Table of Contents

Namespace SigStat.Benchmark

Classes

Analyser

Analyser.ReportLine

BenchmarkBuilder

GrammarEngine

A simple engine for generating all possible sentences of a formal language and also parsing them based on predefined production rules.

GrammarEngine.ProductionRule

Represents

Structs

GrammarEngine.DerivedSentence

Class Analyser

Inheritance

System.Object

Analyser

Inherited Members

System.Object.Equals(System.Object)

System.Object.Equals(System.Object, System.Object)

System.Object.GetHashCode()

System.Object.GetType()

System.Object.MemberwiseClone()

System.Object.ReferenceEquals(System.Object, System.Object)

System.Object.ToString()

Namespace: SigStat.Benchmark

Assembly: SigStat.Benchmark.dll

Syntax

```
public class Analyser
```

Class Analyser.ReportLine

Inheritance

System.Object
Analyser.ReportLine

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: SigStat.Benchmark

Assembly: SigStat.Benchmark.dll

Syntax

```
public class ReportLine
```

Properties

AER

Declaration

```
public double AER { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Double	

Agent

Declaration

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

Property Value

TYPE	DESCRIPTION
System.String	

Benchmark

Declaration

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

Property Value

TYPE	DESCRIPTION
System.String	

ClassificationResults

Declaration

```
public List<ClassificationResult> ClassificationResults { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Collections.Generic.List<ClassificationResult>	

Classifier

Declaration

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

Property Value

TYPE	DESCRIPTION
System.String	

Database

Declaration

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

Property Value

TYPE	DESCRIPTION
System.String	

Date

Declaration

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

Property Value

TYPE	DESCRIPTION
System.String	

Distance

Declaration

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

Property Value

TYPE	DESCRIPTION
System.String	

Duration

Declaration

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

Property Value

TYPE	DESCRIPTION
System.String	

FAR

Declaration

```
public double FAR { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Double	

Feature

Declaration

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

Property Value

TYPE	DESCRIPTION
System.String	

FillGap

Declaration

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

Property Value

TYPE	DESCRIPTION
System.String	

FillInterpolation

Declaration

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

Property Value

TYPE	DESCRIPTION
System.String	

FilterGap

Declaration

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

Property Value

TYPE	DESCRIPTION
System.String	

FRR

Declaration

```
public double FRR { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Double	

Gap

Declaration

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

Property Value

TYPE	DESCRIPTION
System.String	

Key

Declaration

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

Property Value

TYPE	DESCRIPTION
System.String	

Pipeline

Declaration

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

Property Value

TYPE	DESCRIPTION
System.String	

Resampling

Declaration

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

Property Value

TYPE	DESCRIPTION
System.String	

ResamplingInterpolation

Declaration

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

Property Value

TYPE	DESCRIPTION
System.String	

Rotation

Declaration

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

Property Value

TYPE	DESCRIPTION
System.String	

SampleCount

Declaration

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

Property Value

TYPE	DESCRIPTION
System.String	

Scaling

Declaration

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

Property Value

TYPE	DESCRIPTION
System.String	

Split

Declaration

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

Property Value

TYPE	DESCRIPTION
System.String	

Translation

Declaration

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

Property Value

TYPE	DESCRIPTION
System.String	

Verifier

Declaration

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

Property Value

TYPE	DESCRIPTION
System.String	

Class BenchmarkBuilder

Inheritance

System.Object
BenchmarkBuilder

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: SigStat.Benchmark

Assembly: SigStat.Benchmark.dll

Syntax

```
public class BenchmarkBuilder
```

Constructors

BenchmarkBuilder(String)

Declaration

```
public BenchmarkBuilder(string databasePath = null)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	databasePath	

Methods

Build(Dictionary<String, String>)

Declaration

```
public VerifierBenchmark Build(Dictionary<string, string> config)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Collections.Generic.Dictionary<System.String, System.String>	config	

Returns

TYPE	DESCRIPTION
VerifierBenchmark	

GetLoaders()

Declaration

```
public IEnumerable<KeyValuePair<string, DataSetLoader>> GetLoaders()
```

Returns

TYPE	DESCRIPTION
System.Collections.Generic.IEnumerable<System.Collections.Generic.KeyValuePair<System.String, DataSetLoader>>	

GetSampler(String)

Declaration

```
public Sampler GetSampler(string key)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	key	

Returns

TYPE	DESCRIPTION
Sampler	

ParseFeatures(String)

Declaration

```
public List<FeatureDescriptor<List<double>>> ParseFeatures(string featuresString)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	featuresString	

Returns

TYPE	DESCRIPTION
System.Collections.Generic.List<FeatureDescriptor<System.Collections.Generic.List<System.Double>>>	

Class GrammarEngine

A simple engine for generating all possible sentences of a formal language and also parsing them based on predefined production rules.

Inheritance

System.Object
GrammarEngine

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: SigStat.Benchmark

Assembly: SigStat.Benchmark.dll

Syntax

```
public static class GrammarEngine
```

Methods

GenerateAllSentences(GrammarEngine.ProductionRule[], String, String)

Generates all sentences for a given rule-set.

Declaration

```
public static IEnumerable<GrammarEngine.DerivedSentence> GenerateAllSentences(GrammarEngine.ProductionRule[]
rules, string fragment = null, string derivationPath = "")
```

Parameters

TYPE	NAME	DESCRIPTION
GrammarEngine.ProductionRule[]	rules	An array of production rules.
System.String	fragment	The current sentence fragment.
System.String	derivationPath	The current derivation path.

Returns

TYPE	DESCRIPTION
System.Collections.Generic.IEnumerable<GrammarEngine.DerivedSentence>	

Exceptions

TYPE	CONDITION
System.InvalidOperationException	Expression contains nonterminal symbols without any matching production rule: {fragment}

ParseRule(String)

Expects a production rule in the following format: [Non terminal symbol] -> any combination of [terminal] and nonterminal symbols

Declaration

```
public static GrammarEngine.ProductionRule ParseRule(string ruleString)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	ruleString	The rule represented as a string

Returns

TYPE	DESCRIPTION
GrammarEngine.ProductionRule	

ParseRules(String)

Parses the production rules from the given string. The first nonterminal symbol in the left hand side of the first rule is treated as the start symbol. Nonterminal symbols must be enclosed into square brackets, eg [Symbol]. All other characters are treated as terminal symbols. [Verifier] -> [Feature]_[Classifier] [Feature] -> X,Y,P,XY,YXP,XP,YP [Classifier] -> DTW, HMM

Declaration

```
public static GrammarEngine.ProductionRule[] ParseRules(string rulesString)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	rulesString	The rules string.

Returns

TYPE	DESCRIPTION
GrammarEngine.ProductionRule[]	An array of parsed production rules

Exceptions

TYPE	CONDITION
System.ArgumentException	

ParseSentence(String, GrammarEngine.ProductionRule[], String)

Determines the sequence of production rules used to generate the sentence

Declaration

```
public static Dictionary<string, string> ParseSentence(string sentence, GrammarEngine.ProductionRule[] rules,
string whiteSpaceCharacters = "\t _")
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	sentence	The sentence.
GrammarEngine.ProductionRule[]	rules	The rules of the grammar.
System.String	whiteSpaceCharacters	

Returns

TYPE	DESCRIPTION
System.Collections.Generic.Dictionary<System.String, System.String>	

Exceptions

TYPE	CONDITION
System.Collections.Generic.KeyNotFoundException	Sentence '{sentence}'can not be generated by the rules

Struct GrammarEngine.DerivedSentence

Inherited Members

System.ValueType.Equals(System.Object)

System.ValueType.GetHashCode()

System.ValueType.ToString()

System.Object.Equals(System.Object, System.Object)

System.Object.GetType()

System.Object.ReferenceEquals(System.Object, System.Object)

Namespace: SigStat.Benchmark

Assembly: SigStat.Benchmark.dll

Syntax

```
public struct DerivedSentence
```

Constructors

DerivedSentence(String, String)

Declaration

```
public DerivedSentence(string sentence, string derivationPath)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	sentence	
System.String	derivationPath	

Fields

DerivationPath

Declaration

```
public string DerivationPath
```

Field Value

TYPE	DESCRIPTION
System.String	

Sentence

Declaration

```
public string Sentence
```

Field Value

TYPE	DESCRIPTION
System.String	

Class GrammarEngine.ProductionRule

Represents

Inheritance

System.Object
GrammarEngine.ProductionRule

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)

Namespace: SigStat.Benchmark

Assembly: SigStat.Benchmark.dll

Syntax

```
public class ProductionRule
```

Constructors

ProductionRule(String, String[])

Declaration

```
public ProductionRule(string pattern, params string[] fragments)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	pattern	
System.String[]	fragments	

Fields

IsStartSymbol

Declaration

```
public bool IsStartSymbol
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

Properties

Fragments

Declaration

```
public string[] Fragments { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.String[]	

Pattern

Declaration

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

Property Value

TYPE	DESCRIPTION
System.String	

Methods

ToString()

Declaration

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

Returns

TYPE	DESCRIPTION
System.String	

Overrides

System.Object.ToString()

Namespace SigStat.Benchmark.Execution

Classes

ClassificationResult

Class ClassificationResult

Inheritance

System.Object
ClassificationResult

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: SigStat.Benchmark.Execution

Assembly: SigStat.Benchmark.dll

Syntax

```
public class ClassificationResult
```

Properties

Aer

Declaration

```
public double Aer { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Double	

Algorithm

Declaration

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

Property Value

TYPE	DESCRIPTION
System.String	

Distance

Declaration

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

Property Value

TYPE	DESCRIPTION
System.String	

Far

Declaration

```
public double Far { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Double	

Frr

Declaration

```
public double Frr { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Double	

J

Declaration

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

Property Value

TYPE	DESCRIPTION
System.String	

K

Declaration

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

Property Value

TYPE	DESCRIPTION
System.String	

Namespace SigStat.Benchmark.Helpers

Classes

BenchmarkDatabase

dal for sigstat benchmarks database

Structs

BenchmarkDatabase.ExecutionStatistics

Class BenchmarkDatabase

dal for sigstat benchmarks database

Inheritance

System.Object
BenchmarkDatabase

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: SigStat.Benchmark.Helpers

Assembly: SigStat.Benchmark.dll

Syntax

```
public static class BenchmarkDatabase
```

Methods

ClearExperiment()

Declaration

```
public static Task ClearExperiment()
```

Returns

TYPE	DESCRIPTION
System.Threading.Tasks.Task	Number of deleted configurations.

CountFaulted()

Declaration

```
public static Task<int> CountFaulted()
```

Returns

TYPE	DESCRIPTION
System.Threading.Tasks.Task<System.Int32>	

CountFinished()

Declaration

```
public static Task<int> CountFinished()
```

Returns

TYPE	DESCRIPTION
System.Threading.Tasks.Task<System.Int32>	

CountLocked()

Declaration

```
public static Task<int> CountLocked()
```

Returns

TYPE	DESCRIPTION
System.Threading.Tasks.Task<System.Int32>	

CountQueued()

Declaration

```
public static Task<int> CountQueued()
```

Returns

TYPE	DESCRIPTION
System.Threading.Tasks.Task<System.Int32>	

ExperimentExists()

Declaration

```
public static Task<bool> ExperimentExists()
```

Returns

TYPE	DESCRIPTION
System.Threading.Tasks.Task<System.Boolean>	

GetExecutionStatisticsAsync()

Declaration

```
public static Task<BenchmarkDatabase.ExecutionStatistics> GetExecutionStatisticsAsync()
```

Returns

TYPE	DESCRIPTION
System.Threading.Tasks.Task<BenchmarkDatabase.ExecutionStatistics>	

GetGrammarRules()

Declaration

```
public static Task<string> GetGrammarRules()
```

Returns

TYPE	DESCRIPTION
System.Threading.Tasks.Task<System.String>	

GetResults()

Declaration

```
public static IEnumerable<Analyser.ReportLine> GetResults()
```

Returns

TYPE	DESCRIPTION
System.Collections.Generic.IEnumerable<Analyser.ReportLine>	

InitializeConnection(String)

Declaration

```
public static Task InitializeConnection(string connectionString)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	connectionString	

Returns

TYPE	DESCRIPTION
System.Threading.Tasks.Task	

LockNextBenchmarkReport()

Declaration

```
public static Task<BenchmarkReport> LockNextBenchmarkReport()
```

Returns

TYPE	DESCRIPTION
System.Threading.Tasks.Task<BenchmarkReport>	

LockNextConfig(Int32)

This is atomic. Returns null if no config can be locked

Declaration

```
public static Task<string> LockNextConfig(int procId)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	procId	

Returns

TYPE	DESCRIPTION
System.Threading.Tasks.Task<System.String>	Configuration string

RemoveLocks()

Remove locks from unfinished configurations.

Declaration

```
public static Task<int> RemoveLocks()
```

Returns

TYPE	DESCRIPTION
System.Threading.Tasks.Task<System.Int32>	Number of locks removed.

RequeueFinished()

Declaration

```
public static Task<int> RequeueFinished()
```

Returns

TYPE	DESCRIPTION
System.Threading.Tasks.Task<System.Int32>	

ResetFaulted()

Remove locks and logs from configurations that ran into exceptions.

Declaration

```
public static Task<int> ResetFaulted()
```

Returns

TYPE	DESCRIPTION
System.Threading.Tasks.Task<System.Int32>	

SendErrorLog(String, String)

Send log after exception.

Declaration

```
public static Task SendErrorLog(string benchmarkConfig, string logString)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	benchmarkConfig	
System.String	logString	

Returns

TYPE	DESCRIPTION
System.Threading.Tasks.Task	

SendResults(Int32, String, BenchmarkLogModel)

Add results to a specified benchmark item.

Declaration

```
public static Task SendResults(int procId, string benchmarkConfig, BenchmarkLogModel results)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	procId	
System.String	benchmarkConfig	
BenchmarkLogModel	results	

Returns

TYPE	DESCRIPTION
System.Threading.Tasks.Task	

SetGrammarRules(String)

Declaration

```
public static Task SetGrammarRules(string rulesString)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	rulesString	

Returns

TYPE	DESCRIPTION
System.Threading.Tasks.Task	

ToJson(BsonDocument)

Declaration

```
public static string ToJson(BsonDocument bson)
```

Parameters

TYPE	NAME	DESCRIPTION
MongoDB.Bson.BsonDocument	bson	

Returns

TYPE	DESCRIPTION
System.String	

UpsertConfigs(IEnumerable<String>, Int32, Int32)

Insert congifurations if they don't exist already. Due to the 400 RU/s restriction, documents are sent in small batches and client-side throttling is applied when required. Note: This method does not remove locks, results and logs on existing items.

Declaration

```
public static Task<int> UpsertConfigs(IEnumerable<string> configs, int batchSize = 10, int skipCount = 0)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Collections.Generic.IEnumerable<System.String>	configs	
System.Int32	batchSize	
System.Int32	skipCount	

Returns

TYPE	DESCRIPTION
System.Threading.Tasks.Task<System.Int32>	Number of new configurations inserted.

Struct BenchmarkDatabase.ExecutionStatistics

Inherited Members

System.ValueType.Equals(System.Object)

System.ValueType.GetHashCode()

System.ValueType.ToString()

System.Object.Equals(System.Object, System.Object)

System.Object.GetType()

System.Object.ReferenceEquals(System.Object, System.Object)

Namespace: SigStat.Benchmark.Helpers

Assembly: SigStat.Benchmark.dll

Syntax

```
public struct ExecutionStatistics
```

Fields

Count

Declaration

```
public long Count
```

Field Value

TYPE	DESCRIPTION
System.Int64	

ExpiredLockCount

Declaration

```
public long ExpiredLockCount
```

Field Value

TYPE	DESCRIPTION
System.Int64	

MaxMilliseconds

Declaration

```
public long MaxMilliseconds
```

Field Value

TYPE	DESCRIPTION
System.Int64	

Size

Declaration

```
public long Size
```

Field Value

TYPE	DESCRIPTION
System.Int64	

StorageSize

Declaration

```
public long StorageSize
```

Field Value

TYPE	DESCRIPTION
System.Int64	

TotalMilliseconds

Declaration

```
public long TotalMilliseconds
```

Field Value

TYPE	DESCRIPTION
System.Int64	

Namespace SigStat.Benchmark.Model

Classes

BenchmarkReport

Class BenchmarkReport

Inheritance

System.Object
BenchmarkReport

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: SigStat.Benchmark.Model

Assembly: SigStat.Benchmark.dll

Syntax

```
public class BenchmarkReport
```

Properties

Config

Declaration

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

Property Value

TYPE	DESCRIPTION
System.String	

Database

Declaration

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

Property Value

TYPE	DESCRIPTION
System.String	

Distance

Declaration

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

Property Value

TYPE	DESCRIPTION
System.String	

SignerResults

Declaration

```
public List<SignerResults> SignerResults { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Collections.Generic.List<SignerResults>	

Split

Declaration

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

Property Value

TYPE	DESCRIPTION
System.String	

Namespace SigStat.Benchmark.Options

Classes

AnalyserOptions

Command line options for analyser mode

MonitorOptions

Command line options for monitor mode

OptionsBase

Base class for command line options containing common options

ProcessorOptions

Command line options for analyser mode

Class AnalyserOptions

Command line options for analyser mode

Inheritance

System.Object
OptionsBase
AnalyserOptions

Inherited Members

OptionsBase.Connection
OptionsBase.Experiment
System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: SigStat.Benchmark.Options

Assembly: SigStat.Benchmark.dll

Syntax

```
[Verb("analyse", HelpText = "Analyser mode for analysing benchmark results.")]
public class AnalyserOptions : OptionsBase
```

Properties

ReportFilePath

Declaration

```
[Option('r', "report", Required = false, Default = "report.xlsx", HelpText = "Name of the generated report
file (e.g. something.xlsx)")]
public string ReportFilePath { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.String	

Methods

RunAsync()

Declaration

```
public override Task RunAsync()
```

Returns

TYPE	DESCRIPTION
System.Threading.Tasks.Task	

Overrides

OptionsBase.RunAsync()

Class MonitorOptions

Command line options for monitor mode

Inheritance

System.Object
OptionsBase
MonitorOptions

Inherited Members

OptionsBase.Connection
OptionsBase.Experiment
System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: SigStat.Benchmark.Options

Assembly: SigStat.Benchmark.dll

Syntax

```
[Verb("monitor", HelpText = "Monitoring mode for checking HTCondor jobs or database status.")]
public class MonitorOptions : OptionsBase
```

Methods

RunAsync()

Declaration

```
public override Task RunAsync()
```

Returns

TYPE	DESCRIPTION
System.Threading.Tasks.Task	

Overrides

OptionsBase.RunAsync()

Class OptionsBase

Base class for command line options containing common options

Inheritance

System.Object
OptionsBase
AnalyserOptions
MonitorOptions
ProcessorOptions

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: SigStat.Benchmark.Options

Assembly: SigStat.Benchmark.dll

Syntax

```
public abstract class OptionsBase
```

Properties

Connection

Declaration

```
[Option('c', "connection", Required = false, Default = "mongodb://localhost:27017/", HelpText = "MongoDB
connection string in Uri format (see: https://docs.mongodb.com/manual/reference/connection-string). Defaults
to localhost.")]
public string Connection { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.String	

Experiment

Declaration

```
[Option('e', "experiment", Required = false, Default = "test", HelpText = "Unique name for the experiment.
Default: test")]
public string Experiment { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.String	

Methods

RunAsync()

Declaration

```
public abstract Task RunAsync()
```

Returns

TYPE	DESCRIPTION
System.Threading.Tasks.Task	

Class ProcessorOptions

Command line options for analyser mode

Inheritance

System.Object
OptionsBase
ProcessorOptions

Inherited Members

OptionsBase.Connection
OptionsBase.Experiment
System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: SigStat.Benchmark.Options

Assembly: SigStat.Benchmark.dll

Syntax

```
[Verb("process", HelpText = "Analyser mode for analysing benchmark results.")]
public class ProcessorOptions : OptionsBase
```

Methods

RunAsync()

Declaration

```
public override Task RunAsync()
```

Returns

TYPE	DESCRIPTION
System.Threading.Tasks.Task	

Overrides

OptionsBase.RunAsync()

Namespace SigStat.Common

Classes

ArrayExtension

Helper methods for processing arrays

Baseline

BasicMetadataExtraction

Extracts basic statistical signature (like or Cog) information from an Image

BenchmarkResults

Contains the benchmark results of every Signer and the summarized final results.

