Namespace Core

Classes

Canvas

Canvas implementation for drawing on

CommandFactory

Factory for adding new commands using the command design template

ExpressionEvaluator

Evaluator for expressions

Parser

Parser class for handling BOOSE scripts

StoredProgram

Token

Token used to represent interpretted

Variable

Variable class used to represent a variable when evaluating and executing a stored program

Class Canvas

Namespace: Core

Assembly: Core.dll

Canvas implementation for drawing on

```
public class Canvas : ICanvas
```

Inheritance

object ← Canvas

Implements

ICanvas

Inherited Members

object.Equals(object) , object.Equals(object, object) , object.GetHashCode() , object.GetType() , object.MemberwiseClone() , object.ReferenceEquals(object, object) , object.ToString()

Constructors

Canvas(Panel?)

Canvas constructor for frontend

```
public Canvas(Panel? panel)
```

Parameters

`panel` Panel

Panel used for drawing to, optional for unit testing

Properties

BackgroundColour

Current background colour (for when cleared)

```
public Color BackgroundColour { get; set; }
```

Property Value

Color

Bounds

Canvas boundaries

```
public Rectangle Bounds { get; }
```

Property Value

Rectangle

GraphicsBuffer

The buffered graphics instance for the original graphics instance

```
public BufferedGraphics GraphicsBuffer { get; }
```

Property Value

BufferedGraphics

GraphicsBufferContext

The context of the buffered graphics

```
public BufferedGraphicsContext GraphicsBufferContext { get; }
```

Property Value

BufferedGraphicsContext

IsPainting

Free-drawing status

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

Property Value

bool

IsPenDown

Pen drawing status

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

Property Value

bool

Pen

Pen used for drawing

```
public Pen Pen { get; }
```

Property Value

Pen

PenPosition

Current pen position on canvas

```
public Point PenPosition { get; set; }
```

Property Value

Point

Methods

Clear()

Polymorphic of clear which defaults to background colour

```
public void Clear()
```

Clear(Color)

Clears the graphics buffer and re-renders

```
public void Clear(Color colour)
```

Parameters

`colour` Color

FreeDraw(int, int)

Function for passing to panel for free-drawing

```
public void FreeDraw(int xPos, int yPos)
```

Parameters

`xPos` int

x position to draw at

`yPos` int

y position to draw at

Reset()

Clears canvas and resets pen position

```
public void Reset()
```

Class CommandFactory

Namespace: Core

Assembly: Core.dll

Factory for adding new commands using the command design template

```
public class CommandFactory : ICommandFactory
```

Inheritance

object ← CommandFactory

Implements

ICommandFactory

Inherited Members

object.Equals(object) , object.Equals(object, object) , object.GetHashCode() , object.GetType() , object.MemberwiseClone() , object.ReferenceEquals(object, object) , object.ToString()

Constructors

CommandFactory()

Constructor adds default commands to the factory

```
public CommandFactory()
```

Methods

AddCommand(ICommand)

Add new commands to the factory

```
public void AddCommand(ICommand command)
```

Parameters

`command` ICommand

Object of new command

GetCommand(string)

Get command from within the factory

```
public ICommand? GetCommand(string name)
```

Parameters

`name` string

Command name

Returns

ICommand

Command object

GetCommandsRegex()

Utility function for getting a list of commands as a regex string delimited by the or operator ("|")

```
public string GetCommandsRegex()
```

Returns

string

Regex foratted string

Class ExpressionEvaluator

Namespace: Core

Assembly: Core.dll

Evaluator for expressions

```
public class ExpressionEvaluator : IExpressionEvaluator
```

Inheritance

object ← ExpressionEvaluator

Implements

IExpressionEvaluator

Inherited Members

object.Equals(object) , object.Equals(object, object) , object.GetHashCode() , object.GetType() , object.MemberwiseClone() , object.ReferenceEquals(object, object) , object.ToString()

Fields

_variables

Local instance of variables array

```
public Dictionary<string, Variable> _variables
```

Field Value

Dictionary<string, Variable>

Methods

EvaluateBinaryComparison(string, double, double)

Function for evaluation of binary comparisons for iteration and conditions

```
public bool EvaluateBinaryComparison(string binaryOperator, double firstValue,
double secondValue)
```

Parameters

`binaryOperator` string

The operator used for the comparison

`firstValue` double

First value/evaluated expression being compared to

`secondValue` double

Second value/evaluated expression being compared to

Returns

bool

Boolean of whether the comparison is true or not

Exceptions

StoredProgramException

EvaluateExpression(List<Token>, ref int)

An implementation of expression evaluation without memory of previously assigned variables

```
public double EvaluateExpression(List<Token> tokens, ref int index)
```

Parameters

`tokens` List<Token>

Expession tokens to be evaluated

`index` int

Current index within the list of tokens, as a reference so they aren't double parsed elsewhere

Returns

double

Float result of the evaluation

EvaluateExpression(List<Token>, ref int, Dictionary<string, Variable>)

A polyorphic implementation of expression evaluation hat allows for a variable list to be passed this enables variable resolution for previously assigned variables

```
public double EvaluateExpression(List<Token> tokens, ref int index, Dictionary<string,
Variable> variables)
```

Parameters

`tokens` List<Token>

Expession tokens to be evaluated

`index` int

Current index within the list of tokens, as a reference so they aren't double parsed elsewhere

`variables` Dictionary<string, Variable>

A list of previously assigned variables

Returns

double

Float result of the evaluation

evaluateStringExpression(List<Token>, ref int, Dictionary<string, Variable>)

```
public string evaluateStringExpression(List<Token> tokens, ref int index, Dictionary<string,
Variable> variables)
```

Parameters

`tokens` List<Token>

`index` int

`variables` Dictionary<string, Variable>

Returns

string

Class Parser

Namespace: Core

Assembly: Core.dll

Parser class for handling BOOSE scripts

```
public class Parser : IParser
```

Inheritance

object ← Parser

Implements

IParser

Inherited Members

object.Equals(object) , object.Equals(object, object) , object.GetHashCode() , object.GetType() , object.MemberwiseClone() , object.ReferenceEquals(object, object) , object.ToString()

Constructors

Parser(ICommandFactory, StoredProgram)

Parser class for handling BOOSE scripts

```
public Parser(ICommandFactory commandFactory, StoredProgram storedProgram)
```

Parameters

`commandFactory` ICommandFactory

`storedProgram` StoredProgram

Methods

parseProgram(string)

Takes a script, tokenises it and adds it to the stored program

```
public void parseProgram(string program)
```

Parameters

`program` string

Class StoredProgram

Namespace: Core

Assembly: Core.dll

```
public class StoredProgram : IStoredProgram
```

Inheritance

object ← StoredProgram

Implements

IStoredProgram

Inherited Members

object.Equals(object) , object.Equals(object, object) , object.GetHashCode() , object.GetType() , object.MemberwiseClone() , object.ReferenceEquals(object, object) , object.ToString()

