SourceContext)

```
public UnitModuleExpected(string FileName, SourceContext sc)
```

Parameters

FileName [string](#)

sc SourceContext

Class UnitModuleExpectedLibraryFound

Namespace: [PascalABCCompiler.Errors](#)

Assembly: Compiler.dll

Бросается при компиляции библиотеки не первой

```
public class UnitModuleExpectedLibraryFound : CompilerThrownError, ISerializable, _Exception
```

Inheritance

```
object ↳ ← Exception ↳ ← Error ← LocatedError ← CompilerThrownError ←  
UnitModuleExpectedLibraryFound
```

Implements

```
ISerializable, _Exception
```

Inherited Members

```
CompilerThrownError.ToString\(\), LocatedError.source_context, LocatedError.sourceLocation,  
LocatedError.fileName, LocatedError.SourceLocation, LocatedError.SourceContext,  
LocatedError.FileName, Error.MustThrow, Exception.s EDILock, Exception.className,  
Exception.exceptionMethod, Exception.exceptionMethodString, Exception.message,  
Exception.data, Exception.innerException, Exception.helpURL, Exception.stackTrace,  
Exception.watsonBuckets, Exception.stackTraceString, Exception.remoteStackTraceString,  
Exception.remoteStackIndex, Exception.dynamicMethods, Exception.HResult,  
Exception.source, Exception.xptrs, Exception.xcode, Exception.ipForWatsonBuckets,  
Exception.safeSerializationManager, Exception.COMPlusExceptionCode, Exception.Init\(\),  
Exception.IsImmutableAgileException\(Exception\),  
Exception.AddExceptionDataForRestrictedErrorInfo\(string, string, string, object, bool\),  
Exception.TryGetRestrictedLanguageErrorObject\(out object\), Exception.GetClassName\(\),  
Exception.GetBaseException\(\), ExceptionGetMethodFromStackTrace\(object\),  
Exception.GetExceptionMethodFromStackTrace\(\), Exception.GetTargetSiteInternal\(\),  
Exception.GetStackTrace\(bool\), Exception.SetErrorCode\(int\), Exception.ToString\(bool, bool\),  
Exception.GetExceptionMethodString\(\), Exception.GetExceptionMethodFromString\(\),  
Exception.GetObjectData\(SerializationInfo, StreamingContext\), Exception.PrepForRemoting\(\),  
Exception.OnDeserialized\(StreamingContext\), Exception.InternalPreserveStackTrace\(\),  
Exception.PrepareForForeignExceptionRaise\(\),  
Exception.GetStackTracesDeepCopy\(Exception, out object, out object\),  
Exception.SaveStackTracesFromDeepCopy\(Exception, object, object\),  
Exception.CopyStackTrace\(object\), Exception.CopyDynamicMethods\(object\),
```

[Exception.StripFileInfo\(string, bool\)](#) , [Exception.DeepCopyStackTrace\(object\)](#) ,
[Exception.DeepCopyDynamicMethods\(object\)](#) ,
[Exception.GetStackTracesDeepCopy\(out object, out object\)](#) ,
[Exception.RestoreExceptionDispatchInfo\(ExceptionDispatchInfo\)](#) , [Exception.InternalToString\(\)](#) ,
[Exception.GetType\(\)](#) , [Exception.nlsTransient\(int\)](#) ,
[Exception.GetMessageFromNativeResources\(Exception.ExceptionMessageKind\)](#) ,
[Exception.GetMessageFromNativeResources\(Exception.ExceptionMessageKind, StringHandleOnStack\)](#) ,
[Exception.Message](#) , [Exception.Data](#) , [Exception.InnerException](#) , [Exception.TargetSite](#) ,
[Exception.StackTrace](#) , [Exception.HelpLink](#) , [Exception.Source](#) , [Exception.IPForWatsonBuckets](#) ,
[Exception.WatsonBuckets](#) , [Exception.RemoteStackTrace](#) , [Exception.HResult](#) ,
[Exception.IsTransient](#) , [Exception.SerializeObjectState](#) , [object.Equals\(object\)](#) ,
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.FieldSetter\(string, string, object\)](#) ,
[object.FieldGetter\(string, string, ref object\)](#) , [object.GetFieldInfo\(string, string\)](#)

Constructors

UnitModuleExpectedLibraryFound(string, SourceContext)

```
public UnitModuleExpectedLibraryFound(string FileName, SourceContext sc)
```

Parameters

FileName [string](#)

sc [SourceContext](#)

Class UnitNotFound

Namespace: [PascalABCCompiler.Errors](#)

Assembly: Compiler.dll

Бросается, если модуль (unit) не найден по некоторому пути

```
public class UnitNotFound : CompilerThrownError, ISerializable, _Exception
```

Inheritance

[object](#) ← [Exception](#) ← Error ← LocatedError ← [CompilerThrownError](#) ← UnitNotFound

Implements

[ISerializable](#), [Exception](#)

Inherited Members

[CompilerThrownError.ToString\(\)](#), LocatedError.source_context, LocatedError.sourceLocation, LocatedError.fileName, LocatedError.SourceLocation, LocatedError.SourceContext, LocatedError.FileName, Error.MustThrow, [Exception.s_EDILock](#), [Exception.className](#), [Exception.exceptionMethod](#), [Exception.exceptionMethodString](#), [Exception.message](#), [Exception.data](#), [Exception.innerException](#), [Exception.helpURL](#), [Exception.stackTrace](#), [Exception.watsonBuckets](#), [Exception.stackTraceString](#), [Exception.remoteStackTraceString](#), [Exception.remoteStackIndex](#), [Exception.dynamicMethods](#), [Exception.HResult](#), [Exception.source](#), [Exception.xptrs](#), [Exception.xcode](#), [Exception.ipForWatsonBuckets](#), [Exception.safeSerializationManager](#), [Exception.COMPlusExceptionCode](#), [Exception.Init\(\)](#), [Exception.IsImmutableAgileException\(Exception\)](#), [Exception.AddExceptionDataForRestrictedErrorInfo\(string, string, string, object, bool\)](#), [Exception.TryGetRestrictedLanguageErrorObject\(out object\)](#), [Exception.GetClassName\(\)](#), [Exception.GetBaseException\(\)](#), [Exception.GetMethodFromStackTrace\(object\)](#), [Exception.GetExceptionMethodFromStackTrace\(\)](#), [Exception.GetTargetSiteInternal\(\)](#), [Exception.GetStackTrace\(bool\)](#), [Exception.SetErrorCode\(int\)](#), [Exception.ToString\(bool, bool\)](#), [Exception.GetExceptionMethodString\(\)](#), [Exception.GetExceptionMethodFromString\(\)](#), [Exception.GetObjectData\(SerializationInfo, StreamingContext\)](#), [Exception.PrepForRemoting\(\)](#), [Exception.OnDeserialized\(StreamingContext\)](#), [Exception.InternalPreserveStackTrace\(\)](#), [Exception.PrepareForForeignExceptionRaise\(\)](#), [Exception.GetStackTracesDeepCopy\(Exception, out object, out object\)](#), [Exception.SaveStackTracesFromDeepCopy\(Exception, object, object\)](#), [Exception.CopyStackTrace\(object\)](#), [Exception.CopyDynamicMethods\(object\)](#), [Exception.StripFileInfo\(string, bool\)](#), [Exception.DeepCopyStackTrace\(object\)](#),

[Exception.DeepCopyDynamicMethods\(object\)](#) ,
[Exception.GetStackTracesDeepCopy\(out object, out object\)](#) ,
[Exception.RestoreExceptionDispatchInfo\(ExceptionDispatchInfo\)](#) , [Exception.InternalToString\(\)](#) ,
[Exception.GetType\(\)](#) , [Exception.nIsTransient\(int\)](#) ,
[Exception.GetMessageFromNativeResources\(Exception.ExceptionMessageKind\)](#) ,
[Exception.GetMessageFromNativeResources\(Exception.ExceptionMessageKind, StringHandleOnStack\)](#) ,
[Exception.Message](#) , [Exception.Data](#) , [Exception.InnerException](#) , [Exception.TargetSite](#) ,
[Exception.StackTrace](#) , [Exception.HelpLink](#) , [Exception.Source](#) , [Exception.IPForWatsonBuckets](#) ,
[Exception.WatsonBuckets](#) , [Exception.RemoteStackTrace](#) , [Exception.HResult](#) ,
[Exception.IsTransient](#) , [Exception.SerializeObjectState](#) , [object.Equals\(object\)](#) ,
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.FieldSetter\(string, string, object\)](#) ,
[object.FieldGetter\(string, string, ref object\)](#) , [object.GetFieldInfo\(string, string\)](#)

Constructors

UnitNotFound(string, string, SourceContext)

```
public UnitNotFound(string FileName, string UnitName, SourceContext sc)
```

Parameters

FileName [string](#)

UnitName [string](#)

sc SourceContext

Fields

UnitName

```
public string UnitName
```

Field Value

[string](#)

Class UnsupportedOutputFileType

Namespace: [PascalABCCompiler.Errors](#)

Assembly: Compiler.dll

Бросается, если пользователь указывает неподдерживаемый тип выходного файла (например, в директиве {\$apptype ...})

```
public class UnsupportedOutputFileType : CompilerThrownError, ISerializable, _Exception
```

Inheritance

[object](#) ← [Exception](#) ← Error ← LocatedError ← [CompilerThrownError](#) ← UnsupportedOutputFileType

Implements

[ISerializable](#), [Exception](#)

Inherited Members

[CompilerThrownError.ToString\(\)](#), LocatedError.source_context, LocatedError.sourceLocation, LocatedError.fileName, LocatedError.SourceLocation, LocatedError.SourceContext, LocatedError.FileName, Error.MustThrow, [Exception.s_EDILock](#), [Exception.className](#), [Exception.exceptionMethod](#), [Exception.exceptionMethodString](#), [Exception.message](#), [Exception.data](#), [Exception.innerException](#), [Exception.helpURL](#), [Exception.stackTrace](#), [Exception.watsonBuckets](#), [Exception.stackTraceString](#), [Exception.remoteStackTraceString](#), [Exception.remoteStackIndex](#), [Exception.dynamicMethods](#), [Exception.HResult](#), [Exception.source](#), [Exception.xptrs](#), [Exception.xcode](#), [Exception.ipForWatsonBuckets](#), [Exception.safeSerializationManager](#), [Exception.COMPlusExceptionCode](#), [Exception.Init\(\)](#), [Exception.IsImmutableAgileException\(Exception\)](#), [Exception.AddExceptionDataForRestrictedErrorInfo\(string, string, string, object, bool\)](#), [Exception.TryGetRestrictedLanguageErrorObject\(out object\)](#), [Exception.GetClassName\(\)](#), [Exception.GetBaseException\(\)](#), [ExceptionGetMethodFromStackTrace\(object\)](#), [Exception.GetExceptionMethodFromStackTrace\(\)](#), [Exception.GetTargetSiteInternal\(\)](#), [Exception.GetStackTrace\(bool\)](#), [Exception.SetErrorCode\(int\)](#), [Exception.ToString\(bool, bool\)](#), [Exception.GetExceptionMethodString\(\)](#), [Exception.GetExceptionMethodFromString\(\)](#), [Exception.GetObjectData\(SerializationInfo, StreamingContext\)](#), [Exception.PrepForRemoting\(\)](#), [Exception.OnDeserialized\(StreamingContext\)](#), [Exception.InternalPreserveStackTrace\(\)](#), [Exception.PrepareForForeignExceptionRaise\(\)](#), [Exception.GetStackTracesDeepCopy\(Exception, out object, out object\)](#), [Exception.SaveStackTracesFromDeepCopy\(Exception, object, object\)](#), [Exception.CopyStackTrace\(object\)](#), [Exception.CopyDynamicMethods\(object\)](#),

[Exception.StripFileInfo\(string, bool\)](#) , [Exception.DeepCopyStackTrace\(object\)](#) ,
[Exception.DeepCopyDynamicMethods\(object\)](#) ,
[Exception.GetStackTracesDeepCopy\(out object, out object\)](#) ,
[Exception.RestoreExceptionDispatchInfo\(ExceptionDispatchInfo\)](#) , [Exception.InternalToString\(\)](#) ,
[Exception.GetType\(\)](#) , [Exception.nlsTransient\(int\)](#) ,
[Exception.GetMessageFromNativeResources\(Exception.ExceptionMessageKind\)](#) ,
[Exception.GetMessageFromNativeResources\(Exception.ExceptionMessageKind, StringHandleOnStack\)](#) ,
[Exception.Message](#) , [Exception.Data](#) , [Exception.InnerException](#) , [Exception.TargetSite](#) ,
[Exception.StackTrace](#) , [Exception.HelpLink](#) , [Exception.Source](#) , [Exception.IPForWatsonBuckets](#) ,
[Exception.WatsonBuckets](#) , [Exception.RemoteStackTrace](#) , [Exception.HResult](#) ,
[Exception.IsTransient](#) , [Exception.SerializeObjectState](#) , [object.Equals\(object\)](#) ,
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.FieldSetter\(string, string, object\)](#) ,
[object.FieldGetter\(string, string, ref object\)](#) , [object.GetFieldInfo\(string, string\)](#)

Constructors

UnsupportedOutputFileType(string, SourceContext)

```
public UnsupportedOutputFileType(string outputFileType, SourceContext sc)
```

Parameters

outputFileType [string](#)

sc [SourceContext](#)

Class UnsupportedTargetFramework

Namespace: [PascalABCCompiler.Errors](#)

Assembly: Compiler.dll

Бросается, если пользователь указывает неподдерживаемый целевой framework

```
public class UnsupportedTargetFramework : CompilerThrownError, ISerializable, _Exception
```

Inheritance

[object](#) ← [Exception](#) ← Error ← LocatedError ← [CompilerThrownError](#) ← UnsupportedTargetFramework

Implements

[ISerializable](#), [Exception](#)

Inherited Members

[CompilerThrownError.ToString\(\)](#), LocatedError.source_context, LocatedError.sourceLocation, LocatedError.fileName, LocatedError.SourceLocation, LocatedError.SourceContext, LocatedError.FileName, Error.MustThrow, [Exception.s_EDILock](#), [Exception.className](#), [Exception.exceptionMethod](#), [Exception.exceptionMethodString](#), [Exception.message](#), [Exception.data](#), [Exception.innerException](#), [Exception.helpURL](#), [Exception.stackTrace](#), [Exception.watsonBuckets](#), [Exception.stackTraceString](#), [Exception.remoteStackTraceString](#), [Exception.remoteStackIndex](#), [Exception.dynamicMethods](#), [Exception.HResult](#), [Exception.source](#), [Exception.xptrs](#), [Exception.xcode](#), [Exception.ipForWatsonBuckets](#), [Exception.safeSerializationManager](#), [Exception.COMPlusExceptionCode](#), [Exception.Init\(\)](#), [Exception.IsImmutableAgileException\(Exception\)](#), [Exception.AddExceptionDataForRestrictedErrorInfo\(string, string, string, object, bool\)](#), [Exception.TryGetRestrictedLanguageErrorObject\(out object\)](#), [Exception.GetClassName\(\)](#), [Exception.GetBaseException\(\)](#), [Exception.GetMethodFromStackTrace\(object\)](#), [Exception.GetExceptionMethodFromStackTrace\(\)](#), [Exception.GetTargetSiteInternal\(\)](#), [Exception.GetStackTrace\(bool\)](#), [Exception.SetErrorCode\(int\)](#), [Exception.ToString\(bool, bool\)](#), [Exception.GetExceptionMethodString\(\)](#), [Exception.GetExceptionMethodFromString\(\)](#), [Exception.GetObjectData\(SerializationInfo, StreamingContext\)](#), [Exception.PrepForRemoting\(\)](#), [Exception.OnDeserialized\(StreamingContext\)](#), [Exception.InternalPreserveStackTrace\(\)](#), [Exception.PrepareForForeignExceptionRaise\(\)](#), [Exception.GetStackTracesDeepCopy\(Exception, out object, out object\)](#), [Exception.SaveStackTracesFromDeepCopy\(Exception, object, object\)](#), [Exception.CopyStackTrace\(object\)](#), [Exception.CopyDynamicMethods\(object\)](#), [Exception.StripFileInfo\(string, bool\)](#), [Exception.DeepCopyStackTrace\(object\)](#),

[Exception.DeepCopyDynamicMethods\(object\)](#) ,
[Exception.GetStackTracesDeepCopy\(out object, out object\)](#) ,
[Exception.RestoreExceptionDispatchInfo\(ExceptionDispatchInfo\)](#) , [Exception.InternalToString\(\)](#) ,
[Exception.GetType\(\)](#) , [Exception.nIsTransient\(int\)](#) ,
[Exception.GetMessageFromNativeResources\(Exception.ExceptionMessageKind\)](#) ,
[Exception.GetMessageFromNativeResources\(Exception.ExceptionMessageKind, StringHandleOnStack\)](#) ,
[Exception.Message](#) , [Exception.Data](#) , [Exception.InnerException](#) , [Exception.TargetSite](#) ,
[Exception.StackTrace](#) , [Exception.HelpLink](#) , [Exception.Source](#) , [Exception.IPForWatsonBuckets](#) ,
[Exception.WatsonBuckets](#) , [Exception.RemoteStackTrace](#) , [Exception.HResult](#) ,
[Exception.IsTransient](#) , [Exception.SerializeObjectState](#) , [object.Equals\(object\)](#) ,
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.FieldSetter\(string, string, object\)](#) ,
[object.FieldGetter\(string, string, ref object\)](#) , [object.GetFieldInfo\(string, string\)](#)

Constructors

UnsupportedTargetFramework(string, SourceContext)

```
public UnsupportedTargetFramework(string FrameworkName, SourceContext sc)
```

Parameters

FrameworkName [string](#)

sc SourceContext

Class UnsupportedTargetPlatform

Namespace: [PascalABCCompiler.Errors](#)

Assembly: Compiler.dll

Бросается, если пользователь указывает неподдерживаемую целевую платформу

```
public class UnsupportedTargetPlatform : CompilerThrownError, ISerializable, _Exception
```

Inheritance

[object](#) ← [Exception](#) ← Error ← LocatedError ← [CompilerThrownError](#) ← UnsupportedTargetPlatform

Implements

[ISerializable](#), [Exception](#)

Inherited Members

[CompilerThrownError.ToString\(\)](#), LocatedError.source_context, LocatedError.sourceLocation, LocatedError.fileName, LocatedError.SourceLocation, LocatedError.SourceContext, LocatedError.FileName, Error.MustThrow, [Exception.s_EDILock](#), [Exception.className](#), [Exception.exceptionMethod](#), [Exception.exceptionMethodString](#), [Exception.message](#), [Exception.data](#), [Exception.innerException](#), [Exception.helpURL](#), [Exception.stackTrace](#), [Exception.watsonBuckets](#), [Exception.stackTraceString](#), [Exception.remoteStackTraceString](#), [Exception.remoteStackIndex](#), [Exception.dynamicMethods](#), [Exception.HResult](#), [Exception.source](#), [Exception.xptrs](#), [Exception.xcode](#), [Exception.ipForWatsonBuckets](#), [Exception.safeSerializationManager](#), [Exception.COMPlusExceptionCode](#), [Exception.Init\(\)](#), [Exception.IsImmutableAgileException\(Exception\)](#), [Exception.AddExceptionDataForRestrictedErrorInfo\(string, string, string, object, bool\)](#), [Exception.TryGetRestrictedLanguageErrorObject\(out object\)](#), [Exception.GetClassName\(\)](#), [Exception.GetBaseException\(\)](#), [Exception.GetMethodFromStackTrace\(object\)](#), [Exception.GetExceptionMethodFromStackTrace\(\)](#), [Exception.GetTargetSiteInternal\(\)](#), [Exception.GetStackTrace\(bool\)](#), [Exception.SetErrorCode\(int\)](#), [Exception.ToString\(bool, bool\)](#), [Exception.GetExceptionMethodString\(\)](#), [Exception.GetExceptionMethodFromString\(\)](#), [Exception.GetObjectData\(SerializationInfo, StreamingContext\)](#), [Exception.PrepForRemoting\(\)](#), [Exception.OnDeserialized\(StreamingContext\)](#), [Exception.InternalPreserveStackTrace\(\)](#), [Exception.PrepareForForeignExceptionRaise\(\)](#), [Exception.GetStackTracesDeepCopy\(Exception, out object, out object\)](#), [Exception.SaveStackTracesFromDeepCopy\(Exception, object, object\)](#), [Exception.CopyStackTrace\(object\)](#), [Exception.CopyDynamicMethods\(object\)](#), [Exception.StripFileInfo\(string, bool\)](#), [Exception.DeepCopyStackTrace\(object\)](#),