DistanceMatrix<TRowKey, TColumnKey, TValue>

A Sparse Matrix representation of a distance graph.

FeatureDescriptor

Represents a feature with name and type.

FeatureDescriptor<T>

Represents a feature with the type `T`

Features

Standard set of features.

ILoggerObjectExtensions

ILoggerObject extension methods for common scenarios.

IOExtensions

Extension methods for common IO operations in platform edit##############################

Loop

Represents a loop in a signature

MathHelper

Common mathematical functions used by the SigStat framework

PipelineBase

TODO: Ideiglenes osztaly, C# 8.0 ban ezt atalakitani default implementacios interface be. ILoggerObject, IProgress, IPipelineIO default implementacioja.

Result

Contains the benchmark results of a single Signer

Sampler

Takes samples from a set of Signatures by given sampling strategies. Use this to fine-tune the VerifierBenchmark

Signature

Represents a signature as a collection of features, containing the data that flows in the pipeline.

SignatureHelper

Signer

Represents a person as an ID and a list of Signatures.

SigStatEvents

Standard event identifiers used by the SigStat system

SimpleRenderingTransformation

Renders an image of the signature based on the available online information (X,Y,Dpi)

StrokeHelper

Helper class for locating and manipulating strokes in an online signature

StrokeInterval

Represents a stroke in an online signature

VerifierBenchmark

Benchmarking class to test error rates of a Verifier

Structs

ErrorRate

Represents the ErrorRates achieved in a benchmark

Interfaces

ILoggerObject

Represents a type, that contains an ILogger property that can be used to perform logging.

ITransformation

Allows implementing a pipeline transform item capable of logging, progress tracking and IO rewiring.

Enums

Origin

Represents our knowledge on the origin of a signature.

StrokeType

Describes the type of a stroke

Class ArrayExtension

Helper methods for processing arrays

Inheritance

System.Object
ArrayExtension

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: SigStat.Common

Assembly: SigStat.Common.dll

Syntax

```
public static class ArrayExtension
```

Methods

GetCog(Double[,])

Calculates the center of gravity, assuming that each cell contains a weight value

Declaration

```
public static (int x, int y) GetCog(this double[, ] weightMartix)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Double[,]	weightMartix	

Returns

TYPE	DESCRIPTION
System.ValueTuple<System.Int32, System.Int32>	

GetValues<T>(T[,])

Enumerates all values in a two dimensional array

Declaration

```
public static IEnumerable<T> GetValues<T>(this T[, ] array)
```

Parameters

TYPE	NAME	DESCRIPTION
T[,]	array	The array to enumerate

Returns

TYPE	DESCRIPTION
System.Collections.Generic.IEnumerable<T>	

Type Parameters

NAME	DESCRIPTION
T	Array type

SetValues<T>(T[,], T)

Sets all values in a two dimensional array to `value`

Declaration

```
public static T[, ] SetValues<T>(this T[, ] array, T value)
```

Parameters

TYPE	NAME	DESCRIPTION
T[,]	array	Array
T	value	New value for the array elements

Returns

TYPE	DESCRIPTION
T[,]	A reference to `array` (allows chaining)

Type Parameters

NAME	DESCRIPTION
T	Array type

ToArrays<T>(IEnumerable<T>, Int32)

Enumerates items into arrays of given capacity. If there are less items than 'capacity', a smaller array is returned

Declaration

```
public static IEnumerable<T[]> ToArrays<T>(this IEnumerable<T> items, int capacity)
```

Parameters

TYPE	NAME	DESCRIPTION

TYPE	NAME	DESCRIPTION
System.Collections.Generic.IEnumerable<T>	items	The items.
System.Int32	capacity	The capacity.

Returns

TYPE	DESCRIPTION
System.Collections.Generic.IEnumerable<T[]>	

Type Parameters

NAME	DESCRIPTION
T	

Class Baseline

Inheritance

System.Object
Baseline

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)

Namespace: SigStat.Common

Assembly: SigStat.Common.dll

Syntax

```
[JsonObject(MemberSerialization.OptOut)]
public class Baseline
```

Constructors

Baseline()

Initializes a Baseline instance

Declaration

```
public Baseline()
```

Baseline(Int32, Int32, Int32, Int32)

Initializes a Baseline instance with the given startpoint and endpoint

Declaration

```
public Baseline(int x1, int y1, int x2, int y2)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	x1	x coordinate for the start point
System.Int32	y1	y coordinate for the start point
System.Int32	x2	x coordinate for the endpoint
System.Int32	y2	y coordinate for the endpoint

Properties

End

Endpoint of the baseline

Declaration

```
public PointF End { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Drawing.PointF	

Start

Starting point of the baseline

Declaration

```
public PointF Start { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Drawing.PointF	

Methods

ToString()

Returns a string representation of the baseline

Declaration

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

Returns

TYPE	DESCRIPTION
System.String	

Overrides

System.Object.ToString()

Class BasicMetadataExtraction

Extracts basic statistical signature (like or Cog) information from an Image

Inheritance

System.Object
PipelineBase
BasicMetadataExtraction

Implements

ILoggerObject
IProgress
ITransformation
IPipelineIO

Inherited Members

PipelineBase.PipelineInputs
PipelineBase.PipelineOutputs
PipelineBase.Logger
PipelineBase.Progress
PipelineBase.ProgressChanged
PipelineBase.OnProgressChanged()
System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: SigStat.Common

Assembly: SigStat.Common.dll

Syntax

```
[JsonObject(MemberSerialization.OptOut)]
public class BasicMetadataExtraction : PipelineBase, ILoggerObject, IProgress, ITransformation, IPipelineIO
```

Properties

Trim

Represents theratio of significant pixels that should be trimmed from each side while calculating TrimmedBounds

Declaration

```
public static double Trim { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Double	

Methods

Transform(Signature)

Declaration

```
public void Transform(Signature signature)
```

Parameters

TYPE	NAME	DESCRIPTION
Signature	signature	

Implements

ILoggerObject

IProgress

ITransformation

IPipelineIO

Extension Methods

ILoggerObjectExtensions.LogError(ILoggerObject, String, Object[])

ILoggerObjectExtensions.LogError(ILoggerObject, Exception, String, Object[])

ILoggerObjectExtensions.LogInformation(ILoggerObject, String, Object[])

ILoggerObjectExtensions.LogWarning(ILoggerObject, String, Object[])

ILoggerObjectExtensions.LogWarning(ILoggerObject, Exception, String, Object[])

ILoggerObjectExtensions.LogTrace(ILoggerObject, String, Object[])

ILoggerObjectExtensions.LogTrace<TState>(ILoggerObject, TState, EventId, Exception, Func<TState, Exception, String>)

ILoggerObjectExtensions.LogCritical(ILoggerObject, String, Object[])

ILoggerObjectExtensions.LogDebug(ILoggerObject, String, Object[])

Class BenchmarkResults

Contains the benchmark results of every Signer and the summarized final results.

Inheritance

System.Object
BenchmarkResults

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: SigStat.Common

Assembly: SigStat.Common.dll

Syntax

```
[JsonObject(MemberSerialization.OptOut)]
public class BenchmarkResults
```

Fields

FinalResult

Summarized, final result of the benchmark execution.

Declaration

```
[JsonProperty]
public readonly Result FinalResult
```

Field Value

TYPE	DESCRIPTION
Result	

SignerResults

List that contains the Results for each Signer

Declaration

```
[JsonProperty]
public readonly List<Result> SignerResults
```

Field Value

TYPE	DESCRIPTION
System.Collections.Generic.List<Result>	

Class DistanceMatrix<TRowKey, TColumnKey, TValue>

A Sparse Matrix representation of a distance graph.

Inheritance

System.Object
DistanceMatrix<TRowKey, TColumnKey, TValue>

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: SigStat.Common

Assembly: SigStat.Common.dll

Syntax

```
public class DistanceMatrix<TRowKey, TColumnKey, TValue>
```

Type Parameters

NAME	DESCRIPTION
TRowKey	Type to represent the row indexes
TColumnKey	Type to represent the column indexes
TValue	Type to represent the distances

Properties

Item[TRowKey, TColumnKey]

Gets or sets a distance for a given row and column

Declaration

```
public TValue this[TRowKey row, TColumnKey column] { get; set; }
```

Parameters

TYPE	NAME	DESCRIPTION
TRowKey	row	row
TColumnKey	column	column

Property Value

TYPE	DESCRIPTION
TValue	

Methods

ContainsKey(TRowKey, TColumnKey)

Determines whether the Matrix contains the specified key pair

Declaration

```
public bool ContainsKey(TRowKey row, TColumnKey column)
```

Parameters

TYPE	NAME	DESCRIPTION
TRowKey	row	
TColumnKey	column	

Returns

TYPE	DESCRIPTION
System.Boolean	true if the Matrix contains an element with the specified keys; otherwise, false.

GetDistance(TRowKey, TColumnKey)

Gets or sets a distance for a given row and column

Declaration

```
public TValue GetDistance(TRowKey row, TColumnKey column)
```

Parameters

TYPE	NAME	DESCRIPTION
TRowKey	row	row
TColumnKey	column	column

Returns

TYPE	DESCRIPTION
TValue	

TryGetValue(TRowKey, TColumnKey, out TValue)

Gets the value associated with the specified keys.

Declaration

```
public bool TryGetValue(TRowKey row, TColumnKey column, out TValue value)
```

Parameters

TYPE	NAME	DESCRIPTION
TRowKey	row	
TColumnKey	column	
TValue	value	

Returns

TYPE	DESCRIPTION
System.Boolean	true if the Matrix contains an element with the specified keys; otherwise, false.

Struct ErrorRate

Represents the ErrorRates achieved in a benchmark

Inherited Members

System.ValueType.Equals(System.Object)
System.ValueType.GetHashCode()
System.ValueType.ToString()
System.Object.Equals(System.Object, System.Object)
System.Object.GetType()
System.Object.ReferenceEquals(System.Object, System.Object)

Namespace: SigStat.Common

Assembly: SigStat.Common.dll

Syntax

```
public struct ErrorRate
```

Fields

Far

False Acceptance Rate

Declaration

```
public double Far
```

Field Value

TYPE	DESCRIPTION
System.Double	

Frr

False Rejection Rate

Declaration

```
public double Frr
```

Field Value

TYPE	DESCRIPTION
System.Double	

Properties

Aer

Average Error Rate (calculated from Far and Frr)

Declaration

```
public double Aer { get; }
```

Property Value

TYPE	DESCRIPTION
System.Double	

Class FeatureDescriptor

Represents a feature with name and type.

Inheritance

System.Object
FeatureDescriptor
FeatureDescriptor<T>

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)

Namespace: SigStat.Common

Assembly: SigStat.Common.dll

Syntax

```
public class FeatureDescriptor
```

Constructors

FeatureDescriptor(String, String, Type)

Initializes a new instance of the FeatureDescriptor class, and adds it to the static descriptors. Therefore, the `key` parameter must be unique.

Declaration

```
protected FeatureDescriptor(string name, string key, Type featureType)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	name	
System.String	key	
System.Type	featureType	

Fields

descriptors

The static dictionary of all descriptors.

Declaration

```
protected static readonly Dictionary<string, FeatureDescriptor> descriptors
```

Field Value

TYPE	DESCRIPTION
System.Collections.Generic.Dictionary<System.String, FeatureDescriptor>	

Properties

FeatureType

Gets or sets the type of the feature.

Declaration

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

Property Value

TYPE	DESCRIPTION
System.Type	

IsCollection

Gets whether the type of the feature is List.

Declaration

```
public bool IsCollection { get; }
```

Property Value

TYPE	DESCRIPTION
System.Boolean	

Key

Gets the unique key of the feature.

Declaration

```
public string Key { get; protected set; }
```

Property Value

TYPE	DESCRIPTION
System.String	

Name

Gets or sets a human readable name of the feature.

Declaration

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

Property Value

TYPE	DESCRIPTION
System.String	

Methods

Get(String)

Gets the FeatureDescriptor specified by `key`. Throws System.Collections.Generic.KeyNotFoundException exception if there is no descriptor registered with the given key.

Declaration

```
public static FeatureDescriptor Get(string key)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	key	

Returns

TYPE	DESCRIPTION
FeatureDescriptor	

Get<T>(String)

Gets the FeatureDescriptor<T> specified by `key`. If the key is not registered yet, a new FeatureDescriptor<T> is automatically created with the given key and type.

Declaration

```
public static FeatureDescriptor<T> Get<T>(string key)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	key	

Returns

TYPE	DESCRIPTION
FeatureDescriptor<T>	

Type Parameters

NAME	DESCRIPTION
T	

GetAll()

Gets a dictionary of all registered feature descriptors

Declaration

```
public static Dictionary<string, FeatureDescriptor> GetAll()
```

Returns

TYPE	DESCRIPTION
System.Collections.Generic.Dictionary<System.String, FeatureDescriptor>	

IsRegistered(String)

Returns true, if there is a FeatureDescriptor registered with the given key

Declaration

```
public static bool IsRegistered(string featureKey)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	featureKey	The key to search for

Returns

TYPE	DESCRIPTION
System.Boolean	

Register(String, Type)

Registers a new FeatureDescriptor with a given key. If the FeatureDescriptor is allready registered, this function will return a reference to the originally registered FeatureDescriptor. to the a

Declaration

```
public static FeatureDescriptor Register(string featureKey, Type type)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	featureKey	The key for identifying the FeatureDescriptor
System.Type	type	The type of the actual feature values represented by FeatureDescriptor

Returns

TYPE	DESCRIPTION
FeatureDescriptor	A reference to the registered FeatureDescriptor instance

ToString()

Returns a string represenatation of the FeatureDescriptor

Declaration

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

Returns

TYPE	DESCRIPTION
System.String	

Overrides

System.Object.ToString()

Class FeatureDescriptor<T>

Represents a feature with the type `T`

Inheritance

System.Object
FeatureDescriptor
FeatureDescriptor<T>

Inherited Members

FeatureDescriptor.Name
FeatureDescriptor.Key
FeatureDescriptor.FeatureType
FeatureDescriptor.IsCollection
FeatureDescriptor.descriptors
FeatureDescriptor.IsRegistered(String)
FeatureDescriptor.Register(String, Type)
FeatureDescriptor.GetAll()
FeatureDescriptor.Get<T>(String)
FeatureDescriptor.ToString()
System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)

Namespace: SigStat.Common

Assembly: SigStat.Common.dll

Syntax

```
public class FeatureDescriptor<T> : FeatureDescriptor
```

Type Parameters

NAME	DESCRIPTION
T	Type of the feature.

Methods

Get(String)

Gets the FeatureDescriptor<T> specified by `key`. If the key is not registered yet, a new FeatureDescriptor<T> is automatically created with the given key and type.

Declaration

```
public static FeatureDescriptor<T> Get(string key)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	key	

Returns

TYPE	DESCRIPTION
FeatureDescriptor<T>	

Class Features

Standard set of features.

Inheritance

System.Object
Features

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: SigStat.Common

Assembly: SigStat.Common.dll

Syntax

```
public static class Features
```

Fields

All

Returns a readonly list of all FeatureDescriptors defined in Features

Declaration

```
public static readonly IReadOnlyList<FeatureDescriptor> All
```

Field Value

TYPE	DESCRIPTION
System.Collections.Generic.IReadOnlyList<FeatureDescriptor>	

Altitude

Altitude of an online signature as a function of T

Declaration

```
public static readonly FeatureDescriptor<List<double>> Altitude
```

Field Value

TYPE	DESCRIPTION
FeatureDescriptor<System.Collections.Generic.List<System.Double>>	

Azimuth

Azimuth of an online signature as a function of T

Declaration

```
public static readonly FeatureDescriptor<List<double>> Azimuth
```

Field Value

TYPE	DESCRIPTION
FeatureDescriptor<System.Collections.Generic.List<System.Double>>	

Cog

Center of gravity in a signature

Declaration

```
public static readonly FeatureDescriptor<Point> Cog
```

Field Value

TYPE	DESCRIPTION
FeatureDescriptor<SixLabors.Primitives.Point>	

Dpi

Dots per inch

Declaration

```
public static readonly FeatureDescriptor<int> Dpi
```

Field Value

TYPE	DESCRIPTION
FeatureDescriptor<System.Int32>	

Image

The visaul representation of a signature

Declaration

```
public static readonly FeatureDescriptor<Image<Rgba32>> Image
```

Field Value

TYPE	DESCRIPTION
FeatureDescriptor<SixLabors.ImageSharp.Image<SixLabors.ImageSharp.PixelFormats.Rgba32>>	

PenDown

Pen position of an online signature as a function of T. It is true when the pen touches the paper.

Declaration

```
public static readonly FeatureDescriptor<List<bool>> PenDown
```

Field Value

TYPE	DESCRIPTION
FeatureDescriptor<System.Collections.Generic.List<System.Boolean>>	

PointType

Type of points of an online signature as a function of T. The type of a point is defined by: 0 - Stroke - Internal point of an up or downstroke 1 - Start - Starting point of a downstroke 2 - End - Last point of a downstroke 3 - ShortStroke - First and last point of a downstroke

Declaration

```
public static readonly FeatureDescriptor<List<double>> PointType
```

Field Value

TYPE	DESCRIPTION
FeatureDescriptor<System.Collections.Generic.List<System.Double>>	

Pressure

Pressure of an online signature as a function of T

Declaration

```
public static readonly FeatureDescriptor<List<double>> Pressure
```

Field Value

TYPE	DESCRIPTION
FeatureDescriptor<System.Collections.Generic.List<System.Double>>	

Size

Actual bounds of the signature

Declaration

```
public static readonly FeatureDescriptor<SizeF> Size
```

Field Value

TYPE	DESCRIPTION
FeatureDescriptor<SixLabors.Primitives.SizeF>	

T

Timestamps for online signatures

Declaration

```
public static readonly FeatureDescriptor<List<double>> T
```

Field Value

TYPE	DESCRIPTION
FeatureDescriptor<System.Collections.Generic.List<System.Double>>	

TrimmedBounds

Represents the main body of the signature BasicMetadataExtraction

Declaration

```
public static readonly FeatureDescriptor<Rectangle> TrimmedBounds
```

Field Value

TYPE	DESCRIPTION
FeatureDescriptor<SixLabors.Primitives.Rectangle>	

X

X coordinates of an online signature as a function of T

Declaration

```
public static readonly FeatureDescriptor<List<double>> X
```

Field Value

TYPE	DESCRIPTION
FeatureDescriptor<System.Collections.Generic.List<System.Double>>	

Y

Y coordinates of an online signature as a function of T

Declaration

```
public static readonly FeatureDescriptor<List<double>> Y
```

Field Value

TYPE	DESCRIPTION
FeatureDescriptor<System.Collections.Generic.List<System.Double>>	

Interface ILoggerObject

Represents a type, that contains an ILogger property that can be used to perform logging.

Namespace: SigStat.Common

Assembly: SigStat.Common.dll

Syntax

```
public interface ILoggerObject
```

Properties

Logger

Gets or sets the ILogger implementation used to perform logging

Declaration

```
ILogger Logger { get; set; }
```

Property Value

TYPE	DESCRIPTION
Microsoft.Extensions.Logging.ILogger	

Extension Methods

ILoggerObjectExtensions.LogError(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogError(ILoggerObject, Exception, String, Object[])
ILoggerObjectExtensions.LogInformation(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogWarning(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogWarning(ILoggerObject, Exception, String, Object[])
ILoggerObjectExtensions.LogTrace(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogTrace<TState>(ILoggerObject, TState, EventId, Exception, Func<TState, Exception, String>)
ILoggerObjectExtensions.LogCritical(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogDebug(ILoggerObject, String, Object[])

Class ILoggerObjectExtensions

ILoggerObject extension methods for common scenarios.

Inheritance

System.Object
ILoggerObjectExtensions

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: SigStat.Common

Assembly: SigStat.Common.dll

Syntax

```
public static class ILoggerObjectExtensions
```

Remarks

Note to framework developers: you may extend this class with additional overloads if they are required

Methods

LogCritical(ILoggerObject, String, Object[])

Formats and writes an critical error log message.

Declaration