Constructors

StoredProgram(ICanvas)

```
public StoredProgram(ICanvas _canvas)
```

Parameters

`_canvas` ICanvas

Fields

variables

```
public Dictionary<string, Variable> variables
```

Field Value

Dictionary<string, Variable>

Properties

LineIndex

Index for storing which line is being executed at a given time

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

Property Value

int

canvas

Canvas used when executing commands to draw

```
public ICanvas canvas { get; set; }
```

Property Value

ICanvas

tokens

The tokens representing the program

```
public List<List<Token>> tokens { get; set; }
```

Property Value

List<List<Token>>

Methods

Execute()

Function to evaluate and execute the stored program

```
public CommandResult Execute()
```

Returns

CommandResult

ResetProgram()

Function for resetting indexes and the last-ran command hash values

```
public void ResetProgram()
```

addLine(List<Token>)

Function for adding a line to the stored program

```
public void addLine(List<Token> Line)
```

Parameters

`Line` List<Token>

Class Token

Namespace: Core

Assembly: Core.dll

Token used to represent interpretted

```
public class Token
```

Inheritance

object ← Token

Inherited Members

object.Equals(object) , object.Equals(object, object) , object.GetHashCode() , object.GetType() , object.MemberwiseClone() , object.ReferenceEquals(object, object)

Constructors

Token(TokenType, string)

Constructor for all token objects

```
public Token(TokenType type, string value)
```

Parameters

`type` TokenType

The type of token being created

`value` string

A string formatted value for the token to contain

Token(string, ICommand?)

```
public Token(string value, ICommand? command)
```

Parameters

`value` string

`command` ICommand

Properties

Command

The value said token contains

```
public ICommand? Command { get; }
```

Property Value

ICommand

Type

The type of token being created

```
public TokenType Type { get; }
```

Property Value

TokenType

Value

The value said token contains

```
public string Value { get; }
```

Property Value

string

Methods

ToString()

A function for converting the object values to string form

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

Returns

string

A formatted string containing the contents of the token

Class Variable

Namespace: Core

Assembly: Core.dll

Variable class used to represent a variable when evaluating and executing a stored program

```
public class Variable
```

Inheritance

object ← Variable

Inherited Members

object.Equals(object) , object.Equals(object, object) , object.GetHashCode() , object.GetType() , object.MemberwiseClone() , object.ReferenceEquals(object, object) , object.ToString()

Constructors

Variable(string, VariableType)

Take a name and type to initiate a variable without assignment

```
public Variable(string name, VariableType type)
```

Parameters

`name` string

Variable name

`type` VariableType

Variable type

Variable(string, double)

Real variable constructor

```
public Variable(string name, double value)
```

Parameters

`name` string

Variable name

`value` double

Real Variable Value

Variable(string, int)

Integer variable constructor

```
public Variable(string name, int value)
```

Parameters

`name` string

Variable name

`value` int

Integer Variable Value

Variable(string, string)

String variable constructor

```
public Variable(string name, string value)
```

Parameters

`name` string

Variable name

`value` string

String Variable Value

Properties

IntListValue

Optional integer list for integer array variable

```
public int[]? IntListValue { get; set; }
```

Property Value

int[]

IntValue

Optional integer value for integer type variable

```
public int? IntValue { get; set; }
```

Property Value

int?

Name

The name of the variable

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

Property Value

string

RealListValue

Optional real list for real array variable

```
public double[]? RealListValue { get; set; }
```

Property Value

double[]

RealValue

Optional real value for real type variable

```
public double? RealValue { get; set; }
```

Property Value

double?

StrListValue

Optional string list for string array variable

```
public string[]? StrListValue { get; set; }
```

Property Value

string[]

StrValue

Optional string value for string type variable

```
public string? StrValue { get; set; }
```

Property Value

string

Type

Type of variable

```
public VariableType Type { get; set; }
```

Property Value

VariableType

Namespace Core.Commands

Classes

About
: About command class

Circle
: Circle command class

Clear
: Clear command class

DrawTo
: DrawTo command class

MoveTo
: MoveTo command class

PenColour
: PenColour command class

Rectangle
: Rectangle command class

Triangle
: Triangle command class

Write
: Write command class

Class About

Namespace: Core.Commands

Assembly: Core.dll

About command class

```
public class About : ICommand
```

Inheritance

object ← About

Implements

ICommand

Inherited Members

object.Equals(object) , object.Equals(object, object) , object.GetHashCode() , object.GetType() , object.MemberwiseClone() , object.ReferenceEquals(object, object) , object.ToString()

Constructors

About()

Constructor for the circle command Contains information about the command itself

```
public About()
```

Properties

Description

Description of what the commmand does

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

Property Value

string

Name

Name of command to be set

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

Property Value

string

Usage

Command usage to be set

```
public string Usage { get; }
```

Property Value

string

Methods

Execute(ICanvas, List<object>)

The method which will be invoked when the command is run

```
public CommandResult Execute(ICanvas canvas, List<object> parameters)
```

Parameters

`canvas` ICanvas

`parameters` List<object>

A list of objects for parsing in the function which may be used when running the command

Returns

CommandResult

Result of whether the command was successful or not

Class Circle

Namespace: Core.Commands

Assembly: Core.dll

Circle command class

```
public class Circle : ICommand
```

Inheritance

object ← Circle

Implements

ICommand

Inherited Members

object.Equals(object) , object.Equals(object, object) , object.GetHashCode() , object.GetType() , object.MemberwiseClone() , object.ReferenceEquals(object, object) , object.ToString()

Constructors

Circle()

Constructor for the circle command Contains information about the command itself

```
public Circle()
```

Properties

Description

Description of what the commmand does

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

Property Value

string

Name

Name of command to be set

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

Property Value

string

Usage

Command usage to be set

```
public string Usage { get; }
```

Property Value

string

Methods

Execute(ICanvas, List<object>)

The method which will be invoked when the command is run

```
public CommandResult Execute(ICanvas canvas, List<object> parameters)
```

Parameters

`canvas` ICanvas

`parameters` List<object>

A list of objects for parsing in the function which may be used when running the command

Returns

CommandResult

Result of whether the command was successful or not

Class Clear

Namespace: Core.Commands

Assembly: Core.dll

Clear command class

```
public class Clear : ICommand
```

Inheritance

object ← Clear

Implements

ICommand

Inherited Members

object.Equals(object) , object.Equals(object, object) , object.GetHashCode() , object.GetType() , object.MemberwiseClone() , object.ReferenceEquals(object, object) , object.ToString()

Constructors

Clear()