[Exception.DeepCopyDynamicMethods\(object\)](#) ,
[Exception.GetStackTracesDeepCopy\(out object, out object\)](#) ,
[Exception.RestoreExceptionDispatchInfo\(ExceptionDispatchInfo\)](#) , [Exception.InternalToString\(\)](#) ,
[Exception.GetType\(\)](#) , [Exception.nIsTransient\(int\)](#) ,
[Exception.GetMessageFromNativeResources\(Exception.ExceptionMessageKind\)](#) ,
[Exception.GetMessageFromNativeResources\(Exception.ExceptionMessageKind, StringHandleOnStack\)](#) ,
[Exception.Message](#) , [Exception.Data](#) , [Exception.InnerException](#) , [Exception.TargetSite](#) ,
[Exception.StackTrace](#) , [Exception.HelpLink](#) , [Exception.Source](#) , [Exception.IPForWatsonBuckets](#) ,
[Exception.WatsonBuckets](#) , [Exception.RemoteStackTrace](#) , [Exception.HResult](#) ,
[Exception.IsTransient](#) , [Exception.SerializeObjectState](#) , [object.Equals\(object\)](#) ,
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.FieldSetter\(string, string, object\)](#) ,
[object.FieldGetter\(string, string, ref object\)](#) , [object.GetFieldInfo\(string, string\)](#)

Constructors

UnsupportedTargetPlatform(string, SourceContext)

```
public UnsupportedTargetPlatform(string platformName, SourceContext sc)
```

Parameters

platformName [string](#)

sc SourceContext

Class UsesInWrongName

Namespace: [PascalABCCompiler.Errors](#)

Assembly: Compiler.dll

Бросается при некорректном пути в uses in

```
public class UsesInWrongName : CompilerThrownError, ISerializable, _Exception
```

Inheritance

[object](#) ← [Exception](#) ← Error ← LocatedError ← [CompilerThrownError](#) ← UsesInWrongName

Implements

[ISerializable](#), [Exception](#)

Inherited Members

[CompilerThrownError.ToString\(\)](#), LocatedError.source_context, LocatedError.sourceLocation, LocatedError.fileName, LocatedError.SourceLocation, LocatedError.SourceContext, LocatedError.FileName, Error.MustThrow, [Exception.s_EDILock](#), [Exception.className](#), [Exception.exceptionMethod](#), [Exception.exceptionMethodString](#), [Exception.message](#), [Exception.data](#), [Exception.innerException](#), [Exception.helpURL](#), [Exception.stackTrace](#), [Exception.watsonBuckets](#), [Exception.stackTraceString](#), [Exception.remoteStackTraceString](#), [Exception.remoteStackIndex](#), [Exception.dynamicMethods](#), [Exception.HResult](#), [Exception.source](#), [Exception.xptrs](#), [Exception.xcode](#), [Exception.ipForWatsonBuckets](#), [Exception.safeSerializationManager](#), [Exception.COMPlusExceptionCode](#), [Exception.Init\(\)](#), [Exception.IsImmutableAgileException\(Exception\)](#), [Exception.AddExceptionDataForRestrictedErrorInfo\(string, string, string, object, bool\)](#), [Exception.TryGetRestrictedLanguageErrorObject\(out object\)](#), [Exception.GetClassName\(\)](#), [Exception.GetBaseException\(\)](#), [Exception.GetMethodFromStackTrace\(object\)](#), [Exception.GetExceptionMethodFromStackTrace\(\)](#), [Exception.GetTargetSiteInternal\(\)](#), [Exception.GetStackTrace\(bool\)](#), [Exception.SetErrorCode\(int\)](#), [Exception.ToString\(bool, bool\)](#), [Exception.GetExceptionMethodString\(\)](#), [Exception.GetExceptionMethodFromString\(\)](#), [Exception.GetObjectData\(SerializationInfo, StreamingContext\)](#), [Exception.PrepForRemoting\(\)](#), [Exception.OnDeserialized\(StreamingContext\)](#), [Exception.InternalPreserveStackTrace\(\)](#), [Exception.PrepareForForeignExceptionRaise\(\)](#), [Exception.GetStackTracesDeepCopy\(Exception, out object, out object\)](#), [Exception.SaveStackTracesFromDeepCopy\(Exception, object, object\)](#), [Exception.CopyStackTrace\(object\)](#), [Exception.CopyDynamicMethods\(object\)](#), [Exception.StripFileInfo\(string, bool\)](#), [Exception.DeepCopyStackTrace\(object\)](#),

[Exception.DeepCopyDynamicMethods\(object\)](#) ,
[Exception.GetStackTracesDeepCopy\(out object, out object\)](#) ,
[Exception.RestoreExceptionDispatchInfo\(ExceptionDispatchInfo\)](#) , [Exception.InternalToString\(\)](#) ,
[Exception.GetType\(\)](#) , [Exception.nIsTransient\(int\)](#) ,
[Exception.GetMessageFromNativeResources\(Exception.ExceptionMessageKind\)](#) ,
[Exception.GetMessageFromNativeResources\(Exception.ExceptionMessageKind, StringHandleOnStack\)](#) ,
[Exception.Message](#) , [Exception.Data](#) , [Exception.InnerException](#) , [Exception.TargetSite](#) ,
[Exception.StackTrace](#) , [Exception.HelpLink](#) , [Exception.Source](#) , [Exception.IPForWatsonBuckets](#) ,
[Exception.WatsonBuckets](#) , [Exception.RemoteStackTrace](#) , [Exception.HResult](#) ,
[Exception.IsTransient](#) , [Exception.SerializeObjectState](#) , [object.Equals\(object\)](#) ,
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.FieldSetter\(string, string, object\)](#) ,
[object.FieldGetter\(string, string, ref object\)](#) , [object.GetFieldInfo\(string, string\)](#)

Constructors

UsesInWrongName(string, string, string, SourceContext)

```
public UsesInWrongName(string FileName, string UnitName1, string UnitName2,  
SourceContext sc)
```

Parameters

FileName [string](#)

UnitName1 [string](#)

UnitName2 [string](#)

sc [SourceContext](#)

Fields

UnitName1

```
public string UnitName1
```

Field Value

[string](#) ↗

UnitName2

```
public string UnitName2
```

Field Value

[string](#) ↗

Namespace PascalABCCompiler.PCU

Classes

[ClassInfo](#)

[DotAdditInfo](#)

[DotNetNameRef](#)

[ImportedEntity](#)

[InvalidPCUFile](#)

[NameRef](#)

[PCUConsts](#)

[PCUFile](#)

Класс, описывающий заголовок PCU-файла

[PCUFileVersion](#)

[PCUReader](#)

[PCUReader.PCUFileHeadState](#)

Читает начало заголовка PCU файла

[PCUReader.TypeSpec](#)

[PCUWriter](#)

Класс, создающий PCU-модуль

[WrappedClassScope](#)

[WrappedInterfaceScope](#)

[WrappedUnitImplementationScope](#)

[WrappedUnitInterfaceScope](#)

[wrapped_common_type_node](#)

[wrapped_expression](#)

[wrapped_function_body](#)

[wrapped_type_synonym](#)

Enums

[DotNetKind](#)

[GenericParamKind](#)

[ImportKind](#)

[PCUReaderWriterState](#)

[TypeKind](#)

Delegates

[PCUReader.ChangeStateDelegate](#)

[PCUWriter.ChangeStateDelegate](#)

Class ClassInfo

Namespace: [PascalABCCompiler.PCU](#)

Assembly: Compiler.dll

```
public class ClassInfo
```

Inheritance

[object](#) ← ClassInfo

Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.FieldSetter\(string, string, object\)](#) ,
[object.FieldGetter\(string, string, ref object\)](#) , [object.GetFieldInfo\(string, string\)](#)

Constructors

ClassInfo(int, int, int, int, NameRef[])

```
public ClassInfo(int names_pos, int def_prop_off, int base_class_off, int interf_impl_off,  
NameRef[] names)
```

Parameters

names_pos [int](#)

def_prop_off [int](#)

base_class_off [int](#)

interf_impl_off [int](#)

names [NameRef](#)[]

Fields

base_class_off

```
public int base_class_off
```

Field Value

[int↗](#)

def_prop_off

```
public int def_prop_off
```

Field Value

[int↗](#)

interf_impl_off

```
public int interf_impl_off
```

Field Value

[int↗](#)

names

```
public NameRef[] names
```

Field Value

[NameRef\[\]](#)

names_pos

```
public int names_pos
```

Field Value

[int ↗](#)

Class DotAdditInfo

Namespace: [PascalABCCompiler.PCU](#)

Assembly: Compiler.dll

```
public class DotAdditInfo
```

Inheritance

[object](#) ← DotAdditInfo

Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.FieldSetter\(string, string, object\)](#) ,
[object.FieldGetter\(string, string, ref object\)](#) , [object.GetFieldInfo\(string, string\)](#)

Constructors

DotAdditInfo()

```
public DotAdditInfo()
```

Fields

offset

```
public int offset
```

Field Value

[int](#)

Enum DotNetKind

Namespace: [PascalABCCompiler.PCU](#)

Assembly: Compiler.dll

```
public enum DotNetKind
```

Fields

Constructor = 2

Field = 0

Method = 1

Property = 4

Type = 3

Class DotNameRef

Namespace: [PascalABCCompiler.PCU](#)

Assembly: Compiler.dll

```
public class DotNameRef
```

Inheritance

[object](#) ← DotNameRef

Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.FieldSetter\(string, string, object\)](#) ,
[object.FieldGetter\(string, string, ref object\)](#) , [object.GetFieldInfo\(string, string\)](#)

Constructors

DotNameRef()

```
public DotNameRef()
```

Fields

addit

```
public DotAdditInfo[] addit
```

Field Value

[DotAdditInfo\[\]](#)

kind

```
public DotNetKind kind
```

Field Value

[DotNetKind](#)

name

```
public string name
```

Field Value

[string](#) ↗

Enum GenericParamKind

Namespace: [PascalABCCompiler.PCU](#)

Assembly: Compiler.dll

```
public enum GenericParamKind
```

Fields

Class = 1

None = 0

Value = 2

Enum ImportKind

Namespace: [PascalABCCompiler.PCU](#)

Assembly: Compiler.dll

```
public enum ImportKind
```

Fields

Common = 0

DotNet = 1

Class ImportedEntity

Namespace: [PascalABCCompiler.PCU](#)

Assembly: Compiler.dll

```
public class ImportedEntity
```

Inheritance

[object](#) ← ImportedEntity

Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.FieldSetter\(string, string, object\)](#) ,
[object.FieldGetter\(string, string, ref object\)](#) , [object.GetFieldInfo\(string, string\)](#)

Constructors

ImportedEntity()

```
public ImportedEntity()
```

Fields

flag

```
public ImportKind flag
```

Field Value

[ImportKind](#)

index

```
public int index
```

Field Value

[int](#)

num_unit

```
public int num_unit
```

Field Value

[int](#)

offset

```
public int offset
```

Field Value

[int](#)

Methods

GetClassSize()

```
public static int GetClassSize()
```

Returns

[int](#)

GetSize()

```
public int GetSize()
```

Returns

[int](#)

Class InvalidPCUFile

Namespace: [PascalABCCompiler.PCU](#)

Assembly: Compiler.dll

```
public class InvalidPCUFile : LocatedError, ISerializable, _Exception
```

Inheritance

[object](#) ← [Exception](#) ← Error ← LocatedError ← InvalidPCUFile

Implements

[ISerializable](#), [Exception](#)

Inherited Members

LocatedError.source_context , LocatedError.sourceLocation , LocatedError.fileName , LocatedError.SourceLocation , LocatedError.SourceContext , LocatedError.FileName , Error.MustThrow , [Exception.s_EDILock](#) , [Exception.className](#) , [Exception.exceptionMethod](#) , [Exception.exceptionMethodString](#) , [Exception.message](#) , [Exception.data](#) , [Exception.innerException](#) , [Exception.helpURL](#) , [Exception.stackTrace](#) , [Exception.watsonBuckets](#) , [Exception.stackTraceString](#) , [Exception.remoteStackTraceString](#) , [Exception.remoteStackIndex](#) , [Exception.dynamicMethods](#) , [Exception.HResult](#) , [Exception.source](#) , [Exception.xptrs](#) , [Exception.xcode](#) , [Exception.ipForWatsonBuckets](#) , [Exception.safeSerializationManager](#) , [Exception.COMPlusExceptionCode](#) , [Exception.Init\(\)](#) , [Exception.IsImmutableAgileException](#)([Exception](#)) , [Exception.AddExceptionDataForRestrictedErrorInfo](#)([string](#), [string](#), [string](#), [object](#), [bool](#)) , [Exception.TryGetRestrictedLanguageErrorObject](#)([out object](#)) , [Exception.GetClassName\(\)](#) , [Exception.GetBaseException\(\)](#) , [ExceptionGetMethodFromStackTrace](#)([object](#)) , [Exception.GetExceptionMethodFromStackTrace](#)() , [Exception.GetTargetSiteInternal](#)() , [Exception.GetStackTrace](#)([bool](#)) , [Exception.SetErrorCode](#)([int](#)) , [Exception.ToString](#)([bool](#), [bool](#)) , [Exception.GetExceptionMethodString](#)() , [Exception.GetExceptionMethodFromString](#)() , [Exception.GetObjectData](#)([SerializationInfo](#), [StreamingContext](#)) , [Exception.PrepForRemoting](#)() , [Exception.OnDeserialized](#)([StreamingContext](#)) , [Exception.InternalPreserveStackTrace](#)() , [Exception.PrepareForForeignExceptionRaise](#)() , [Exception.GetStackTracesDeepCopy](#)([Exception](#), [out object](#), [out object](#)) , [Exception.SaveStackTracesFromDeepCopy](#)([Exception](#), [object](#), [object](#)) , [Exception.CopyStackTrace](#)([object](#)) , [Exception.CopyDynamicMethods](#)([object](#)) , [Exception.StripFileInfo](#)([string](#), [bool](#)) , [Exception.DeepCopyStackTrace](#)([object](#)) , [Exception.DeepCopyDynamicMethods](#)([object](#)) , [Exception.GetStackTracesDeepCopy](#)([out object](#), [out object](#)) ,

[Exception.RestoreExceptionDispatchInfo\(ExceptionDispatchInfo\)](#) , [Exception.InternalToString\(\)](#) ,
[Exception.GetType\(\)](#) , [Exception.nIsTransient\(int\)](#) ,
[Exception.GetMessageFromNativeResources\(Exception.ExceptionMessageKind\)](#) ,
[Exception.GetMessageFromNativeResources\(Exception.ExceptionMessageKind, StringHandleOnStack\)](#) ,
[Exception.Message](#) , [Exception.Data](#) , [Exception.InnerException](#) , [Exception.TargetSite](#) ,
[Exception.StackTrace](#) , [Exception.HelpLink](#) , [Exception.Source](#) , [Exception.IPForWatsonBuckets](#) ,
[Exception.WatsonBuckets](#) , [Exception.RemoteStackTrace](#) , [Exception.HResult](#) ,
[Exception.IsTransient](#) , [Exception.SerializeObjectState](#) , [object.Equals\(object\)](#) ,
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.FieldSetter\(string, string, object\)](#) ,
[object.FieldGetter\(string, string, ref object\)](#) , [object.GetFieldInfo\(string, string\)](#)

Constructors

InvalidPCUFile(string)

```
public InvalidPCUFile(string UnitName)
```

Parameters

UnitName [string](#)

Fields

UnitName

```
internal string UnitName
```

Field Value

[string](#)

Methods

ToString()

```
public override string ToString()
```

Returns

[string](#)

Class NameRef

Namespace: [PascalABCCompiler.PCU](#)

Assembly: Compiler.dll

```
public class NameRef
```

Inheritance

[object](#) ← NameRef

Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.FieldSetter\(string, string, object\)](#) ,
[object.FieldGetter\(string, string, ref object\)](#) , [object.GetFieldInfo\(string, string\)](#)

Constructors

NameRef(string, int)

```
public NameRef(string name, int index)
```

Parameters

name [string](#)

index [int](#)

NameRef(string, int, access_level, semantic_node_type)

```
public NameRef(string name, int index, access_level access_level,  
semantic_node_type semantic_node_type)
```

Parameters

name [string](#)

`index` [int](#)

`access_level` `access_level`

`semantic_node_type` `semantic_node_type`

Fields

`access_level`

`public access_level access_level`

Field Value

`access_level`

`always_restore`

`public bool always_restore`

Field Value

[bool](#)

`index`

`public int index`

Field Value

[int](#)

`is_static`

```
public bool is_static
```

Field Value

[bool](#)

name

```
public string name
```

Field Value

[string](#)

offset

```
public int offset
```

Field Value

[int](#)

semantic_node_type

```
public semantic_node_type semantic_node_type
```

Field Value

semantic_node_type

special_scope

```
public byte special_scope
```

Field Value

[byte](#) ↗

symbol_kind

```
public symbol_kind symbol_kind
```

Field Value

symbol_kind

Properties

Size

```
public int Size { get; }
```

Property Value

[int](#) ↗

Class PCUConsts

Namespace: [PascalABCCompiler.PCU](#)

Assembly: Compiler.dll

```
public class PCUConsts
```

Inheritance

[object](#) ← PCUConsts

Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.FieldSetter\(string, string, object\)](#) ,
[object.FieldGetter\(string, string, ref object\)](#) , [object.GetFieldInfo\(string, string\)](#)

Constructors

PCUConsts()

```
public PCUConsts()
```

Fields

common_method_generic

```
public const byte common_method_generic = 230
```

Field Value

[byte](#)

common_namespace_generic

```
public const byte common_namespace_generic = 228
```

Field Value

[byte](#) ↗

compiled_method_generic

```
public const byte compiled_method_generic = 229
```

Field Value

[byte](#) ↗

generic_ctor

```
public const byte generic_ctor = 239
```

Field Value

[byte](#) ↗

generic_field

```
public const byte generic_field = 238
```

Field Value

[byte](#) ↗

generic_meth

```
public const byte generic_meth = 240
```

Field Value

[byte](#) ↗

generic_param_ctor

```
public const byte generic_param_ctor = 236
```

Field Value

[byte](#) ↗

generic_prop

```
public const byte generic_prop = 237
```

Field Value

[byte](#) ↗

method_instance_as_compiled_function_node

```
public const int method_instance_as_compiled_function_node = -1
```

Field Value

[int](#) ↗

template_field

```
public const byte template_field = 249
```

Field Value

[byte](#) ↗

template_meth

```
public const byte template_meth = 250
```

Field Value

[byte](#) ↗

template_prop

```
public const byte template_prop = 248
```

Field Value

[byte](#) ↗

Class PCUFile

Namespace: [PascalABCCompiler.PCU](#)

Assembly: Compiler.dll

Класс, описывающий заголовок PCU-файла

```
public class PCUFile
```

Inheritance

[object](#) ← PCUFile

Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.FieldSetter\(string, string, object\)](#) ,
[object.FieldGetter\(string, string, ref object\)](#) , [object.GetFieldInfo\(string, string\)](#)

Constructors

PCUFile()

```
public PCUFile()
```

Fields

CRC

```
public long CRC
```

Field Value

[long](#)

CRCOffset

```
public static int CRCOffset
```

Field Value

[int↗](#)

Header

```
public static char[] Header
```

Field Value

[char↗\[\]](#)

IncludeDebugInfo

```
public bool IncludeDebugInfo
```

Field Value

[bool↗](#)

Revision

```
public int Revision
```

Field Value

[int↗](#)

SourceFileName

```
public string SourceFileName
```

Field Value

[string](#)