```
public static void LogCritical(this ILoggerObject obj, string message, params object[] args)
```

Parameters

TYPE	NAME	DESCRIPTION
ILoggerObject	obj	The SigStat.Common.ILoggerObject containing the Logger to write to.
System.String	message	Format string of the log message in message template format. Example: "User {User} logged in from {Address}"
System.Object[]	args	An object array that contains zero or more objects to format.

LogDebug(ILoggerObject, String, Object[])

Formats and writes an debug log message.

Declaration

```
public static void LogDebug(this ILoggerObject obj, string message, params object[] args)
```

Parameters

TYPE	NAME	DESCRIPTION
ILoggerObject	obj	The SigStat.Common.ILoggerObject containing the Logger to write to.
System.String	message	Format string of the log message in message template format. Example: "User {User} logged in from {Address}"
System.Object[]	args	An object array that contains zero or more objects to format.

LogError(ILoggerObject, Exception, String, Object[])

Formats and writes an error log message.

Declaration

```
public static void LogError(this ILoggerObject obj, Exception exception, string message, params object[] args)
```

Parameters

TYPE	NAME	DESCRIPTION
ILoggerObject	obj	The SigStat.Common.ILoggerObject containing the Logger to write to.
System.Exception	exception	The exception to log.
System.String	message	Format string of the log message in message template format. Example: "User {User} logged in from {Address}"
System.Object[]	args	An object array that contains zero or more objects to format.

LogError(ILoggerObject, String, Object[])

Formats and writes an error log message.

Declaration

```
public static void LogError(this ILoggerObject obj, string message, params object[] args)
```

Parameters

TYPE	NAME	DESCRIPTION
ILoggerObject	obj	The SigStat.Common.ILoggerObject containing the Logger to write to.

TYPE	NAME	DESCRIPTION
System.String	message	Format string of the log message in message template format. Example: "User {User} logged in from {Address}"
System.Object[]	args	An object array that contains zero or more objects to format.

LogInformation(ILoggerObject, String, Object[])

Formats and writes an informational log message.

Declaration

```
public static void LogInformation(this ILoggerObject obj, string message, params object[] args)
```

Parameters

TYPE	NAME	DESCRIPTION
ILoggerObject	obj	The SigStat.Common.ILoggerObject containing the Logger to write to.
System.String	message	Format string of the log message in message template format. Example: "User {User} logged in from {Address}"
System.Object[]	args	An object array that contains zero or more objects to format.

LogTrace(ILoggerObject, String, Object[])

Formats and writes a trace log message.

Declaration

```
public static void LogTrace(this ILoggerObject obj, string message, params object[] args)
```

Parameters

TYPE	NAME	DESCRIPTION
ILoggerObject	obj	The SigStat.Common.ILoggerObject containing the Logger to write to.
System.String	message	Format string of the log message in message template format. Example: "User {User} logged in from {Address}"
System.Object[]	args	An object array that contains zero or more objects to format.

LogTrace<TState>(ILoggerObject, TState, EventId, Exception, Func<TState, Exception, String>)

Formats and writes a trace log message with state.

Declaration

```
public static void LogTrace<TState>(this ILoggerObject obj, TState state, EventId eventId = default(EventId),
Exception exception = null, Func<TState, Exception, string> formatter = null)
```

Parameters

TYPE	NAME	DESCRIPTION
ILoggerObject	obj	The SigStat.Common.ILoggerObject containing the Logger to write to.
TState	state	The entry to be written.
Microsoft.Extensions.Logging.EventId	eventId	Id of the event.
System.Exception	exception	The exception related to this entry.
System.Func<TState, System.Exception, System.String>	formatter	Function to create a String message of the state and exception.

Type Parameters

NAME	DESCRIPTION
TState	The type of the object to be written (preferably a descendant of SigstatLogState).

LogWarning(ILoggerObject, Exception, String, Object[])

Formats and writes an warning log message.

Declaration

```
public static void LogWarning(this ILoggerObject obj, Exception exception, string message, params object[]
args)
```

Parameters

TYPE	NAME	DESCRIPTION
ILoggerObject	obj	The SigStat.Common.ILoggerObject containing the Logger to write to.
System.Exception	exception	The exception to log.
System.String	message	Format string of the log message in message template format. Example: "User {User} logged in from {Address}"

TYPE	NAME	DESCRIPTION
System.Object[]	args	An object array that contains zero or more objects to format.

LogWarning(ILoggerObject, String, Object[])

Formats and writes an warning log message.

Declaration

```
public static void LogWarning(this ILoggerObject obj, string message, params object[] args)
```

Parameters

TYPE	NAME	DESCRIPTION
ILoggerObject	obj	The SigStat.Common.ILoggerObject containing the Logger to write to.
System.String	message	Format string of the log message in message template format. Example: "User {User} logged in from {Address}"
System.Object[]	args	An object array that contains zero or more objects to format.

Class IOExtensions

Extension methods for common IO operations in platform edit##############################

Inheritance

System.Object
IOExtensions

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: SigStat.Common

Assembly: SigStat.Common.dll

Syntax

```
public static class IOExtensions
```

Methods

GetPath(String)

Gets the given relative or absolute path in a platform neutral form

Declaration

```
public static string GetPath(this string path)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	path	

Returns

TYPE	DESCRIPTION
System.String	

Interface ITransformation

Allows implementing a pipeline transform item capable of logging, progress tracking and IO rewiring.

Inherited Members

IPipelineIO.PipelineInputs
IPipelineIO.PipelineOutputs

Namespace: SigStat.Common

Assembly: SigStat.Common.dll

Syntax

```
public interface ITransformation : IPipelineIO
```

Methods

Transform(Signature)

Executes the transform on the `signature` parameter. This function gets called by the pipeline.

Declaration

```
void Transform(Signature signature)
```

Parameters

TYPE	NAME	DESCRIPTION
Signature	signature	The Signature with a set of features to be transformed.

Class Loop

Represents a loop in a signature

Inheritance

System.Object
Loop

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)

Namespace: SigStat.Common

Assembly: SigStat.Common.dll

Syntax

```
[JsonObject(MemberSerialization.OptOut)]
public class Loop
```

Constructors

Loop()

Creates a Loop instance

Declaration

```
public Loop()
```

Loop(Single, Single)

Creates a Loop instance and initializes the Center property

Declaration

```
public Loop(float centerX, float centerY)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Single	centerX	
System.Single	centerY	

Properties

Bounds

The bounding rectangle of the loop

Declaration

```
[JsonConverter(typeof(RectangleFConverter))]
public RectangleF Bounds { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Drawing.RectangleF	

Center

The geometrical center of the looop

Declaration

```
public PointF Center { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Drawing.PointF	

Points

A list of defining points of the loop

Declaration

```
public List<PointF> Points { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Collections.Generic.List<System.Drawing.PointF>	

Methods

ToString()

Returns a string representation of the loop

Declaration

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

Returns

TYPE	DESCRIPTION
System.String	

Overrides

System.Object.ToString()

Class MathHelper

Common mathematical functions used by the SigStat framework

Inheritance

System.Object
MathHelper

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: SigStat.Common

Assembly: SigStat.Common.dll

Syntax

```
public static class MathHelper
```

Methods

Median(IEnumerable<Double>)

Calculates the median of the given data series

Declaration

```
public static double Median(this IEnumerable<double> values)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Collections.Generic.IEnumerable<System.Double>	values	The data series

Returns

TYPE	DESCRIPTION
System.Double	

Min(Double, Double, Double)

Returns the smallest of the three double parameters

Declaration

```
public static double Min(double d1, double d2, double d3)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Double	d1	

TYPE	NAME	DESCRIPTION
System.Double	d2	
System.Double	d3	

Returns

TYPE	DESCRIPTION
System.Double	

StdDiviation(IEnumerable<Double>)

return standard diviation of a feature values

Declaration

```
public static double StdDiviation(this IEnumerable<double> values)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Collections.Generic.IEnumerable<System.Double>	values	

Returns

TYPE	DESCRIPTION
System.Double	

Enum Origin

Represents our knowledge on the origin of a signature.

Namespace: SigStat.Common

Assembly: SigStat.Common.dll

Syntax

```
public enum Origin
```

Fields

NAME	DESCRIPTION
Forged	The Signature is a forgery.
Genuine	The Signature's origin is verified to be from Signer
Unknown	Use this in practice before a signature is verified.

Class PipelineBase

TODO: Ideiglenes osztaly, C# 8.0 ban ezt atalakitani default implementacios interface be. ILoggerObject, IProgress, IPipelineIO default implementacioja.

Inheritance

System.Object

PipelineBase

BasicMetadataExtraction

ParallelTransformPipeline

SequentialTransformPipeline

DtwClassifier

NearestNeighborEerClassifier

OneClassNearestNeighborClassifier

OptimalDtwClassifier

WeightedClassifier

FillPenUpDurations

FilterPoints

NormalizeRotation

NormalizeRotation2

NormalizeRotation3

NormalizeRotationForX

OrthognalRotation

RelativeScale

ResampleSamplesCountBased

SampleRate

Scale

TranslatePreproc

UniformScale

ZNormalization

SimpleRenderingTransformation

AddConst

AddVector

ApproximateOnlineFeatures

Binarization

BinaryRasterizer

CentroidExtraction

ComponentExtraction

ComponentSorter

ComponentsToFeatures

EndpointExtraction

Extrema

HSCPThinning

ImageGenerator

Map

Multiply

Normalize

OnePixelThinning

RealisticImageGenerator

Resize

TangentExtraction

Trim

OnlineToOfflineFeature
StrokeMerging
ReSamplingFeatureExtraction
AlterDtwPairing
DistanceMatrixViewer
StrokePairingDistances
StrokePairSaver
XYSaver

Implements

ILoggerObject
IProgress
IPipelineIO

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: SigStat.Common

Assembly: SigStat.Common.dll

Syntax

```
public abstract class PipelineBase : ILoggerObject, IProgress, IPipelineIO
```

Constructors

PipelineBase()

Initializes a new instance of the PipelineBase class.

Declaration

```
public PipelineBase()
```

Properties

Logger

Declaration

```
public ILogger Logger { get; set; }
```

Property Value

TYPE	DESCRIPTION
Microsoft.Extensions.Logging.ILogger	

PipelineInputs

A collection of inputs for the pipeline elements

Declaration

```
public virtual List<PipelineInput> PipelineInputs { get; }
```

Property Value

TYPE	DESCRIPTION
System.Collections.Generic.List<PipelineInput>	

PipelineOutputs

A collection of outputs for the pipeline elements

Declaration

```
public virtual List<PipelineOutput> PipelineOutputs { get; }
```

Property Value

TYPE	DESCRIPTION
System.Collections.Generic.List<PipelineOutput>	

Progress

Declaration

```
public int Progress { get; protected set; }
```

Property Value

TYPE	DESCRIPTION
System.Int32	

Methods

OnProgressChanged()

Raises the ProgressChanged event

Declaration

```
protected virtual void OnProgressChanged()
```

Events

ProgressChanged

The event is raised whenever the value of Progress changes

Declaration

```
public event EventHandler<int> ProgressChanged
```

Event Type

TYPE	DESCRIPTION
System.EventHandler<System.Int32>	

Implements

ILoggerObject
IProgress

IPipelineIO

Extension Methods

ILoggerObjectExtensions.LogError(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogError(ILoggerObject, Exception, String, Object[])
ILoggerObjectExtensions.LogInformation(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogWarning(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogWarning(ILoggerObject, Exception, String, Object[])
ILoggerObjectExtensions.LogTrace(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogTrace<TState>(ILoggerObject, TState, EventId, Exception, Func<TState, Exception, String>)
ILoggerObjectExtensions.LogCritical(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogDebug(ILoggerObject, String, Object[])

Class Result

Contains the benchmark results of a single Signer

Inheritance

System.Object
Result

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: SigStat.Common

Assembly: SigStat.Common.dll

Syntax

```
public class Result
```

Fields

Model

Declaration

```
public readonly ISignerModel Model
```

Field Value

TYPE	DESCRIPTION
ISignerModel	

Properties

Aer

Average Error Rate

Declaration

```
[JsonProperty]
public double Aer { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Double	

Far

False Acceptance Rate

Declaration

```
[JsonProperty]
public double Far { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Double	

Frr

False Rejection Rate

Declaration

```
[JsonProperty]
public double Frr { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Double	

Signer

Identifier of the Signer

Declaration

```
[JsonProperty]
public string Signer { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.String	

Class Sampler

Takes samples from a set of Signatures by given sampling strategies. Use this to fine-tune the VerifierBenchmark

Inheritance

System.Object
Sampler
EvenNSampler
FirstNSampler
LastNSampler
OddNSampler
TestingSampler
UniversalSampler

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: SigStat.Common

Assembly: SigStat.Common.dll

Syntax

```
public class Sampler
```

Constructors

Sampler(Func<List<Signature>, List<Signature>>, Func<List<Signature>, List<Signature>>, Func<List<Signature>, List<Signature>>)

Initialize a new instance of the Sampler class by given sampling strategies.

Declaration

```
public Sampler(Func<List<Signature>, List<Signature>> references, Func<List<Signature>, List<Signature>>
genuineTests, Func<List<Signature>, List<Signature>> forgeryTests)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Func<System.Collections.Generic.List<Signature>, System.Collections.Generic.List<Signature>>	references	Strategy to sample genuine signatures to be used for training.
System.Func<System.Collections.Generic.List<Signature>, System.Collections.Generic.List<Signature>>	genuineTests	Strategy to sample genuine signatures to be used for testing.
System.Func<System.Collections.Generic.List<Signature>, System.Collections.Generic.List<Signature>>	forgeryTests	Strategy to sample forged signatures to be used for testing.

Properties

ForgeryTestFilter

Declaration

```
public Func<List<Signature>, List<Signature>> ForgeryTestFilter { get; protected set; }
```

Property Value

TYPE	DESCRIPTION
System.Func<System.Collections.Generic.List<Signature>, System.Collections.Generic.List<Signature>>	

GenuineTestFilter

Declaration

```
public Func<List<Signature>, List<Signature>> GenuineTestFilter { get; protected set; }
```

Property Value

TYPE	DESCRIPTION
System.Func<System.Collections.Generic.List<Signature>, System.Collections.Generic.List<Signature>>	

TrainingFilter

Declaration

```
public Func<List<Signature>, List<Signature>> TrainingFilter { get; protected set; }
```

Property Value

TYPE	DESCRIPTION
System.Func<System.Collections.Generic.List<Signature>, System.Collections.Generic.List<Signature>>	

Methods

SampleForgeryTests(List<Signature>)

Samples a batch of forged signatures to test on.

Declaration

```
public List<Signature> SampleForgeryTests(List<Signature> signatures)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Collections.Generic.List<Signature>	signatures	

Returns

TYPE	DESCRIPTION
System.Collections.Generic.List<Signature>	Forged signatures to test on.

SampleGenuineTests(List<Signature>)

Samples a batch of genuine test signatures to test on.

Declaration

```
public List<Signature> SampleGenuineTests(List<Signature> signatures)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Collections.Generic.List<Signature>	signatures	

Returns

TYPE	DESCRIPTION
System.Collections.Generic.List<Signature>	Genuine signatures to test on.

SampleReferences(List<Signature>)

Samples a batch of genuine reference signatures to train on.

Declaration

```
public List<Signature> SampleReferences(List<Signature> signatures)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Collections.Generic.List<Signature>	signatures	

Returns

TYPE	DESCRIPTION
System.Collections.Generic.List<Signature>	Genuine reference signatures to train on.

Class Signature

Represents a signature as a collection of features, containing the data that flows in the pipeline.

Inheritance

System.Object
Signature

Implements

System.Collections.Generic.IEnumerable<System.Collections.Generic.KeyValuePair<FeatureDescriptor, System.Object>>
System.Collections.IEnumerable

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)

Namespace: SigStat.Common

Assembly: SigStat.Common.dll

Syntax

```
[JsonObject(MemberSerialization.OptOut)]
public class Signature : IEnumerable<KeyValuePair<FeatureDescriptor, object>>, IEnumerable
```

Constructors

Signature()

Initializes a signature instance

Declaration

```
public Signature()
```

Signature(String, Origin, Signer)

Initializes a signature instance with the given properties

Declaration

```
public Signature(string signatureID, Origin origin = Origin.Unknown, Signer signer = null)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	signatureID	
Origin	origin	
Signer	signer	

Properties

ID

An identifier for the Signature. Keep it unique to be useful for logs.

Declaration

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

Property Value

TYPE	DESCRIPTION
System.String	

Item[FeatureDescriptor]

Gets or sets the specified feature.

Declaration

```
public object this[FeatureDescriptor featureDescriptor] { get; set; }
```

Parameters

TYPE	NAME	DESCRIPTION
FeatureDescriptor	featureDescriptor	

Property Value

TYPE	DESCRIPTION
System.Object	The feature object without cast.

Item[String]

Gets or sets the specified feature.

Declaration

```
public object this[string featureKey] { get; set; }
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	featureKey	

Property Value

TYPE	DESCRIPTION
System.Object	The feature object without cast.

Origin

Represents our knowledge on the origin of the signature. Unknown may be used in practice before it is verified.

Declaration

```
public Origin Origin { get; set; }
```

Property Value

TYPE	DESCRIPTION
Origin	

Signer

A reference to the Signer who this signature belongs to. (The origin is not constrained to be genuine.)

Declaration

```
public Signer Signer { get; set; }
```

Property Value

TYPE	DESCRIPTION
Signer	

Methods

GetAggregateFeature(List<FeatureDescriptor>)

Aggregate multiple features into one. Example: X, Y features -> P.xy feature. Use this for example at DTW algorithm input.

Declaration

```
public List<double[]> GetAggregateFeature(List<FeatureDescriptor> fs)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Collections.Generic.List<FeatureDescriptor>	fs	List of features to aggregate.

Returns

TYPE	DESCRIPTION
System.Collections.Generic.List<System.Double[]>	Aggregated feature value

GetEnumerator()

Returns an enumerator that iterates through the features.

Declaration

```
public IEnumerator<KeyValuePair<FeatureDescriptor, object>> GetEnumerator()
```

Returns

TYPE	DESCRIPTION
System.Collections.Generic.IEnumerator<System.Collections.Generic.KeyValuePair<FeatureDescriptor, System.Object>>	An enumerator that can be used to iterate through the features.

GetFeature<T>(FeatureDescriptor)

Gets the specified feature. This is the preferred way.

Declaration

```
public T GetFeature<T>(FeatureDescriptor featureDescriptor)
```

Parameters

TYPE	NAME	DESCRIPTION
FeatureDescriptor	featureDescriptor	

Returns

TYPE	DESCRIPTION
T	The casted feature object

Type Parameters

NAME	DESCRIPTION
T	

GetFeature<T>(FeatureDescriptor<T>)

Gets the specified feature. This is the preferred way.

Declaration

```
public T GetFeature<T>(FeatureDescriptor<T> featureDescriptor)
```

Parameters

TYPE	NAME	DESCRIPTION
FeatureDescriptor<T>	featureDescriptor	

Returns

TYPE	DESCRIPTION
T	The casted feature object

Type Parameters

NAME	DESCRIPTION
T	

GetFeature<T>(String)

Gets the specified feature.

Declaration

```
public T GetFeature<T>(string featureKey)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	featureKey	

Returns

TYPE	DESCRIPTION
T	The casted feature object

Type Parameters

NAME	DESCRIPTION
T	

GetFeatureDescriptors()

Gets a collection of FeatureDescriptors that are used in this signature.

Declaration

```
public IEnumerable<FeatureDescriptor> GetFeatureDescriptors()
```

Returns

TYPE	DESCRIPTION
System.Collections.Generic.IEnumerable<FeatureDescriptor>	A collection of FeatureDescriptors.

HasFeature(FeatureDescriptor)

Returns true if the signature contains the specified feature

Declaration

```
public bool HasFeature(FeatureDescriptor featureDescriptor)
```

Parameters

TYPE	NAME	DESCRIPTION
FeatureDescriptor	featureDescriptor	

Returns

TYPE	DESCRIPTION
System.Boolean	

HasFeature(String)

Returns true if the signature contains the specified feature

Declaration

```
public bool HasFeature(string featureKey)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	featureKey	

Returns

TYPE	DESCRIPTION
System.Boolean	

SetFeature<T>(FeatureDescriptor, T)

Sets the specified feature.