Constructor for the clear command Contains information about the command itself

```
public Clear()
```

Properties

Description

Description of what the commmand does

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

Property Value

string

Name

Name of command to be set

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

Property Value

string

Usage

Command usage to be set

```
public string Usage { get; }
```

Property Value

string

Methods

Execute(ICanvas, List<object>)

The method which will be invoked when the command is run

```
public CommandResult Execute(ICanvas canvas, List<object> parameters)
```

Parameters

`canvas` ICanvas

`parameters` List<object>

A list of objects for parsing in the function which may be used when running the command

Returns

CommandResult

Result of whether the command was successful or not

Class DrawTo

Namespace: Core.Commands

Assembly: Core.dll

DrawTo command class

```
public class DrawTo : ICommand
```

Inheritance

object ← DrawTo

Implements

ICommand

Inherited Members

object.Equals(object) , object.Equals(object, object) , object.GetHashCode() , object.GetType() , object.MemberwiseClone() , object.ReferenceEquals(object, object) , object.ToString()

Constructors

DrawTo()

Constructor for the drawto command Contains information about the command itself

```
public DrawTo()
```

Properties

Description

Description of what the commmand does

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

Property Value

string

Name

Name of command to be set

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

Property Value

string

Usage

Command usage to be set

```
public string Usage { get; }
```

Property Value

string

Methods

Execute(ICanvas, List<object>)

The method which will be invoked when the command is run

```
public CommandResult Execute(ICanvas canvas, List<object> parameters)
```

Parameters

`canvas` ICanvas

`parameters` List<object>

A list of objects for parsing in the function which may be used when running the command

Returns

CommandResult

Result of whether the command was successful or not

Class MoveTo

Namespace: Core.Commands

Assembly: Core.dll

MoveTo command class

```
public class MoveTo : ICommand
```

Inheritance

object ← MoveTo

Implements

ICommand

Inherited Members

object.Equals(object) , object.Equals(object, object) , object.GetHashCode() , object.GetType() , object.MemberwiseClone() , object.ReferenceEquals(object, object) , object.ToString()

Constructors

MoveTo()

Constructor for the moveto command Contains information about the command itself

```
public MoveTo()
```

Properties

Description

Description of what the commmand does

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

Property Value

string

Name

Name of command to be set

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

Property Value

string

Usage

Command usage to be set

```
public string Usage { get; }
```

Property Value

string

Methods

Execute(ICanvas, List<object>)

The method which will be invoked when the command is run

```
public CommandResult Execute(ICanvas canvas, List<object> parameters)
```

Parameters

`canvas` ICanvas

`parameters` List<object>

A list of objects for parsing in the function which may be used when running the command

Returns

CommandResult

Result of whether the command was successful or not

Class PenColour

Namespace: Core.Commands

Assembly: Core.dll

PenColour command class

```
public class PenColour : ICommand
```

Inheritance

object ← PenColour

Implements

ICommand

Inherited Members

object.Equals(object) , object.Equals(object, object) , object.GetHashCode() , object.GetType() , object.MemberwiseClone() , object.ReferenceEquals(object, object) , object.ToString()

Constructors

PenColour()

Constructor for the pencolour command Contains information about the command itself

```
public PenColour()
```

Properties

Description

Description of what the commmand does

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

Property Value

string

Name

Name of command to be set

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

Property Value

string

Usage

Command usage to be set

```
public string Usage { get; }
```

Property Value

string

Methods

Execute(ICanvas, List<object>)

The method which will be invoked when the command is run

```
public CommandResult Execute(ICanvas canvas, List<object> parameters)
```

Parameters

`canvas` ICanvas

`parameters` List<object>

A list of objects for parsing in the function which may be used when running the command

Returns

CommandResult

Result of whether the command was successful or not

Class Rectangle

Namespace: Core.Commands

Assembly: Core.dll

Rectangle command class

```
public class Rectangle : ICommand
```

Inheritance

object ← Rectangle

Implements

ICommand

Inherited Members

object.Equals(object) , object.Equals(object, object) , object.GetHashCode() , object.GetType() , object.MemberwiseClone() , object.ReferenceEquals(object, object) , object.ToString()

Constructors

Rectangle()

Constructor for the rectangle command Contains information about the command itself

```
public Rectangle()
```

Properties

Description

Description of what the commmand does

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

Property Value

string

Name

Name of command to be set

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

Property Value

string

Usage

Command usage to be set

```
public string Usage { get; }
```

Property Value

string

Methods

Execute(ICanvas, List<object>)

The method which will be invoked when the command is run

```
public CommandResult Execute(ICanvas canvas, List<object> parameters)
```

Parameters

`canvas` ICanvas

`parameters` List<object>

A list of objects for parsing in the function which may be used when running the command

Returns

CommandResult

Result of whether the command was successful or not

Class Triangle

Namespace: Core.Commands

Assembly: Core.dll

Triangle command class

```
public class Triangle : ICommand
```

Inheritance

object ← Triangle

Implements

ICommand

Inherited Members

object.Equals(object) , object.Equals(object, object) , object.GetHashCode() , object.GetType() , object.MemberwiseClone() , object.ReferenceEquals(object, object) , object.ToString()

Constructors

Triangle()

Constructor for the triangle command Contains information about the command itself

```
public Triangle()
```

Properties

Description

Description of what the commmand does

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

Property Value

string

Name

Name of command to be set

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

Property Value

string

Usage

Command usage to be set

```
public string Usage { get; }
```

Property Value

string

Methods

Execute(ICanvas, List<object>)

The method which will be invoked when the command is run

```
public CommandResult Execute(ICanvas canvas, List<object> parameters)
```

Parameters

`canvas` ICanvas

`parameters` List<object>

A list of objects for parsing in the function which may be used when running the command

Returns

CommandResult

Result of whether the command was successful or not

Class Write

Namespace: Core.Commands

Assembly: Core.dll

Write command class

```
public class Write : ICommand
```

Inheritance

object ← Write

Implements

ICommand

Inherited Members

object.Equals(object) , object.Equals(object, object) , object.GetHashCode() , object.GetType() , object.MemberwiseClone() , object.ReferenceEquals(object, object) , object.ToString()

Constructors

Write()

Constructor for the Write command Contains information about the command itself

```
public Write()
```

Properties

Description

Description of what the commmand does

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

Property Value

string

Name

Name of command to be set

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

Property Value

string

Usage

Command usage to be set

```
public string Usage { get; }
```

Property Value

string

Methods

Execute(ICanvas, List<object>)

The method which will be invoked when the command is run

```
public CommandResult Execute(ICanvas canvas, List<object> parameters)
```

Parameters

`canvas` ICanvas

`parameters` List<object>

A list of objects for parsing in the function which may be used when running the command