SupportedRevision

```
public static int SupportedRevision
```

Field Value

[int](#)

SupportedVersion

```
public static short SupportedVersion
```

Field Value

[short](#)

UseRtlDll

```
public bool UseRtlDll
```

Field Value

[bool](#)

Version

```
public short Version
```

Field Value

[short](#)

compiler_directives

```
public List<compiler_directive> compiler_directives
```

Field Value

[List](#)<compiler_directive>

dotnet_names

```
public DotNetNameRef[] dotnet_names
```

Field Value

[DotNetNameRef](#)[]

imp_entities

```
public ImportedEntity[] imp_entities
```

Field Value

[ImportedEntity](#)[]

implementation_names

```
public NameRef[] implementation_names
```

Field Value

[NameRef\[\]](#)

implementation_synonyms_offset

```
public int implementation_synonyms_offset
```

Field Value

[int ↗](#)

incl_modules

```
public string[] incl_modules
```

Field Value

[string\[\] ↗](#)

interface_synonyms_offset

```
public int interface_synonyms_offset
```

Field Value

[int ↗](#)

interface_uses_count

```
public int interface_uses_count
```

Field Value

[int](#)

names

```
public NameRef[] names
```

Field Value

[NameRef\[\]](#)

ref_assemblies

```
public string[] ref_assemblies
```

Field Value

[string\[\]](#)

used_namespaces

```
public string[] used_namespaces
```

Field Value

[string\[\]](#)

Class PCUFormatVersion

Namespace: [PascalABCCompiler.PCU](#)

Assembly: Compiler.dll

```
public static class PCUFormatVersion
```

Inheritance

[object](#) ← PCUFormatVersion

Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.FieldSetter\(string, string, object\)](#) ,
[object.FieldGetter\(string, string, ref object\)](#) , [object.GetFieldInfo\(string, string\)](#)

Fields

Version

```
public static short Version
```

Field Value

[short](#)

Class PCUReader

Namespace: [PascalABCCompiler.PCU](#)

Assembly: Compiler.dll

```
public class PCUReader : BasePCUReader
```

Inheritance

[object](#) ← BasePCUReader ← PCUReader

Inherited Members

[BasePCUReader.RestoreSymbolsInterfaceMember\(List<SymbolInfo>, string\)](#) ,
[BasePCUReader.RestoreSymbol\(SymbolInfo, string, int\)](#) ,
[BasePCUReader.RestoreSymbols\(List<SymbolInfo>, string, int\)](#) ,
[BasePCUReader.RestoreSymbols\(SymbolInfo, wrapped_definition_node, string\)](#) , [object.ToString\(\)](#) ,
[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) ,
[object.GetHashCode\(\)](#) , [object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) ,
[object.FieldSetter\(string, string, object\)](#) , [object.FieldGetter\(string, string, ref object\)](#) ,
[object.GetFieldInfo\(string, string\)](#)

Constructors

PCUReader(Compiler, ChangeStateDelegate)

```
public PCUReader(Compiler comp, PCUReader.ChangeStateDelegate changeState)
```

Parameters

comp [Compiler](#)

changeState [PCUReader.ChangeStateDelegate](#)

PCUReader(PCUReader)

```
public PCUReader(PCUReader Reader)
```

Parameters

Reader [PCUReader](#)

Fields

AllReadOrWrittenDefinitionNodesOffsets

```
internal static Dictionary<definition_node, int> AllReadOrWrittenDefinitionNodesOffsets
```

Field Value

[Dictionary](#)<definition_node, [int](#)>

AllReaders

```
public static List<PCUReader> AllReaders
```

Field Value

[List](#)<[PCUReader](#)>

FileName

```
public string FileName
```

Field Value

[string](#)

FinalizationMethodOffset

```
private int FinalizationMethodOffset
```

Field Value

[int](#)

InitializationMethodOffset

```
private int InitializationMethodOffset
```

Field Value

[int](#)

already_compiled

```
private Hashtable already_compiled
```

Field Value

[Hashtable](#)

assemblies

```
private Dictionary<string, Assembly> assemblies
```

Field Value

[Dictionary](#)<[string](#), [Assembly](#)>

br

```
private BinaryReader br
```

Field Value

class_names

```
private Dictionary<common_type_node, string[]> class_names
```

Field Value

[Dictionary](#)<common_type_node, [string](#)[]>

comp

```
internal Compiler comp
```

Field Value

[Compiler](#)

cun

```
private common_unit_node cun
```

Field Value

common_unit_node

cur_doc

```
private document cur_doc
```

Field Value

document

dir

```
private string dir
```

Field Value

[string](#)

dot_net_cache

```
private Dictionary<int, MemberInfo> dot_net_cache
```

Field Value

[Dictionary](#)<[int](#), [MemberInfo](#)>

dot_net_type_cache

```
private Dictionary<int, Type> dot_net_type_cache
```

Field Value

[Dictionary](#)<[int](#), [Type](#)>

ext_members

```
private Dictionary<int, definition_node> ext_members
```

Field Value

[Dictionary](#)<[int](#), [definition_node](#)>

ext_pos

```
private int ext_pos
```

Field Value

[int](#)

impl_members

```
private List<definition_node> impl_members
```

Field Value

[List](#)<definition_node>

impl_type_list

```
private SortedDictionary<int, common_type_node> impl_type_list
```

Field Value

[SortedDictionary](#)<[int](#), common_type_node>

impl_var_list

```
private SortedDictionary<int, var_definition_node> impl_var_list
```

Field Value

[SortedDictionary](#)<[int](#), var_definition_node>

int_members

```
private List<definition_node> int_members
```

Field Value

[List](#) <definition_node>

interf_type_list

```
private SortedDictionary<int, common_type_node> interf_type_list
```

Field Value

[SortedDictionary](#) <int, common_type_node>

interf_var_list

```
private SortedDictionary<int, var_definition_node> interf_var_list
```

Field Value

[SortedDictionary](#) <int, var_definition_node>

members

```
private Dictionary<int, definition_node> members
```

Field Value

[Dictionary](#) <int, definition_node>

ms

```
private MemoryStream ms
```

Field Value

[MemoryStream](#) ↗

need

```
public bool need
```

Field Value

[bool](#) ↗

pcu_file

```
private PCUFile pcu_file
```

Field Value

[PCUFile](#)

readDebugInfo

```
private bool readDebugInfo
```

Field Value

[bool](#) ↗

start_pos

```
private int start_pos
```

Field Value

[int](#)

unit

```
private CompilationUnit unit
```

Field Value

[CompilationUnit](#)

unit_name

```
private string unit_name
```

Field Value

[string](#)

units

```
public static Hashtable units
```

Field Value

[Hashtable](#)

used_units

```
private Hashtable used_units
```

Field Value

[Hashtable](#)

waited_method_codes

```
private Dictionary<common_method_node, int> waited_method_codes
```

Field Value

[Dictionary](#)<common_method_node, [int](#)>

waited_method_restoring

```
private bool waited_method_restoring
```

Field Value

[bool](#)

waited_types_to_restore_fields

```
internal List<common_type_node> waited_types_to_restore_fields
```

Field Value

[List](#)<common_type_node>

Methods

AddAlreadyCompiledUnit(string)

```
public void AddAlreadyCompiledUnit(string name)
```

Parameters

name [string](#)

AddClassMemberNames(WrappedClassScope)

```
private string[] AddClassMemberNames(WrappedClassScope scope)
```

Parameters

scope [WrappedClassScope](#)

Returns

[string](#)[]

AddEnumOperators(common_type_node)

```
private void AddEnumOperators(common_type_node tctn)
```

Parameters

tctn common_type_node

AddFinalizationMethod()

```
private void AddFinalizationMethod()
```

AddImplTypeToOrderList(common_type_node, int)

```
internal void AddImplTypeToOrderList(common_type_node ctn, int ind)
```

Parameters

ctn common_type_node

ind [int](#)

AddImplVarToOrderList(var_definition_node, int)

```
internal void AddImplVarToOrderList(var_definition_node vdn, int ind)
```

Parameters

vdn var_definition_node

ind [int](#)

AddImplementationNames()

```
private void AddImplementationNames()
```

AddInitFinalMethods()

```
public void AddInitFinalMethods()
```

AddInitializationMethod()

```
private void AddInitializationMethod()
```

AddInterfaceNames()

```
private void AddInterfaceNames()
```

AddMember(definition_node, int)

```
internal void AddMember(definition_node dn, int offset)
```

Parameters

dn definition_node

offset [int](#)

AddMembersToNamespace()

```
public void AddMembersToNamespace()
```

AddNames(NameRef[], Scope)

```
private void AddNames(NameRef[] names, Scope Scope)
```

Parameters

names [NameRef](#)[]

Scope Scope

AddNamespaces()

```
private void AddNamespaces()
```

AddReadOrWrittenDefinitionNode(definition_node, int)

```
internal static void AddReadOrWrittenDefinitionNode(definition_node dn, int offset)
```

Parameters

dn definition_node

offset [int](#)

AddTypeSynonyms(int, Scope)

```
private void AddTypeSynonyms(int offset, Scope scope)
```

Parameters

offset [int](#)

scope Scope

AddTypeToOrderList(common_type_node, int)

```
internal void AddTypeToOrderList(common_type_node ctn, int ind)
```

Parameters

ctn common_type_node

ind [int](#)

AddUsedMembersInAllUnits()

```
public static void AddUsedMembersInAllUnits()
```

AddVarToOrderList(var_definition_node, int)

```
internal void AddVarToOrderList(var_definition_node vdn, int ind)
```

Parameters

vdn var_definition_node

ind [int](#)

AddWaitedMethodCode(common_method_node, int)

```
public void AddWaitedMethodCode(common_method_node cmn, int offset)
```

Parameters

cmn common_method_node

offset [int](#)

AlreadyCompiled(string)

```
public bool AlreadyCompiled(string name)
```

Parameters

name [string](#)

Returns

[bool](#)

CanReadObject()

```
private bool CanReadObject()
```

Returns

[bool](#)

ChooseConstructor(Type, TypeSpec[])

```
private ConstructorInfo ChooseConstructor(Type t, PCUReader.TypeSpec[] param_types)
```

Parameters

t [Type](#)

param_types [TypeSpec](#)[]

Returns

[ConstructorInfo](#)

ChooseMethod(Type, IList<MethodInfo>, TypeSpec[])

```
private MethodInfo ChooseMethod(Type t, IList<MethodInfo> mis,
PCUReader.TypeSpec[] param_types)
```

Parameters

t [Type](#)

mis [IList](#)<[MethodInfo](#)>

param_types [TypeSpec](#)[]

Returns

[MethodInfo](#)

CloseUnit()

```
public void CloseUnit()
```

CloseUnits()

```
public static void CloseUnits()
```

CreateArrayConst()

```
private array_const CreateArrayConst()
```

Returns

array_const

CreateArrayInitializer()

```
private array_initializer CreateArrayInitializer()
```

Returns

array_initializer

CreateAsNode()

```
private as_node CreateAsNode()
```

Returns

as_node

CreateBasicFunctionCall()

```
private expression_node CreateBasicFunctionCall()
```

Returns

expression_node

CreateBasicFunctionCallAsConstant()

```
private expression_node CreateBasicFunctionCallAsConstant()
```

Returns

expression_node

CreateBoolConstNode()

```
private expression_node CreateBoolConstNode()
```

Returns

expression_node

CreateByteConstNode()

```
private expression_node CreateByteConstNode()
```

Returns

expression_node

CreateCaseVariant()

```
private case_variant_node CreateCaseVariant()
```

Returns

case_variant_node

CreateCharConstNode()

```
private expression_node CreateCharConstNode()
```

Returns

expression_node

CreateClassFieldReference()

```
private expression_node CreateClassFieldReference()
```

Returns

expression_node

CreateCommonConstructorCall()

```
private expression_node CreateCommonConstructorCall()
```

Returns

expression_node

CreateCommonConstructorCallAsConstant()

```
private expression_node CreateCommonConstructorCallAsConstant()
```

Returns

expression_node

CreateCommonInFuncFuncCall()

```
private expression_node CreateCommonInFuncFuncCall()
```

Returns

expression_node

CreateCommonNamespaceFunctionCall()

```
private expression_node CreateCommonNamespaceFunctionCall()
```

Returns

expression_node

CreateCommonNamespaceFunctionCallNodeAsConstant()

```
private common_namespace_function_call_as_constant  
CreateCommonNamespaceFunctionCallNodeAsConstant()
```

Returns

common_namespace_function_call_as_constant

CreateCommonParameterReference()

```
private expression_node CreateCommonParameterReference()
```

Returns

expression_node

CreateCommonStaticMethodCallNodeAsConstant()

```
private common_static_method_call_as_constant CreateCommonStaticMethodCallNodeAsConstant()
```

Returns

common_static_method_call_as_constant

CreateCompiledConstructorCall()

```
private compiled_constructor_call CreateCompiledConstructorCall()
```

Returns

compiled_constructor_call

CreateCompiledConstructorCallAsConstant()

```
private compiled_constructor_call_as_constant CreateCompiledConstructorCallAsConstant()
```

Returns

compiled_constructor_call_as_constant

CreateCompiledFunctionCall()

```
private compiled_function_call CreateCompiledFunctionCall()
```

Returns

compiled_function_call

CreateCompiledStaticFieldReferenceAsConstant()

```
private expression_node CreateCompiledStaticFieldReferenceAsConstant()
```

Returns

expression_node

CreateCompiledStaticMethodCall()

```
private compiled_static_method_call CreateCompiledStaticMethodCall()
```

Returns

compiled_static_method_call

CreateCompiledStaticMethodCallNodeAsConstant()

```
private compiled_static_method_call_as_constant  
CreateCompiledStaticMethodCallNodeAsConstant()
```

Returns

compiled_static_method_call_as_constant

CreateCompiledTypeNode(int)

```
private compiled_type_node CreateCompiledTypeNode(int offset)
```

Parameters

offset [int](#)

Returns

compiled_type_node

CreateCompiledVariableReference()

```
private compiled_variable_reference CreateCompiledVariableReference()
```

Returns

compiled_variable_reference

CreateDefaultOperator()

```
private default_operator_node CreateDefaultOperator()
```

Returns

default_operator_node

CreateDefaultOperatorAsConstant()

```
private default_operator_node_as_constant CreateDefaultOperatorAsConstant()
```

Returns

default_operator_node_as_constant

CreateDerefNode()

```
private expression_node CreateDerefNode()
```

Returns

expression_node

CreateDoubleConstNode()

```
private expression_node CreateDoubleConstNode()
```

Returns

expression_node

CreateDoubleQuestionColonExpression()

```
private double_question_colon_expression CreateDoubleQuestionColonExpression()
```

Returns

double_question_colon_expression

CreateEmpty()

```
private statement_node CreateEmpty()
```

Returns

statement_node

CreateEnumConstNode()

```
private expression_node CreateEnumConstNode()
```

Returns

expression_node

CreateExitProcedure()

```
private exit_procedure CreateExitProcedure()
```

Returns

exit_procedure

CreateExpression()

```
private expression_node CreateExpression()
```

Returns

expression_node

CreateExpression(semantic_node_type)

```
private expression_node CreateExpression(semantic_node_type snt)
```

Parameters

snt semantic_node_type

Returns

expression_node

CreateExpressionWithOffset()

```
private expression_node CreateExpressionWithOffset()
```

Returns

expression_node

CreateExpressionWithOffset(int)

```
private expression_node CreateExpressionWithOffset(int pos)
```

Returns

pos int ↗

Returns

expression_node

CreateExternalStatement()

```
private external_statement CreateExternalStatement()
```

Returns

external_statement

CreateFloatConst()

```
private float_const_node CreateFloatConst()
```

Returns

float_const_node

CreateFor()

```
private statement_node CreateFor()
```

Returns

statement_node

CreateForBreakNode()

```
private expression_node CreateForBreakNode()
```

Returns

expression_node

CreateForContinueNode()

```
private expression_node CreateForContinueNode()
```

Returns

expression_node

CreateForeach()

```
private statement_node CreateForeach()
```

Returns

statement_node

CreateForeachBreakNode()

```
private expression_node CreateForeachBreakNode()
```

Returns

expression_node

CreateForeachContinueNode()

```
private expression_node CreateForeachContinueNode()
```

Returns

expression_node

CreateFunctionConstantReference()

```
private expression_node CreateFunctionConstantReference()
```

Returns

expression_node

CreateGetAddrNode()

```
private expression_node CreateGetAddrNode()
```

Returns

expression_node

CreateGoto()

```
private statement_node CreateGoto()
```

Returns

statement_node

CreateIf()

```
private statement_node CreateIf()
```

Returns

statement_node

CreateImplementationMember(int, bool)

```
public override definition_node CreateImplementationMember(int offset, bool restore_code  
= true)
```

Parameters

offset [int](#)

restore_code [bool](#)

Returns

definition_node

CreateIntConstNode()

```
private expression_node CreateIntConstNode()
```

Returns

expression_node

CreateInterfaceClassConstantDefinition(string, int)

```
private class_constant_definition CreateInterfaceClassConstantDefinition(string name,  
int offset)
```

Parameters

`name` [string](#)

`offset` [int](#)

Returns

`class_constant_definition`

CreateInterfaceClassField(string, int)

```
private definition_node CreateInterfaceClassField(string name, int offset)
```

Parameters

`name` [string](#)

`offset` [int](#)

Returns

`definition_node`

CreateInterfaceCommonType(string, int)

```
private definition_node CreateInterfaceCommonType(string name, int offset)
```

Parameters

`name` [string](#)

`offset` [int](#)

Returns

`definition_node`

CreateInterfaceConstantDefinition(string, int)

```
private namespace_constant_definition CreateInterfaceConstantDefinition(string name,  
int offset)
```

Parameters

`name` [string](#)

`offset` [int](#)

Returns

`namespace_constant_definition`

CreateInterfaceEvent(string, int)

```
private definition_node CreateInterfaceEvent(string name, int offset)
```

Parameters

`name` [string](#)

`offset` [int](#)

Returns

`definition_node`

CreateInterfaceInClassMember(int, string)

```
public override definition_node CreateInterfaceInClassMember(int offset, string name)
```

Parameters

`offset` [int](#)

`name` [string](#)

Returns

definition_node

CreateInterfaceMember(int, string)

```
public override definition_node CreateInterfaceMember(int offset, string name)
```

Parameters

offset [int](#)

name [string](#)

Returns

definition_node

CreateInterfaceMethod(string, int, bool)

```
private common_method_node CreateInterfaceMethod(string name, int offset, bool  
not_restore_code = false)
```

Parameters

name [string](#)

offset [int](#)

not_restore_code [bool](#)

Returns

common_method_node

CreateInterfaceNamespaceEvent(string, int)

```
private common_namespace_event CreateInterfaceNamespaceEvent(string name, int offset)
```

Parameters

name [string](#)

offset [int](#)

Returns

common_namespace_event

CreateInterfaceNamespaceFunction(string, int)

```
private common_namespace_function_node CreateInterfaceNamespaceFunction(string name,  
int offset)
```

Parameters

name [string](#)

offset [int](#)

Returns

common_namespace_function_node

CreateInterfaceNamespaceVariable(string, int)

```
private namespace_variable CreateInterfaceNamespaceVariable(string name, int offset)
```

Parameters

name [string](#)

offset [int](#)

Returns

namespace_variable

CreateInterfaceProperty(string, int)

```
private definition_node CreateInterfaceProperty(string name, int offset)
```

Parameters

name [string](#)

offset [int](#)

Returns

definition_node

CreateIsNode()

```
private is_node CreateIsNode()
```

Returns

is_node

CreateLabeledStatement()

```
private statement_node CreateLabeledStatement()
```

Returns

statement_node

CreateLocalBlockVariable(statements_list)

```
private local_block_variable CreateLocalBlockVariable(statements_list stmt)
```

Returns

local_block_variable

Returns

local_block_variable

CreateLocalBlockVariableReference()

```
private expression_node CreateLocalBlockVariableReference()
```

Returns

expression_node

CreateLocalVariableReference()

```
private expression_node CreateLocalVariableReference()
```