Declaration

```
public Signature SetFeature<T>(FeatureDescriptor featureDescriptor, T feature)
```

Parameters

TYPE	NAME	DESCRIPTION
FeatureDescriptor	featureDescriptor	The feature to put the new value in.
T	feature	The value to set.

Returns

TYPE	DESCRIPTION
Signature	

Type Parameters

NAME	DESCRIPTION
T	

SetFeature<T>(String, T)

Sets the specified feature.

Declaration

```
public Signature SetFeature<T>(string featureKey, T feature)
```

Parameters

TYPE	NAME	DESCRIPTION

TYPE	NAME	DESCRIPTION
System.String	featureKey	The unique key of the feature.
T	feature	The value to set.

Returns

TYPE	DESCRIPTION
Signature	

Type Parameters

NAME	DESCRIPTION
T	

ToString()

Returns a string representation of the signature

Declaration

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

Returns

TYPE	DESCRIPTION
System.String	

Overrides

System.Object.ToString()

Explicit Interface Implementations

IEnumerable.GetEnumerator()

Declaration

```
IEnumerator IEnumerable.GetEnumerator()
```

Returns

TYPE	DESCRIPTION
System.Collections.IEnumerator	

Implements

System.Collections.Generic.IEnumerable<T>
System.Collections.IEnumerable

Extension Methods

ArrayExtension.ToArrays<T>(IEnumerable<T>, Int32)
StrokeHelper.GetStrokes(Signature)

Class SignatureHelper

Inheritance

System.Object
SignatureHelper

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: SigStat.Common

Assembly: SigStat.Common.dll

Syntax

```
public static class SignatureHelper
```

Methods

GetSignatureLength(Signature)

Return the signature length using Eculidan distance

Declaration

```
public static double GetSignatureLength(Signature signature)
```

Parameters

TYPE	NAME	DESCRIPTION
Signature	signature	

Returns

TYPE	DESCRIPTION
System.Double	

SaveImage(Signature, String)

Save online signature as file

Declaration

```
public static void SaveImage(Signature sig, string fileName)
```

Parameters

TYPE	NAME	DESCRIPTION
Signature	sig	
System.String	fileName	

Class Signer

Represents a person as an ID and a list of Signatures.

Inheritance

System.Object
Signer

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)

Namespace: SigStat.Common

Assembly: SigStat.Common.dll

Syntax

```
public class Signer
```

Fields

bestFrr

best frr for the signer, used to find the best sampling frequency and step for each signer

Declaration

```
public double bestFrr
```

Field Value

TYPE	DESCRIPTION
System.Double	

bestSampleRate

best sampling frequency for the signer

Declaration

```
public int bestSampleRate
```

Field Value

TYPE	DESCRIPTION
System.Int32	

bestStep

best step (nmber of skipped points) for the signer

Declaration

```
public int bestStep
```

Field Value

TYPE	DESCRIPTION
System.Int32	

Properties

ID

An identifier for the Signer. Keep it unique to be useful for logs.

Declaration

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

Property Value

TYPE	DESCRIPTION
System.String	

Signatures

List of signatures that belong to the signer. (Their origin is not constrained to be genuine.)

Declaration

```
public virtual List<Signature> Signatures { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Collections.Generic.List<Signature>	

Methods

ToString()

Returns a string representation of a Signer

Declaration

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

Returns

TYPE	DESCRIPTION
System.String	

Overrides

System.Object.ToString()

Extension Methods

SignerStatisticsHelper.GetLengthAverage(Signer)
SignerStatisticsHelper.GetWidthAvg(Signer)
SignerStatisticsHelper.GetHeightAvg(Signer)
SignerStatisticsHelper.GetPointsAvg(Signer)

SignerStatisticsHelper.GetMinSignaturePoints(Signer)

SignerStatisticsHelper.GetMaxSignaturePoints(Signer)

Class SigStatEvents

Standard event identifiers used by the SigStat system

Inheritance

System.Object
SigStatEvents

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: SigStat.Common

Assembly: SigStat.Common.dll

Syntax

```
public static class SigStatEvents
```

Fields

BenchmarkEvent

Events originating from a benchmark

Declaration

```
public static readonly EventId BenchmarkEvent
```

Field Value

TYPE	DESCRIPTION
Microsoft.Extensions.Logging.EventId	

VerifierEvent

Events originating from a verifier

Declaration

```
public static readonly EventId VerifierEvent
```

Field Value

TYPE	DESCRIPTION
Microsoft.Extensions.Logging.EventId	

Class SimpleRenderingTransformation

Renders an image of the signature based on the available online information (X,Y,Dpi)

Inheritance

System.Object
PipelineBase
SimpleRenderingTransformation

Implements

ILoggerObject
IProgress
ITransformation
IPipelineIO

Inherited Members

PipelineBase.PipelineInputs
PipelineBase.PipelineOutputs
PipelineBase.Logger
PipelineBase.Progress
PipelineBase.ProgressChanged
PipelineBase.OnProgressChanged()
System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: SigStat.Common

Assembly: SigStat.Common.dll

Syntax

```
[JsonObject(MemberSerialization.OptOut)]
public class SimpleRenderingTransformation : PipelineBase, ILoggerObject, IProgress, ITransformation,
IPipelineIO
```

Methods

Transform(Signature)

Declaration

```
public void Transform(Signature signature)
```

Parameters

TYPE	NAME	DESCRIPTION
Signature	signature	

Implements

ILoggerObject
IProgress
ITransformation
IPipelineIO

Extension Methods

ILoggerObjectExtensions.LogError(ILoggerObject, String, Object[])

ILoggerObjectExtensions.LogError(ILoggerObject, Exception, String, Object[])

ILoggerObjectExtensions.LogInformation(ILoggerObject, String, Object[])

ILoggerObjectExtensions.LogWarning(ILoggerObject, String, Object[])

ILoggerObjectExtensions.LogWarning(ILoggerObject, Exception, String, Object[])

ILoggerObjectExtensions.LogTrace(ILoggerObject, String, Object[])

ILoggerObjectExtensions.LogTrace<TState>(ILoggerObject, TState, EventId, Exception, Func<TState, Exception, String>)

ILoggerObjectExtensions.LogCritical(ILoggerObject, String, Object[])

ILoggerObjectExtensions.LogDebug(ILoggerObject, String, Object[])

Class StrokeHelper

Helper class for locating and manipulating strokes in an online signature

Inheritance

System.Object
StrokeHelper

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: SigStat.Common

Assembly: SigStat.Common.dll

Syntax

```
public static class StrokeHelper
```

Methods

GetStrokes(Signature)

Gets the strokes from an online signature with standard features. Note that the signature has to contain T and Pressure

Declaration

```
public static List<StrokeInterval> GetStrokes(this Signature signature)
```

Parameters

TYPE	NAME	DESCRIPTION
Signature	signature	An online signature with standard features

Returns

TYPE	DESCRIPTION
System.Collections.Generic.List<StrokeInterval>	

Class StrokeInterval

Represents a stroke in an online signature

Inheritance

System.Object
StrokeInterval

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: SigStat.Common

Assembly: SigStat.Common.dll

Syntax

```
public class StrokeInterval
```

Constructors

StrokeInterval(Int32, Int32, StrokeType)

Initializes a new instance of the StrokeInterval struct.

Declaration

```
public StrokeInterval(int startIndex, int endIndex, StrokeType strokeType)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	startIndex	The index of the firs element
System.Int32	endIndex	The index of the last element
StrokeType	strokeType	Type of the stroke.

Fields

EndIndex

The index of the last element

Declaration

```
public int EndIndex
```

Field Value

TYPE	DESCRIPTION
System.Int32	

StartIndex

The index of the firs element

Declaration

```
public int StartIndex
```

Field Value

TYPE	DESCRIPTION
System.Int32	

StrokeType

The StrokeType of the stroke.

Declaration

```
public StrokeType StrokeType
```

Field Value

TYPE	DESCRIPTION
StrokeType	

Enum StrokeType

Describes the type of a stroke

Namespace: SigStat.Common

Assembly: SigStat.Common.dll

Syntax

```
public enum StrokeType
```

Fields

NAME	DESCRIPTION
Down	The stroke was made on the writing surface (tablet, paper etc.)
Unknown	The type of the stroke is not known
Up	The stroke was made in the air (the pen did not tuch the tablet/paper)

Class VerifierBenchmark

Benchmarking class to test error rates of a Verifier

Inheritance

System.Object
VerifierBenchmark

Implements

ILoggerObject

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: SigStat.Common

Assembly: SigStat.Common.dll

Syntax

```
[JsonObject(MemberSerialization.OptOut)]
public class VerifierBenchmark : ILoggerObject
```

Constructors

VerifierBenchmark()

Initializes a new instance of the VerifierBenchmark class. Sets the Sampler to the default FirstNSampler.

Declaration

```
public VerifierBenchmark()
```

Fields

SignerModels

An optional dictionary of fully or partially precalculated signer models. You may fill itt before executing a benchmark if you have saved the models previously

Declaration

```
public List<ISignerModel> SignerModels
```

Field Value

TYPE	DESCRIPTION
System.Collections.Generic.List<ISignerModel>	

Properties

Loader

The loader that will provide the database for benchmarking

Declaration

```
public IDataSetLoader Loader { get; set; }
```

Property Value

TYPE	DESCRIPTION
IDataSetLoader	

Logger

Gets or sets the attached Microsoft.Extensions.Logging.ILogger object used to log messages. Hands it over to the verifier.

Declaration

```
public ILogger Logger { get; set; }
```

Property Value

TYPE	DESCRIPTION
Microsoft.Extensions.Logging.ILogger	

Parameters

A key value store that can be used to store custom information about the benchmark

Declaration

```
public List<KeyValuePair<string, string>> Parameters { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Collections.Generic.List<System.Collections.Generic.KeyValuePair<System.String, System.String>>	

Progress

Declaration

```
[JsonIgnore]
public int Progress { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Int32	

Sampler

The Sampler to be used for benchmarking

Declaration

```
public Sampler Sampler { get; set; }
```

Property Value

TYPE	DESCRIPTION
Sampler	

Verifier

Gets or sets the Verifier to be benchmarked.

Declaration

```
public Verifier Verifier { get; set; }
```

Property Value

TYPE	DESCRIPTION
Verifier	

Methods

Dump(String, IEnumerable<KeyValuePair<String, String>>)

Dumps the results of the benchmark in a file.

Declaration

```
public void Dump(string filename, IEnumerable<KeyValuePair<string, string>> parameters)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	filename	The filename.
System.Collections.Generic.IEnumerable<System.Collections.Generic.KeyValuePair<System.String, System.String>>	parameters	The custom parameters of the benchmark (to be included in the dump)

Execute(Boolean)

Execute the benchmarking process.

Declaration

```
public BenchmarkResults Execute(bool ParallelMode = true)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Boolean	ParallelMode	

Returns

TYPE	DESCRIPTION
BenchmarkResults	

Execute(Int32)

Execute the benchmarking process with a degree of parallelism.

Declaration

```
public BenchmarkResults Execute(int degreeOfParallelism)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	degreeOfParallelism	Degree of parallelism is the maximum number of concurrently executing tasks.

Returns

TYPE	DESCRIPTION
BenchmarkResults	

Events

ProgressChanged

Declaration

```
public event EventHandler<int> ProgressChanged
```

Event Type

TYPE	DESCRIPTION
System.EventHandler<System.Int32>	

Implements

ILoggerObject

Extension Methods

ILoggerObjectExtensions.LogError(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogError(ILoggerObject, Exception, String, Object[])
ILoggerObjectExtensions.LogInformation(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogWarning(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogWarning(ILoggerObject, Exception, String, Object[])
ILoggerObjectExtensions.LogTrace(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogTrace<TState>(ILoggerObject, TState, EventId, Exception, Func<TState, Exception, String>)
ILoggerObjectExtensions.LogCritical(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogDebug(ILoggerObject, String, Object[])

Namespace SigStat.Common.Algorithms

Classes

DtwImplementations

A simple implementation of the DTW algorithm.

HSCPThinningStep

HSCP thinning algorithm http://www.ppgia.pucpr.br/~facon/Afinamento/1987holt.pdf

PatternMatching3x3

Binary 3x3 pattern matcher with rotating option.

Class DtwImplementations

A simple implementation of the DTW algorithm.

Inheritance

System.Object
DtwImplementations

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: SigStat.Common.Algorithms

Assembly: SigStat.Common.dll

Syntax

```
public static class DtwImplementations
```

Methods

ConstrainedDTw<P>(IEnumerable<P>, IEnumerable<P>, Func<P, P, Double>, Int32)

Constrained DTW implementation (Abdullah Mueen, Eamonn J. Keogh)

Declaration

```
public static double ConstrainedDTw<P>(IEnumerable<P> sequence1, IEnumerable<P> sequence2, Func<P, P, double>
distance, int w)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Collections.Generic.IEnumerable<P>	sequence1	The sequence1.
System.Collections.Generic.IEnumerable<P>	sequence2	The sequence2.
System.Func<P, P, System.Double>	distance	The distance.
System.Int32	w	The w.

Returns

TYPE	DESCRIPTION
System.Double	

Type Parameters

NAME	DESCRIPTION
P	

Remarks

Bases on: Abdullah Mueen, Eamonn J. Keogh: Extracting Optimal Performance from Dynamic Time Warping.KDD 2016: 2129-2130

ConstrainedDtwWikipedia<P>(IEnumerable<P>, IEnumerable<P>, Func<P, P, Double>, Int32)

Constrained DTW implementation (Wikipedia)

Declaration

```
public static double ConstrainedDtwWikipedia<P>(IEnumerable<P> sequence1, IEnumerable<P> sequence2, Func<P, P,
double> distance, int w)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Collections.Generic.IEnumerable<P>	sequence1	The sequence1.
System.Collections.Generic.IEnumerable<P>	sequence2	The sequence2.
System.Func<P, P, System.Double>	distance	The distance.
System.Int32	w	The w.

Returns

TYPE	DESCRIPTION
System.Double	

Type Parameters

NAME	DESCRIPTION
P	

Remarks

https://en.wikipedia.org/wiki/Dynamic_time_warping

ExactDtw<P>(IEnumerable<P>, IEnumerable<P>, Func<P, P, Double>)

Exact DTW implementation (Abdullah Mueen, Eamonn J. Keogh)

Declaration

```
public static double ExactDtw<P>(IEnumerable<P> sequence1, IEnumerable<P> sequence2, Func<P, P, double>
distance)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Collections.Generic.IEnumerable<P>	sequence1	The sequence1.
System.Collections.Generic.IEnumerable<P>	sequence2	The sequence2.
System.Func<P, P, System.Double>	distance	The distance.

Returns

TYPE	DESCRIPTION
System.Double	

Type Parameters

NAME	DESCRIPTION
P	

Remarks

Bases on: Abdullah Mueen, Eamonn J. Keogh: Extracting Optimal Performance from Dynamic Time Warping.KDD 2016: 2129-2130

ExactDtwWikipedia<P>(IEnumerable<P>, IEnumerable<P>, Func<P, P, Double>)

Exact DTW implementation (Wikipedia)

Declaration

```
public static double ExactDtwWikipedia<P>(IEnumerable<P> sequence1, IEnumerable<P> sequence2, Func<P, P,
double> distance)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Collections.Generic.IEnumerable<P>	sequence1	The sequence1.
System.Collections.Generic.IEnumerable<P>	sequence2	The sequence2.
System.Func<P, P, System.Double>	distance	The distance.

Returns

TYPE	DESCRIPTION
System.Double	

Type Parameters

NAME	DESCRIPTION
P	

Remarks

https://en.wikipedia.org/wiki/Dynamic_time_warping

OptimizedDtw<P>(IEnumerable<P>, IEnumerable<P>, Func<P, P, Double>, Int32, Int32)

Complex, optimized DTW calculation (Abdullah Mueen, Eamonn J. Keogh)

Declaration

```
public static double OptimizedDtw<P>(IEnumerable<P> sequence1, IEnumerable<P> sequence2, Func<P, P, double>
distance, int m = 0, int r = 0)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Collections.Generic.IEnumerable<P>	sequence1	
System.Collections.Generic.IEnumerable<P>	sequence2	
System.Func<P, P, System.Double>	distance	
System.Int32	m	
System.Int32	r	

Returns

TYPE	DESCRIPTION
System.Double	

Type Parameters

NAME	DESCRIPTION
P	

Remarks

Bases on: Abdullah Mueen, Eamonn J. Keogh: Extracting Optimal Performance from Dynamic Time Warping.KDD 2016: 2129-2130

Class HSCPThinningStep

HSCP thinning algorithm http://www.ppgia.pucpr.br/~facon/Afinamento/1987holt.pdf

Inheritance

System.Object
HSCPThinningStep

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: SigStat.Common.Algorithms

Assembly: SigStat.Common.dll

Syntax

```
public class HSCPThinningStep
```

Properties

ResultChanged

Gets whether the last Scan(Boolean[,]) call was effective.

Declaration

```
public bool? ResultChanged { get; }
```

Property Value

TYPE	DESCRIPTION
System.Nullable<System.Boolean>	

Methods

Scan(Boolean[,])

Does one step of the thinning. Call it iteratively while ResultChanged.

Declaration

```
public bool[, ] Scan(bool[, ] b)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Boolean[,]	b	Binary raster.

Returns

TYPE	DESCRIPTION
System.Boolean[,]	Thinned binary raster.

Class PatternMatching3x3

Binary 3x3 pattern matcher with rotating option.

Inheritance

System.Object
PatternMatching3x3

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: SigStat.Common.Algorithms

Assembly: SigStat.Common.dll

Syntax

```
public class PatternMatching3x3
```

Constructors

PatternMatching3x3(Nullable<Boolean>[,])

Initializes a new instance of the PatternMatching3x3 class with given pattern.

Declaration

```
public PatternMatching3x3(bool? [, ] pattern)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Nullable<System.Boolean>[,]	pattern	3x3 pattern. null: don't care.

Methods

Match(Boolean[,])

Match the 3x3 input with the 3x3 pattern.

Declaration

```
public bool Match(bool[, ] input)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Boolean[,]	input	

Returns

TYPE	DESCRIPTION
System.Boolean	True if the pattern matches.

RotMatch(Boolean[,])

Match the 3x3 input with the 3x3 pattern from all 4 directions.

Declaration

```
public bool RotMatch(bool[, ] input)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Boolean[,]	input	

Returns

TYPE	DESCRIPTION
System.Boolean	True if the pattern matches from at least one direction.

Namespace SigStat.Common.Algorithms.Classifiers

Classes

Ocjknn

One Class JKNN classifier based on: Khan, Shehroz Saeed. "Kernels for one-class nearest neighbour classification and comparison of chemical spectral data." College of Engineering and Informatics, National University of Ireland (2010). https://cs.uwaterloo.ca/~s255khan/files/Kernels_for_One-Class_Nearest_Neighbour_Classification_and_Comparison_of_Chemical_Spectral_Data-libre.pdf

Class Ocjknn

One Class JKNN classifier based on: Khan, Shehroz Saeed. "Kernels for one-class nearest neighbour classification and comparison of chemical spectral data." College of Engineering and Informatics, National University of Ireland (2010). https://cs.uwaterloo.ca/~s255khan/files/Kernels_for_One-Class_Nearest_Neighbour_Classification_and_Comparison_of_Chemical_Spectral_Data-libre.pdf

Inheritance

System.Object
Ocjknn

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: SigStat.Common.Algorithms.Classifiers

Assembly: SigStat.Common.dll

Syntax

```
public static class Ocjknn
```

Methods

Test<T>(T, IEnumerable<T>, Int32, Int32, Double, Func<T, T, Double>)

Step 1: find the `j` nearest neighbors of `testItem` in the set of `targetItems`. Step 2: for each neighbor, if (distance from test) / (average distance from `k` nearest neighbors) < `threshold` accept++ Steo 3: return accept / `j`

Declaration

```
public static double Test<T>(T testItem, IEnumerable<T> targetItems, int j, int k, double threshold, Func<T, T, double> distanceFunction)
```

Parameters

TYPE	NAME	DESCRIPTION
T	testItem	The item, that we want to classify
System.Collections.Generic.IEnumerable<T>	targetItems	Items belonging to the target class
System.Int32	j	See algorithm description for details
System.Int32	k	See algorithm description for details
System.Double	threshold	See algorithm description for details

TYPE	NAME	DESCRIPTION
System.Func<T, T, System.Double>	distanceFunction	Calculates the distance between two items of type `T`

Returns

TYPE	DESCRIPTION
System.Double	If the result is 0.5 or greater, then `testItem` should be accepted as a member of target class

Type Parameters

NAME	DESCRIPTION
T	Item type (typically a vector or a label, that `distanceFunction` can work with)

Namespace SigStat.Common.Algorithms.Distances

Classes

DtwDistance

Calculates the distance between two vector sequences using Dynamic Time Warping

EuclideanDistance

ManhattanDistance

Interfaces

IDistance<P>

An abstract base class for the calculation of the distance of two entities (points, sequences etc.)