Returns

CommandResult

Result of whether the command was successful or not

Namespace Core.Enums

Enums

CommandResult

Enumerable for results of commands

TokenType

Token types enumerable

VariableType

Variable types enumerable

Enum CommandResult

Namespace: Core.Enums

Assembly: Core.dll

Enumerable for results of commands

```
public enum CommandResult
```

Fields

`Error = -1`

`Finished = 2`

`Success = 1`

Enum TokenType

Namespace: Core.Enums

Assembly: Core.dll

Token types enumerable

```
public enum TokenType
```

Fields

`EOF = -2`

`String = 9`

`array_manipulation = 10`

`command = 0`

`conditional = 1`

`integer = 5`

`invalid = -1`

`iteration = 2`

`operation = 7`

`punctuation = 8`

`real = 6`

`variableName = 4`

`variableType = 3`

Enum VariableType

Namespace: Core.Enums

Assembly: Core.dll

Variable types enumerable

```
public enum VariableType
```

Fields

`Integer = 0`

`ListInt = 3`

`ListReal = 4`

`ListStr = 5`

`Real = 1`

`String = 2`

Namespace Core.Exceptions

Classes

BooseException

Generic Exception Class

CommandException

Exception within command execution

ParserException

Error within the parsing and tokenising of a script

StoredProgramException

Errors occuring within the stored program

VariableException

Exceptions for variable assignment

Class BooseException

Namespace: Core.Exceptions

Assembly: Core.dll

Generic Exception Class

```
public class BooseException : Exception, ISerializable
```

Inheritance

object ← Exception ← BooseException

Implements

ISerializable

Derived

ParserException, StoredProgramException

Inherited Members

Exception.GetBaseException() , Exception.GetType() , Exception.ToString() , Exception.Data , Exception.HelpLink , Exception.HResult , Exception.InnerException , Exception.Message , Exception.Source , Exception.StackTrace , Exception.TargetSite , Exception.SerializeObjectState , object.Equals(object) , object.Equals(object, object) , object.GetHashCode() , object.MemberwiseClone() , object.ReferenceEquals(object, object)

Constructors

BooseException(string)

```
public BooseException(string message)
```

Parameters

`message` string

Class CommandException

Namespace: Core.Exceptions

Assembly: Core.dll

Exception within command execution

```
public class CommandException : StoredProgramException, ISerializable
```

Inheritance

object ← Exception ← BooseException ← StoredProgramException ← CommandException

Implements

ISerializable

Inherited Members

Exception.GetBaseException() , Exception.GetType() , Exception.ToString() , Exception.Data , Exception.HelpLink , Exception.HResult , Exception.InnerException , Exception.Message , Exception.Source , Exception.StackTrace , Exception.TargetSite , Exception.SerializeObjectState , object.Equals(object) , object.Equals(object, object) , object.GetHashCode() , object.MemberwiseClone() , object.ReferenceEquals(object, object)

Constructors

CommandException(string)

```
public CommandException(string message)
```

Parameters

`message` string

Class ParserException

Namespace: Core.Exceptions

Assembly: Core.dll

Error within the parsing and tokenising of a script

```
public class ParserException : BooseException, ISerializable
```

Inheritance

object ← Exception ← BooseException ← ParserException

Implements

ISerializable

Inherited Members

Exception.GetBaseException() , Exception.GetType() , Exception.ToString() , Exception.Data , Exception.HelpLink , Exception.HResult , Exception.InnerException , Exception.Message , Exception.Source , Exception.StackTrace , Exception.TargetSite , Exception.SerializeObjectState , object.Equals(object) , object.Equals(object, object) , object.GetHashCode() , object.MemberwiseClone() , object.ReferenceEquals(object, object)

Constructors

ParserException(string)

```
public ParserException(string message)
```

Parameters

`message` string

Class StoredProgramException

Namespace: Core.Exceptions

Assembly: Core.dll

Errors occuring within the stored program

```
public class StoredProgramException : BooseException, ISerializable
```

Inheritance

object ← Exception ← BooseException ← StoredProgramException

Implements

ISerializable

Derived

CommandException, VariableException

Inherited Members

Exception.GetBaseException() , Exception.GetType() , Exception.ToString() , Exception.Data , Exception.HelpLink , Exception.HResult , Exception.InnerException , Exception.Message , Exception.Source , Exception.StackTrace , Exception.TargetSite , Exception.SerializeObjectState , object.Equals(object) , object.Equals(object, object) , object.GetHashCode() , object.MemberwiseClone() , object.ReferenceEquals(object, object)

Constructors

StoredProgramException(string)

```
public StoredProgramException(string message)
```

Parameters

`message` string

Class VariableException

Namespace: Core.Exceptions

Assembly: Core.dll

Exceptions for variable assignment

```
public class VariableException : StoredProgramException, ISerializable
```

Inheritance

object ← Exception ← BooseException ← StoredProgramException ← VariableException

Implements

ISerializable

Inherited Members

Exception.GetBaseException() , Exception.GetType() , Exception.ToString() , Exception.Data , Exception.HelpLink , Exception.HResult , Exception.InnerException , Exception.Message , Exception.Source , Exception.StackTrace , Exception.TargetSite , Exception.SerializeObjectState , object.Equals(object) , object.Equals(object, object) , object.GetHashCode() , object.MemberwiseClone() , object.ReferenceEquals(object, object)

Constructors

VariableException(string)

```
public VariableException(string message)
```

Parameters

`message` string

Namespace Core.Interfaces

Interfaces

ICanvas

Canvas interface for drawing

ICommand

Base command class used to implement commands

ICommandFactory

Base commandfactory class used for adding custom commands

IExpressionEvaluator

Base expression evaluation class for resolving expressions from tokens

IParser

Parser interface for parsing and tokenising scripts inputted by a user

IStoredProgram

Base storedprogram class used to store and execute tokens for a given script

Interface ICanvas

Namespace: Core.Interfaces

Assembly: Core.dll

Canvas interface for drawing

```
public interface ICanvas
```

Properties

BackgroundColour

Current background colour (for when cleared)

```
Color BackgroundColour { get; set; }
```

Property Value

Color

Bounds

Canvas boundaries

```
Rectangle Bounds { get; }
```

Property Value

Rectangle

GraphicsBuffer

The buffered graphics instance for the original graphics instance

```
BufferedGraphics GraphicsBuffer { get; }
```

Property Value

BufferedGraphics

GraphicsBufferContext

The context of the buffered graphics

```
BufferedGraphicsContext GraphicsBufferContext { get; }
```

Property Value

BufferedGraphicsContext

IsPainting

Free-drawing status

```
bool IsPainting { get; set; }
```

Property Value

bool

IsPenDown

Pen drawing status

```
bool IsPenDown { get; set; }
```

Property Value

bool

Pen

Pen used for drawing

```
Pen Pen { get; }
```

Property Value

Pen

PenPosition

Current pen position on canvas

```
Point PenPosition { get; set; }
```

Property Value

Point

Methods

Clear()