Returns

expression_node

CreateLock()

```
private statement_node CreateLock()
```

Returns

statement_node

CreateLongConstNode()

```
private expression_node CreateLongConstNode()
```

Returns

expression_node

CreateMethodCall()

```
private expression_node CreateMethodCall()
```

Returns

expression_node

CreateNamespaceConstantReference()

```
private expression_node CreateNamespaceConstantReference()
```

Returns

expression_node

CreateNamespaceVariableReference()

```
private expression_node CreateNamespaceVariableReference()
```

Returns

expression_node

CreateNonStaticEventReference()

```
private expression_node CreateNonStaticEventReference()
```

Returns

expression_node

CreateNonStaticPropertyReference()

```
private expression_node CreateNonStaticPropertyReference()
```

Returns

expression_node

CreateNullConstNode()

```
private expression_node CreateNullConstNode()
```

Returns

expression_node

CreatePInvokeStatement()

```
private statement_node CreatePInvokeStatement()
```

Returns

statement_node

CreateQuestionColonExpression()

```
private question_colon_expression CreateQuestionColonExpression()
```

Returns

question_colon_expression

CreateRecordConst()

```
private record_constant CreateRecordConst()
```

Returns

record_constant

CreateRecordInitializer()

```
private record_initializer CreateRecordInitializer()
```

Returns

record_initializer

CreateRefType(int)

```
private ref_type_node CreateRefType(int offset)
```

Parameters

offset [int](#)

Returns

ref_type_node

CreateRepeat()

```
private statement_node CreateRepeat()
```

Returns

statement_node

CreateRepeatBreakNode()

```
private expression_node CreateRepeatBreakNode()
```

Returns

expression_node

CreateRepeatContinueNode()

```
private expression_node CreateRepeatContinueNode()
```

Returns

expression_node

CreateRethrow()

```
private statement_node CreateRethrow()
```

Returns

statement_node

CreateReturnNode()

```
private statement_node CreateReturnNode()
```

Returns

statement_node

CreateRuntimeStatement()

```
private runtime_statement CreateRuntimeStatement()
```

Returns

runtime_statement

CreateSByteConstNode()

```
private expression_node CreateSByteConstNode()
```

Returns

expression_node

CreateShortConstNode()

```
private expression_node CreateShortConstNode()
```

Returns

expression_node

CreateSimpleArrayIndexing()

```
private simple_array_indexing CreateSimpleArrayIndexing()
```

Returns

simple_array_indexing

CreateSizeOfOperator()

```
private sizeof_operator CreateSizeOfOperator()
```

Returns

sizeof_operator

CreateSizeOfOperatorAsConstant()

```
private sizeof_operator_as_constant CreateSizeOfOperatorAsConstant()
```

Returns

sizeof_operator_as_constant

CreateStatement()

```
private statement_node CreateStatement()
```

Returns

statement_node

CreateStatementList()

```
private statement_node CreateStatementList()
```

Returns

statement_node

CreateStatementsExpressionNode()

```
private statements_expression_node CreateStatementsExpressionNode()
```

Returns

statements_expression_node

CreateStaticClassFieldReference()

```
private expression_node CreateStaticClassFieldReference()
```

Returns

expression_node

CreateStaticCompiledVariableReference()

```
private static_compiled_variable_reference CreateStaticCompiledVariableReference()
```

Returns

static_compiled_variable_reference

CreateStaticEventReference()

```
private expression_node CreateStaticEventReference()
```

Returns

expression_node

CreateStaticMethodCall()

```
private common_static_method_call CreateStaticMethodCall()
```

Returns

common_static_method_call

CreateStringConstNode()

```
private expression_node CreateStringConstNode()
```

Returns

expression_node

CreateSwitchNode()

```
private statement_node CreateSwitchNode()
```

Returns

statement_node

CreateTemplateClass(int)

```
private template_class CreateTemplateClass(int offset)
```

Returns

template_class

CreateThisNode()

```
private expression_node CreateThisNode()
```

Returns

expression_node

CreateThrow()

```
private statement_node CreateThrow()
```

Returns

statement_node

CreateTryBlock()

```
private try_block CreateTryBlock()
```

Returns

try_block

CreateTypeOfOperator()

```
private typeof_operator CreateTypeOfOperator()
```

Returns

typeof_operator

CreateTypeSynonim(int, string)

```
public override definition_node CreateTypeSynonim(int offset, string name)
```

Parameters

offset [int](#)

name [string](#)

Returns

definition_node

CreateTypeSynonym()

```
private wrapped_type_synonym CreateTypeSynonym()
```

Returns

[wrapped_type_synonym](#)

CreateUIntConstNode()

```
private expression_node CreateUIntConstNode()
```

Returns

expression_node

CreateULongConstNode()

```
private expression_node CreateULongConstNode()
```

Returns

expression_node

CreateUShortConstNode()

```
private expression_node CreateUShortConstNode()
```

Returns

expression_node

CreateWhile()

```
private statement_node CreateWhile()
```

Returns

statement_node

CreateWhileBreakNode()

```
private expression_node CreateWhileBreakNode()
```

Returns

expression_node

CreateWhileContinueNode()

```
private expression_node CreateWhileContinueNode()
```

Returns

expression_node

~PCUReader()

```
protected ~PCUReader()
```

FindConstructorByHandle(Type, int)

```
private ConstructorInfo FindConstructorByHandle(Type t, int off)
```

Parameters

t [Type](#)

off [int](#)

Returns

[ConstructorInfo](#)

FindEventByHandle(Type, int)

```
private EventInfo FindEventByHandle(Type t, int off)
```

Parameters

t [Type](#)

off [int](#)

Returns

[EventInfo](#)

FindFieldByHandle(Type, int)

```
private FieldInfo FindFieldByHandle(Type t, int off)
```

Parameters

t [Type](#)

off [int](#)

Returns

[FieldInfo](#)

FindMethodByHandle(Type, int)

```
private MethodInfo FindMethodByHandle(Type t, int off)
```

Parameters

t [Type](#)

off [int](#)

Returns

[MethodInfo](#)

FindPropertyByHandle(Type, int)

```
private PropertyInfo FindPropertyByHandle(Type t, int off)
```

Parameters

t [Type](#)

off [int](#)

Returns

[PropertyInfo](#)

FindSpecTypeByHandle(int)

```
private PCUReader.TypeSpec FindSpecTypeByHandle(int off)
```

Parameters

off [int](#)

Returns

[PCUReader.TypeSpec](#)

FindTypeByHandle(int)

```
private Type FindTypeByHandle(int off)
```

Parameters

off [int](#)

Returns

[Type](#)

GetAttribute()

```
private attribute_node GetAttribute()
```

Returns

attribute_node

GetAttributes()

```
private attributes_list GetAttributes()
```

Returns

attributes_list

GetClassField(int)

```
public class_field GetClassField(int offset)
```

Parameters

offset [int](#)

Returns

class_field

GetClassFieldByOffset()

```
private class_field GetClassFieldByOffset()
```

Returns

class_field

GetClassFieldByOffset(int)

```
private class_field GetClassFieldByOffset(int offset)
```

Parameters

offset [int](#)

Returns

class_field

GetClassMethod(int, bool)

```
private common_method_node GetClassMethod(int offset, bool not_restore_code = false)
```

Parameters

offset [int](#)

not_restore_code [bool](#)

Returns

common_method_node

GetCode(int)

```
public statement_node GetCode(int offset)
```

Parameters

offset [int](#)

Returns

statement_node

GetCodeWithOverridedMethod(common_method_node, int)

```
public statement_node GetCodeWithOverridedMethod(common_method_node meth, int offset)
```

Parameters

meth common_method_node

offset [int ↗](#)

Returns

statement_node

GetCommonGenericMethodInstanceReference()

```
private generic_method_instance_node GetCommonGenericMethodInstanceReference()
```

Returns

generic_method_instance_node

GetCommonType(int)

```
public common_type_node GetCommonType(int offset)
```

Parameters

offset [int ↗](#)

Returns

common_type_node

GetCompilationUnit(string, bool)

```
public CompilationUnit GetCompilationUnit(string FileName, bool readDebugInfo)
```

Parameters

FileName [string](#)

readDebugInfo [bool](#)

Returns

[CompilationUnit](#)

GetCompiledConstructor(int)

```
private compiled_constructor_node GetCompiledConstructor(int offset)
```

Parameters

offset [int](#)

Returns

compiled_constructor_node

GetCompiledEvent(int)

```
private compiled_event GetCompiledEvent(int offset)
```

Parameters

offset [int](#)

Returns

compiled_event

GetCompiledGenericMethodInstanceReference()

```
private function_node GetCompiledGenericMethodInstanceReference()
```

Returns

function_node

GetCompiledMethod(int)

```
private compiled_function_node GetCompiledMethod(int offset)
```

Parameters

offset [int](#)

Returns

compiled_function_node

GetCompiledProperty(int)

```
private compiled_property_node GetCompiledProperty(int offset)
```

Parameters

offset [int](#)

Returns

compiled_property_node

GetCompiledVariable(int)

```
private compiled_variable_definition GetCompiledVariable(int offset)
```

Parameters

offset [int](#)

Returns

compiled_variable_definition

GetConstantDefinition(int)

```
private namespace_constant_definition GetConstantDefinition(int offset)
```

Parameters

offset [int](#)

Returns

namespace_constant_definition

GetEventByOffset()

```
private event_node GetEventByOffset()
```

Returns

event_node

GetEventNode(int)

```
public common_event GetEventNode(int offset)
```

Parameters

offset [int](#)

Returns

common_event

GetExpression(int)

```
public expression_node GetExpression(int offset)
```

Parameters

offset [int](#)

Returns

expression_node

GetExtClassFieldByOffset(int)

```
private class_field GetExtClassFieldByOffset(int offset)
```

Parameters

offset [int](#)

Returns

class_field

GetExtEventNode(int)

```
private common_event GetExtEventNode(int offset)
```

Parameters

offset [int](#)

Returns

common_event

GetExtMethodByOffset(int)

```
private common_method_node GetExtMethodByOffset(int offset)
```

Parameters

offset [int](#)

Returns

common_method_node

GetExtNamespaceConstantByOffset(int)

```
private namespace_constant_definition GetExtNamespaceConstantByOffset(int offset)
```

Parameters

offset [int](#)

Returns

namespace_constant_definition

GetExtNamespaceEvent(int)

```
private common_namespace_event GetExtNamespaceEvent(int offset)
```

Parameters

offset [int ↗](#)

Returns

common_namespace_event

GetExtNamespaceFunctionByOffset(int)

```
private common_namespace_function_node GetExtNamespaceFunctionByOffset(int offset)
```

Parameters

offset [int ↗](#)

Returns

common_namespace_function_node

GetExtNamespaceVariableByOffset(int)

```
private namespace_variable GetExtNamespaceVariableByOffset(int offset)
```

Parameters

offset [int ↗](#)

Returns

namespace_variable

GetExtPropertyByOffset(int)

```
private common_property_node GetExtPropertyByOffset(int offset)
```

Parameters

`offset int`

Returns

`common_property_node`

GetFieldReference()

```
private var_definition_node GetFieldReference()
```

Returns

`var_definition_node`

GetFullUnitName(string)

```
public string GetFullUnitName(string UnitName)
```

Parameters

`UnitName string`

Returns

`string`

GetFunctionConstant(common_function_node)

```
private function_constant_definition GetFunctionConstant(common_function_node func)
```

Parameters

`func common_function_node`

Returns

function_constant_definition

GetFunctionConstantByOffset(int)

```
private function_constant_definition GetFunctionConstantByOffset(int offset)
```

Parameters

offset [int](#)

Returns

function_constant_definition

GetGenericInstance()

```
private generic_instance_type_node GetGenericInstance()
```

Returns

generic_instance_type_node

GetGenericInstanceConstructor()

```
private common_method_node GetGenericInstanceConstructor()
```

Returns

common_method_node

GetGenericInstanceField()

```
private class_field GetGenericInstanceField()
```

Returns

class_field

GetGenericInstanceMethod()

```
private common_method_node GetGenericInstanceMethod()
```

Returns

common_method_node

GetGenericInstanceProperty()

```
private common_property_node GetGenericInstanceProperty()
```

Returns

common_property_node

GetGenericNamespaceFunctionReference()

```
private generic_namespace_function_instance_node GetGenericNamespaceFunctionReference()
```

Returns

generic_namespace_function_instance_node

GetGenericParamConstructor()

```
private common_method_node GetGenericParamConstructor()
```

Returns

common_method_node

GetGenericParameterOfFunction()

```
private common_type_node GetGenericParameterOfFunction()
```

Returns

common_type_node

GetGenericParameterOfMethod()

```
private common_type_node GetGenericParameterOfMethod()
```

Returns

common_type_node

GetGenericParameterOfType()

```
private common_type_node GetGenericParameterOfType()
```

Returns

common_type_node

GetLabel(int)

```
private label_node GetLabel(int offset)
```

Parameters

offset [int](#)

Returns

label_node

GetLocalBlockVariableByOffset(int)

```
private local_block_variable GetLocalBlockVariableByOffset(int offset)
```

Parameters

offset [int](#)

Returns

local_block_variable

GetLocalOrBlockVariableByOffset(int)

```
private var_definition_node GetLocalOrBlockVariableByOffset(int offset)
```

Parameters

offset [int](#)

Returns

var_definition_node

GetLocalOrNamespaceVariableByOffset(int)

```
private var_definition_node GetLocalOrNamespaceVariableByOffset(int offset)
```

Parameters

offset [int](#)

Returns

var_definition_node

GetLocalVariable(common_function_node)

```
private local_variable GetLocalVariable(common_function_node func)
```

Parameters

func common_function_node

Returns

local_variable

GetLocalVariableByOffset(int)

```
private local_variable GetLocalVariableByOffset(int offset)
```

Parameters

offset [int](#)

Returns

local_variable

GetLocalVariableLazy(common_function_node)

```
private Tuple<local_variable, int> GetLocalVariableLazy(common_function_node func)
```

Parameters

func common_function_node

Returns

[Tuple](#)<local_variable, [int](#)>

GetMethodByOffset()

```
private common_method_node GetMethodByOffset()
```

Returns

common_method_node

GetMethodByOffset(int)

```
private common_method_node GetMethodByOffset(int offset)
```

Parameters

offset [int](#)

Returns

common_method_node

GetMethodReference()

```
private function_node GetMethodReference()
```

Returns

function_node

GetNamespaceConstantByOffset(int)

```
private namespace_constant_definition GetNamespaceConstantByOffset(int offset)
```

Parameters

offset [int](#)

Returns

namespace_constant_definition

GetNamespaceEvent(int)

```
public common_namespace_event GetNamespaceEvent(int offset)
```

Parameters

offset [int](#)

Returns

common_namespace_event

GetNamespaceEventNode(int)

```
public common_namespace_event GetNamespaceEventNode(int offset)
```

Parameters

offset [int](#)

Returns

common_namespace_event

GetNamespaceFunction(int, bool)

```
private common_namespace_function_node GetNamespaceFunction(int offset, bool restore_code  
= true)
```

Parameters

offset [int](#)

restore_code [bool](#)

Returns

common_namespace_function_node

GetNamespaceFunctionByOffset()

```
private common_namespace_function_node GetNamespaceFunctionByOffset()
```

Returns

common_namespace_function_node

GetNamespaceFunctionByOffset(int)

```
private common_namespace_function_node GetNamespaceFunctionByOffset(int offset)
```

Parameters

offset [int](#)

Returns

common_namespace_function_node

GetNamespaceFunctionWithImplementation(int)

```
private common_namespace_function_node GetNamespaceFunctionWithImplementation(int offset)
```

Parameters

offset [int](#)

Returns

common_namespace_function_node

GetNamespaceVariable(int)

```
public namespace_variable GetNamespaceVariable(int offset)
```

Parameters

offset [int](#)

Returns

namespace_variable

GetNamespaceVariableByOffset(int)

```
private namespace_variable GetNamespaceVariableByOffset(int offset)
```

Parameters

offset [int](#)

Returns

namespace_variable

GetNestedFunction()

```
private common_in_function_function_node GetNestedFunction()
```

Returns

common_in_function_function_node

GetNestedFunctionByOffset(int)

```
private common_in_function_function_node GetNestedFunctionByOffset(int offset)
```

Parameters

offset [int](#)

Returns

common_in_function_function_node

GetNetExtType(int)

```
private compiled_type_node GetNetExtType(int offset)
```

Parameters

offset [int](#)

Returns

compiled_type_node

GetPCUFileHeadState(string)

```
public static PCUReader.PCUFileHeadState GetPCUFileHeadState(string fileName)
```

Parameters

fileName [string](#)

Returns

[PCUReader.PCUFileHeadState](#)

GetPCUReaderForUnitId(int)

```
private PCUReader GetPCUReaderForUnitId(int id)
```

Parameters

id [int](#)

Returns

[PCUReader](#)

GetParameter()

```
private common_parameter GetParameter()
```

Returns

common_parameter

GetParameter(common_function_node)

```
private common_parameter GetParameter(common_function_node func)
```

Parameters

func common_function_node

Returns

common_parameter

GetParameterByOffset(int)

```
private common_parameter GetParameterByOffset(int offset)
```

Parameters

offset [int](#)

Returns

common_parameter

GetPropertyByOffset()

```
private common_property_node GetPropertyByOffset()
```

Returns

common_property_node

GetPropertyByOffset(int)

```
private common_property_node GetPropertyByOffset(int offset)
```

Parameters

offset [int](#)

Returns

common_property_node

GetPropertyNode(int)

```
private common_property_node GetPropertyNode(int offset)
```

Parameters

offset [int](#)

Returns

common_property_node

GetPropertyReference()

```
private property_node GetPropertyReference()
```

Returns

property_node

GetShortStringType()

```
private short_string_type_node GetShortStringType()
```

Returns

short_string_type_node

GetSpecialTypeReference(int)

```
private type_node GetSpecialTypeReference(int offset)
```

Parameters

offset [int](#)

Returns

type_node

GetStreamPosition()

```
public int GetStreamPosition()
```

Returns

[int](#)

GetString(int)

```
private string GetString(int index)
```

Parameters

index [int](#)

Returns

[string](#)

GetStringInClass(common_type_node, int)

```
private string GetStringInClass(common_type_node type, int name_off)
```

Parameters

`type` common_type_node

`name_off` [int](#)

Returns

[string](#)

GetTemplateClass(int)

```
private template_class GetTemplateClass(int offset)
```

Parameters

`offset` [int](#)

Returns

template_class

GetTemplateClassReference()

```
private template_class GetTemplateClassReference()
```

Returns

template_class

GetTemplateInstance()

```
private common_type_node GetTemplateInstance()
```

Returns

common_type_node

GetTemplateInstanceField()

```
private class_field GetTemplateInstanceField()
```

Returns

class_field

GetTemplateInstanceMethod()

```
private common_method_node GetTemplateInstanceMethod()
```

Returns

common_method_node

GetTemplateInstanceProperty()

```
private common_property_node GetTemplateInstanceProperty()
```

Returns

common_property_node

GetTypeReference()

```
private type_node GetTypeReference()
```

Returns

type_node

GetTypeReference(int)

```
public type_node GetTypeReference(int offset)
```

Parameters

offset [int](#)

Returns

type_node

GetTypesList()

```
private List<type_node> GetTypesList()
```

Returns

[List](#)<type_node>

GetVariables(common_function_node, int)

```
private void GetVariables(common_function_node cfn, int num_var)
```

Parameters

cfn common_function_node

num_var [int](#)

InvalidUnitDetected()

```
private void InvalidUnitDetected()
```

MakeTopScopeArray(using_namespace_list, int)

```
public Scope[] MakeTopScopeArray(using_namespace_list unl, int uses_count)
```

Parameters

unl using_namespace_list

uses_count [int](#)

Returns

Scope[]

MakeTypeAsOrdinal(common_type_node, int, int)

```
private void MakeTypeAsOrdinal(common_type_node ctn, int low_val, int upper_val)
```

Parameters

ctn common_type_node

low_val [int](#)

upper_val [int](#)

NeedRecompiled()

```
public bool NeedRecompiled()
```

Returns

[bool](#)

OpenUnit()

```
public void OpenUnit()
```

ProcessWaitedToRestoreFields()

```
public void ProcessWaitedToRestoreFields()
```