Class DtwDistance

Calculates the distance between two vector sequences using Dynamic Time Warping

Inheritance

System.Object
DtwDistance

Implements

IDistance<System.Double[][]>

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: SigStat.Common.Algorithms.Distances

Assembly: SigStat.Common.dll

Syntax

```
public class DtwDistance : IDistance<double[][]>
```

Remarks

Based on: Abdullah Mueen, Eamonn J. Keogh: Extracting Optimal Performance from Dynamic Time Warping.KDD 2016: 2129-2130

Constructors

DtwDistance(IDistance<Double[]>)

Initializes a new instance of the DtwDistance class with default settings

Declaration

```
public DtwDistance(IDistance<double[]> localDistance = null)
```

Parameters

TYPE	NAME	DESCRIPTION
IDistance<System.Double[]>	localDistance	The distance function used to calculate the distance between two individual points of the squences. Set the parameter to 'null' to use the default EuclideanDistance

Properties

LocalDistance

The local distance function to use, when calculating the distance between two sueqence-points. Default is EuclideanDistance

Declaration

```
public IDistance<double[]> LocalDistance { get; set; }
```

Property Value

TYPE	DESCRIPTION
IDistance<System.Double[]>	

Methods

Calculate(Double[][], Double[][])

Declaration

```
public double Calculate(double[][] p1, double[][] p2)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Double[][]	p1	
System.Double[][]	p2	

Returns

TYPE	DESCRIPTION
System.Double	

Implements

IDistance<P>

Class EuclideanDistance

Inheritance

System.Object
EuclideanDistance

Implements

IDistance<System.Double[]>

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: SigStat.Common.Algorithms.Distances

Assembly: SigStat.Common.dll

Syntax

```
public class EuclideanDistance : IDistance<double[]>
```

Methods

Calculate(Double[], Double[])

Gets the Euclidean distance between two points.

Declaration

```
public double Calculate(double[] x, double[] y)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Double[]	x	A point in space.
System.Double[]	y	A point in space.

Returns

TYPE	DESCRIPTION
System.Double	The Euclidean distance between x and y.

Implements

IDistance<P>

Interface IDistance<P>

An abstract base class for the calculation of the distance of two entities (points, sequences etc.)

Namespace: SigStat.Common.Algorithms.Distances

Assembly: SigStat.Common.dll

Syntax

```
public interface IDistance<in P>
```

Type Parameters

NAME	DESCRIPTION
P	Entity type

Methods

Calculate(P, P)

Calculates the distance between the two parameters

Declaration

```
double Calculate(P p1, P p2)
```

Parameters

TYPE	NAME	DESCRIPTION
P	p1	Firs parameter
P	p2	Second parameter

Returns

TYPE	DESCRIPTION
System.Double	

Class ManhattanDistance

Inheritance

System.Object
ManhattanDistance

Implements

IDistance<System.Double[]>

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: SigStat.Common.Algorithms.Distances

Assembly: SigStat.Common.dll

Syntax

```
public class ManhattanDistance : IDistance<double[]>
```

Methods

Calculate(Double[], Double[])

Gets the Manhattan distance between two points.

Declaration

```
public double Calculate(double[] x, double[] y)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Double[]	x	A point in space.
System.Double[]	y	A point in space.

Returns

TYPE	DESCRIPTION
System.Double	The Manhattan distance between x and y.

Implements

IDistance<P>

Namespace SigStat.Common.Framework.Samplers

Classes

EvenNSampler

Selects the first N signatures with even index for training

FirstNSampler

Selects the first N signatures for training

LastNSampler

Selects the first N signatures for training

OddNSampler

Selects the first N signatures with odd index for training

TestingSampler

Testing sampler for signer dependent sampling frequency verification system

UniversalSampler

Selects a given number of signatures for training and testing

Class EvenNSampler

Selects the first N signatures with even index for training

Inheritance

System.Object
Sampler
EvenNSampler

Inherited Members

Sampler.TrainingFilter
Sampler.GenuineTestFilter
Sampler.ForgeryTestFilter
Sampler.SampleReferences(List<Signature>)
Sampler.SampleGenuineTests(List<Signature>)
Sampler.SampleForgeryTests(List<Signature>)
System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: SigStat.Common.Framework.Samplers

Assembly: SigStat.Common.dll

Syntax

```
public class EvenNSampler : Sampler
```

Constructors

EvenNSampler(Int32)

Constructor

Declaration

```
public EvenNSampler(int n = 10)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	n	count of signatures used for training

Properties

N

Count of signatures used for training

Declaration

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

Property Value

TYPE	DESCRIPTION
System.Int32	

Class FirstNSampler

Selects the first N signatures for training

Inheritance

System.Object
Sampler
FirstNSampler

Inherited Members

Sampler.TrainingFilter
Sampler.GenuineTestFilter
Sampler.ForgeryTestFilter
Sampler.SampleReferences(List<Signature>)
Sampler.SampleGenuineTests(List<Signature>)
Sampler.SampleForgeryTests(List<Signature>)
System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: SigStat.Common.Framework.Samplers

Assembly: SigStat.Common.dll

Syntax

```
public class FirstNSampler : Sampler
```

Constructors

FirstNSampler(Int32)

Constructor

Declaration

```
public FirstNSampler(int n = 10)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	n	count of signatures used for training

Properties

N

Count of signatures used for training

Declaration

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

Property Value

TYPE	DESCRIPTION
System.Int32	

Class LastNSampler

Selects the first N signatures for training

Inheritance

System.Object

Sampler

LastNSampler

Inherited Members

Sampler.TrainingFilter

Sampler.GenuineTestFilter

Sampler.ForgeryTestFilter

Sampler.SampleReferences(List<Signature>)

Sampler.SampleGenuineTests(List<Signature>)

Sampler.SampleForgeryTests(List<Signature>)

System.Object.Equals(System.Object)

System.Object.Equals(System.Object, System.Object)

System.Object.GetHashCode()

System.Object.GetType()

System.Object.MemberwiseClone()

System.Object.ReferenceEquals(System.Object, System.Object)

System.Object.ToString()

Namespace: SigStat.Common.Framework.Samplers

Assembly: SigStat.Common.dll

Syntax

```
public class LastNSampler : Sampler
```

Constructors

LastNSampler(Int32)

Constructor

Declaration

```
public LastNSampler(int n = 10)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	n	Count of signatures used for training

Properties

N

Count of signatures used for training

Declaration

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

Property Value

TYPE	DESCRIPTION
System.Int32	

Class OddNSampler

Selects the first N signatures with odd index for training

Inheritance

System.Object
Sampler
OddNSampler

Inherited Members

Sampler.TrainingFilter
Sampler.GenuineTestFilter
Sampler.ForgeryTestFilter
Sampler.SampleReferences(List<Signature>)
Sampler.SampleGenuineTests(List<Signature>)
Sampler.SampleForgeryTests(List<Signature>)
System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: SigStat.Common.Framework.Samplers

Assembly: SigStat.Common.dll

Syntax

```
public class OddNSampler : Sampler
```

Constructors

OddNSampler(Int32)

Constructor

Declaration

```
public OddNSampler(int n = 10)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	n	Count of signatures used for training

Properties

N

Count of signatures used for training

Declaration

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

Property Value

TYPE	DESCRIPTION
System.Int32	

Class TestingSampler

Testing sampler for signer dependent sampling frequency verification system

Inheritance

System.Object

Sampler

TestingSampler

Inherited Members

Sampler.TrainingFilter

Sampler.GenuineTestFilter

Sampler.ForgeryTestFilter

Sampler.SampleReferences(List<Signature>)

Sampler.SampleGenuineTests(List<Signature>)

Sampler.SampleForgeryTests(List<Signature>)

System.Object.Equals(System.Object)

System.Object.Equals(System.Object, System.Object)

System.Object.GetHashCode()

System.Object.GetType()

System.Object.MemberwiseClone()

System.Object.ReferenceEquals(System.Object, System.Object)

System.Object.ToString()

Namespace: SigStat.Common.Framework.Samplers

Assembly: SigStat.Common.dll

Syntax

```
public class TestingSampler : Sampler
```

Constructors

TestingSampler(Int32)

Constructor

Declaration

```
public TestingSampler(int n = 5)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	n	count of signatures used for training

Properties

N

Count of signatures used for training

Declaration

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

Property Value

TYPE	DESCRIPTION
System.Int32	

Class UniversalSampler

Selects a given number of signatures for training and testing

Inheritance

System.Object
Sampler
UniversalSampler

Inherited Members

Sampler.TrainingFilter
Sampler.GenuineTestFilter
Sampler.ForgeryTestFilter
Sampler.SampleReferences(List<Signature>)
Sampler.SampleGenuineTests(List<Signature>)
Sampler.SampleForgeryTests(List<Signature>)
System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: SigStat.Common.Framework.Samplers

Assembly: SigStat.Common.dll

Syntax

```
public class UniversalSampler : Sampler
```

Constructors

UniversalSampler(Int32, Int32)

Constructor

Declaration

```
public UniversalSampler(int trainingCount, int testCount)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	trainingCount	Count of signatures to use for training
System.Int32	testCount	Count of signatures to use for testing

Properties

TestCount

Count of signatures to use for testing

Declaration

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

Property Value

TYPE	DESCRIPTION
System.Int32	

TrainingCount

Count of signatures to use for training

Declaration

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

Property Value

TYPE	DESCRIPTION
System.Int32	

Namespace SigStat.Common.Helpers

Classes

BenchmarkConfig

Represents a configuration for a benchmark

DataCleaningHelper

Helper class for cleaning online signature data in loaders

ExcelHelper

Extension methods for common EPPlus tasks

FeatureDescriptorJsonConverter

Custom serializer for FeatureDescriptor objects

FeatureDescriptorTJsonConverter

Custom serializer for FeatureDescriptor<T> objects

HierarchyElement

Hierarchical structure to store object

ProgressHelper

A helper class for tracking progress of an operation.

SerializationHelper

Json serialization and deserialization using the custom resolver VerifierResolver

SignerStatisticsHelper

Interfaces

IProgress

Enables progress tracking by expsoing the Progress property and the ProgressChanged event.

Class BenchmarkConfig

Represents a configuration for a benchmark

Inheritance

System.Object

BenchmarkConfig

Inherited Members

System.Object.Equals(System.Object)

System.Object.Equals(System.Object, System.Object)

System.Object.GetHashCode()

System.Object.GetType()

System.Object.MemberwiseClone()

System.Object.ReferenceEquals(System.Object, System.Object)

System.Object.ToString()

Namespace: SigStat.Common.Helpers

Assembly: SigStat.Common.dll

Syntax

```
[Obsolete("This class was created for a specific benchmark and will be removed in the future")]
public class BenchmarkConfig
```

Constructors

BenchmarkConfig()

Declaration

```
public BenchmarkConfig()
```

BenchmarkConfig(BenchmarkConfig)

Declaration

```
public BenchmarkConfig(BenchmarkConfig c)
```

Parameters

TYPE	NAME	DESCRIPTION
BenchmarkConfig	c	

Properties

Classifier

Declaration

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

Property Value

TYPE	DESCRIPTION
System.String	

Database

Declaration

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

Property Value

TYPE	DESCRIPTION
System.String	

Distance

Declaration

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

Property Value

TYPE	DESCRIPTION
System.String	

Features

Declaration

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

Property Value

TYPE	DESCRIPTION
System.String	

Interpolation

Declaration

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

Property Value

TYPE	DESCRIPTION
System.String	

ResamplingParam

Declaration

```
public double ResamplingParam { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Double	

ResamplingType_Filter

Declaration

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

Property Value

TYPE	DESCRIPTION
System.String	

Rotation

Declaration

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

Property Value

TYPE	DESCRIPTION
System.Boolean	

Sampling

Declaration

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

Property Value

TYPE	DESCRIPTION
System.String	

Translation_Scaling

Declaration

```
public (string Translation, string Scaling) Translation_Scaling { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.ValueTuple<System.String, System.String>	

Methods

FromJsonFile(String)

Helper

Declaration

```
public BenchmarkConfig FromJsonFile(string path)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	path	

Returns

TYPE	DESCRIPTION
BenchmarkConfig	

FromJsonString(String)

helper

Declaration

```
public static BenchmarkConfig FromJsonString(string jsonString)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	jsonString	

Returns

TYPE	DESCRIPTION
BenchmarkConfig	

GenerateConfigurations()

Helper

Declaration

```
public static List<BenchmarkConfig> GenerateConfigurations()
```

Returns

TYPE	DESCRIPTION
System.Collections.Generic.List<BenchmarkConfig>	

ToJsonString()

Helper

Declaration

```
public string ToJsonString()
```

Returns

TYPE	DESCRIPTION
System.String	

ToKeyValuePairs()

Helper

Declaration

```
public IEnumerable<KeyValuePair<string, string>> ToKeyValuePairs()
```

Returns

TYPE	DESCRIPTION
System.Collections.Generic.IEnumerable<System.Collections.Generic.KeyValuePair<System.String, System.String>>	

ToShortString()

Helper

Declaration

```
public string ToShortString()
```

Returns

TYPE	DESCRIPTION
System.String	

Class DataCleaningHelper

Helper class for cleaning online signature data in loaders

Inheritance

System.Object
DataCleaningHelper

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: SigStat.Common.Helpers

Assembly: SigStat.Common.dll

Syntax

```
public static class DataCleaningHelper
```

Methods

GeneratePointTypeValuesFromPressure(Double[])

Generate point type values of an online signature based on its pressure values (zero pressure points are required)

Declaration

```
public static double[] GeneratePointTypeValuesFromPressure(double[] pressure)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Double[]	pressure	The preussure values of an online signature

Returns

TYPE	DESCRIPTION
System.Double[]	

InitializeTimestamps(Signature, Double)

Initialize timestamps of an online signature which does not have captured timestamps

Declaration

```
public static void InitializeTimestamps(Signature signature, double unitTimeSlot)
```

Parameters

TYPE	NAME	DESCRIPTION

TYPE	NAME	DESCRIPTION
Signature	signature	The online signature which's timestamps are initialized
System.Double	unitTimeSlot	The unit time slot between two points of the signature

Class ExcelHelper

Extension methods for common EPPlus tasks

Inheritance

System.Object
ExcelHelper

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: SigStat.Common.Helpers

Assembly: SigStat.Common.dll

Syntax

```
public static class ExcelHelper
```

Methods

FormatAsTable(ExcelRange, ExcelColor, Boolean, Boolean)

Format cells in the range into a table

Declaration

```
public static void FormatAsTable(this ExcelRange range, ExcelColor color = ExcelColor.Primary, bool
showColumnHeader = true, bool showRowHeader = true)
```

Parameters

TYPE	NAME	DESCRIPTION
OfficeOpenXml.ExcelRange	range	The table's cells
ExcelColor	color	Color palette of the table
System.Boolean	showColumnHeader	Defines if the table has column header
System.Boolean	showRowHeader	Defines if the table has row header

FormatAsTableWithTitle(ExcelRange, String, ExcelColor, Boolean, Boolean)

Format cells in the range into a table with possible title

Declaration

```
public static int FormatAsTableWithTitle(this ExcelRange range, string title, ExcelColor color =
ExcelColor.Primary, bool showColumnHeader = true, bool showRowHeader = true)
```

Parameters

TYPE	NAME	DESCRIPTION
OfficeOpenXml.ExcelRange	range	The table's cells
System.String	title	The table's title, if null, the table won't have title
ExcelColor	color	Color palette of the table
System.Boolean	showColumnHeader	Defines if the table has column header
System.Boolean	showRowHeader	Defines if the table has row header

Returns

TYPE	DESCRIPTION
System.Int32	

InsertColumnChart(ExcelWorksheet, ExcelRange, Int32, Int32, String, String, String, ExcelRange, Int32, Int32, String)

Draws a column chart for the given data

Declaration

```
public static void InsertColumnChart(this ExcelWorksheet ws, ExcelRange range, int row, int col, string name,
string xLabel = null, string yLabel = null, ExcelRange serieLabels = null, int width = -1, int height = -1,
string title = null)
```

Parameters

TYPE	NAME	DESCRIPTION
OfficeOpenXml.ExcelWorksheet	ws	Worksheet in wich the graph is inserted
OfficeOpenXml.ExcelRange	range	Range containing the data (first row for x axis other rows for series)
System.Int32	row	The graph inserted starts at this row
System.Int32	col	The graph inserted starts at this column
System.String	name	Id and default title of the graph

TYPE	NAME	DESCRIPTION
System.String	xLabel	Label for x axis of the graph
System.String	yLabel	Label for y axis of the graph
OfficeOpenXml.ExcelRange	serieLabels	If the graph hase more than one series, each can be named separately
System.Int32	width	Graph's width in px
System.Int32	height	Graph's height in px
System.String	title	Title of the graph if the defauolt name has to be overwritten

InsertDictionary<TKey, TValue>(ExcelWorksheet, Int32, Int32, IEnumerable<KeyValuePair<TKey, TValue>>, String, ExcelColor, String)

Insert table from key-value pairs

Declaration

```
public static ExcelRange InsertDictionary<TKey, TValue>(this ExcelWorksheet ws, int row, int col,
IEnumerable<KeyValuePair<TKey, TValue>> data, string title = null, ExcelColor color = ExcelColor.Primary,
string Name = null)
```

Parameters

TYPE	NAME	DESCRIPTION
OfficeOpenXml.ExcelWorksheet	ws	Worksheet in wich the table is created
System.Int32	row	Starting row of the table
System.Int32	col	Starting column of the table
System.Collections.Generic.IEnumerable<System.Collections.Generic.KeyValuePair<TKey, TValue>>	data	IEnumerable of key-value pairs in wich the data to insert is stored
System.String	title	The table's title
ExcelColor	color	The table's color

TYPE	NAME	DESCRIPTION
System.String	Name	If given, creates a named range, with this name

Returns

TYPE	DESCRIPTION
OfficeOpenXml.ExcelRange	Range of the inserted data

Type Parameters

NAME	DESCRIPTION
TKey	
TValue	

InsertHierarchicalList(ExcelWorksheet, Int32, Int32, HierarchyElement, String, ExcelColor)

Insert a hierarchical list in tree style into the worksheet

Declaration

```
public static void InsertHierarchicalList(this ExcelWorksheet ws, int row, int col, HierarchyElement root,
string title = null, ExcelColor color = ExcelColor.Primary)
```

Parameters

TYPE	NAME	DESCRIPTION
OfficeOpenXml.ExcelWorksheet	ws	Worksheet in wich the list is inserted
System.Int32	row	Starting row of the list
System.Int32	col	Starting column of the list
HierarchyElement	root	Root element of the list
System.String	title	Title of the list
ExcelColor	color	color of the list

InsertLegend(ExcelRange, String, String, ExcelColor)

Insert legend

Declaration

```
public static void InsertLegend(this ExcelRange range, string text, string title = null, ExcelColor color =
ExcelColor.Info)
```

Parameters

TYPE	NAME	DESCRIPTION
OfficeOpenXml.ExcelRange	range	Range of the legend
System.String	text	Text of the legend
System.String	title	Title of the legend (can be null)
ExcelColor	color	Color of the legend

InsertLineChart(ExcelWorksheet, ExcelRange, Int32, Int32, String, String, String, ExcelRange, Int32, Int32, String)

Draws a line chart for the given data

Declaration

```
public static void InsertLineChart(this ExcelWorksheet ws, ExcelRange range, int row, int col, string name,
string xLabel = null, string yLabel = null, ExcelRange SerieLabels = null, int width = -1, int height = -1,
string title = null)
```