Polymorphic of clear which defaults to background colour

```
void Clear()
```

Clear(Color)

Clears the graphics buffer and re-renders

```
void Clear(Color colour)
```

Parameters

`colour` Color

FreeDraw(int, int)

Function for passing to panel for free-drawing

```
void FreeDraw(int xPos, int yPos)
```

Parameters

`xPos` int

x position to draw at

`yPos` int

y position to draw at

Reset()

Clears canvas and resets pen position

```
void Reset()
```

Interface ICommand

Namespace: Core.Interfaces

Assembly: Core.dll

Base command class used to implement commands

```
public interface ICommand
```

Properties

Description

Description of what the commmand does as part of the help message

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

Property Value

string

Name

Name used for adding command to list

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

Property Value

string

Usage

Usage used to help generate help message

```
string Usage { get; }
```

Property Value

string

Methods

Execute(ICanvas, List<object>)

Function containing the code executed when the command is run

```
CommandResult Execute(ICanvas canvas, List<object> parameters)
```

Parameters

`canvas` ICanvas

`parameters` List<object>

List of parameters as un-parsed objects

Returns

CommandResult

Interface ICommandFactory

Namespace: Core.Interfaces

Assembly: Core.dll

Base commandfactory class used for adding custom commands

```
public interface ICommandFactory
```

Methods

AddCommand(ICommand)

Function for adding a commands to the factory

```
void AddCommand(ICommand command)
```

Parameters

`command` ICommand

Command to add

GetCommand(string)

Function for returning a command stored in the factory

```
ICommand? GetCommand(string name)
```

Parameters

`name` string

Command name to fetch

Returns

ICommand

GetCommandsRegex()

Returns a regex string used for listing commands

```
string GetCommandsRegex()
```

Returns

string

Interface IExpressionEvaluator

Namespace: Core.Interfaces

Assembly: Core.dll

Base expression evaluation class for resolving expressions from tokens

```
public interface IExpressionEvaluator
```

Methods

EvaluateExpression(List<Token>, ref int)

Function for evaluating expressions to doubles

```
double EvaluateExpression(List<Token> tokens, ref int index)
```

Parameters

`tokens` List<Token>

List of tokens

`index` int

Current index in list

Returns

double

Interface IParser

Namespace: Core.Interfaces

Assembly: Core.dll

Parser interface for parsing and tokenising scripts inputted by a user

```
public interface IParser
```

Methods

parseProgram(string)

Formats and interprets BOOSE script

```
void parseProgram(string program)
```

Parameters

`program` string

The raw program as a string

Interface IStoredProgram

Namespace: Core.Interfaces

Assembly: Core.dll

Base storedprogram class used to store and execute tokens for a given script

```
public interface IStoredProgram
```

Properties

LineIndex

Index for storing which line is being executed at a given time

```
int LineIndex { get; set; }
```

Property Value

int

canvas

Canvas used when executing commands to draw

```
ICanvas canvas { get; set; }
```

Property Value

ICanvas

tokens

The tokens representing the program

```
List<List<Token>> tokens { get; set; }
```

Property Value

List<List<Token>>

Methods

Execute()

Function to evaluate and execute the stored program

```
CommandResult Execute()
```

Returns

CommandResult

ResetProgram()

Function for resetting indexes and the last-ran command hash values

```
void ResetProgram()
```

addLine(List<Token>)

Function for adding a line to the stored program

```
void addLine(List<Token> tokens)
```

Parameters

`tokens` List<Token>

A list of tokens representing the line

Namespace CoreUnitTest

Classes

CoreUnitTest

Unit tests for the BOOSE core

TestCommand

Command class for testing commandFactory

Class CoreUnitTest

Namespace: CoreUnitTest

Assembly: CoreUnitTest.dll

Unit tests for the BOOSE core

```
public class CoreUnitTest
```

Inheritance

object ← CoreUnitTest

Inherited Members

object.Equals(object) , object.Equals(object, object) , object.GetHashCode() , object.GetType() , object.MemberwiseClone() , object.ReferenceEquals(object, object) , object.ToString()

Constructors

CoreUnitTest()

Setup constructor for creating canvas, storedProgram, commandFactory and parser

```
public CoreUnitTest()
```

Methods

ArrayVariableTest()

```
[Fact]
public void ArrayVariableTest()
```

CommandFactoryTest()

Test command factory implementation

```
[Fact]
public void CommandFactoryTest()
```

ConditionalTest()

Test conditional blocks

```
[Fact]
public void ConditionalTest()
```

DrawToTest()

Test DrawTo command

```
[Fact]
public void DrawToTest()
```

ForIterationTest()

Test for blocks

```
[Fact]
public void ForIterationTest()
```

IntegerVariableTest()

Test integer variable assignment

```
[Fact]
public void IntegerVariableTest()
```

MethodTest()

```
[Fact]
public void MethodTest()
```

MoveToTest()

Test MoveTo command

```
[Fact]
public void MoveToTest()
```

MultiLineTest()

Test multi-line script functionality

```
[Fact]
public void MultiLineTest()
```

RealVariableTest()

Test real variable assignment

```
[Fact]
public void RealVariableTest()
```

WhileIterationTest()

Test while blocks

```
[Fact]
public void WhileIterationTest()
```

Class TestCommand

Namespace: CoreUnitTest

Assembly: CoreUnitTest.dll

Command class for testing commandFactory

```
public class TestCommand : ICommand
```

Inheritance

object ← TestCommand

Implements

ICommand

Inherited Members

object.Equals(object) , object.Equals(object, object) , object.GetHashCode() , object.GetType() , object.MemberwiseClone() , object.ReferenceEquals(object, object) , object.ToString()

Constructors

TestCommand()

Constructor for the moveto command Contains information about the command itself

```
public TestCommand()
```

Properties

Description

Description of what the commmand does

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

Property Value

string

Name

Name of command to be set

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

Property Value

string

Usage

Command usage to be set

```
public string Usage { get; }
```

Property Value

string

Methods

Execute(ICanvas, List<object>)

The method which will be invoked when the command is run

```
public CommandResult Execute(ICanvas canvas, List<object> parameters)
```

Parameters

`canvas` ICanvas

`parameters` List<object>

A list of objects for parsing in the function which may be used when running the command

Returns

CommandResult

Result of whether the command was successful or not

Namespace Frontend

Classes

MainWindow

Class MainWindow

Namespace: Frontend

Assembly: Frontend.dll

```
public class MainWindow : Form, IDropTarget, ISynchronizeInvoke, IWin32Window,
IBindableComponent, IComponent, IDisposable, IContainerControl
```