ReadAllAssemblies()

```
private void ReadAllAssemblies()
```

ReadCommonExtEvent()

```
private common_event ReadCommonExtEvent()
```

Returns

common_event

ReadCommonExtField()

```
private class_field ReadCommonExtField()
```

Returns

class_field

ReadCommonExtMethod()

```
private common_method_node ReadCommonExtMethod()
```

Returns

common_method_node

ReadCommonExtNamespaceFunc()

```
private common_namespace_function_node ReadCommonExtNamespaceFunc()
```

Returns

common_namespace_function_node

ReadCommonExtProperty()

```
private common_property_node ReadCommonExtProperty()
```

Returns

common_property_node

ReadCommonExtType()

```
private type_node ReadCommonExtType()
```

Returns

type_node

ReadCommonNamespaceExtEvent()

```
private common_namespace_event ReadCommonNamespaceExtEvent()
```

Returns

common_namespace_event

ReadDebugInfo()

```
private location ReadDebugInfo()
```

Returns

location

ReadExtNamespaceConstant()

```
private namespace_constant_definition ReadExtNamespaceConstant()
```

Returns

namespace_constant_definition

ReadExtNamespaceVariable()

```
private namespace_variable ReadExtNamespaceVariable()
```

Returns

namespace_variable

ReadGenericFunctionInformation(common_function_node)

```
private void ReadGenericFunctionInformation(common_function_node func)
```

Parameters

func common_function_node

ReadGenericParams(common_namespace_node)

```
private List<ICommonTypeNode> ReadGenericParams(common_namespace_node cur_nn)
```

Parameters

cur_nn common_namespace_node

Returns

[List](#)<ICommonTypeNode>

ReadImplementingInterfaces()

```
private List<ITypeNode> ReadImplementingInterfaces()
```

Returns

[List](#)<ITypeNode>

ReadNetExtType()

```
private compiled_type_node ReadNetExtType()
```

Returns

compiled_type_node

ReadPCUHead(PCUFile, BinaryReader)

```
private static bool ReadPCUHead(PCUFile pcu_file, BinaryReader br)
```

Parameters

pcu_file [PCUFile](#)

`br` [BinaryReader](#)

Returns

[bool](#)

ReadPCUHeader()

```
private void ReadPCUHeader()
```

ReadTemplateExtClass()

```
private template_class ReadTemplateExtClass()
```

Returns

`template_class`

ReadTypeParameterEliminations(common_type_node)

```
private void ReadTypeParameterEliminations(common_type_node par)
```

Parameters

`par` `common_type_node`

RemoveMember(int, definition_node)

```
internal void RemoveMember(int offset, definition_node dn)
```

Parameters

`offset` [int](#)

`dn` definition_node

RestoreAllFields(common_type_node)

```
private void RestoreAllFields(common_type_node ctn)
```

Parameters

`ctn` common_type_node

RestoreOperators(common_type_node)

```
private void RestoreOperators(common_type_node ctn)
```

Parameters

`ctn` common_type_node

RestoreWaitedMethodCodes()

```
public void RestoreWaitedMethodCodes()
```

SeekInExternal(int)

```
private void SeekInExternal(int pos)
```

Parameters

`pos` [int](#)

SetStreamPosition(int)

```
public void SetStreamPosition(int pos)
```

Parameters

pos [int](#)

compareTypesDeeply(Type, Type)

```
private bool compareTypesDeeply(Type t1, Type t2)
```

Parameters

t1 [Type](#)

t2 [Type](#)

Returns

[bool](#)

Events

ChangeState

```
public event PCUReader.ChangeStateDelegate ChangeState
```

Event Type

[PCUReader.ChangeStateDelegate](#)

Delegate PCUReader.ChangeStateDelegate

Namespace: [PascalABCCompiler.PCU](#)

Assembly: Compiler.dll

```
public delegate void PCUReader.ChangeStateDelegate(object Sender, PCUReaderWriterState  
State, object obj)
```

Parameters

Sender [object](#)

State [PCUReaderWriterState](#)

obj [object](#)

Constructors

ChangeStateDelegate(object, IntPtr)

```
public ChangeStateDelegate(object @object, IntPtr method)
```

Parameters

object [object](#)

method [IntPtr](#)

Methods

BeginInvoke(object, PCUReaderWriterState, object, AsyncCallback, object)

```
public virtual IAsyncResult BeginInvoke(object Sender, PCUReaderWriterState State, object  
obj, AsyncCallback callback, object @object)
```

Parameters

Sender [object](#)

State [PCUReaderWriterState](#)

obj [object](#)

callback [AsyncCallback](#)

object [object](#)

Returns

[IAsyncResult](#)

EndInvoke(IAsyncResult)

```
public virtual void EndInvoke(IAsyncResult result)
```

Parameters

result [IAsyncResult](#)

Invoke(object, PCUReaderWriterState, object)

```
public virtual void Invoke(object Sender, PCUReaderWriterState State, object obj)
```

Parameters

Sender [object](#)

State [PCUReaderWriterState](#)

obj [object](#)

Class PCUReader.PCUFileHeadState

Namespace: [PascalABCCompiler.PCU](#)

Assembly: Compiler.dll

Читает начало заголовка PCU файла

```
public class PCUReader.PCUFileHeadState
```

Inheritance

[object](#) ← PCUReader.PCUFileHeadState

Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.FieldSetter\(string, string, object\)](#) ,
[object.FieldGetter\(string, string, ref object\)](#) , [object.GetFieldInfo\(string, string\)](#)

Constructors

PCUFileHeadState()

```
public PCUFileHeadState()
```

Fields

FileHead

```
public PCUFile FileHead
```

Field Value

[PCUFile](#)

IncludetDebugInfo

```
public bool IncludetDebugInfo
```

Field Value

[bool ↗](#)

IsPCUFile

```
public bool IsPCUFile
```

Field Value

[bool ↗](#)

SupportedVersion

```
public bool SupportedVersion
```

Field Value

[bool ↗](#)

Class PCUReader.TypeSpec

Namespace: [PascalABCCompiler.PCU](#)

Assembly: Compiler.dll

```
private class PCUReader.TypeSpec
```

Inheritance

[object](#) ← PCUReader.TypeSpec

Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.FieldSetter\(string, string, object\)](#) ,
[object.FieldGetter\(string, string, ref object\)](#) , [object.GetFieldInfo\(string, string\)](#)

Constructors

TypeSpec()

```
public TypeSpec()
```

Fields

name

```
public string name
```

Field Value

[string](#)

t

```
public Type t
```

Field Value

Type ↗

Enum PCUReaderWriterState

Namespace: [PascalABCCompiler.PCU](#)

Assembly: Compiler.dll

```
public enum PCUReaderWriterState
```

Fields

BeginReadTree = 0

BeginSaveTree = 2

EndReadTree = 1

EndSaveTree = 3

ErrorSaveTree = 4

Class PCUWriter

Namespace: [PascalABCCompiler.PCU](#)

Assembly: Compiler.dll

Класс, создающий PCU-модуль

```
public class PCUWriter
```

Inheritance

[object](#) ← PCUWriter

Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.FieldSetter\(string, string, object\)](#) ,
[object.FieldGetter\(string, string, ref object\)](#) , [object.GetFieldInfo\(string, string\)](#)

Constructors

PCUWriter(Compiler, ChangeStateDelegate)

```
public PCUWriter(Compiler compiler, PCUWriter.ChangeStateDelegate changeState)
```

Parameters

compiler [Compiler](#)

changeState [PCUWriter.ChangeStateDelegate](#)

Fields

AllWriters

```
internal static List<PCUWriter> AllWriters
```

Field Value

[List](#) <PCUWriter>

FileName

`public string FileName`

Field Value

[string](#)

FinalizationMethodOffset

`private int FinalizationMethodOffset`

Field Value

[int](#)

InitializationMethodOffset

`private int InitializationMethodOffset`

Field Value

[int](#)

added_indirect_types

`private HashSet<type_node> added_indirect_types`

Field Value

[HashSet](#) <type_node>

assm_refs

`private Dictionary<Assembly, int> assm_refs`

Field Value

[Dictionary](#) <[Assembly](#), [int](#)>

attr_dict

`private Dictionary<definition_node, List<int>> attr_dict`

Field Value

[Dictionary](#) <definition_node, [List](#) <[int](#)>>

bw

`private BinaryWriter bw`

Field Value

[BinaryWriter](#)

class_info

`private Dictionary<definition_node, ClassInfo> class_info`

Field Value

[Dictionary](#) <definition_node, [ClassInfo](#)>

compiler

```
private Compiler compiler
```

Field Value

[Compiler](#)

const_positions

```
private Hashtable const_positions
```

Field Value

[Hashtable](#)

cun

```
private common_unit_node cun
```

Field Value

common_unit_node

cur_cnn

```
private common_namespace_node cur_cnn
```

Field Value

common_namespace_node

dot_net_name_list

```
private List<DotNetNameRef> dot_net_name_list
```

Field Value

[List](#)<[DotNetNameRef](#)>

entity_index

```
private int entity_index
```

Field Value

[int](#)

exprs_cache

```
private Dictionary<expression_node, List<int>> exprs_cache
```

Field Value

[Dictionary](#)<expression_node, [List](#)<[int](#)>>

ext_members

```
private Dictionary<definition_node, ImportedEntity> ext_members
```

Field Value

[Dictionary](#)<definition_node, [ImportedEntity](#)>

ext_offsets

```
private Dictionary<definition_node, int> ext_offsets
```

Field Value

[Dictionary](#)<definition_node, [int](#)>

func_codes

```
private Dictionary<definition_node, int> func_codes
```

Field Value

[Dictionary](#)<definition_node, [int](#)>

function_references

```
private Dictionary<int, common_namespace_function_node> function_references
```

Field Value

[Dictionary](#)<[int](#), common_namespace_function_node>

gl_members

```
private static Dictionary<definition_node, int> gl_members
```

Field Value

[Dictionary](#)<definition_node, [int](#)>

imp_entities

```
private List<ImportedEntity> imp_entitles
```

Field Value

[List](#)<[ImportedEntity](#)>

impl_name_pool

```
private Dictionary<definition_node, NameRef> impl_name_pool
```

Field Value

[Dictionary](#)<definition_node, [NameRef](#)>

is_interface

```
private bool is_interface
```

Field Value

[bool](#)

members

```
private Dictionary<definition_node, int> members
```

Field Value

[Dictionary](#)<definition_node, [int](#)>

ms

```
private MemoryStream ms
```

Field Value

[MemoryStream](#)

name

```
private string name
```

Field Value

[string](#)

name_pool

```
private Dictionary<definition_node, NameRef> name_pool
```

Field Value

[Dictionary](#) <definition_node, [NameRef](#)>

not_comp_members

```
private static Dictionary<definition_node, List<PCUWriter>> not_comp_members
```

Field Value

[Dictionary](#) <definition_node, [List](#) <[PCUWriter](#)>>

pcu_file

```
private PCUFile pcu_file
```

Field Value

[PCUFile](#)

pcu_reader

```
private PCUReader pcu_reader
```

Field Value

[PCUReader](#)

ref_assemblies

```
private List<string> ref_assemblies
```

Field Value

[List](#) <[string](#)>

tokens

```
private Hashtable tokens
```

Field Value

[Hashtable](#)

type_entity_index

```
private int type_entity_index
```

Field Value

[int](#)

type_references

```
private Dictionary<int, common_type_node> type_references
```

Field Value

[Dictionary](#)<[int](#), common_type_node>

unit

```
private CompilationUnit unit
```

Field Value

[CompilationUnit](#)

used_units

```
private Dictionary<common_unit_node, int> used_units
```

Field Value

[Dictionary](#)<common_unit_node, [int](#)>

var_positions

```
private Hashtable var_positions
```

Field Value

[Hashtable](#) ↗

Methods

AddAttributes()

```
private void AddAttributes()
```

AddExternalMember(definition_node, int)

```
public static void AddExternalMember(definition_node dn, int offset)
```

Parameters

dn definition_node

offset [int](#) ↗

AddIndirectImplementationUsedUnits()

```
private void AddIndirectImplementationUsedUnits()
```

AddIndirectInteraceUsedUnits()

```
private void AddIndirectInteraceUsedUnits()
```

AddIndirectUsedUnitsForFunction(common_namespace_function_node, Dictionary<common_namespace_node, bool>, bool)

```
private void AddIndirectUsedUnitsForFunction(common_namespace_function_node cnfn,  
Dictionary<common_namespace_node, bool> ns_dict, bool interf)
```

Parameters

cnfn common_namespace_function_node

ns_dict [Dictionary](#)<common_namespace_node, [bool](#)>

interf [bool](#)

AddIndirectUsedUnitsForType(type_node, Dictionary<common_namespace_node, bool>, bool)

```
private void AddIndirectUsedUnitsForType(type_node tn, Dictionary<common_namespace_node,  
bool> ns_dict, bool interf)
```

Parameters

tn type_node

ns_dict [Dictionary](#)<common_namespace_node, [bool](#)>

interf [bool](#)

AddIndirectUsedUnitsInStatement(statement_node, Dictionary<common_namespace_node, bool>, bool)

```
private void AddIndirectUsedUnitsInStatement(statement_node stmt,  
Dictionary<common_namespace_node, bool> ns_dict, bool interf)
```

Parameters

stmt statement_node

```
ns_dict Dictionary<common_namespace_node, bool>
```

```
interf bool
```

AddIndirectUsedUnitsInVariables<T>(IEnumerable<T>, Dictionary<common_namespace_node, bool>, bool)

```
private void AddIndirectUsedUnitsInVariables<T>(IEnumerable<T> variables,  
Dictionary<common_namespace_node, bool> ns_dict, bool interf) where T : var_definition_node
```

Parameters

```
variables IEnumerable<T>
```

```
ns_dict Dictionary<common_namespace_node, bool>
```

```
interf bool
```

Type Parameters

T

AddOffsetForMembers(definition_node, int)

```
private void AddOffsetForMembers(definition_node dn, int offset)
```

Parameters

dn definition_node

offset [int](#)

AddPCUToOpen(PCUReader)

```
public void AddPCUToOpen(PCUReader pr)
```

Parameters

pr [PCUReader](#)

CanWriteObject(object)

```
private bool CanWriteObject(object obj)
```

Parameters

[obj](#) [object](#)

Returns

[bool](#)

Clear()

```
public static void Clear()
```

CloseWriter()

```
internal void CloseWriter()
```

FillNames(NameRef[], Dictionary<definition_node, NameRef>)

```
private void FillNames(NameRef[] name_array, Dictionary<definition_node, NameRef> pools)
```

Parameters

[name_array](#) [NameRef](#)[]

[pools](#) [Dictionary](#)<[definition_node](#), [NameRef](#)>

FixupCode(function_node)

```
private void FixupCode(function_node fn)
```

Parameters

fn function_node

FixupExpressionPosition(expression_node)

```
private void FixupExpressionPosition(expression_node expr)
```

Parameters

expr expression_node

GetAssemblyToken(Assembly)

```
private int GetAssemblyToken(Assembly a)
```

Parameters

a [Assembly](#)

Returns

[int](#)

GetCompiledConstructor(compiled_constructor_node)

```
private int GetCompiledConstructor(compiled_constructor_node ccn)
```

Parameters

`ccn` compiled_constructor_node

Returns

[int↗](#)

GetCompiledMethod(compiled_function_node)

`private int GetCompiledMethod(compiled_function_node cfn)`

Parameters

`cfn` compiled_function_node

Returns

[int↗](#)

GetCompiledProperty(compiled_property_node)

`private int GetCompiledProperty(compiled_property_node cpn)`

Parameters

`cpn` compiled_property_node

Returns

[int↗](#)

GetCompiledTypeReference(compiled_type_node)

`private int GetCompiledTypeReference(compiled_type_node ctn)`

Parameters

`ctn` compiled_type_node

Returns

[int↗](#)

GetCompiledVariable(compiled_variable_definition)

`private int GetCompiledVariable(compiled_variable_definition cvd)`

Parameters

`cvd` compiled_variable_definition

Returns

[int↗](#)

GetConstantReference(namespace_constant_definition, ref byte)

`private int GetConstantReference(namespace_constant_definition nv, ref byte is_def)`

Parameters

`nv` namespace_constant_definition

`is_def` [byte↗](#)

Returns

[int↗](#)

GetCountOfImplementationMembers()

`private void GetCountOfImplementationMembers()`

GetCountOfMembers()

```
private void GetCountOfMembers()
```

GetEventReference(event_node, ref byte)

```
private int GetEventReference(event_node ev, ref byte is_def)
```

Parameters

`ev` event_node

`is_def` byte ↗

Returns

int ↗

GetExtTypeReference(common_type_node)

```
private int GetExtTypeReference(common_type_node ctn)
```

Parameters

`ctn` common_type_node

Returns

int ↗

GetExternalOffset(definition_node)

```
private int GetExternalOffset(definition_node dn)
```

Parameters

`dn definition_node`

Returns

[int ↗](#)

GetFieldReference(class_field, ref byte)

`private int GetFieldReference(class_field field, ref byte is_def)`

Parameters

`field class_field`

`is_def byte ↗`

Returns

[int ↗](#)

GetFunctionReference(common_namespace_function_node, ref byte)

`private int GetFunctionReference(common_namespace_function_node fn, ref byte is_def)`

Parameters

`fn common_namespace_function_node`

`is_def byte ↗`

Returns

[int ↗](#)

GetMemberOffset(definition_node)

```
private int GetMemberOffset(definition_node dn)
```

Parameters

dn definition_node

Returns

[int↗](#)

GetMethodReference(common_method_node, ref byte)

```
private int GetMethodReference(common_method_node meth, ref byte is_def)
```

Parameters

meth common_method_node

is_def [byte↗](#)

Returns

[int↗](#)

GetNameIndex(definition_node)

```
private int GetNameIndex(definition_node dn)
```

Parameters

dn definition_node

Returns

[int↗](#)

GetPropertyReference(common_property_node, ref byte)

```
private int GetPropertyReference(common_property_node prop, ref byte is_def)
```

Parameters

prop common_property_node

is_def byte ↗

Returns

int ↗

GetSizeOfReference(type_node)

```
private int GetSizeOfReference(type_node tn)
```

Parameters

tn type_node

Returns

int ↗

GetSynonymName(common_namespace_node, compiled_type_node)

```
private string GetSynonymName(common_namespace_node cnn, compiled_type_node ctn)
```

Parameters

cnn common_namespace_node

ctn compiled_type_node

Returns

[string](#)

GetTemplateTypeReference(template_class, ref byte)

```
private int GetTemplateTypeReference(template_class tc, ref byte is_def)
```

Parameters

tc template_class

is_def [byte](#)

Returns

[int](#)

GetTokenForNetEntity(ConstructorInfo)

```
private int GetTokenForNetEntity(ConstructorInfo val)
```

Parameters

val [ConstructorInfo](#)

Returns

[int](#)

GetTokenForNetEntity(FieldInfo)

```
private int GetTokenForNetEntity(FieldInfo val)
```

Parameters

val [FieldInfo](#)

Returns

[int](#)

GetTokenForNetEntity(MethodInfo)

```
private int GetTokenForNetEntity(MethodInfo val)
```

Parameters

val [MethodInfo](#)

Returns

[int](#)

GetTokenForNetEntity(PropertyInfo)

```
private int GetTokenForNetEntity(PropertyInfo val)
```

Parameters

val [PropertyInfo](#)

Returns

[int](#)

GetTokenForNetEntity(Type)

```
private int GetTokenForNetEntity(Type val)
```

Parameters

`val Type`

Returns

`int`

GetTypeReference(type_node, ref byte)

```
private int GetTypeReference(type_node tn, ref byte is_def)
```

Parameters

`tn` type_node

`is_def` byte

Returns

`int`

GetUnitReference(namespace_node)

```
private int GetUnitReference(namespace_node nn)
```

Parameters

`nn` namespace_node

Returns

`int`

GetUnitToken(common_namespace_node)

```
private int GetUnitToken(common_namespace_node ns)
```

Parameters

ns common_namespace_node

Returns

[int](#)