Parameters

TYPE	NAME	DESCRIPTION
OfficeOpenXml.ExcelWorksheet	ws	Worksheet in wich the graph is inserted
OfficeOpenXml.ExcelRange	range	Range containing the data (first row for x axis other rows for series)
System.Int32	row	The graph inserted starts at this row
System.Int32	col	The graph inserted starts at this column
System.String	name	Id and default title of the graph
System.String	xLabel	Label for x axis of the graph

TYPE	NAME	DESCRIPTION
System.String	yLabel	Label for y axis of the graph
OfficeOpenXml.ExcelRange	SerieLabels	Label of the series
System.Int32	width	Graph's width in px
System.Int32	height	Graph's height in px
System.String	title	Title of the graph if the defauolt name has to be overwritten

InsertLink(ExcelRange, String)

Creates a link to given sheet

Declaration

```
public static void InsertLink(this ExcelRange range, string sheet)
```

Parameters

TYPE	NAME	DESCRIPTION
OfficeOpenXml.ExcelRange	range	Cells to place the link in
System.String	sheet	Destination sheet's name

InsertLink(ExcelRange, String, String)

Creates a link to selected cells in given sheet

Declaration

```
public static void InsertLink(this ExcelRange range, string sheet, string cells)
```

Parameters

TYPE	NAME	DESCRIPTION
OfficeOpenXml.ExcelRange	range	Cells to place the link in
System.String	sheet	Destination sheet's name

TYPE	NAME	DESCRIPTION
System.String	cells	Destination cells' address

InsertTable(ExcelWorksheet, Int32, Int32, IEnumerable<IEnumerable<Object>>, IEnumerable<String>, String, ExcelColor, String)

Insert a table filled with data from an IEnumerable

Declaration

```
public static ExcelRange InsertTable(this ExcelWorksheet ws, int row, int col,
IEnumerable<IEnumerable<object>> data, IEnumerable<string> headers, string title = null, ExcelColor color =
ExcelColor.Primary, string name = null)
```

Parameters

TYPE	NAME	DESCRIPTION
OfficeOpenXml.ExcelWorksheet	ws	Worksheet in wich the table is created
System.Int32	row	Starting row of the table
System.Int32	col	Starting column of the table
System.Collections.Generic.IEnumerable<System.Collections.Generic.IEnumerable<System.Object>>	data	IEnumerable in wich the data to insert is stored
System.Collections.Generic.IEnumerable<System.String>	headers	Defines if the table has header
System.String	title	The table's title
ExcelColor	color	The table's color
System.String	name	If given, creates a named range, with this name

Returns

TYPE	DESCRIPTION

TYPE	DESCRIPTION
OfficeOpenXml.ExcelRange	Range of the inserted data

InsertTable(ExcelWorksheet, Int32, Int32, Double[,], String, ExcelColor, Boolean, Boolean, String)

Insert table filled with data from a 2D array

Declaration

```
public static ExcelRange InsertTable(this ExcelWorksheet ws, int row, int col, double[, ] data, string title =
null, ExcelColor color = ExcelColor.Primary, bool hasRowHeader = true, bool hasColumnHeader = true, string
name = null)
```

Parameters

TYPE	NAME	DESCRIPTION
OfficeOpenXml.ExcelWorksheet	ws	Worksheet in wich the table is created
System.Int32	row	Starting row of the table
System.Int32	col	Starting column of the table
System.Double[,]	data	2D array in wich the data to insert is stored (double values)
System.String	title	The table's title
ExcelColor	color	The table's color
System.Boolean	hasRowHeader	Defines if the table has row header
System.Boolean	hasColumnHeader	Defines if the table has column header
System.String	name	If given, creates a named range, with this name

Returns

TYPE	DESCRIPTION
OfficeOpenXml.ExcelRange	Range of the inserted data

InsertTable(ExcelWorksheet, Int32, Int32, Object[,], String, ExcelColor, Boolean, Boolean, String)

Insert table filled with data from a 2D array

Declaration

```
public static ExcelRange InsertTable(this ExcelWorksheet ws, int row, int col, object[, ] data, string title =
null, ExcelColor color = ExcelColor.Primary, bool hasRowHeader = true, bool hasColumnHeader = true, string
name = null)
```

Parameters

TYPE	NAME	DESCRIPTION
OfficeOpenXml.ExcelWorksheet	ws	Worksheet in wich the table is created
System.Int32	row	Starting row of the table
System.Int32	col	Starting column of the table
System.Object[,]	data	2D array in wich the data to insert is stored
System.String	title	The table's title
ExcelColor	color	The table's color
System.Boolean	hasRowHeader	Defines if the table has row header
System.Boolean	hasColumnHeader	Defines if the table has column header
System.String	name	If given, creates a named range, with this name

Returns

TYPE	DESCRIPTION
OfficeOpenXml.ExcelRange	Range of the inserted data

InsertTable<T>(ExcelWorksheet, Int32, Int32, IEnumerable<T>, String, ExcelColor, Boolean, String)

Insert a table filled with data from an IEnumerable

Declaration

```
public static ExcelRange InsertTable<T>(this ExcelWorksheet ws, int row, int col, IEnumerable<T> data, string
title = null, ExcelColor color = ExcelColor.Primary, bool showHeader = true, string Name = null)
```

Parameters

TYPE	NAME	DESCRIPTION
OfficeOpenXml.ExcelWorksheet	ws	Worksheet in wich the table is created
System.Int32	row	Starting row of the table
System.Int32	col	Starting column of the table
System.Collections.Generic.IEnumerable<T>	data	IEnumerable in wich the data to insert is stored
System.String	title	The table's title
ExcelColor	color	The table's color
System.Boolean	showHeader	Defines if the table has header
System.String	Name	If given, creates a named range, with this name

Returns

TYPE	DESCRIPTION
OfficeOpenXml.ExcelRange	Range of the inserted data

Type Parameters

NAME	DESCRIPTION
T	Type of inserted objects

InsertText(ExcelWorksheet, Int32, Int32, String, TextLevel)

Inserts text into the defined cell, and format to match text level

Declaration

```
public static void InsertText(this ExcelWorksheet ws, int row, int col, string text, TextLevel level =
TextLevel.Normal)
```

Parameters

TYPE	NAME	DESCRIPTION
OfficeOpenXml.ExcelWorksheet	ws	Worksheet in wich the text is inserted
System.Int32	row	Row of the cell
System.Int32	col	Column of the cell
System.String	text	Text to insert
TextLevel	level	Level of text

Merge(ExcelRangeBase)

Merge all cells into one in the range.

Declaration

```
public static void Merge(this ExcelRangeBase range)
```

Parameters

TYPE	NAME	DESCRIPTION
OfficeOpenXml.ExcelRangeBase	range	Cells to merge

Class FeatureDescriptorJsonConverter

Custom serializer for FeatureDescriptor objects

Inheritance

System.Object
Newtonsoft.Json.JsonConverter
FeatureDescriptorJsonConverter

Inherited Members

Newtonsoft.Json.JsonConverter.CanRead
Newtonsoft.Json.JsonConverter.CanWrite
System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: SigStat.Common.Helpers

Assembly: SigStat.Common.dll

Syntax

```
public class FeatureDescriptorJsonConverter : JsonConverter
```

Methods

CanConvert(Type)

Tells if the current object is of the correct type

Declaration

```
public override bool CanConvert(Type objectType)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Type	objectType	The type of the object

Returns

TYPE	DESCRIPTION
System.Boolean	If the object can be converted or not

Overrides

Newtonsoft.Json.JsonConverter.CanConvert(System.Type)

ReadJson(JsonReader, Type, Object, JsonSerializer)

Overwrite of the Newtonsoft.Json.JsonConverter method Deserializes the FeatureDescriptor json created by the this class

Declaration

```
public override object ReadJson(JsonReader reader, Type objectType, object existingValue, JsonSerializer
serializer)
```

Parameters

TYPE	NAME	DESCRIPTION
Newtonsoft.Json.JsonReader	reader	
System.Type	objectType	
System.Object	existingValue	
Newtonsoft.Json.JsonSerializer	serializer	

Returns

TYPE	DESCRIPTION
System.Object	

Overrides

Newtonsoft.Json.JsonConverter.ReadJson(Newtonsoft.Json.JsonReader, System.Type, System.Object, Newtonsoft.Json.JsonSerializer)

WriteJson(JsonWriter, Object, JsonSerializer)

Overwrite of the Newtonsoft.Json.JsonConverter method Serializes the FeatureDescriptor to json with type depending on if it was serialized earlier or not

Declaration

```
public override void WriteJson(JsonWriter writer, object value, JsonSerializer serializer)
```

Parameters

TYPE	NAME	DESCRIPTION
Newtonsoft.Json.JsonWriter	writer	
System.Object	value	
Newtonsoft.Json.JsonSerializer	serializer	

Overrides

Newtonsoft.Json.JsonConverter.WriteJson(Newtonsoft.Json.JsonWriter, System.Object, Newtonsoft.Json.JsonSerializer)

Class FeatureDescriptorTJsonConverter

Custom serializer for FeatureDescriptor<T> objects

Inheritance

System.Object
Newtonsoft.Json.JsonConverter
FeatureDescriptorTJsonConverter

Inherited Members

Newtonsoft.Json.JsonConverter.CanRead
Newtonsoft.Json.JsonConverter.CanWrite
System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: SigStat.Common.Helpers

Assembly: SigStat.Common.dll

Syntax

```
public class FeatureDescriptorTJsonConverter : JsonConverter
```

Methods

CanConvert(Type)

Tells if the current object is of the correct type

Declaration

```
public override bool CanConvert(Type objectType)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Type	objectType	The type of the object

Returns

TYPE	DESCRIPTION
System.Boolean	If the object can be converted or not

Overrides

Newtonsoft.Json.JsonConverter.CanConvert(System.Type)

ReadJson(JsonReader, Type, Object, JsonSerializer)

Overwrite of the Newtonsoft.Json.JsonConverter method Deserializes the FeatureDescriptor<T> json created by the this class

Declaration

```
public override object ReadJson(JsonReader reader, Type objectType, object existingValue, JsonSerializer
serializer)
```

Parameters

TYPE	NAME	DESCRIPTION
Newtonsoft.Json.JsonReader	reader	
System.Type	objectType	
System.Object	existingValue	
Newtonsoft.Json.JsonSerializer	serializer	

Returns

TYPE	DESCRIPTION
System.Object	

Overrides

Newtonsoft.Json.JsonConverter.ReadJson(Newtonsoft.Json.JsonReader, System.Type, System.Object, Newtonsoft.Json.JsonSerializer)

WriteJson(JsonWriter, Object, JsonSerializer)

Overwrite of the Newtonsoft.Json.JsonConverter method Serializes the FeatureDescriptor<T> to json with type depending on if it was serialized earlier or not

Declaration

```
public override void WriteJson(JsonWriter writer, object value, JsonSerializer serializer)
```

Parameters

TYPE	NAME	DESCRIPTION
Newtonsoft.Json.JsonWriter	writer	
System.Object	value	
Newtonsoft.Json.JsonSerializer	serializer	

Overrides

Newtonsoft.Json.JsonConverter.WriteJson(Newtonsoft.Json.JsonWriter, System.Object, Newtonsoft.Json.JsonSerializer)

Class HierarchyElement

Hierarchical structure to store object

Inheritance

System.Object
HierarchyElement

Implements

System.Collections.Generic.IEnumerable<HierarchyElement>
System.Collections.IEnumerable

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)

Namespace: SigStat.Common.Helpers

Assembly: SigStat.Common.dll

Syntax

```
public class HierarchyElement : IEnumerable<HierarchyElement>, IEnumerable
```

Constructors

HierarchyElement()

Create an emty element

Declaration

```
public HierarchyElement()
```

HierarchyElement(Object)

Create a new element with content

Declaration

```
public HierarchyElement(object Content)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Object	Content	Content of the new element

Properties

Children

Gets the children.

Declaration

```
public List<HierarchyElement> Children { get; }
```

Property Value

TYPE	DESCRIPTION
System.Collections.Generic.List<HierarchyElement>	

Content

Gets or sets the content.

Declaration

```
public object Content { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Object	

Methods

Add(HierarchyElement)

Adds the specified element as a child

Declaration

```
public void Add(HierarchyElement child)
```

Parameters

TYPE	NAME	DESCRIPTION
HierarchyElement	child	

GetCount()

Returns number of elements under this node and itself

Declaration

```
public int GetCount()
```

Returns

TYPE	DESCRIPTION
System.Int32	

GetDepth()

Return the hierarchy's depth from this node

Declaration

```
public int GetDepth()
```

Returns

TYPE	DESCRIPTION
System.Int32	

GetEnumerator()

Returns an enumerator that iterates through the collection.

Declaration

```
public IEnumerator<HierarchyElement> GetEnumerator()
```

Returns

TYPE	DESCRIPTION
System.Collections.Generic.IEnumerator<HierarchyElement>	

ToString()

Converts to string.

Declaration

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

Returns

TYPE	DESCRIPTION
System.String	A System.String that represents this instance.

Overrides

System.Object.ToString()

Explicit Interface Implementations

IEnumerable.GetEnumerator()

Declaration

```
IEnumerator IEnumerable.GetEnumerator()
```

Returns

TYPE	DESCRIPTION
System.Collections.IEnumerator	

Implements

System.Collections.Generic.IEnumerable<T>
System.Collections.IEnumerable

Extension Methods

ArrayExtension.ToArrays<T>(IEnumerable<T>, Int32)

Interface IProgress

Enables progress tracking by expsoing the Progress property and the ProgressChanged event.

Namespace: SigStat.Common.Helpers

Assembly: SigStat.Common.dll

Syntax

```
public interface IProgress
```

Properties

Progress

Gets the current progress in percentage.

Declaration

```
int Progress { get; }
```

Property Value

TYPE	DESCRIPTION
System.Int32	

Events

ProgressChanged

Invoked whenever the Progress property is changed.

Declaration

```
event EventHandler<int> ProgressChanged
```

Event Type

TYPE	DESCRIPTION
System.EventHandler<System.Int32>	

Class ProgressHelper

A helper class for tracking progress of an operation.

Inheritance

System.Object
ProgressHelper

Implements

System.IDisposable

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: SigStat.Common.Helpers

Assembly: SigStat.Common.dll

Syntax

```
public class ProgressHelper : IDisposable
```

Constructors

ProgressHelper()

Initializes an instance of ProgressHelper

Declaration

```
protected ProgressHelper()
```

Properties

Elapsed

Gets the total elapsed time measured by the current instance.

Declaration

```
public TimeSpan Elapsed { get; }
```

Property Value

TYPE	DESCRIPTION
System.TimeSpan	

Eta

Gets the estimated time of completion assuming linear progress.

Declaration

```
public DateTime Eta { get; }
```

Property Value

TYPE	DESCRIPTION
System.DateTime	

Maximum

The total number of individual items to be processed.

Declaration

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

Property Value

TYPE	DESCRIPTION
System.Int32	

Remaining

Gets the estimated remaining time till completion assuming linear progress.

Declaration

```
public TimeSpan Remaining { get; }
```

Property Value

TYPE	DESCRIPTION
System.TimeSpan	

ReportIntervallSeconds

If larger than 0, ReportProgress event will be executed periodically after ReportIntervallSeconds when the Value property changes.

Declaration

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

Property Value

TYPE	DESCRIPTION
System.Int32	

Value

The actual number of processed items.

Declaration

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

Property Value

TYPE	DESCRIPTION
System.Int32	

TYPE	DESCRIPTION

Methods

Dispose()

Declaration

```
public void Dispose()
```

StartNew(Int32, Int32, Action<ProgressHelper>)

Initializes an instance of ProgressHelper with the given parameters. Make sure to manually set the Value property during operation.

Declaration

```
public static ProgressHelper StartNew(int maximum, int reportIntervallSeconds = 0, Action<ProgressHelper>
reportProgress = null)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	maximum	The total number of individual items to be processed.
System.Int32	reportIntervallSeconds	If larger than 0, ReportProgress event will be executed periodically after ReportIntervallSeconds when the Value property changes.
System.Action<ProgressHelper>	reportProgress	

Returns

TYPE	DESCRIPTION
ProgressHelper	

Events

ReportProgress

Event will be executed periodically after ReportIntervallSeconds when the Value property changes.

Declaration

```
public event Action<ProgressHelper> ReportProgress
```

Event Type

TYPE	DESCRIPTION
System.Action<ProgressHelper>	

Remarks

If ReportIntervallSeconds is set to 0, this event will be suppressed

Implements

System.IDisposable

Class SerializationHelper

Json serialization and deserialization using the custom resolver VerifierResolver

Inheritance

System.Object
SerializationHelper

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: SigStat.Common.Helpers

Assembly: SigStat.Common.dll

Syntax

```
public class SerializationHelper
```

Methods

Deserialize<T>(String)

Constructs object from strings that were serialized previously

Declaration

```
public static T Deserialize<T>(string s)

    where T : class
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	s	The serialized string

Returns

TYPE	DESCRIPTION
T	The object that was serialized

Type Parameters

NAME	DESCRIPTION
T	A type which has a public parameterless constructor

DeserializeFromFile<T>(String)

Constructs object from file given by a path

Declaration

```
public static T DeserializeFromFile<T>(string path)

    where T : class
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	path	Relative path to the file

Returns

TYPE	DESCRIPTION
T	The object that was serialized to the file

Type Parameters

NAME	DESCRIPTION
T	A type which has a public parameterless constructor

GetSettings(Boolean)

Settings used for the serialization methods

Declaration

```
public static JsonSerializerSettings GetSettings(bool compactFeatures = false)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Boolean	compactFeatures	

Returns

TYPE	DESCRIPTION
Newtonsoft.Json.JsonSerializerSettings	A new settings object

JsonSerialize<T>(T, Boolean)

Creates json string from object

Declaration

```
public static string JsonSerialize<T>(T o, bool compactFeatures = false)
```

Parameters

TYPE	NAME	DESCRIPTION
T	o	The object
System.Boolean	compactFeatures	

Returns

TYPE	DESCRIPTION
System.String	The json string constructed from the object

Type Parameters

NAME	DESCRIPTION
T	The type of the object

JsonSerializeToFile<T>(T, String, Boolean)

Writes object to file to the given by path in json format

Declaration

```
public static void JsonSerializeToFile<T>(T o, string path, bool compactFeatures = false)
```

Parameters

TYPE	NAME	DESCRIPTION
T	o	The object
System.String	path	Relative path
System.Boolean	compactFeatures	

Type Parameters

NAME	DESCRIPTION
T	The type of the object

Class SignerStatisticsHelper

Inheritance

System.Object
SignerStatisticsHelper

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: SigStat.Common.Helpers

Assembly: SigStat.Common.dll

Syntax

```
public static class SignerStatisticsHelper
```

Methods

GetHeightAvg(Signer)

return signer height average

Declaration

```
public static double GetHeightAvg(this Signer signer)
```

Parameters

TYPE	NAME	DESCRIPTION
Signer	signer	

Returns

TYPE	DESCRIPTION
System.Double	

GetLengthAverage(Signer)

Return the average od signatures points number

Declaration

```
public static double GetLengthAverage(this Signer signer)
```

Parameters

TYPE	NAME	DESCRIPTION
Signer	signer	

Returns

TYPE	DESCRIPTION
System.Double	

GetMaxSignaturePoints(Signer)

return the min signature points number of a signer

Declaration

```
public static double GetMaxSignaturePoints(this Signer signer)
```

Parameters

TYPE	NAME	DESCRIPTION
Signer	signer	

Returns

TYPE	DESCRIPTION
System.Double	

GetMinSignaturePoints(Signer)

return the min signature points number of a signer

Declaration

```
public static double GetMinSignaturePoints(this Signer signer)
```

Parameters

TYPE	NAME	DESCRIPTION
Signer	signer	

Returns

TYPE	DESCRIPTION
System.Double	

GetPointsAvg(Signer)

return signer points average

Declaration

```
public static double GetPointsAvg(this Signer signer)
```

Parameters

TYPE	NAME	DESCRIPTION
Signer	signer	

Returns

TYPE	DESCRIPTION
System.Double	

GetWidthAvg(Signer)

Return signer width average

Declaration

```
public static double GetWidthAvg(this Signer signer)
```

Parameters

TYPE	NAME	DESCRIPTION
Signer	signer	

Returns

TYPE	DESCRIPTION
System.Double	

Namespace SigStat.Common.Helpers.Excel

Classes

Palette

Enums

ExcelColor

Predefined color schemes for Excel

TextLevel

Paragraph style setting

Enum ExcelColor

Predefined color schemes for Excel

Namespace: SigStat.Common.Helpers.Excel

Assembly: SigStat.Common.dll

Syntax

```
public enum ExcelColor
```

Fields

NAME	DESCRIPTION
Danger	Danger color
Info	Info color
Primary	Primary color
Secondary	Secondary color
Succes	Succes color
Transparent	Transparent color
Warning	Warning color

Class Palette

Inheritance

System.Object
Palette

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: SigStat.Common.Helpers.Excel

Assembly: SigStat.Common.dll

Syntax

```
public class Palette
```

Constructors

Palette(Color, Color, Color)

Initializes a new instance of the Palette class.