Inheritance

object ← MarshalByRefObject ← Component ← Control ← ScrollableControl ← ContainerControl ← Form ← MainWindow

Implements

IDropTarget, ISynchronizeInvoke, IWin32Window, IBindableComponent, IComponent, IDisposable, IContainerControl

Inherited Members

Form.SetVisibleCore(bool) , Form.Activate() , Form.ActivateMdiChild(Form) , Form.AddOwnedForm(Form) , Form.AdjustFormScrollbars(bool) , Form.Close() , Form.CreateAccessibilityInstance() , Form.CreateControlsInstance() , Form.CreateHandle() , Form.DefWndProc(ref Message) , Form.ProcessMnemonic(char) , Form.CenterToParent() , Form.CenterToScreen() , Form.LayoutMdi(MdiLayout) , Form.OnActivated(EventArgs) , Form.OnBackgroundImageChanged(EventArgs) , Form.OnBackgroundImageLayoutChanged(EventArgs) , Form.OnClosing(CancelEventArgs) , Form.OnClosed(EventArgs) , Form.OnFormClosing(FormClosingEventArgs) , Form.OnFormClosed(FormClosedEventArgs) , Form.OnCreateControl() , Form.OnDeactivate(EventArgs) , Form.OnEnabledChanged(EventArgs) , Form.OnEnter(EventArgs) , Form.OnFontChanged(EventArgs) , Form.OnGotFocus(EventArgs) , Form.OnHandleCreated(EventArgs) , Form.OnHandleDestroyed(EventArgs) , Form.OnHelpButtonClicked(CancelEventArgs) , Form.OnLayout(LayoutEventArgs) , Form.OnLoad(EventArgs) , Form.OnMaximizedBoundsChanged(EventArgs) , Form.OnMaximumSizeChanged(EventArgs) , Form.OnMinimumSizeChanged(EventArgs) , Form.OnInputLanguageChanged(InputLanguageChangedEventArgs) , Form.OnInputLanguageChanging(InputLanguageChangingEventArgs) , Form.OnVisibleChanged(EventArgs) , Form.OnMdiChildActivate(EventArgs) , Form.OnMenuStart(EventArgs) , Form.OnMenuComplete(EventArgs) , Form.OnPaint(PaintEventArgs) , Form.OnResize(EventArgs) , Form.OnDpiChanged(DpiChangedEventArgs) , Form.OnGetDpiScaledSize(int, int, ref Size) , Form.OnRightToLeftLayoutChanged(EventArgs) , Form.OnShown(EventArgs) , Form.OnTextChanged(EventArgs) , Form.ProcessCmdKey(ref Message, Keys) , Form.ProcessDialogKey(Keys) , Form.ProcessDialogChar(char) , Form.ProcessKeyPreview(ref Message) , Form.ProcessTabKey(bool) , Form.RemoveOwnedForm(Form) , Form.Select(bool, bool) , Form.ScaleMinMaxSize(float, float, bool) , Form.GetScaledBounds(Rectangle, SizeF, BoundsSpecified) , Form.ScaleControl(SizeF, BoundsSpecified) , Form.SetBoundsCore(int, int, int, int, BoundsSpecified) , Form.SetClientSizeCore(int, int) , Form.SetDesktopBounds(int, int, int, int) , Form.SetDesktopLocation(int, int) , Form.Show(IWin32Window) , Form.ShowDialog() , Form.ShowDialog(IWin32Window) , Form.ToString() , Form.UpdateDefaultButton() , Form.OnResizeBegin(EventArgs) , Form.OnResizeEnd(EventArgs) , Form.OnStyleChanged(EventArgs) , Form.ValidateChildren() , Form.ValidateChildren(ValidationConstraints) , Form.WndProc(ref Message) , Form.AcceptButton , Form.ActiveForm , Form.ActiveMdiChild , Form.AllowTransparency , Form.AutoScroll , Form.AutoSize , Form.AutoSizeMode , Form.AutoValidate , Form.BackColor , Form.FormBorderStyle , Form.CancelButton , Form.ClientSize , Form.ControlBox , Form.CreateParams , Form.DefaultImeMode , Form.DefaultSize , Form.DesktopBounds , Form.DesktopLocation , Form.DialogResult , Form.HelpButton , Form.Icon , Form.IsMdiChild , Form.IsMdiContainer , Form.IsRestrictedWindow , Form.KeyPreview , Form.Location , Form.MaximizedBounds , Form.MaximumSize , Form.MainMenuStrip , Form.MinimumSize , Form.MaximizeBox , Form.MdiChildren , Form.MdiChildrenMinimizedAnchorBottom , Form.MdiParent , Form.MinimizeBox , Form.Modal , Form.Opacity , Form.OwnedForms , Form.Owner , Form.RestoreBounds , Form.RightToLeftLayout , Form.ShowInTaskbar , Form.ShowIcon , Form.ShowWithoutActivation , Form.Size , Form.SizeGripStyle , Form.StartPosition , Form.Text , Form.TopLevel , Form.TopMost , Form.TransparencyKey , Form.WindowState , Form.AutoSizeChanged , Form.AutoValidateChanged , Form.HelpButtonClicked , Form.MaximizedBoundsChanged , Form.MaximumSizeChanged , Form.MinimumSizeChanged , Form.Activated , Form.Deactivate , Form.FormClosing , Form.FormClosed , Form.Load , Form.MdiChildActivate , Form.MenuComplete , Form.MenuStart , Form.InputLanguageChanged , Form.InputLanguageChanging , Form.RightToLeftLayoutChanged , Form.Shown , Form.DpiChanged , Form.ResizeBegin , Form.ResizeEnd , ContainerControl.OnAutoValidateChanged(EventArgs) , ContainerControl.OnMove(EventArgs) , ContainerControl.OnParentChanged(EventArgs) , ContainerControl.PerformAutoScale() , ContainerControl.RescaleConstantsForDpi(int, int) , ContainerControl.Validate() , ContainerControl.Validate(bool) , ContainerControl.AutoScaleDimensions , ContainerControl.AutoScaleFactor , ContainerControl.AutoScaleMode , ContainerControl.BindingContext , ContainerControl.CanEnableIme , ContainerControl.ActiveControl , ContainerControl.CurrentAutoScaleDimensions , ContainerControl.ParentForm , ScrollableControl.ScrollStateAutoScrolling , ScrollableControl.ScrollStateHScrollVisible ,