GetUsedUnits()

```
private void GetUsedUnits()
```

GetVariableReference(namespace_variable, ref byte)

```
private int GetVariableReference(namespace_variable nv, ref byte is_def)
```

Parameters

nv namespace_variable

is_def [byte](#)

Returns

[int](#)

HasPCUAlwaysRestoreAttribute(definition_node)

```
private bool HasPCUAlwaysRestoreAttribute(definition_node dn)
```

Parameters

dn definition_node

Returns

[bool](#)

IsDefined(definition_node)

```
private bool IsDefined(definition_node dn)
```

Parameters

dn definition_node

Returns

[bool](#)

IsSaved()

```
public bool IsSaved()
```

Returns

[bool](#)

RemoveSelf()

```
internal void RemoveSelf()
```

SaveAttribute(attribute_node)

```
private void SaveAttribute(attribute_node attr)
```

Parameters

attr attribute_node

SaveAttributes(attributes_list, List<int>)

```
private void SaveAttributes(attributes_list attrs, List<int> offs)
```

Parameters

attrs attributes_list

offs [List<int>](#)

SaveCodeReference(function_node)

```
private void SaveCodeReference(function_node fn)
```

Parameters

fn function_node

SaveConstantReferencePosition(namespace_constant_definition)

```
private void SaveConstantReferencePosition(namespace_constant_definition nv)
```

Parameters

nv namespace_constant_definition

SaveExpressionAndOffset(expression_node)

```
private void SaveExpressionAndOffset(expression_node expr)
```

Parameters

expr expression_node

SaveOffsetForAttribute(definition_node)

```
private void SaveOffsetForAttribute(definition_node dn)
```

Parameters

dn definition_node

SavePosition(definition_node)

```
private int SavePosition(definition_node dn)
```

Parameters

dn definition_node

Returns

[int↗](#)

SavePositionAndConstPool(definition_node)

```
private int SavePositionAndConstPool(definition_node dn)
```

Parameters

dn definition_node

Returns

[int↗](#)

SavePositionAndImplementationPool(definition_node)

```
private int SavePositionAndImplementationPool(definition_node dn)
```

Parameters

`dn` `definition_node`

Returns

[int](#)

SaveSemanticTree(CompilationUnit, string, bool)

```
public void SaveSemanticTree(CompilationUnit Unit, string TargetFileName,  
bool IncludeDebugInfo)
```

Parameters

`Unit` [CompilationUnit](#)

`TargetFileName` [string](#)

`IncludeDebugInfo` [bool](#)

SaveVariableReferencePosition(namespace_variable)

```
private void SaveVariableReferencePosition(namespace_variable nv)
```

Parameters

`nv` `namespace_variable`

VisitArrayConst(array_const)

```
private void VisitArrayConst(array_const node)
```

Parameters

`node` `array_const`

VisitArrayInitializer(array_initializer)

```
private void VisitArrayInitializer(array_initializer node)
```

Parameters

`node` array_initializer

VisitAsNode(as_node)

```
private void VisitAsNode(as_node node)
```

Parameters

`node` as_node

VisitBasicFunctionCall(basic_function_call)

```
private void VisitBasicFunctionCall(basic_function_call expr)
```

Parameters

`expr` basic_function_call

VisitBasicFunctionCallAsConstant(basic_function_call_as_constant)

```
private void VisitBasicFunctionCallAsConstant(basic_function_call_as_constant expr)
```

Parameters

`expr` basic_function_call_as_constant

VisitBoolConstNode(bool_const_node)

```
private void VisitBoolConstNode(bool_const_node expr)
```

Parameters

expr bool_const_node

VisitByteConstNode(byte_const_node)

```
private void VisitByteConstNode(byte_const_node expr)
```

Parameters

expr byte_const_node

VisitCaseVariantNode(case_variant_node)

```
private void VisitCaseVariantNode(case_variant_node cvn)
```

Parameters

cvn case_variant_node

VisitCharConstNode(char_const_node)

```
private void VisitCharConstNode(char_const_node expr)
```

Parameters

expr char_const_node

VisitClassFieldReference(class_field_reference)

```
private void VisitClassFieldReference(class_field_reference expr)
```

Parameters

expr class_field_reference

VisitCommonConstructorCall(common_constructor_call)

```
private void VisitCommonConstructorCall(common_constructor_call expr)
```

Parameters

expr common_constructor_call

VisitCommonConstructorCallAsConstant(common_constructor_call_as_constant)

```
private void VisitCommonConstructorCallAsConstant(common_constructor_call_as_constant expr)
```

Parameters

expr common_constructor_call_as_constant

VisitCommonInFuncFuncCall(common_in_function_function_call)

```
private void VisitCommonInFuncFuncCall(common_in_function_function_call expr)
```

Parameters

expr common_in_function_function_call

VisitCommonMethodCall(common_method_call)

```
private void VisitCommonMethodCall(common_method_call expr)
```

Parameters

`expr` common_method_call

VisitCommonNamespaceFunctionCall(common_namespace_function_call)

```
private void VisitCommonNamespaceFunctionCall(common_namespace_function_call expr)
```

Parameters

`expr` common_namespace_function_call

VisitCommonNamespaceFunctionCallNode AsConstant(common_namespace_function_call_as_constant)

```
private void  
VisitCommonNamespaceFunctionCallNodeAsConstant(common_namespace_function_call_as_constant  
node)
```

Parameters

`node` common_namespace_function_call_as_constant

VisitCommonParameterReference(common_parameter_reference)

```
private void VisitCommonParameterReference(common_parameter_reference expr)
```

Parameters

`expr` common_parameter_reference

VisitCommonStaticMethodCall(common_static_method_call)

```
private void VisitCommonStaticMethodCall(common_static_method_call expr)
```

Parameters

`expr` common_static_method_call

VisitCommonStaticMethodCallNodeAsConstant(common_static_method_call_as_constant)

```
private void VisitCommonStaticMethodCallNodeAsConstant(common_static_method_call_as_constant node)
```

Parameters

`node` common_static_method_call_as_constant

VisitCompiledConstructorCall(compiled_constructor_call)

```
private void VisitCompiledConstructorCall(compiled_constructor_call expr)
```

Parameters

`expr` compiled_constructor_call

VisitCompiledConstructorCallAsConstant(compiled_constructor_call_as_constant)

```
private void VisitCompiledConstructorCallAsConstant(compiled_constructor_call_as_constant node)
```

Parameters

`node` `compiled_constructor_call_as_constant`

VisitCompiledFunctionCall(`compiled_function_call`)

```
private void VisitCompiledFunctionCall(compiled_function_call expr)
```

Parameters

`expr` `compiled_function_call`

VisitCompiledStaticCall(`compiled_static_method_call`)

```
private void VisitCompiledStaticCall(compiled_static_method_call expr)
```

Parameters

`expr` `compiled_static_method_call`

VisitCompiledStaticFieldReferenceAsConstant(`compiled_static_field_reference_as_constant`)

```
private void  
VisitCompiledStaticFieldReferenceAsConstant(compiled_static_field_reference_as_constant en)
```

Parameters

`en` `compiled_static_field_reference_as_constant`

VisitCompiledStaticMethodCall(`compiled_static_method_call`)

```
private void VisitCompiledStaticMethodCall(compiled_static_method_call expr)
```

Parameters

`expr` `compiled_static_method_call`

VisitCompiledStaticMethodCallNodeAsConstant(`compiled_static_method_call_as_constant`)

```
private void VisitCompiledStaticMethodCallNodeAsConstant(CompiledStaticMethodCallAsConstant node)
```

Parameters

`node` `compiled_static_method_call_as_constant`

VisitCompiledTypeDefinition(`compiled_type_node`)

```
private void VisitCompiledTypeDefinition(CompiledTypeNode type)
```

Parameters

`type` `compiled_type_node`

VisitCompiledVariableReference(`compiled_variable_reference`)

```
private void VisitCompiledVariableReference(CompiledVariableReference expr)
```

Parameters

`expr` `compiled_variable_reference`

VisitConstantDefinition(`namespace_constant_definition`)

```
private void VisitConstantDefinition(NamespaceConstantDefinition cnst)
```

Parameters

`cnst` namespace_constant_definition

VisitConstantDefinitions()

```
private void VisitConstantDefinitions()
```

VisitConstantInTypeDefinition(class_constant_definition, int)

```
private void VisitConstantInTypeDefinition(class_constant_definition cdn, int offset)
```

Parameters

`cdn` class_constant_definition

`offset` `int`

VisitConstantInTypeDefinitions(common_type_node)

```
private void VisitConstantInTypeDefinitions(common_type_node ctn)
```

Parameters

`ctn` common_type_node

VisitDefaultOperator(default_operator_node)

```
private void VisitDefaultOperator(default_operator_node expr)
```

Parameters

`expr` default_operator_node

VisitDefaultOperatorAsConstant(default_operator_node_as_constant)

```
private void VisitDefaultOperatorAsConstant(default_operator_node_as_constant expr)
```

Parameters

expr default_operator_node_as_constant

VisitDerefNode(dereference_node)

```
private void VisitDerefNode(dereference_node expr)
```

Parameters

expr dereference_node

VisitDoubleConstNode(double_const_node)

```
private void VisitDoubleConstNode(double_const_node expr)
```

Parameters

expr double_const_node

VisitDoubleQuestionColonExpression(double_question_colon_expression)

```
private void VisitDoubleQuestionColonExpression(double_question_colon_expression node)
```

Parameters

node double_question_colon_expression

VisitEmpty(empty_statement)

```
private void VisitEmpty(empty_statement stmt)
```

Parameters

`stmt` empty_statement

VisitEnumConstNode(enum_const_node)

```
private void VisitEnumConstNode(enum_const_node en)
```

Parameters

`en` enum_const_node

VisitEventDefinition(common_event, int)

```
private void VisitEventDefinition(common_event _event, int offset)
```

Parameters

`_event` common_event

`offset` [int](#)

VisitEventDefinitions()

```
private void VisitEventDefinitions()
```

VisitEventDefinitions(common_type_node)

```
private void VisitEventDefinitions(common_type_node ctn)
```

Parameters

`ctn` common_type_node

VisitExpression(expression_node)

```
private void VisitExpression(expression_node en)
```

Parameters

`en` expression_node

VisitExternalStatementNode(external_statement)

```
private void VisitExternalStatementNode(external_statement stmt)
```

Parameters

`stmt` external_statement

VisitFieldDefinition(class_field, int)

```
private void VisitFieldDefinition(class_field field, int offset)
```

Parameters

`field` class_field

`offset` int

VisitFieldDefinitions(common_type_node)

```
private void VisitFieldDefinitions(common_type_node ctn)
```

Parameters

`ctn` common_type_node

VisitFloatConst(float_const_node)

```
private void VisitFloatConst(float_const_node node)
```

Parameters

`node` float_const_node

VisitFor(for_node)

```
private void VisitFor(for_node stmt)
```

Parameters

`stmt` for_node

VisitForBreakNode(for_break_node)

```
private void VisitForBreakNode(for_break_node expr)
```

Parameters

`expr` for_break_node

VisitForContinueNode(for_continue_node)

```
private void VisitForContinueNode(for_continue_node expr)
```

Parameters

`expr` for_continue_node

VisitForeach(foreach_node)

```
private void VisitForeach(foreach_node fn)
```

Parameters

`fn` foreach_node

VisitForeachBreak(foreach_break_node)

```
private void VisitForeachBreak(foreach_break_node sn)
```

Parameters

`sn` foreach_break_node

VisitForeachContinue(foreach_continue_node)

```
private void VisitForeachContinue(foreach_continue_node sn)
```

Parameters

`sn` foreach_continue_node

VisitFunctionConstant(function_constant_definition)

```
private void VisitFunctionConstant(function_constant_definition cnst)
```

Parameters

`cnst` function_constant_definition

VisitFunctionConstantReference(function_constant_reference)

```
private void VisitFunctionConstantReference(function_constant_reference expr)
```

Parameters

`expr` function_constant_reference

VisitFunctionDefinition(common_namespace_function_node)

```
private void VisitFunctionDefinition(common_namespace_function_node func)
```

Parameters

`func` common_namespace_function_node

VisitFunctionDefinitions()

```
private void VisitFunctionDefinitions()
```

VisitFunctionImplementation(common_namespace_function_node)

```
private void VisitFunctionImplementation(common_namespace_function_node func)
```

Parameters

`func` common_namespace_function_node

VisitFunctionWithImplementation(common_namespace_function_node)

```
private int VisitFunctionWithImplementation(common_namespace_function_node func)
```

Parameters

func common_namespace_function_node

Returns

[int](#)

VisitGetAddrNode(get_addr_node)

```
private void VisitGetAddrNode(get_addr_node expr)
```

Parameters

expr get_addr_node

VisitGoto(goto_statement)

```
private void VisitGoto(goto_statement gs)
```

Parameters

gs goto_statement

VisitIf(if_node)

```
private void VisitIf(if_node stmt)
```

Parameters

stmt if_node

VisitIntConstNode(int_const_node)

```
private void VisitIntConstNode(int_const_node expr)
```

Parameters

`expr` int_const_node

VisitIsNode(is_node)

```
private void VisitIsNode(is_node node)
```

Parameters

`node` is_node

VisitLabelDeclaration(label_node)

```
private void VisitLabelDeclaration(label_node ln)
```

Parameters

`ln` label_node

VisitLabelDeclarations(List<label_node>)

```
private void VisitLabelDeclarations(List<label_node> labels)
```

Parameters

`labels` [List](#)<label_node>

VisitLabeledStatement(labeled_statement)

```
private void VisitLabeledStatement(labeled_statement ls)
```

Parameters

ls labeled_statement

VisitLocalBlockVariable(local_block_variable)

```
private void VisitLocalBlockVariable(local_block_variable lv)
```

Parameters

lv local_block_variable

VisitLocalBlockVariableReference(local_block_variable_reference)

```
private void VisitLocalBlockVariableReference(local_block_variable_reference expr)
```

Parameters

expr local_block_variable_reference

VisitLocalVariable(local_variable)

```
private void VisitLocalVariable(local_variable var)
```

Parameters

var local_variable

VisitLocalVariableReference(local_variable_reference)

```
private void VisitLocalVariableReference(local_variable_reference expr)
```

Parameters

expr local_variable_reference

VisitLock(lock_statement)

```
private void VisitLock(lock_statement node)
```

Parameters

node lock_statement

VisitLongConstNode(long_const_node)

```
private void VisitLongConstNode(long_const_node expr)
```

Parameters

expr long_const_node

VisitMethodDefinition(common_method_node, int)

```
private void VisitMethodDefinition(common_method_node meth, int offset)
```

Parameters

meth common_method_node

offset [int](#)

VisitMethodDefinitions(common_type_node)

```
private void VisitMethodDefinitions(common_type_node ctn)
```

Parameters

ctn common_type_node

VisitMethodImplementation(common_method_node)

```
private void VisitMethodImplementation(common_method_node meth)
```

Parameters

meth common_method_node

VisitNamespaceConstantReference(namespace_constant_reference)

```
private void VisitNamespaceConstantReference(namespace_constant_reference expr)
```

Parameters

expr namespace_constant_reference

VisitNamespaceEventDefinition(common_namespace_event)

```
private void VisitNamespaceEventDefinition(common_namespace_event _event)
```

Parameters

_event common_namespace_event

VisitNamespaceVariableReference(namespace_variable_reference)

```
private void VisitNamespaceVariableReference(namespace_variable_reference expr)
```

Parameters

`expr` namespace_variable_reference

VisitNestedFunctionDefinition(common_in_function_function_node)

```
private void VisitNestedFunctionDefinition(common_in_function_function_node func)
```

Parameters

`func` common_in_function_function_node

VisitNestedFunctionImplementation(common_in_function_function_node)

```
private void VisitNestedFunctionImplementation(common_in_function_function_node func)
```

Parameters

`func` common_in_function_function_node

VisitNonStaticEventReference(nonstatic_event_reference)

```
private void VisitNonStaticEventReference(nonstatic_event_reference en)
```

Parameters

`en` nonstatic_event_reference

VisitNonStaticPropertyReference(non_static_property_reference)

```
private void VisitNonStaticPropertyReference(non_static_property_reference expr)
```

Parameters

`expr` non_static_property_reference

VisitNullConstNode(null_const_node)

```
private void VisitNullConstNode(null_const_node expr)
```

Parameters

`expr` null_const_node

VisitPInvokeStatement(pinvoke_statement)

```
private void VisitPInvokeStatement(pinvoke_statement stmt)
```

Parameters

`stmt` pinvoke_statement

VisitParameter(common_parameter)

```
private void VisitParameter(common_parameter p)
```

Parameters

`p` common_parameter

VisitPropertyDefinition(common_property_node, int)

```
private void VisitPropertyDefinition(common_property_node prop, int offset)
```

Parameters

prop common_property_node

offset int ↗

VisitPropertyDefinitions(common_type_node)

```
private void VisitPropertyDefinitions(common_type_node ctn)
```

Parameters

ctn common_type_node

VisitQuestionColonExpression(question_colon_expression)

```
private void VisitQuestionColonExpression(question_colon_expression node)
```

Parameters

node question_colon_expression

VisitRecordConst(record_constant)

```
private void VisitRecordConst(record_constant node)
```

Parameters

node record_constant

VisitRecordInitializer(record_initializer)

```
private void VisitRecordInitializer(record_initializer expr)
```

Parameters

`expr` record_initializer

VisitRefTypeDefinition(ref_type_node)

```
private void VisitRefTypeDefinition(ref_type_node node)
```

Parameters

`node` ref_type_node

VisitRefTypeDefinitions()

```
private void VisitRefTypeDefinitions()
```

VisitRepeat(repeat_node)

```
private void VisitRepeat(repeat_node stmt)
```

Parameters

`stmt` repeat_node

VisitRepeatBreakNode(repeat_break_node)

```
private void VisitRepeatBreakNode(repeat_break_node expr)
```

Parameters

expr repeat_break_node

VisitRepeatContinueNode(repeat_continue_node)

```
private void VisitRepeatContinueNode(repeat_continue_node expr)
```

Parameters

expr repeat_continue_node

VisitRethrow(rethrow_statement_node)

```
private void VisitRethrow(rethrow_statement_node sn)
```

Parameters

sn rethrow_statement_node

VisitReturnNode(return_node)

```
private void VisitReturnNode(return_node expr)
```

Parameters

expr return_node

VisitRuntimeStatement(runtime_statement)

```
private void VisitRuntimeStatement(runtime_statement rts)
```

Parameters

`rts` runtime_statement

VisitSByteConstNode(sbyte_const_node)

```
private void VisitSByteConstNode(sbyte_const_node expr)
```

Parameters

`expr` sbyte_const_node

VisitShortConstNode(short_const_node)

```
private void VisitShortConstNode(short_const_node expr)
```

Parameters

`expr` short_const_node

VisitSimpleArrayIndexing(simple_array_indexing)

```
private void VisitSimpleArrayIndexing(simple_array_indexing expr)
```

Parameters

`expr` simple_array_indexing

VisitSizeOfOperator(sizeof_operator)

```
private void VisitSizeOfOperator(sizeof_operator node)
```

Parameters

`node` sizeof_operator

VisitSizeOfOperatorAsConstant(sizeof_operator_as_constant)

```
private void VisitSizeOfOperatorAsConstant(sizeof_operator_as_constant node)
```

Parameters

node sizeof_operator_as_constant

VisitStatement(statement_node)

```
private void VisitStatement(statement_node sn)
```

Parameters

sn statement_node

VisitStatementList(statements_list)

```
private void VisitStatementList(statements_list stmt)
```

Parameters

stmt statements_list

VisitStatementsExpressionNode(statements_expression_node)

```
private void VisitStatementsExpressionNode(statements_expression_node node)
```

Parameters

node statements_expression_node

VisitStaticClassFieldReference(static_class_field_reference)

```
private void VisitStaticClassFieldReference(static_class_field_reference expr)
```

Parameters

expr static_class_field_reference

VisitStaticCompiledVariableReference(static_compiled_variable_reference)

```
private void VisitStaticCompiledVariableReference(static_compiled_variable_reference expr)
```

Parameters

expr static_compiled_variable_reference

VisitStaticEventReference(static_event_reference)

```
private void VisitStaticEventReference(static_event_reference en)
```