Declaration

```
public Palette(Color main, Color dark, Color light)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Drawing.Color	main	The main color
System.Drawing.Color	dark	The dark color
System.Drawing.Color	light	The light color

Properties

DarkColor

Gets or sets the color for rendering darker elements

Declaration

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

Property Value

TYPE	DESCRIPTION
System.Drawing.Color	

LightColor

Gets or sets the color for rendering bright elements

Declaration

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

Property Value

TYPE	DESCRIPTION
System.Drawing.Color	

MainColor

Gets or sets the main color used in the palette

Declaration

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

Property Value

TYPE	DESCRIPTION
System.Drawing.Color	

Enum TextLevel

Paragraph style setting

Namespace: SigStat.Common.Helpers.Excel

Assembly: SigStat.Common.dll

Syntax

```
public enum TextLevel
```

Fields

NAME	DESCRIPTION
Heading1	Level 1 heading
Heading2	Level 2 heading
Heading3	Level 3 heading
Normal	Normal document body style
Title	Main title

Namespace SigStat.Common.Helpers.Serialization

Classes

DistanceFunctionJsonConverter

Helper class for serializing distance functions

DistanceMatrixConverter

Serializes/Deserializes a object using its ToArray() and FromArray() methods.

FeatureDescriptorDictionaryConverter

Custom serializer for a Dictionary of FeatureDescriptor

FeatureDescriptorListJsonConverter

Custom serializer for lists containing FeatureDescriptor or FeatureDescriptor<T> objects

FeatureStreamingContextState

SerializationContext for serializing SigStat objects

RectangleFConverter

Custom serializer for System.Drawing.RectangleF objects

VerifierResolver

Custom resolver for customizing the json serialization

Class DistanceFunctionJsonConverter

Helper class for serializing distance functions

Inheritance

System.Object
Newtonsoft.Json.JsonConverter
Newtonsoft.Json.JsonConverter<System.Func<System.Double[], System.Double[], System.Double>>
DistanceFunctionJsonConverter

Inherited Members

Newtonsoft.Json.JsonConverter<System.Func<System.Double[], System.Double[], System.Double>>.WriteJson(Newtonsoft.Json.JsonWriter, System.Object, Newtonsoft.Json.JsonSerializer)
Newtonsoft.Json.JsonConverter<System.Func<System.Double[], System.Double[], System.Double>>.WriteJson(Newtonsoft.Json.JsonWriter, System.Func<System.Double[], System.Double[], System.Double>, Newtonsoft.Json.JsonSerializer)
Newtonsoft.Json.JsonConverter<System.Func<System.Double[], System.Double[], System.Double>>.ReadJson(Newtonsoft.Json.JsonReader, System.Type, System.Object, Newtonsoft.Json.JsonSerializer)
Newtonsoft.Json.JsonConverter<System.Func<System.Double[], System.Double[], System.Double>>.ReadJson(Newtonsoft.Json.JsonReader, System.Type, System.Func<System.Double[], System.Double[], System.Double>, System.Boolean, Newtonsoft.Json.JsonSerializer)
Newtonsoft.Json.JsonConverter<System.Func<System.Double[], System.Double[], System.Double>>.CanConvert(System.Type)
Newtonsoft.Json.JsonConverter.WriteJson(Newtonsoft.Json.JsonWriter, System.Object, Newtonsoft.Json.JsonSerializer)
Newtonsoft.Json.JsonConverter.ReadJson(Newtonsoft.Json.JsonReader, System.Type, System.Object, Newtonsoft.Json.JsonSerializer)
Newtonsoft.Json.JsonConverter.CanConvert(System.Type)
Newtonsoft.Json.JsonConverter.CanRead
Newtonsoft.Json.JsonConverter.CanWrite
System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: SigStat.Common.Helpers.Serialization

Assembly: SigStat.Common.dll

Syntax

```
public class DistanceFunctionJsonConverter : JsonConverter<Func<double[], double[], double>>
```

Methods

ReadJson(JsonReader, Type, Func<Double[], Double[], Double>, Boolean, JsonSerializer)

Declaration

```
public override Func<double[], double[], double> ReadJson(JsonReader reader, Type objectType, Func<double[],
double[], double> existingValue, bool hasExistingValue, JsonSerializer serializer)
```

Parameters

TYPE	NAME	DESCRIPTION
Newtonsoft.Json.JsonReader	reader	

TYPE	NAME	DESCRIPTION
System.Type	objectType	
System.Func<System.Double[], System.Double[], System.Double>	existingValue	
System.Boolean	hasExistingValue	
Newtonsoft.Json.JsonSerializer	serializer	

Returns

TYPE	DESCRIPTION
System.Func<System.Double[], System.Double[], System.Double>	

Overrides

Newtonsoft.Json.JsonConverter<System.Func<System.Double[], System.Double[], System.Double>>.ReadJson(Newtonsoft.Json.JsonReader, System.Type, System.Func<System.Double[], System.Double[], System.Double>, System.Boolean, Newtonsoft.Json.JsonSerializer)

WriteJson(JsonWriter, Func<Double[], Double[], Double>, JsonSerializer)

Declaration

```
public override void WriteJson(JsonWriter writer, Func<double[], double[], double> value, JsonSerializer
serializer)
```

Parameters

TYPE	NAME	DESCRIPTION
Newtonsoft.Json.JsonWriter	writer	
System.Func<System.Double[], System.Double[], System.Double>	value	
Newtonsoft.Json.JsonSerializer	serializer	

Overrides

Newtonsoft.Json.JsonConverter<System.Func<System.Double[], System.Double[], System.Double>>.WriteJson(Newtonsoft.Json.JsonWriter, System.Func<System.Double[], System.Double[], System.Double>, Newtonsoft.Json.JsonSerializer)

See Also

Newtonsoft.Json.JsonConverter

Class DistanceMatrixConverter

Serializes/Deserializes a object using its ToArray() and FromArray() methods.

Inheritance

System.Object
Newtonsoft.Json.JsonConverter
Newtonsoft.Json.JsonConverter<DistanceMatrix<System.String, System.String, System.Double>>
DistanceMatrixConverter

Inherited Members

Newtonsoft.Json.JsonConverter<SigStat.Common.DistanceMatrix<System.String, System.String, System.Double>>.WriteJson(Newtonsoft.Json.JsonWriter, System.Object, Newtonsoft.Json.JsonSerializer)
Newtonsoft.Json.JsonConverter<SigStat.Common.DistanceMatrix<System.String, System.String, System.Double>>.WriteJson(Newtonsoft.Json.JsonWriter, SigStat.Common.DistanceMatrix<System.String, System.String, System.Double>, Newtonsoft.Json.JsonSerializer)
Newtonsoft.Json.JsonConverter<SigStat.Common.DistanceMatrix<System.String, System.String, System.Double>>.ReadJson(Newtonsoft.Json.JsonReader, System.Type, System.Object, Newtonsoft.Json.JsonSerializer)
Newtonsoft.Json.JsonConverter<SigStat.Common.DistanceMatrix<System.String, System.String, System.Double>>.ReadJson(Newtonsoft.Json.JsonReader, System.Type, SigStat.Common.DistanceMatrix<System.String, System.String, System.Double>, System.Boolean, Newtonsoft.Json.JsonSerializer)
Newtonsoft.Json.JsonConverter<SigStat.Common.DistanceMatrix<System.String, System.String, System.Double>>.CanConvert(System.Type)
Newtonsoft.Json.JsonConverter.WriteJson(Newtonsoft.Json.JsonWriter, System.Object, Newtonsoft.Json.JsonSerializer)
Newtonsoft.Json.JsonConverter.ReadJson(Newtonsoft.Json.JsonReader, System.Type, System.Object, Newtonsoft.Json.JsonSerializer)
Newtonsoft.Json.JsonConverter.CanConvert(System.Type)
Newtonsoft.Json.JsonConverter.CanRead
Newtonsoft.Json.JsonConverter.CanWrite
System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: SigStat.Common.Helpers.Serialization

Assembly: SigStat.Common.dll

Syntax

```
public class DistanceMatrixConverter : JsonConverter<DistanceMatrix<string, string, double>>
```

Methods

ReadJson(JsonReader, Type, DistanceMatrix<String, String, Double>, Boolean, JsonSerializer)

Declaration

```
public override DistanceMatrix<string, string, double> ReadJson(JsonReader reader, Type objectType,
DistanceMatrix<string, string, double> existingValue, bool hasExistingValue, JsonSerializer serializer)
```

Parameters

TYPE	NAME	DESCRIPTION
Newtonsoft.Json.JsonReader	reader	
System.Type	objectType	
DistanceMatrix<System.String, System.String, System.Double>	existingValue	
System.Boolean	hasExistingValue	
Newtonsoft.Json.JsonSerializer	serializer	

Returns

TYPE	DESCRIPTION
DistanceMatrix<System.String, System.String, System.Double>	

Overrides

Newtonsoft.Json.JsonConverter<SigStat.Common.DistanceMatrix<System.String, System.String, System.Double>>.ReadJson(Newtonsoft.Json.JsonReader, System.Type, SigStat.Common.DistanceMatrix<System.String, System.String, System.Double>, System.Boolean, Newtonsoft.Json.JsonSerializer)

WriteJson(JsonWriter, DistanceMatrix<String, String, Double>, JsonSerializer)

Declaration

```
public override void WriteJson(JsonWriter writer, DistanceMatrix<string, string, double> value, JsonSerializer
serializer)
```

Parameters

TYPE	NAME	DESCRIPTION
Newtonsoft.Json.JsonWriter	writer	
DistanceMatrix<System.String, System.String, System.Double>	value	
Newtonsoft.Json.JsonSerializer	serializer	

Overrides

Newtonsoft.Json.JsonConverter<SigStat.Common.DistanceMatrix<System.String, System.String, System.Double>>.WriteJson(Newtonsoft.Json.JsonWriter, SigStat.Common.DistanceMatrix<System.String, System.String, System.Double>, Newtonsoft.Json.JsonSerializer)

Class FeatureDescriptorDictionaryConverter

Custom serializer for a Dictionary of FeatureDescriptor

Inheritance

System.Object
Newtonsoft.Json.JsonConverter
FeatureDescriptorDictionaryConverter

Inherited Members

Newtonsoft.Json.JsonConverter.CanRead
Newtonsoft.Json.JsonConverter.CanWrite
System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: SigStat.Common.Helpers.Serialization

Assembly: SigStat.Common.dll

Syntax

```
public class FeatureDescriptorDictionaryConverter : JsonConverter
```

Methods

CanConvert(Type)

Tells if the current object is of the correct type

Declaration

```
public override bool CanConvert(Type objectType)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Type	objectType	The type of the object

Returns

TYPE	DESCRIPTION
System.Boolean	If the object can be converted or not

Overrides

Newtonsoft.Json.JsonConverter.CanConvert(System.Type)

ReadJson(JsonReader, Type, Object, JsonSerializer)

Overwrite of the Newtonsoft.Json.JsonConverter method Deserializes the dictionary of FeatureDescriptor created by the this class

Declaration

```
public override object ReadJson(JsonReader reader, Type objectType, object existingValue, JsonSerializer
serializer)
```

Parameters

TYPE	NAME	DESCRIPTION
Newtonsoft.Json.JsonReader	reader	
System.Type	objectType	
System.Object	existingValue	
Newtonsoft.Json.JsonSerializer	serializer	

Returns

TYPE	DESCRIPTION
System.Object	

Overrides

Newtonsoft.Json.JsonConverter.ReadJson(Newtonsoft.Json.JsonReader, System.Type, System.Object, Newtonsoft.Json.JsonSerializer)

WriteJson(JsonWriter, Object, JsonSerializer)

Overwrite of the Newtonsoft.Json.JsonConverter method Serializes the dictionary FeatureDescriptor with type of the descriptor

Declaration

```
public override void WriteJson(JsonWriter writer, object value, JsonSerializer serializer)
```

Parameters

TYPE	NAME	DESCRIPTION
Newtonsoft.Json.JsonWriter	writer	
System.Object	value	
Newtonsoft.Json.JsonSerializer	serializer	

Overrides

Newtonsoft.Json.JsonConverter.WriteJson(Newtonsoft.Json.JsonWriter, System.Object, Newtonsoft.Json.JsonSerializer)

Class FeatureDescriptorListJsonConverter

Custom serializer for lists containing FeatureDescriptor or FeatureDescriptor<T> objects

Inheritance

System.Object
Newtonsoft.Json.JsonConverter
FeatureDescriptorListJsonConverter

Inherited Members

Newtonsoft.Json.JsonConverter.CanRead
Newtonsoft.Json.JsonConverter.CanWrite
System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: SigStat.Common.Helpers.Serialization

Assembly: SigStat.Common.dll

Syntax

```
public class FeatureDescriptorListJsonConverter : JsonConverter
```

Methods

CanConvert(Type)

Tells if the current object is of the correct type

Declaration

```
public override bool CanConvert(Type objectType)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Type	objectType	The type of the object

Returns

TYPE	DESCRIPTION
System.Boolean	If the object can be converted or not

Overrides

Newtonsoft.Json.JsonConverter.CanConvert(System.Type)

ReadJson(JsonReader, Type, Object, JsonSerializer)

Overwrite of the Newtonsoft.Json.JsonConverter method Deserializes the list of FeatureDescriptor objects

Declaration

```
public override object ReadJson(JsonReader reader, Type objectType, object existingValue, JsonSerializer
serializer)
```

Parameters

TYPE	NAME	DESCRIPTION
Newtonsoft.Json.JsonReader	reader	
System.Type	objectType	
System.Object	existingValue	
Newtonsoft.Json.JsonSerializer	serializer	

Returns

TYPE	DESCRIPTION
System.Object	

Overrides

Newtonsoft.Json.JsonConverter.ReadJson(Newtonsoft.Json.JsonReader, System.Type, System.Object, Newtonsoft.Json.JsonSerializer)

WriteJson(JsonWriter, Object, JsonSerializer)

Overwrite of the Newtonsoft.Json.JsonConverter method Serializes the list of FeatureDescriptor objects to json

Declaration

```
public override void WriteJson(JsonWriter writer, object value, JsonSerializer serializer)
```

Parameters

TYPE	NAME	DESCRIPTION
Newtonsoft.Json.JsonWriter	writer	
System.Object	value	
Newtonsoft.Json.JsonSerializer	serializer	

Overrides

Newtonsoft.Json.JsonConverter.WriteJson(Newtonsoft.Json.JsonWriter, System.Object, Newtonsoft.Json.JsonSerializer)

Class FeatureStreamingContextState

SerializationContext for serializing SigStat objects

Inheritance

System.Object
FeatureStreamingContextState

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: SigStat.Common.Helpers.Serialization

Assembly: SigStat.Common.dll

Syntax

```
public class FeatureStreamingContextState
```

Constructors

FeatureStreamingContextState(Boolean)

Constructor

Declaration

```
public FeatureStreamingContextState(bool compactFeatures)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Boolean	compactFeatures	

Properties

CompactFeatures

Declaration

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

Property Value

TYPE	DESCRIPTION
System.Boolean	

KnownFeatureKeys

A list of already serialized FeatureDescriptor keys

Declaration

```
public List<string> KnownFeatureKeys { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Collections.Generic.List<System.String>	

Class RectangleFConverter

Custom serializer for System.Drawing.RectangleF objects

Inheritance

System.Object
Newtonsoft.Json.JsonConverter
RectangleFConverter

Inherited Members

Newtonsoft.Json.JsonConverter.CanRead
Newtonsoft.Json.JsonConverter.CanWrite
System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: SigStat.Common.Helpers.Serialization

Assembly: SigStat.Common.dll

Syntax

```
public class RectangleFConverter : JsonConverter
```

Methods

CanConvert(Type)

Tells if the current object is of the correct type

Declaration

```
public override bool CanConvert(Type objectType)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Type	objectType	The type of the object

Returns

TYPE	DESCRIPTION
System.Boolean	If the object can be converted or not

Overrides

Newtonsoft.Json.JsonConverter.CanConvert(System.Type)

ReadJson(JsonReader, Type, Object, JsonSerializer)

Overwrite of the Newtonsoft.Json.JsonConverter method Deserializes the System.Drawing.RectangleF json created by the same class

Declaration

```
public override object ReadJson(JsonReader reader, Type objectType, object existingValue, JsonSerializer
serializer)
```

Parameters

TYPE	NAME	DESCRIPTION
Newtonsoft.Json.JsonReader	reader	
System.Type	objectType	
System.Object	existingValue	
Newtonsoft.Json.JsonSerializer	serializer	

Returns

TYPE	DESCRIPTION
System.Object	

Overrides

Newtonsoft.Json.JsonConverter.ReadJson(Newtonsoft.Json.JsonReader, System.Type, System.Object, Newtonsoft.Json.JsonSerializer)

WriteJson(JsonWriter, Object, JsonSerializer)

Overwrite of the Newtonsoft.Json.JsonConverter method Serializes the System.Drawing.RectangleF to json

Declaration

```
public override void WriteJson(JsonWriter writer, object value, JsonSerializer serializer)
```

Parameters

TYPE	NAME	DESCRIPTION
Newtonsoft.Json.JsonWriter	writer	
System.Object	value	
Newtonsoft.Json.JsonSerializer	serializer	

Overrides

Newtonsoft.Json.JsonConverter.WriteJson(Newtonsoft.Json.JsonWriter, System.Object, Newtonsoft.Json.JsonSerializer)

Class VerifierResolver

Custom resolver for customizing the json serialization

Inheritance

System.Object
Newtonsoft.Json.Serialization.DefaultContractResolver
VerifierResolver

Implements

Newtonsoft.Json.Serialization.IContractResolver

Inherited Members

Newtonsoft.Json.Serialization.DefaultContractResolver.ResolveContract(System.Type)
Newtonsoft.Json.Serialization.DefaultContractResolver.GetSerializableMembers(System.Type)
Newtonsoft.Json.Serialization.DefaultContractResolver.CreateObjectContract(System.Type)
Newtonsoft.Json.Serialization.DefaultContractResolver.CreateConstructorParameters(System.Reflection.ConstructorInfo, Newtonsoft.Json.Serialization.JsonPropertyCollection)
Newtonsoft.Json.Serialization.DefaultContractResolver.CreatePropertyFromConstructorParameter(Newtonsoft.Json.Serialization.JsonProperty, System.Reflection.ParameterInfo)
Newtonsoft.Json.Serialization.DefaultContractResolver.ResolveContractConverter(System.Type)
Newtonsoft.Json.Serialization.DefaultContractResolver.CreateDictionaryContract(System.Type)
Newtonsoft.Json.Serialization.DefaultContractResolver.CreateArrayContract(System.Type)
Newtonsoft.Json.Serialization.DefaultContractResolver.CreatePrimitiveContract(System.Type)
Newtonsoft.Json.Serialization.DefaultContractResolver.CreateLinqContract(System.Type)
Newtonsoft.Json.Serialization.DefaultContractResolver.CreateISerializableContract(System.Type)
Newtonsoft.Json.Serialization.DefaultContractResolver.CreateDynamicContract(System.Type)
Newtonsoft.Json.Serialization.DefaultContractResolver.CreateStringContract(System.Type)
Newtonsoft.Json.Serialization.DefaultContractResolver.CreateContract(System.Type)
Newtonsoft.Json.Serialization.DefaultContractResolver.CreateMemberValueProvider(System.Reflection.MemberInfo)
Newtonsoft.Json.Serialization.DefaultContractResolver.ResolvePropertyName(System.String)
Newtonsoft.Json.Serialization.DefaultContractResolver.ResolveExtensionDataName(System.String)
Newtonsoft.Json.Serialization.DefaultContractResolver.ResolveDictionaryKey(System.String)
Newtonsoft.Json.Serialization.DefaultContractResolver.GetResolvedPropertyName(System.String)
Newtonsoft.Json.Serialization.DefaultContractResolver.DynamicCodeGeneration
Newtonsoft.Json.Serialization.DefaultContractResolver.DefaultMembersSearchFlags
Newtonsoft.Json.Serialization.DefaultContractResolver.SerializeCompilerGeneratedMembers
Newtonsoft.Json.Serialization.DefaultContractResolver.IgnoreSerializableInterface
Newtonsoft.Json.Serialization.DefaultContractResolver.IgnoreSerializableAttribute
Newtonsoft.Json.Serialization.DefaultContractResolver.IgnoreIsSpecifiedMembers
Newtonsoft.Json.Serialization.DefaultContractResolver.IgnoreShouldSerializeMembers
Newtonsoft.Json.Serialization.DefaultContractResolver.NamingStrategy
System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: SigStat.Common.Helpers.Serialization

Assembly: SigStat.Common.dll

Syntax

```
public class VerifierResolver : DefaultContractResolver, IContractResolver
```

Methods

CreateProperties(Type, MemberSerialization)

Decides if the current property should be serialized or not

Declaration

```
protected override IList<JsonProperty> CreateProperties(Type type, MemberSerialization memberSerialization)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Type	type	The type of the current property
Newtonsoft.Json.MemberSerialization	memberSerialization	The type of member serialization in Json.NET

Returns

TYPE	DESCRIPTION
System.Collections.Generic.IList<Newtonsoft.Json.Serialization.JsonProperty>	A bool

Overrides

Newtonsoft.Json.Serialization.DefaultContractResolver.CreateProperties(System.Type, Newtonsoft.Json.MemberSerialization)

CreateProperty(MemberInfo, MemberSerialization)

Selects which JsonConverter should be used for the property

Declaration

```
protected override JsonProperty CreateProperty(MemberInfo member, MemberSerialization memberSerialization)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Reflection.MemberInfo	member	A System.Reflection.MemberInfo
Newtonsoft.Json.MemberSerialization	memberSerialization	The type of member serialization in Json.NET

Returns

TYPE	DESCRIPTION
Newtonsoft.Json.Serialization.JsonProperty	

Overrides

Newtonsoft.Json.Serialization.DefaultContractResolver.CreateProperty(System.Reflection.MemberInfo, Newtonsoft.Json.MemberSerialization)

Implements

Newtonsoft.Json.Serialization.IContractResolver

Namespace SigStat.Common.Loaders

Classes

BenchmarkBuilder

DataSetLoader

Abstract loader class to inherit from. Implements ILogger.