ScrollableControl.ScrollStateVScrollVisible , ScrollableControl.ScrollStateUserHasScrolled ,
ScrollableControl.ScrollStateFullDrag , ScrollableControl.GetScrollState(int) ,
ScrollableControl.OnMouseWheel(MouseEventArgs) ,
ScrollableControl.OnRightToLeftChanged(EventArgs) ,
ScrollableControl.OnPaintBackground(PaintEventArgs) ,
ScrollableControl.OnPaddingChanged(EventArgs) , ScrollableControl.SetDisplayRectLocation(int, int) ,
ScrollableControl.ScrollControlIntoView(Control) , ScrollableControl.ScrollToControl(Control) ,
ScrollableControl.OnScroll(ScrollEventArgs) , ScrollableControl.SetAutoScrollMargin(int, int) ,
ScrollableControl.SetScrollState(int, bool) , ScrollableControl.AutoScrollMargin ,
ScrollableControl.AutoScrollPosition , ScrollableControl.AutoScrollMinSize ,
ScrollableControl.DisplayRectangle , ScrollableControl.HScroll , ScrollableControl.HorizontalScroll ,
ScrollableControl.VScroll , ScrollableControl.VerticalScroll , ScrollableControl.Scroll ,
Control.GetAccessibilityObjectById(int) , Control.SetAutoSizeMode(AutoSizeMode) ,
Control.GetAutoSizeMode() , Control.GetPreferredSize(Size) ,
Control.AccessibilityNotifyClients(AccessibleEvents, int) ,
Control.AccessibilityNotifyClients(AccessibleEvents, int, int) , Control.BeginInvoke(Delegate) ,
Control.BeginInvoke(Action) , Control.BeginInvoke(Delegate, params object[]) ,
Control.BringToFront() , Control.Contains(Control) , Control.CreateGraphics() ,
Control.CreateControl() , Control.DestroyHandle() , Control.DoDragDrop(object, DragDropEffects) ,
Control.DoDragDrop(object, DragDropEffects, Bitmap, Point, bool) ,
Control.DrawToBitmap(Bitmap, Rectangle) , Control.EndInvoke(IAsyncResult) , Control.FindForm() ,
Control.GetTopLevel() , Control.RaiseKeyEvent(object, KeyEventArgs) ,
Control.RaiseMouseEvent(object, MouseEventArgs) , Control.Focus() ,
Control.FromChildHandle(nint) , Control.FromHandle(nint) ,
Control.GetChildAtPoint(Point, GetChildAtPointSkip) , Control.GetChildAtPoint(Point) ,
Control.GetContainerControl() , Control.GetNextControl(Control, bool) ,
Control.GetStyle(ControlStyles) , Control.Hide() , Control.InitLayout() , Control.Invalidate(Region) ,
Control.Invalidate(Region, bool) , Control.Invalidate() , Control.Invalidate(bool) ,
Control.Invalidate(Rectangle) , Control.Invalidate(Rectangle, bool) , Control.Invoke(Action) ,
Control.Invoke(Delegate) , Control.Invoke(Delegate, params object[]) ,
Control.Invoke<T>(Func<T>) , Control.InvokePaint(Control, PaintEventArgs) ,
Control.InvokePaintBackground(Control, PaintEventArgs) , Control.IsKeyLocked(Keys) ,
Control.IsInputChar(char) , Control.IsInputKey(Keys) , Control.IsMnemonic(char, string) ,
Control.LogicalToDeviceUnits(int) , Control.LogicalToDeviceUnits(Size) ,
Control.ScaleBitmapLogicalToDevice(ref Bitmap) , Control.NotifyInvalidate(Rectangle) ,
Control.InvokeOnClick(Control, EventArgs) , Control.OnAutoSizeChanged(EventArgs) ,
Control.OnBackColorChanged(EventArgs) , Control.OnBindingContextChanged(EventArgs) ,
Control.OnCausesValidationChanged(EventArgs) , Control.OnContextMenuStripChanged(EventArgs) ,
Control.OnCursorChanged(EventArgs) , Control.OnDataContextChanged(EventArgs) ,
Control.OnDockChanged(EventArgs) , Control.OnForeColorChanged(EventArgs) ,

Control.OnNotifyMessage(Message) , Control.OnParentBackColorChanged(EventArgs) ,
Control.OnParentBackgroundImageChanged(EventArgs) ,
Control.OnParentBindingContextChanged(EventArgs) , Control.OnParentCursorChanged(EventArgs) ,
Control.OnParentDataContextChanged(EventArgs) , Control.OnParentEnabledChanged(EventArgs) ,
Control.OnParentFontChanged(EventArgs) , Control.OnParentForeColorChanged(EventArgs) ,
Control.OnParentRightToLeftChanged(EventArgs) , Control.OnParentVisibleChanged(EventArgs) ,
Control.OnPrint(PaintEventArgs) , Control.OnTabIndexChanged(EventArgs) ,
Control.OnTabStopChanged(EventArgs) , Control.OnClick(EventArgs) ,
Control.OnClientSizeChanged(EventArgs) , Control.OnControlAdded(ControlEventArgs) ,
Control.OnControlRemoved(ControlEventArgs) , Control.OnLocationChanged(EventArgs) ,
Control.OnDoubleClick(EventArgs) , Control.OnDragEnter(DragEventArgs) ,
Control.OnDragOver(DragEventArgs) , Control.OnDragLeave(EventArgs) ,
Control.OnDragDrop(DragEventArgs) , Control.OnGiveFeedback(GiveFeedbackEventArgs) ,
Control.InvokeGotFocus(Control, EventArgs) , Control.OnHelpRequested(HelpEventArgs) ,
Control.OnInvalidated(InvalidateEventArgs) , Control.OnKeyDown(KeyEventArgs) ,
Control.OnKeyPress(KeyPressEventArgs) , Control.OnKeyUp(KeyEventArgs) ,
Control.OnLeave(EventArgs) , Control.InvokeLostFocus(Control, EventArgs) ,
Control.OnLostFocus(EventArgs) , Control.OnMarginChanged(EventArgs) ,
Control.OnMouseDoubleClick(MouseEventArgs) , Control.OnMouseClick(MouseEventArgs) ,
Control.OnMouseCaptureChanged(EventArgs) , Control.OnMouseDown(MouseEventArgs) ,
Control.OnMouseEnter(EventArgs) , Control.OnMouseLeave(EventArgs) ,
Control.OnDpiChangedBeforeParent(EventArgs) , Control.OnDpiChangedAfterParent(EventArgs) ,
Control.OnMouseHover(EventArgs) , Control.OnMouseMove(MouseEventArgs) ,
Control.OnMouseUp(MouseEventArgs) ,
Control.OnQueryContinueDrag(QueryContinueDragEventArgs) ,
Control.OnRegionChanged(EventArgs) , Control.OnPreviewKeyDown(PreviewKeyDownEventArgs) ,
Control.OnSizeChanged(EventArgs) , Control.OnChangeUICues(UICuesEventArgs) ,
Control.OnSystemColorsChanged(EventArgs) , Control.OnValidating(CancelEventArgs) ,
Control.OnValidated(EventArgs) , Control.PerformLayout() , Control.PerformLayout(Control, string) ,
Control.PointToClient(Point) , Control.PointToScreen(Point) ,
Control.PreProcessMessage(ref Message) , Control.PreProcessControlMessage(ref Message) ,
Control.ProcessKeyEventArgs(ref Message) , Control.ProcessKeyMessage(ref Message) ,
Control.RaiseDragEvent(object, DragEventArgs) , Control.RaisePaintEvent(object, PaintEventArgs) ,
Control.RecreateHandle() , Control.RectangleToClient(Rectangle) ,
Control.RectangleToScreen(Rectangle) , Control.ReflectMessage(nint, ref Message) ,
Control.Refresh() , Control.ResetMouseEventArgs() , Control.ResetText() , Control.ResumeLayout() ,
Control.ResumeLayout(bool) , Control.Scale(SizeF) , Control.Select() ,
Control.SelectNextControl(Control, bool, bool, bool, bool) , Control.SendToBack() ,
Control.SetBounds(int, int, int, int) , Control.SetBounds(int, int, int, int, BoundsSpecified) ,
Control.SizeFromClientSize(Size) , Control.SetStyle(ControlStyles, bool) , Control.SetTopLevel(bool) ,