Parameters

en static_event_reference

VisitStringConstNode(string_const_node)

```
private void VisitStringConstNode(string_const_node expr)
```

Parameters

expr string_const_node

VisitSwitchNode(switch_node)

```
private void VisitSwitchNode(switch_node stmt)
```

Parameters

stmt switch_node

VisitTemplateClassDefinition(template_class)

```
private void VisitTemplateClassDefinition(template_class tclass)
```

Parameters

tclass template_class

VisitTemplateClasses()

```
private void VisitTemplateClasses()
```

VisitThisNode(this_node)

```
private void VisitThisNode(this_node expr)
```

Parameters

expr this_node

VisitThrow(throw_statement_node)

```
private void VisitThrow(throw_statement_node stmt)
```

Parameters

`stmt` throw_statement_node

VisitTryBlock(try_block)

`private void VisitTryBlock(try_block tryblock)`

Parameters

`tryblock` try_block

VisitTypeDefinition(common_type_node)

`private void VisitTypeDefinition(common_type_node type)`

Parameters

`type` common_type_node

VisitTypeDefinitions()

`private void VisitTypeDefinitions()`

VisitTypeImplementation(common_type_node)

`private void VisitTypeImplementation(common_type_node type)`

Parameters

`type` common_type_node

VisitTypeMemberDefinition(common_type_node)

```
private void VisitTypeMemberDefinition(common_type_node ctn)
```

Parameters

ctn common_type_node

VisitTypeOfOperator(typeof_operator)

```
private void VisitTypeOfOperator(typeof_operator node)
```

Parameters

node typeof_operator

VisitTypeSynonym(type_synonym)

```
private void VisitTypeSynonym(type_synonym synonym)
```

Parameters

synonym type_synonym

VisitTypeSynonyms()

```
private void VisitTypeSynonyms()
```

VisitUIntConstNode(uint_const_node)

```
private void VisitUIntConstNode(uint_const_node expr)
```

Parameters

```
expr uint_const_node
```

VisitULongConstNode(ulong_const_node)

```
private void VisitULongConstNode(ulong_const_node expr)
```

Parameters

```
expr ulong_const_node
```

VisitUShortConstNode(ushort_const_node)

```
private void VisitUShortConstNode(ushort_const_node expr)
```

Parameters

```
expr ushort_const_node
```

VisitVariableDefinition(namespace_variable)

```
private void VisitVariableDefinition(namespace_variable var)
```

Parameters

```
var namespace_variable
```

VisitVariableDefinitions()

```
private void VisitVariableDefinitions()
```

VisitWhile(while_node)

```
private void VisitWhile(while_node stmt)
```

Parameters

stmt while_node

VisitWhileBreakNode(while_break_node)

```
private void VisitWhileBreakNode(while_break_node expr)
```

Parameters

expr while_break_node

VisitWhileContinueNode(while_continue_node)

```
private void VisitWhileContinueNode(while_continue_node expr)
```

Parameters

expr while_continue_node

WriteArrayType(simple_array)

```
private void WriteArrayType(simple_array type)
```

Parameters

type simple_array

WriteBoundedArray(bounded_array_interface)

```
private void WriteBoundedArray(bounded_array_interface bai)
```

Parameters

bai bounded_array_interface

WriteCRC32InHeader(byte[], BinaryWriter)

```
private void WriteCRC32InHeader(byte[] buf, BinaryWriter bw)
```

Parameters

buf byte[]

bw BinaryWriter

WriteCompiledConstructor(compiled_constructor_node)

```
private void WriteCompiledConstructor(compiled_constructor_node ccn)
```

Parameters

ccn compiled_constructor_node

WriteCompiledMethod(compiled_function_node)

```
private void WriteCompiledMethod(compiled_function_node cfn)
```

Parameters

cfn compiled_function_node

WriteCompiledProperty(compiled_property_node)

```
private void WriteCompiledProperty(compiled_property_node cpn)
```

Parameters

cpn compiled_property_node

WriteCompiledVariable(compiled_variable_definition)

```
private void WriteCompiledVariable(compiled_variable_definition cvd)
```

Parameters

cvd compiled_variable_definition

WriteConstantPositions()

```
private void WriteConstantPositions()
```

WriteConstantReference(namespace_constant_definition)

```
private void WriteConstantReference(namespace_constant_definition nv)
```

Parameters

nv namespace_constant_definition

WriteDebugInfo(ILocation)

```
private void WriteDebugInfo(ILocation loc)
```

Parameters

loc ILocation

WriteDebugInfo(BinaryWriter, ILocation)

```
private void WriteDebugInfo(BinaryWriter bw, ILocation loc)
```

Parameters

bw [BinaryWriter](#)

loc ILocation

WriteEventReference(event_node)

```
private void WriteEventReference(event_node ev)
```

Parameters

ev event_node

WriteFieldReference(class_field)

```
private void WriteFieldReference(class_field field)
```

Parameters

field class_field

WriteFieldReference(var_definition_node)

```
private void WriteFieldReference(var_definition_node vdn)
```

Parameters

```
vdn var_definition_node
```

WriteFunctionReference(common_namespace_function_node)

```
private void WriteFunctionReference(common_namespace_function_node fn)
```

Parameters

fn common_namespace_function_node

WriteFunctionReferenceWithDelay(common_namespace_function_node)

```
private void WriteFunctionReferenceWithDelay(common_namespace_function_node fn)
```

Parameters

fn common_namespace_function_node

WriteFunctionReferences()

```
private void WriteFunctionReferences()
```

WriteGenericFunctionInformation(common_function_node)

```
private void WriteGenericFunctionInformation(common_function_node func)
```

Parameters

func common_function_node

WriteGenericMethodReference(function_node)

```
private void WriteGenericMethodReference(function_node meth)
```

Parameters

meth function_node

WriteGenericNamespaceFunctionReference(generic_namespace_function_instance_node)

```
private void WriteGenericNamespaceFunctionReference(generic_namespace_function_instance_node func)
```

Parameters

func generic_namespace_function_instance_node

WriteGenericParameter(common_type_node)

```
private void WriteGenericParameter(common_type_node type)
```

Parameters

type common_type_node

WriteGenericTypeInstance(generic_instance_type_node)

```
private void WriteGenericTypeInstance(generic_instance_type_node gitn)
```

Parameters

gitn generic_instance_type_node

WriteImplementingInterfaces(common_type_node)

```
private void WriteImplementingInterfaces(common_type_node type)
```

Parameters

type common_type_node

WriteInitExpressions()

```
private void WriteInitExpressions()
```

WriteMethodReference(common_method_node)

```
private void WriteMethodReference(common_method_node meth)
```

Parameters

meth common_method_node

WriteMethodReference(function_node)

```
private void WriteMethodReference(function_node fn)
```

Parameters

fn function_node

WritePCUHeader(BinaryWriter)

```
private void WritePCUHeader(BinaryWriter fbw)
```

Parameters

fbw [BinaryWriter](#) ↗

WritePointerType(ref_type_node)

```
private void WritePointerType(ref_type_node type)
```

Parameters

type ref_type_node

WritePropertyReference(common_property_node)

```
private void WritePropertyReference(common_property_node prop)
```

Parameters

prop common_property_node

WritePropertyReference(property_node)

```
private void WritePropertyReference(property_node pn)
```

Parameters

pn property_node

WriteShortStringType(short_string_type_node)

```
private void WriteShortStringType(short_string_type_node type)
```

Parameters

`type` short_string_type_node

WriteTemplateClassReference(template_class)

`private void WriteTemplateClassReference(template_class tc)`

Parameters

`tc` template_class

WriteTemplateInstance(common_type_node)

`private void WriteTemplateInstance(common_type_node cnode)`

Parameters

`cnode` common_type_node

WriteTypeList(List<type_node>)

`private void WriteTypeList(List<type_node> types)`

Parameters

`types` [List](#)<type_node>

WriteTypeParamsEliminations(List<ICommonTypeNode>)

`private void WriteTypeParamsEliminations(List<ICommonTypeNode> tpars)`

Parameters

`tpars` [List](#)<ICommonTypeNode>

WriteTypeReference(type_node)

```
private void WriteTypeReference(type_node type)
```

Parameters

type type_node

WriteTypeReferenceWithDelay(common_type_node)

```
private void WriteTypeReferenceWithDelay(common_type_node type)
```

Parameters

type common_type_node

WriteTypeReferences()

```
private void WriteTypeReferences()
```

WriteUnit()

```
private void WriteUnit()
```

WriteUnsizedArrayType(type_node, array_internal_interface)

```
private void WriteUnsizedArrayType(type_node type, array_internal_interface aii)
```

Parameters

type type_node

aii array_internal_interface

WriteVariablePositions()

```
private void WriteVariablePositions()
```

WriteVariableReference(namespace_variable)

```
private void WriteVariableReference(namespace_variable nv)
```

Parameters

nv namespace_variable

add_not_comp_members(definition_node, PCUWriter)

```
private static void add_not_comp_members(definition_node dn, PCUWriter pw)
```

Parameters

dn definition_node

pw [PCUWriter](#)

convert_field_access_level(field_access_level)

```
private access_level convert_field_access_level(field_access_level fal)
```

Parameters

fal field_access_level

Returns

access_level

Events

ChangeState

```
public event PCUWriter.ChangeStateDelegate ChangeState
```

Event Type

[PCUWriter.ChangeStateDelegate](#)

Delegate PCUWriter.ChangeStateDelegate

Namespace: [PascalABCCompiler.PCU](#)

Assembly: Compiler.dll

```
public delegate void PCUWriter.ChangeStateDelegate(object Sender, PCUReaderWriterState  
State, object obj)
```

Parameters

Sender [object](#)

State [PCUReaderWriterState](#)

obj [object](#)

Constructors

ChangeStateDelegate(object, IntPtr)

```
public ChangeStateDelegate(object @object, IntPtr method)
```

Parameters

object [object](#)

method [IntPtr](#)

Methods

BeginInvoke(object, PCUReaderWriterState, object, AsyncCallback, object)

```
public virtual IAsyncResult BeginInvoke(object Sender, PCUReaderWriterState State, object  
obj, AsyncCallback callback, object @object)
```

Parameters

Sender [object](#)

State [PCUReaderWriterState](#)

obj [object](#)

callback [AsyncCallback](#)

object [object](#)

Returns

[IAsyncResult](#)

EndInvoke(IAsyncResult)

```
public virtual void EndInvoke(IAsyncResult result)
```

Parameters

result [IAsyncResult](#)

Invoke(object, PCUReaderWriterState, object)

```
public virtual void Invoke(object Sender, PCUReaderWriterState State, object obj)
```

Parameters

Sender [object](#)

State [PCUReaderWriterState](#)

obj [object](#)

Enum TypeKind

Namespace: [PascalABCCompiler.PCU](#)

Assembly: Compiler.dll

```
public enum TypeKind
```

Fields

Array = 2

BoundedArray = 5

GenericInstance = 7

GenericParameterOfFunction = 10

GenericParameterOfMethod = 11

GenericParameterOfType = 9

LambdaAnyType = 12

Pointer = 3

ShortString = 8

TemplateInstance = 6

UnsizedArray = 4

Class WrappedClassScope

Namespace: [PascalABCCompiler.PCU](#)

Assembly: Compiler.dll

```
public class WrappedClassScope : ClassScope
```

Inheritance

[object](#) ← BaseScope ← Scope ← ClassScope ← WrappedClassScope

Derived

[WrappedInterfaceScope](#)

Inherited Members

ClassScope.PartialScope , ClassScope.class_type , ClassScope.BaseClassScope ,
[ClassScope.FindOnlyInType\(string, Scope\)](#) , Scope.Name , Scope.Symbols , Scope.InternalScopes ,
Scope.SymbolTable , Scope.CaseSensitive , Scope.AddStatementsToFront , Scope.TopScope ,
Scope.ScopeNum , Scope.ToString() , Scope.ClearScope() , [Scope.FindOnlyInScopeAndBlocks\(string\)](#) ,
[Scope.FindOnlyInScope\(string\)](#) , [Scope.AddSymbol\(string, SymbolInfo\)](#) , [object.Equals\(object\)](#) ,
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.FieldSetter\(string, string, object\)](#) ,
[object.FieldGetter\(string, string, ref object\)](#) , [object.GetFieldInfo\(string, string\)](#)

Constructors

WrappedClassScope(PCUReader, Scope, Scope)

```
public WrappedClassScope(PCUReader pr, Scope top_scope, Scope up_scope)
```

Parameters

pr [PCUReader](#)

top_scope Scope

up_scope Scope

Fields

ctn

```
internal common_type_node ctn
```

Field Value

common_type_node

pr

```
protected PCUReader pr
```

Field Value

[PCUReader](#)

Methods

Find(string)

```
public override List<SymbolInfo> Find(string name)
```

Parameters

name [string](#)

Returns

[List](#)<SymbolInfo>

Find(string, Scope)

```
public override List<SymbolInfo> Find(string name, Scope CurrentScope)
```

Parameters

name [string](#)

CurrentScope [Scope](#)

Returns

[List](#) <SymbolInfo>

FindWithoutCreation(string)

```
public List<SymbolInfo> FindWithoutCreation(string name)
```

Parameters

name [string](#)

Returns

[List](#) <SymbolInfo>

RestoreMembers(string)

```
public void RestoreMembers(string name)
```

Parameters

name [string](#)

hasNotRestoreAttribute()

```
private bool hasNotRestoreAttribute()
```

Returns

[bool](#)

needRestore(PCUSymbolInfo, string)

```
private bool needRestore(PCUSymbolInfo pcu_tsi, string name)
```

Parameters

[pcu_tsi](#) PCUSymbolInfo

[name](#) [string](#)

Returns

[bool](#)

Class WrappedInterfaceScope

Namespace: [PascalABCCompiler.PCU](#)

Assembly: Compiler.dll

```
public class WrappedInterfaceScope : WrappedClassScope, IInterfaceScope
```

Inheritance

[object](#) ← BaseScope ← Scope ← ClassScope ← [WrappedClassScope](#) ← WrappedInterfaceScope

Implements

IInterfaceScope

Inherited Members

[WrappedClassScope.pr](#) , [WrappedClassScope.ctn](#) , [WrappedClassScope.Find\(string\)](#) ,
[WrappedClassScope.hasNotRestoreAttribute\(\)](#) ,
[WrappedClassScope.needRestore\(PCUSymbolInfo, string\)](#) , [WrappedClassScope.RestoreMembers\(string\)](#) ,
[WrappedClassScope.Find\(string, Scope\)](#) , [WrappedClassScope.FindWithoutCreation\(string\)](#) ,
ClassScope.PartialScope , ClassScope.class_type , ClassScope.BaseClassScope ,
[ClassScope.FindOnlyInType\(string, Scope\)](#) , Scope.Name , Scope.Symbols , Scope.InternalScopes ,
Scope.SymbolTable , Scope.CaseSensitive , Scope.AddStatementsToFront , Scope.TopScope ,
Scope.ScopeNum , Scope.ToString() , Scope.ClearScope() , [Scope.FindOnlyInScopeAndBlocks\(string\)](#) ,
[Scope.FindOnlyInScope\(string\)](#) , [Scope.AddSymbol\(string, SymbolInfo\)](#) , [object.Equals\(object\)](#) ,
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.FieldSetter\(string, string, object\)](#) ,
[object.FieldGetter\(string, string, ref object\)](#) , [object.GetFieldInfo\(string, string\)](#)

Constructors

WrappedInterfaceScope(PCUReader, Scope, Scope, Scope[])

```
public WrappedInterfaceScope(PCUReader pr, Scope TopScope, Scope BaseClassScope,  
Scope[] vTopInterfaceScopeArray)
```

Parameters

pr [PCUReader](#)

TopScope Scope

BaseClassScope Scope

vTopInterfaceScopeArray Scope[]

WrappedInterfaceScope(PCUReader, Scope, Scope[])

```
public WrappedInterfaceScope(PCUReader pr, Scope TopScope, Scope[] vTopInterfaceScopeArray)
```

Parameters

pr [PCUReader](#)

TopScope Scope

vTopInterfaceScopeArray Scope[]

Fields

_TopInterfaceScopeArray

```
private Scope[] _TopInterfaceScopeArray
```

Field Value

Scope[]

Properties

TopInterfaceScopeArray

```
public virtual Scope[] TopInterfaceScopeArray { get; set; }
```

Property Value

Scope[]

Class WrappedUnitImplementationScope

Namespace: [PascalABCCompiler.PCU](#)

Assembly: Compiler.dll

```
public class WrappedUnitImplementationScope : UnitImplementationScope
```

Inheritance

[object](#) ← BaseScope ← Scope ← UnitPartScope ← UnitImplementationScope ← WrappedUnitImplementationScope

Inherited Members

UnitPartScope.TopScopeArray , Scope.Name , Scope.Symbols , Scope.InternalScopes ,
Scope.SymbolTable , Scope.CaseSensitive , Scope.AddStatementsToFront , Scope.TopScope ,
Scope.ScopeNum , Scope.ToString() , Scope.ClearScope() , [Scope.Find\(string, Scope\)](#) ,
[Scope.FindOnlyInScopeAndBlocks\(string\)](#) , [Scope.FindOnlyInType\(string, Scope\)](#) ,
[Scope.AddSymbol\(string, SymbolInfo\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.FieldSetter\(string, string, object\)](#) ,
[object.FieldGetter\(string, string, ref object\)](#) , [object.GetFieldInfo\(string, string\)](#)

Constructors

WrappedUnitImplementationScope(PCUReader, Scope)

```
public WrappedUnitImplementationScope(PCUReader pr, Scope TopScope)
```

Parameters

pr [PCUReader](#)

TopScope Scope

Fields

pr

```
protected PCUReader pr
```

Field Value

[PCUReader](#)

Methods

Find(string)

```
public override List<SymbolInfo> Find(string name)
```

Parameters

name [string](#)

Returns

[List](#)<SymbolInfo>

FindOnlyInScope(string)

```
public override List<SymbolInfo> FindOnlyInScope(string name)
```

Parameters

name [string](#)

Returns

[List](#)<SymbolInfo>

FindWithoutCreation(string)

```
public List<SymbolInfo> FindWithoutCreation(string name)
```

Parameters

name [string](#)

Returns

[List](#) <SymbolInfo>

Class WrappedUnitInterfaceScope

Namespace: [PascalABCCompiler.PCU](#)

Assembly: Compiler.dll

```
public class WrappedUnitInterfaceScope : UnitInterfaceScope
```

Inheritance

[object](#) ← BaseScope ← Scope ← UnitPartScope ← UnitInterfaceScope ← WrappedUnitInterfaceScope

Inherited Members

UnitPartScope.TopScopeArray , Scope.Name , Scope.Symbols , Scope.InternalScopes ,
Scope.SymbolTable , Scope.CaseSensitive , Scope.AddStatementsToFront , Scope.TopScope ,
Scope.ScopeNum , Scope.ToString() , Scope.ClearScope() , [Scope.Find\(string, Scope\)](#) ,
[Scope.FindOnlyInScopeAndBlocks\(string\)](#) , [Scope.FindOnlyInType\(string, Scope\)](#) ,
[Scope.AddSymbol\(string, SymbolInfo\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.FieldSetter\(string, string, object\)](#) ,
[object.FieldGetter\(string, string, ref object\)](#) , [object.GetFieldInfo\(string, string\)](#)

Constructors

WrappedUnitInterfaceScope(PCUReader)

```
public WrappedUnitInterfaceScope(PCUReader pr)
```

Parameters

pr [PCUReader](#)

Fields

pr

```
protected PCUReader pr
```

Field Value

[PCUReader](#)

Methods

Find(string)

```
public override List<SymbolInfo> Find(string name)
```

Parameters

name [string](#)

Returns

[List](#)<SymbolInfo>

FindOnlyInScope(string)

```
public override List<SymbolInfo> FindOnlyInScope(string name)
```

Parameters

name [string](#)

Returns

[List](#)<SymbolInfo>

FindWithoutCreation(string)

```
public List<SymbolInfo> FindWithoutCreation(string name)
```

Parameters

name [string](#)