Control.RtlTranslateAlignment(HorizontalAlignment) ,
Control.RtlTranslateAlignment(LeftRightAlignment) ,
Control.RtlTranslateAlignment(ContentAlignment) ,
Control.RtlTranslateHorizontal(HorizontalAlignment) ,
Control.RtlTranslateLeftRight(LeftRightAlignment) , Control.RtlTranslateContent(ContentAlignment) ,
Control.Show() , Control.SuspendLayout() , Control.Update() , Control.UpdateBounds() ,
Control.UpdateBounds(int, int, int, int) , Control.UpdateBounds(int, int, int, int, int, int) ,
Control.UpdateZOrder() , Control.UpdateStyles() , Control.OnImeModeChanged(EventArgs) ,
Control.AccessibilityObject , Control.AccessibleDefaultActionDescription ,
Control.AccessibleDescription , Control.AccessibleName , Control.AccessibleRole ,
Control.AllowDrop , Control.Anchor , Control.AutoScrollOffset , Control.LayoutEngine ,
Control.DataContext , Control.BackgroundImage , Control.BackgroundImageLayout ,
Control.Bottom , Control.Bounds , Control.CanFocus , Control.CanRaiseEvents ,
Control.CanSelect , Control.Capture , Control.CausesValidation ,
Control.CheckForIllegalCrossThreadCalls , Control.ClientRectangle , Control.CompanyName ,
Control.ContainsFocus , Control.ContextMenuStrip , Control.Controls , Control.Created ,
Control.Cursor , Control.DataBindings , Control.DefaultBackColor , Control.DefaultCursor ,
Control.DefaultFont , Control.DefaultForeColor , Control.DefaultMargin ,
Control.DefaultMaximumSize , Control.DefaultMinimumSize , Control.DefaultPadding ,
Control.DeviceDpi , Control.IsDisposed , Control.Disposing , Control.Dock ,
Control.DoubleBuffered , Control.Enabled , Control.Focused , Control.Font ,
Control.FontHeight , Control.ForeColor , Control.Handle , Control.HasChildren , Control.Height ,
Control.IsHandleCreated , Control.InvokeRequired , Control.IsAccessible ,
Control.IsAncestorSiteInDesignMode , Control.IsMirrored , Control.Left , Control.Margin ,
Control.ModifierKeys , Control.MouseButtons , Control.MousePosition , Control.Name ,
Control.Parent , Control.ProductName , Control.ProductVersion , Control.RecreatingHandle ,
Control.Region , Control.RenderRightToLeft , Control.ResizeRedraw , Control.Right ,
Control.RightToLeft , Control.ScaleChildren , Control.Site , Control.TabIndex , Control.TabStop ,
Control.Tag , Control.Top , Control.TopLevelControl , Control.ShowKeyboardCues ,
Control.ShowFocusCues , Control.UseWaitCursor , Control.Visible , Control.Width ,
Control.PreferredSize , Control.Padding , Control.ImeMode , Control.ImeModeBase ,
Control.PropagatingImeMode , Control.BackColorChanged , Control.BackgroundImageChanged ,
Control.BackgroundImageLayoutChanged , Control.BindingContextChanged ,
Control.CausesValidationChanged , Control.ClientSizeChanged ,
Control.ContextMenuStripChanged , Control.CursorChanged , Control.DockChanged ,
Control.EnabledChanged , Control.FontChanged , Control.ForeColorChanged ,
Control.LocationChanged , Control.MarginChanged , Control.RegionChanged ,
Control.RightToLeftChanged , Control.SizeChanged , Control.TabIndexChanged ,
Control.TabStopChanged , Control.TextChanged , Control.VisibleChanged , Control.Click ,
Control.ControlAdded , Control.ControlRemoved , Control.DataContextChanged ,

Control.DragDrop , Control.DragEnter , Control.DragOver , Control.DragLeave , Control.GiveFeedback , Control.HandleCreated , Control.HandleDestroyed , Control.HelpRequested , Control.Invalidated , Control.PaddingChanged , Control.Paint , Control.QueryContinueDrag , Control.QueryAccessibilityHelp , Control.DoubleClick , Control.Enter , Control.GotFocus , Control.KeyDown , Control.KeyPress , Control.KeyUp , Control.Layout , Control.Leave , Control.LostFocus , Control.MouseClick , Control.MouseDoubleClick , Control.MouseCaptureChanged , Control.MouseDown , Control.MouseEnter , Control.MouseLeave , Control.DpiChangedBeforeParent , Control.DpiChangedAfterParent , Control.MouseHover , Control.MouseMove , Control.MouseUp , Control.MouseWheel , Control.Move , Control.PreviewKeyDown , Control.Resize , Control.ChangeUICues , Control.StyleChanged , Control.SystemColorsChanged , Control.Validating , Control.Validated , Control.ParentChanged , Control.ImeModeChanged , Component.Dispose() , Component.GetService(Type) , Component.Container , Component.DesignMode , Component.Events , Component.Disposed , MarshalByRefObject.GetLifetimeService() , MarshalByRefObject.InitializeLifetimeService() , MarshalByRefObject.MemberwiseClone(bool) , object.Equals(object) , object.Equals(object, object) , object.GetHashCode() , object.GetType() , object.MemberwiseClone() , object.ReferenceEquals(object, object)

Constructors

MainWindow()

```
public MainWindow()
```

Methods

Dispose(bool)

Clean up any resources being used.

```
protected override void Dispose(bool disposing)
```

Parameters

disposing bool

true if managed resources should be disposed; otherwise, false.