Returns

[List](#)<SymbolInfo>

Class wrapped_common_type_node

Namespace: [PascalABCCompiler.PCU](#)

Assembly: Compiler.dll

```
public class wrapped_common_type_node : common_type_node, ICommonTypeNode, ITypeNode,  
IDefinitionNode, ISemanticNode, INamespaceMemberNode, ILocated
```

Inheritance

[object](#) ← semantic_node ← definition_node ← type_node ← wrapped_type ← common_type_node ← wrapped_common_type_node

Implements

ICommonTypeNode, ITypeNode, IDefinitionNode, ISemanticNode, INamespaceMemberNode, ILocated

Inherited Members

common_type_node._name , common_type_node._type_access_level , common_type_node._base_type ,
common_type_node._comprehensive_namespace , common_type_node._methods ,
common_type_node._fields , common_type_node._properties , common_type_node._const_defs ,
common_type_node._events , common_type_node._default_property_node , common_type_node._loc ,
common_type_node.defined_in_scope , common_type_node._is_value , common_type_node._is_class ,
common_type_node._scope , common_type_node.has_default_constructor ,
common_type_node.has_explicit_default_constructor ,
common_type_node.has_user_defined_constructor , common_type_node._has_static_constructor ,
common_type_node._static_constr , common_type_node._is_generic_type_definition ,
common_type_node._base_generic_instance , common_type_node._runtime_initialization_marker ,
common_type_node._generic_type_container , common_type_node._generic_function_container ,
common_type_node._generic_params , common_type_node._is_interface ,
common_type_node._is_delegate , common_type_node._forward_declaration_only ,
common_type_node._implementing_interfaces , common_type_node._original_template ,
common_type_node._depended_from_indefinite , common_type_node._type_special_kind ,
common_type_node._sealed , common_type_node._is_abstract , common_type_node.abstract_reason ,
common_type_node._is_partial , common_type_node._is_static , common_type_node._is_ficive ,
common_type_node._parameters_eliminations ,
[common_type_node.SetImplementingInterfaces\(List<ITypeNode>\)](#) ,
[common_type_node.SetDependedFromIndefinite\(bool\)](#) , [common_type_node.SetIsSealed\(bool\)](#) ,
[common_type_node.SetIsAbstract\(bool, ClassAbstractReason\)](#) , [common_type_node.SetIsStatic\(bool\)](#) ,
common_type_node.has_flags_attribute() , common_type_node.Merge(common_type_node) ,
common_type_node.add_additional_enum_operations() , common_type_node.SetBaseType(type_node) ,

common_type_node.SetBaseTypeIgnoringScope(type_node) , [common_type_node.SetName\(string\)](#) ,
[common_type_node.find\(string, bool\)](#) , [common_type_node.find\(string, Scope, bool\)](#) ,
[common_type_node.find_first_in_type\(string, bool\)](#) ,
[common_type_node.add_name\(string, SymbolInfo\)](#) ,
[common_type_node.add_generated_name\(string, SymbolInfo\)](#) ,
common_type_node.visit(ISemanticVisitor) , common_type_node.get_implicit_conversion_to(type_node) ,
common_type_node.get_implicit_conversion_from(type_node) ,
common_type_node.get_explicit_conversion_to(type_node) ,
common_type_node.get_explicit_conversion_from(type_node) ,
[common_type_node.get_instance\(List<type_node>\)](#) , common_type_node.has_static_constructor ,
common_type_node.static_constr , common_type_node.ICommonTypeNode.static_constructor ,
common_type_node.is_generic_type_definition , common_type_node.base_generic_instance ,
common_type_node.runtime_initialization_marker ,
common_type_node.ICommonTypeNode.runtime_initialization_marker ,
common_type_node.generic_type_container , common_type_node.generic_function_container ,
common_type_node.is_generic_parameter , common_type_node.generic_param_index ,
common_type_node.generic_params , common_type_node.IsInterface , common_type_node.IsEnum ,
common_type_node.IsDelegate , common_type_node.ForwardDeclarationOnly ,
common_type_node.ImplementingInterfaces , common_type_node.original_template ,
common_type_node.depended_from_indefinite , common_type_node.IsPartial ,
common_type_node.IsStatic , common_type_node.IsSealed , common_type_node.IsAbstract ,
common_type_node.AbstractReason , common_type_node.rank , common_type_node.type_special_kind ,
common_type_node.internal_type_special_kind , common_type_node.default_property ,
common_type_node.default_property_node , common_type_node.scope ,
common_type_node.const_defs , common_type_node.loc , common_type_node.location ,
common_type_node.Location , common_type_node.methods , common_type_node.fields ,
common_type_node.properties , common_type_node.events , common_type_node.internal_is_value ,
common_type_node.type_access_level , common_type_node.base_type , common_type_node.Scope ,
common_type_node.name , common_type_node.full_name , common_type_node.BaseFullName ,
common_type_node.PrintableName , common_type_node.low_bound ,
common_type_node.upper_bound , common_type_node.comprehensive_namespace ,
common_type_node.node_kind , common_type_node.semantic_node_type ,
common_type_node.ICommonTypeNode.methods , common_type_node.ICommonTypeNode.fields ,
common_type_node.ICommonTypeNode.properties , common_type_node.ICommonTypeNode.events ,
common_type_node.ITypeNode.base_type ,
common_type_node.INamespaceMemberNode.comprehensive_namespace ,
common_type_node.constants , common_type_node.ICommonTypeNode.default_property ,
common_type_node.ICommonTypeNode.lower_value ,
common_type_node.ICommonTypeNode.upper_value , common_type_node.is_value ,
common_type_node.is_class , common_type_node.IsFictive ,

common_type_node.parameters_eliminations , wrapped_type.additional_names ,
wrapped_type.additional_generated_names ,
[wrapped_type.add_additional_name\(string, SymbolInfo, bool\)](#) ,
wrapped_type.clear_generated_names() , [wrapped_type.find_in_additional_names\(string\)](#) ,
type_node.type_intersections , type_node.generated_type_intersections , type_node.internal_interfaces ,
type_node.is_ref_inited , type_node.is_nullable_inited , type_node._is_nullable_type , type_node.ref_type ,
type_node._element_type , type_node.element_type_recived , type_node.ToString() ,
[type_node.add_intersection_node\(type_node, type_intersection_node, bool\)](#) ,
type_node.clear_generated_intersections() , type_node.get_type_intersection(type_node) ,
type_node.add_internal_interface(internal_interface) ,
type_node.get_internal_interface(internal_interface_kind) ,
[type_node.find_first_in_type\(string, Scope, bool\)](#) , type_node.is_standard_type ,
type_node.is_nullable_type , type_node.ITypeNode.common_generic_function_container ,
type_node.IsPointer , type_node.ImplementingInterfacesOrEmpty , type_node.is_generic_type_instance ,
type_node.instance_params , type_node.original_generic , type_node.ref_type ,
type_node.general_node_type , type_node.is_value_type , type_node.ITypeNode.type_special_kind ,
type_node.element_type , type_node.ITypeNode.element_type , definition_node.doc ,
definition_node._attributes , [definition_node.find_by_location\(int, int\)](#) , definition_node.attributes ,
definition_node.Attributes , definition_node.documentation ,
definition_node.IDefinitionNode.Documentation , [object.Equals\(object\)](#) ,
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.FieldSetter\(string, string, object\)](#) ,
[object.FieldGetter\(string, string, ref object\)](#) , [object.GetFieldInfo\(string, string\)](#)

Constructors

wrapped_common_type_node(PCUReader, type_node, string,
type_access_level, common_namespace_node, ClassScope,
location, int)

```
public wrapped_common_type_node(PCUReader pr, type_node base_type, string name,  
type_access_level type_access_level, common_namespace_node comprehensive_namespace,  
ClassScope cs, location loc, int offset)
```

Parameters

pr [PCUReader](#)

base_type type_node

`name` [string](#)

`type_access_level` `type_access_level`

`comprehensive_namespace` `common_namespace_node`

`cs` `ClassScope`

`loc` `location`

`offset` [int](#)

Fields

`offset`

`public int offset`

Field Value

[int](#)

`pr`

`private PCUReader pr`

Field Value

[PCUReader](#)

Methods

`find_in_type(string, Scope, type_node, bool)`

`public override List<SymbolInfo> find_in_type(string name, Scope CurrentScope, type_node orig_generic_or_null = null, bool no_search_in_extension_methods = false)`

Parameters

`name` [string](#)

`CurrentScope` `Scope`

`orig_generic_or_null` `type_node`

`no_search_in_extension_methods` [bool](#)

Returns

[List](#) <`SymbolInfo`>

`find_in_type(string, bool)`

```
public override List<SymbolInfo> find_in_type(string name, bool  
no_search_in_extension_methods = false)
```

Parameters

`name` [string](#)

`no_search_in_extension_methods` [bool](#)

Returns

[List](#) <`SymbolInfo`>

Class wrapped_expression

Namespace: [PascalABCCompiler.PCU](#)

Assembly: Compiler.dll

```
public class wrapped_expression : expression_node, IExpressionNode, IStatementNode,  
ISemanticNode, ILocated
```

Inheritance

[object](#) ← semantic_node ← statement_node ← expression_node ← wrapped_expression

Implements

IExpressionNode, IStatementNode, ISemanticNode, ILocated

Inherited Members

expression_node._tn , expression_node._conversion_tn , expression_node.visit(ISemanticVisitor) ,
expression_node.type , expression_node.conversion_type , expression_node.general_node_type ,
expression_node.is_addressed , expression_node.IExpressionNode.type ,
expression_node.IExpressionNode.conversion_type , statement_node._loc , statement_node.location ,
statement_node.ILocated.Location , [object.ToString\(\)](#) , [object.Equals\(object\)](#) ,
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.FieldSetter\(string, string, object\)](#) ,
[object.FieldGetter\(string, string, ref object\)](#) , [object.GetFieldInfo\(string, string\)](#)

Constructors

wrapped_expression(PCUReader, int)

```
public wrapped_expression(PCUReader pr, int offset)
```

Parameters

pr [PCUReader](#)

offset [int](#)

Fields

offset

```
public int offset
```

Field Value

[int↗](#)

pr

```
private PCUReader pr
```

Field Value

[PCUReader](#)

Properties

semantic_node_type

```
public override semantic_node_type semantic_node_type { get; }
```

Property Value

semantic_node_type

Methods

restore()

```
public expression_node restore()
```

Returns

expression_node

Class wrapped_function_body

Namespace: [PascalABCCompiler.PCU](#)

Assembly: Compiler.dll

```
public class wrapped_function_body : wrapped_statement, IStatementNode,  
ISemanticNode, ILocated
```

Inheritance

[object](#) ← semantic_node ← statement_node ← wrapped_statement ← wrapped_function_body

Implements

IStatementNode, ISemanticNode, ILocated

Inherited Members

wrapped_statement.visit(ISemanticVisitor) , wrapped_statement.semantic_node_type ,
statement_node._loc , statement_node.location , statement_node.ILocated.Location ,
statement_node.general_node_type , [object.ToString\(\)](#) , [object.Equals\(object\)](#) ,
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.FieldSetter\(string, string, object\)](#) ,
[object.FieldGetter\(string, string, ref object\)](#) , [object.GetFieldInfo\(string, string\)](#)

Constructors

wrapped_function_body(PCUReader, int)

```
public wrapped_function_body(PCUReader pr, int offset)
```

Parameters

pr [PCUReader](#)

offset [int](#)

Fields

offset

`public int offset`

Field Value

[int↗](#)

pr

`private PCUReader pr`

Field Value

[PCUReader](#)

Methods

restore()

`public override statement_node restore()`

Returns

`statement_node`

Class wrapped_type_synonym

Namespace: [PascalABCCompiler.PCU](#)

Assembly: Compiler.dll

```
public class wrapped_type_synonym : wrapped_definition_node, IDefinitionNode, ISemanticNode
```

Inheritance

[object](#) ← semantic_node ← definition_node ← wrapped_definition_node ← wrapped_type_synonym

Implements

IDefinitionNode, ISemanticNode

Inherited Members

wrapped_definition_node.offset , wrapped_definition_node.is_synonym ,
wrapped_definition_node.PCUREader , wrapped_definition_node.general_node_type ,
wrapped_definition_node.semantic_node_type , definition_node.doc , definition_node._attributes ,
definition_node.visit(ISemanticVisitor) , [definition_node.find_by_location\(int,int\)](#) ,
definition_node.attributes , definition_node.Attributes , definition_node.documentation ,
definition_node.IDefinitionNode.Documentation , definition_node.location , [object.ToString\(\)](#) ,
[object.Equals\(object\)](#) , [object.Equals\(object,object\)](#) , [object.ReferenceEquals\(object,object\)](#) ,
[object.GetHashCode\(\)](#) , [object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) ,
[object.FieldSetter\(string,string,object\)](#) , [object.FieldGetter\(string,string,ref object\)](#) ,
[object.GetFieldInfo\(string,string\)](#)

Constructors

wrapped_type_synonym(PCUREader, string, int)

```
public wrapped_type_synonym(PCUREader pr, string name, int offset)
```

Parameters

pr [PCUREader](#)

name [string](#)

offset [int](#)

Fields

name

```
public string name
```

Field Value

[string](#) ↗

Namespace PascalABCCompiler.SemanticTreeConverters

Classes

[SemanticTreeConvertersController](#)

Interfaces

[ISemanticTreeConverter](#)

Enums

[ConverterType](#)

[ExecutionOrder](#)

[SemanticTreeConvertersController.State](#)

Delegates

[SemanticTreeConvertersController.ChangeStateDelegate](#)

Enum ConverterType

Namespace: [PascalABCCompiler.SemanticTreeConverters](#)

Assembly: Compiler.dll

```
public enum ConverterType
```

Fields

Analysis = 1

CodeGenerator = 2

Conversion = 0

Enum ExecutionOrder

Namespace: [PascalABCCompiler.SemanticTreeConverters](#)

Assembly: Compiler.dll

```
public enum ExecutionOrder
```

Fields

First = 1

Last = 2

Undefined = 0

Interface ISemanticTreeConverter

Namespace: [PascalABCCompiler.SemanticTreeConverters](#)

Assembly: Compiler.dll

```
public interface ISemanticTreeConverter
```

Properties

ConverterType

```
ConverterType ConverterType { get; }
```

Property Value

[ConverterType](#)

Copyright

```
string Copyright { get; }
```

Property Value

[string](#)

Description

```
string Description { get; }
```

Property Value

[string](#)

ExecutionOrder

```
ExecutionOrder ExecutionOrder { get; }
```

Property Value

[ExecutionOrder](#)

Name

```
string Name { get; }
```

Property Value

[string](#) ↗

Version

```
string Version { get; }
```

Property Value

[string](#) ↗

Methods

Convert(ICompiler, IProgramNode)

```
IProgramNode Convert(ICompiler Compiler, IProgramNode ProgramNode)
```

Parameters

Compiler [ICompiler](#)

ProgramNode IProgramNode

Returns

IProgramNode

Class SemanticTreeConvertersController

Namespace: [PascalABCCompiler.SemanticTreeConverters](#)

Assembly: Compiler.dll

```
public class SemanticTreeConvertersController
```

Inheritance

[object](#) ← SemanticTreeConvertersController

Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.FieldSetter\(string, string, object\)](#) ,
[object.FieldGetter\(string, string, ref object\)](#) , [object.GetFieldInfo\(string, string\)](#)

Constructors

SemanticTreeConvertersController(ICompiler)

```
public SemanticTreeConvertersController(ICompiler Compiler)
```

Parameters

Compiler [ICompiler](#)

Fields

Compiler

```
private ICompiler Compiler
```

Field Value

[ICompiler](#)

semanticTreeConverters

```
private List<ISemanticTreeConverter> semanticTreeConverters
```

Field Value

[List](#) <[ISemanticTreeConverter](#)>

Properties

SemanticTreeConverters

```
public List<ISemanticTreeConverter> SemanticTreeConverters { get; }
```

Property Value

[List](#) <[ISemanticTreeConverter](#)>

Methods

AddConverters()

```
public void AddConverters()
```

AddConverters(string)

```
private void AddConverters(string DirectoryName)
```

Parameters

DirectoryName [string](#)

Convert(IProgramNode)

```
public IProgramNode Convert(IProgramNode ProgramNode)
```

Parameters

ProgramNode IProgramNode

Returns

IProgramNode

Events

ChangeState

```
public event SemanticTreeConvertersController.ChangeStateDelegate ChangeState
```

Event Type

[SemanticTreeConvertersController.ChangeStateDelegate](#)

Delegate SemanticTreeConvertersController.ChangeStateDelegate

Namespace: [PascalABCCompiler.SemanticTreeConverters](#)

Assembly: Compiler.dll

```
public delegate void  
SemanticTreeConvertersController.ChangeStateDelegate(SemanticTreeConvertersController.State  
State, ISemanticTreeConverter SemanticTreeConverter)
```

Parameters

State [SemanticTreeConvertersController.State](#)

SemanticTreeConverter [ISemanticTreeConverter](#)

Constructors

ChangeStateDelegate(object, IntPtr)

```
public ChangeStateDelegate(object @object, IntPtr method)
```

Parameters

object [object](#)

method [IntPtr](#)

Methods

BeginInvoke(State, ISemanticTreeConverter, AsyncCallback, object)

```
public virtual IAsyncResult BeginInvoke(SemanticTreeConvertersController.State State,  
ISemanticTreeConverter SemanticTreeConverter, AsyncCallback callback, object @object)
```

Parameters

State [SemanticTreeConvertersController.State](#)

SemanticTreeConverter [ISemanticTreeConverter](#)

callback [AsyncCallback](#)

object [object](#)

Returns

[IAsyncResult](#)

EndInvoke(IAsyncResult)

public virtual void EndInvoke(IAsyncResult result)

Parameters

result [IAsyncResult](#)

Invoke(State, ISemanticTreeConverter)

public virtual void Invoke(SemanticTreeConvertersController.State State,
ISemanticTreeConverter SemanticTreeConverter)

Parameters

State [SemanticTreeConvertersController.State](#)

SemanticTreeConverter [ISemanticTreeConverter](#)

Enum SemanticTreeConvertersController.State

Namespace: [PascalABCCompiler.SemanticTreeConverters](#)

Assembly: Compiler.dll

```
public enum SemanticTreeConvertersController.State
```

Fields

ConnectConverter = 0

Convert = 1

Namespace PascalABCCompiler.TreeConverter

Classes

[SyntaxTreeToSemanticTreeConverter](#)

Class SyntaxTreeToSemanticTreeConverter

Namespace: [PascalABCCompiler.TreeConverter](#)

Assembly: Compiler.dll

```
public class SyntaxTreeToSemanticTreeConverter
```

Inheritance

[object](#) ← SyntaxTreeToSemanticTreeConverter

Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.FieldSetter\(string, string, object\)](#) ,
[object.FieldGetter\(string, string, ref object\)](#) , [object.GetFieldInfo\(string, string\)](#)

Constructors

SyntaxTreeToSemanticTreeConverter()

```
public SyntaxTreeToSemanticTreeConverter()
```

Fields

syntaxTreeVisitor

```
private syntax_tree_visitor syntaxTreeVisitor
```

Field Value

syntax_tree_visitor

Properties

SymbolTable

```
public TreeConverterSymbolTable SymbolTable { get; }
```

Property Value

TreeConverterSymbolTable

Methods

CompileImplementation(ILanguage,
InitializationDataForCompilingImplementation,
List<var_definition_node>)

```
public void CompileImplementation(ILanguage language,  
InitializationDataForCompilingImplementation initializationData,  
List<var_definition_node> CompiledVariables)
```

Parameters

language ILanguage

initializationData InitializationDataForCompilingImplementation

CompiledVariables [List](#)<var_definition_node>

CompileInterface(ILanguage,
InitializationDataForCompilingInterface,
List<var_definition_node>)

```
public common_unit_node CompileInterface(ILanguage language,  
InitializationDataForCompilingInterface initializationData, List<var_definition_node>  
CompiledVariables)
```

Parameters

`language` `ILanguage`

`initializationData` `InitializationDataForCompilingInterface`

`CompiledVariables` `List<var_definition_node>`

Returns

`common_unit_node`

`Reset()`

`public void Reset()`