ImageLoader

DataSetLoader for Image type databases. Similar format to Svc2004Loader, but finds png images.

ImageSaver

Get the Image of a Signature and save it as png file.

MCYTLoader

DataSetLoader for the MCYT dataset

MCYTLoader.MCYT

Set of features containing raw data loaded from MCYT-format database.

MemoryDataSetLoader

Stores and enumerates Signer data that has already been loaded

SigComp11ChineseLoader

DataSetLoader for the SigComp11Chinese dataset

SigComp11ChineseLoader.SigComp11Ch

Set of features containing raw data loaded from SigComp11Chinese-format database.

SigComp11DutchLoader

DataSetLoader for the SigComp11Dutch dataset

SigComp11DutchLoader.SigComp11

Set of features containing raw data loaded from MCYT-format database.

SigComp13JapaneseLoader

DataSetLoader for the SigComp13Japanese dataset

SigComp13JapaneseLoader.SigComp13Japanese

Set of features containing raw data loaded from SigComp13Japanese-format database.

SigComp15GermanLoader

DataSetLoader for the SigComp15German dataset

SigComp15GermanLoader.SigComp15

Set of features containing raw data loaded from SigComp15German-format database.

SigComp19OnlineLoader

DataSetLoader for the SigComp19 dataset

SigComp19OnlineLoader.SigComp19

Set of features containing raw data loaded from SigComp19-format database.

Svc2004

Set of features containing raw data loaded from SVC2004-format database.

Svc2004Loader

Loads SVC2004-format database from .zip

Interfaces

IDataSetLoader

Exposes a function to enable loading collections of Signers. Base abstract class: DataSetLoader.

Class BenchmarkBuilder

Inheritance

System.Object
BenchmarkBuilder

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: SigStat.Common.Loaders

Assembly: SigStat.Common.dll

Syntax

```
[Obsolete]
public static class BenchmarkBuilder
```

Methods

Build(BenchmarkConfig, String)

Declaration

```
public static VerifierBenchmark Build(BenchmarkConfig config, string databasePath = null)
```

Parameters

TYPE	NAME	DESCRIPTION
BenchmarkConfig	config	
System.String	databasePath	

Returns

TYPE	DESCRIPTION
VerifierBenchmark	

Class DataSetLoader

Abstract loader class to inherit from. Implements ILogger.

Inheritance

System.Object
DataSetLoader
ImageLoader
MCYTLoader
MemoryDataSetLoader
SigComp11ChineseLoader
SigComp11DutchLoader
SigComp13JapaneseLoader
SigComp15GermanLoader
SigComp19OnlineLoader
Svc2004Loader
BiosecureIDOfflineLoader
BiosecureIDOnlineLoader
MemoryLoader
Svc2004OfflineLoader
Svc2004OnlineLoader

Implements

IDataSetLoader
ILoggerObject

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: SigStat.Common.Loaders

Assembly: SigStat.Common.dll

Syntax

```
public abstract class DataSetLoader : IDataSetLoader, ILoggerObject
```

Properties

Logger

Declaration

```
public ILogger Logger { get; set; }
```

Property Value

TYPE	DESCRIPTION
Microsoft.Extensions.Logging.ILogger	

SamplingFrequency

Sampling frequency for each database

Declaration

```
public abstract int SamplingFrequency { get; }
```

Property Value

TYPE	DESCRIPTION
System.Int32	

Methods

EnumerateSigners()

Declaration

```
public IEnumerable<Signer> EnumerateSigners()
```

Returns

TYPE	DESCRIPTION
System.Collections.Generic.IEnumerable<Signer>	

EnumerateSigners(Predicate<Signer>)

Declaration

```
public abstract IEnumerable<Signer> EnumerateSigners(Predicate<Signer> signerFilter)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Predicate<Signer>	signerFilter	

Returns

TYPE	DESCRIPTION
System.Collections.Generic.IEnumerable<Signer>	

Explicit Interface Implementations

IDataSetLoader.SignerFilter

Ignores any signers during the loading, that do not match the predicate

Declaration

```
Predicate<Signer> IDataSetLoader.SignerFilter { get; set; }
```

Returns

TYPE	DESCRIPTION
System.Predicate<Signer>	

Implements

IDataSetLoader

ILoggerObject

Extension Methods

ILoggerObjectExtensions.LogError(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogError(ILoggerObject, Exception, String, Object[])
ILoggerObjectExtensions.LogInformation(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogWarning(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogWarning(ILoggerObject, Exception, String, Object[])
ILoggerObjectExtensions.LogTrace(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogTrace<TState>(ILoggerObject, TState, EventId, Exception, Func<TState, Exception, String>)
ILoggerObjectExtensions.LogCritical(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogDebug(ILoggerObject, String, Object[])

Interface IDataSetLoader

Exposes a function to enable loading collections of Signers. Base abstract class: DataSetLoader.

Namespace: SigStat.Common.Loaders

Assembly: SigStat.Common.dll

Syntax

```
public interface IDataSetLoader
```

Properties

SignerFilter

Ignores any signers during the loading, that do not match the predicate

Declaration

```
Predicate<Signer> SignerFilter { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Predicate<Signer>	

Methods

EnumerateSigners()

Enumerates all signers of the database

Declaration

```
IEnumerable<Signer> EnumerateSigners()
```

Returns

TYPE	DESCRIPTION
System.Collections.Generic.IEnumerable<Signer>	

EnumerateSigners(Predicate<Signer>)

Enumerates all Signers that match the `signerFilter`.

Declaration

```
IEnumerable<Signer> EnumerateSigners(Predicate<Signer> signerFilter)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Predicate<Signer>	signerFilter	Filter to specify which Signers to load. Example: (p=>p=="01")

Returns

TYPE	DESCRIPTION
System.Collections.Generic.IEnumerable<Signer>	Collection of Signers that match the `signerFilter`

Class ImageLoader

DataSetLoader for Image type databases. Similar format to Svc2004Loader, but finds png images.

Inheritance

System.Object
DataSetLoader
ImageLoader

Implements

IDataSetLoader
ILoggerObject

Inherited Members

DataSetLoader.Logger
DataSetLoader.EnumerateSigners()
DataSetLoader.IDataSetLoader.SignerFilter
System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: SigStat.Common.Loaders

Assembly: SigStat.Common.dll

Syntax

```
public class ImageLoader : DataSetLoader, IDataSetLoader, ILoggerObject
```

Constructors

ImageLoader(String)

Initializes a new instance of the ImageLoader class with specified database.

Declaration

```
public ImageLoader(string databasePath)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	databasePath	File path to the database.

Properties

SamplingFrequency

Declaration

```
public override int SamplingFrequency { get; }
```

Property Value

TYPE	DESCRIPTION
System.Int32	

Overrides

DataSetLoader.SamplingFrequency

Methods

EnumerateSigners(Predicate<Signer>)

Declaration

```
public override IEnumerable<Signer> EnumerateSigners(Predicate<Signer> signerFilter)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Predicate<Signer>	signerFilter	

Returns

TYPE	DESCRIPTION
System.Collections.Generic.IEnumerable<Signer>	

Overrides

DataSetLoader.EnumerateSigners(Predicate<Signer>)

LoadImage(Signature, String)

Load one image.

Declaration

```
protected static void LoadImage(Signature signature, string file)
```

Parameters

TYPE	NAME	DESCRIPTION
Signature	signature	The signature that receives the new Image
System.String	file	File path to the image to be loaded.

LoadSignature(String)

Declaration

```
public static Signature LoadSignature(string file)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	file	

Returns

TYPE	DESCRIPTION
Signature	

Implements

IDataSetLoader
ILoggerObject

Extension Methods

ILoggerObjectExtensions.LogError(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogError(ILoggerObject, Exception, String, Object[])
ILoggerObjectExtensions.LogInformation(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogWarning(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogWarning(ILoggerObject, Exception, String, Object[])
ILoggerObjectExtensions.LogTrace(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogTrace<TState>(ILoggerObject, TState, EventId, Exception, Func<TState, Exception, String>)
ILoggerObjectExtensions.LogCritical(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogDebug(ILoggerObject, String, Object[])

Class ImageSaver

Get the Image of a Signature and save it as png file.

Inheritance

System.Object
ImageSaver

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: SigStat.Common.Loaders

Assembly: SigStat.Common.dll

Syntax

```
public static class ImageSaver
```

Methods

Save(Signature, String)

Saves a png image file to the specified `path`.

Declaration

```
public static void Save(Signature signature, string path)
```

Parameters

TYPE	NAME	DESCRIPTION
Signature	signature	Input signature containing Image.
System.String	path	Output file path of the png image.

Class MCYTLoader

DataSetLoader for the MCYT dataset

Inheritance

System.Object
DataSetLoader
MCYTLoader

Implements

IDataSetLoader
ILoggerObject

Inherited Members

DataSetLoader.Logger
DataSetLoader.EnumerateSigners()
DataSetLoader.IDataSetLoader.SignerFilter
System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: SigStat.Common.Loaders

Assembly: SigStat.Common.dll

Syntax

```
[JsonObject(MemberSerialization.OptOut)]
public class MCYTLoader : DataSetLoader, IDataSetLoader, ILoggerObject
```

Constructors

MCYTLoader(String, Boolean)

Initializes a new instance of the MCYTLoader class.

Declaration

```
public MCYTLoader(string databasePath, bool standardFeatures)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	databasePath	The database path.
System.Boolean	standardFeatures	if set to `true` features will be also stored in Features.

MCYTLoader(String, Boolean, Predicate<Signer>)

Initializes a new instance of the MCYTLoader class with specified database.

Declaration

```
public MCYTLoader(string databasePath, bool standardFeatures, Predicate<Signer> signerFilter = null)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	databasePath	Represents the path, to load the signatures from. It supports two basic approaches: • •
System.Boolean	standardFeatures	Convert loaded data to standard Features.
System.Predicate<Signer>	signerFilter	Sets the SignerFilter property

Properties

DatabasePath

Gets or sets the database path.

Declaration

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

Property Value

TYPE	DESCRIPTION
System.String	

SamplingFrequency

Set MCYT sampling frequenct to 100hz

Declaration

```
public override int SamplingFrequency { get; }
```

Property Value

TYPE	DESCRIPTION
System.Int32	

Overrides

DataSetLoader.SamplingFrequency

SignerFilter

Ignores any signers during the loading, that do not match the predicate

Declaration

```
public Predicate<Signer> SignerFilter { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Predicate<Signer>	

StandardFeatures

Gets or sets a value indicating whether features are also loaded as Features

Declaration

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

Property Value

TYPE	DESCRIPTION
System.Boolean	

Methods

EnumerateSigners(Predicate<Signer>)

Declaration

```
public override IEnumerable<Signer> EnumerateSigners(Predicate<Signer> signerFilter)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Predicate<Signer>	signerFilter	

Returns

TYPE	DESCRIPTION
System.Collections.Generic.IEnumerable<Signer>	

Overrides

DataSetLoader.EnumerateSigners(Predicate<Signer>)

LoadSignature(Signature, MemoryStream, Boolean)

Loads one signature from specified stream.

Declaration

```
public static void LoadSignature(Signature signature, MemoryStream stream, bool standardFeatures)
```

Parameters

TYPE	NAME	DESCRIPTION
Signature	signature	Signature to write features to.
System.IO.MemoryStream	stream	Stream to read MCYT data from.

TYPE	NAME	DESCRIPTION
System.Boolean	standardFeatures	Convert loaded data to standard Features.

Remarks

Based on Mohammad's MCYT reader.

Implements

IDataSetLoader
ILoggerObject

Extension Methods

ILoggerObjectExtensions.LogError(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogError(ILoggerObject, Exception, String, Object[])
ILoggerObjectExtensions.LogInformation(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogWarning(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogWarning(ILoggerObject, Exception, String, Object[])
ILoggerObjectExtensions.LogTrace(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogTrace<TState>(ILoggerObject, TState, EventId, Exception, Func<TState, Exception, String>)
ILoggerObjectExtensions.LogCritical(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogDebug(ILoggerObject, String, Object[])

Class MCYTLoader.MCYT

Set of features containing raw data loaded from MCYT-format database.

Inheritance

System.Object
MCYTLoader.MCYT

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: SigStat.Common.Loaders

Assembly: SigStat.Common.dll

Syntax

```
public static class MCYT
```

Fields

Altitude

Altitude values from the online signature imported from the MCYT database

Declaration

```
public static readonly FeatureDescriptor<List<int>> Altitude
```

Field Value

TYPE	DESCRIPTION
FeatureDescriptor<System.Collections.Generic.List<System.Int32>>	

Azimuth

Azimuth values from the online signature imported from the MCYT database

Declaration

```
public static readonly FeatureDescriptor<List<int>> Azimuth
```

Field Value

TYPE	DESCRIPTION
FeatureDescriptor<System.Collections.Generic.List<System.Int32>>	

Pressure

Pressure values from the online signature imported from the MCYT database

Declaration

```
public static readonly FeatureDescriptor<List<int>> Pressure
```

Field Value

TYPE	DESCRIPTION
FeatureDescriptor<System.Collections.Generic.List<System.Int32>>	

X

X cooridnates from the online signature imported from the MCYT database

Declaration

```
public static readonly FeatureDescriptor<List<int>> X
```

Field Value

TYPE	DESCRIPTION
FeatureDescriptor<System.Collections.Generic.List<System.Int32>>	

Y

Y cooridnates from the online signature imported from the MCYT database

Declaration

```
public static readonly FeatureDescriptor<List<int>> Y
```

Field Value

TYPE	DESCRIPTION
FeatureDescriptor<System.Collections.Generic.List<System.Int32>>	

Class MemoryDataSetLoader

Stores and enumerates Signer data that has already been loaded

Inheritance

System.Object

DataSetLoader

MemoryDataSetLoader

Implements

IDataSetLoader

ILoggerObject

Inherited Members

DataSetLoader.Logger

DataSetLoader.EnumerateSigners()

DataSetLoader.IDataSetLoader.SignerFilter

System.Object.Equals(System.Object)

System.Object.Equals(System.Object, System.Object)

System.Object.GetHashCode()

System.Object.GetType()

System.Object.MemberwiseClone()

System.Object.ReferenceEquals(System.Object, System.Object)

System.Object.ToString()

Namespace: SigStat.Common.Loaders

Assembly: SigStat.Common.dll

Syntax

```
[JsonObject(MemberSerialization.OptOut)]
public class MemoryDataSetLoader : DataSetLoader, IDataSetLoader, ILoggerObject
```

Constructors

MemoryDataSetLoader(IEnumerable<Signer>)

Initializes a new instance of the Svc2004Loader class with specified database.

Declaration

```
[JsonConstructor]
public MemoryDataSetLoader(IEnumerable<Signer> signers)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Collections.Generic.IEnumerable<Signer>	signers	

Properties

DatabasePath

Gets or sets the database path.

Declaration

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

Property Value

TYPE	DESCRIPTION
System.String	

SamplingFrequency

Declaration

```
public override int SamplingFrequency { get; }
```

Property Value

TYPE	DESCRIPTION
System.Int32	

Overrides

DataSetLoader.SamplingFrequency

SignerFilter

Ignores any signers during the loading, that do not match the predicate

Declaration

```
public Predicate<Signer> SignerFilter { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Predicate<Signer>	

StandardFeatures

Gets or sets a value indicating whether features are also loaded as Features

Declaration

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

Property Value

TYPE	DESCRIPTION
System.Boolean	

Methods

EnumerateSigners(Predicate<Signer>)

Declaration

```
public override IEnumerable<Signer> EnumerateSigners(Predicate<Signer> signerFilter)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Predicate<Signer>	signerFilter	

Returns

TYPE	DESCRIPTION
System.Collections.Generic.IEnumerable<Signer>	

Overrides

DataSetLoader.EnumerateSigners(Predicate<Signer>)

Implements

IDataSetLoader
ILoggerObject

Extension Methods

ILoggerObjectExtensions.LogError(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogError(ILoggerObject, Exception, String, Object[])
ILoggerObjectExtensions.LogInformation(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogWarning(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogWarning(ILoggerObject, Exception, String, Object[])
ILoggerObjectExtensions.LogTrace(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogTrace<TState>(ILoggerObject, TState, EventId, Exception, Func<TState, Exception, String>)
ILoggerObjectExtensions.LogCritical(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogDebug(ILoggerObject, String, Object[])

Class SigComp11ChineseLoader

DataSetLoader for the SigComp11Chinese dataset

Inheritance

System.Object
DataSetLoader
SigComp11ChineseLoader

Implements

IDataSetLoader
ILoggerObject

Inherited Members

DataSetLoader.Logger
DataSetLoader.EnumerateSigners()
DataSetLoader.IDataSetLoader.SignerFilter
System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: SigStat.Common.Loaders

Assembly: SigStat.Common.dll

Syntax

```
public class SigComp11ChineseLoader : DataSetLoader, IDataSetLoader, ILoggerObject
```

Constructors

SigComp11ChineseLoader(String, Boolean)

Initializes a new instance of the SigComp11ChineseLoader class.

Declaration

```
public SigComp11ChineseLoader(string databasePath, bool standardFeatures)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	databasePath	The database path.
System.Boolean	standardFeatures	if set to `true` features will be also stored in Features.

SigComp11ChineseLoader(String, Boolean, Predicate<Signer>)

Initializes a new instance of the SigComp11ChineseLoader class with specified database.

Declaration

```
public SigComp11ChineseLoader(string databasePath, bool standardFeatures, Predicate<Signer> signerFilter =
null)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	databasePath	Represents the path, to load the signatures from. It supports two basic approaches: ● ●
System.Boolean	standardFeatures	Convert loaded data to standard Features.
System.Predicate<Signer>	signerFilter	Sets the SignerFilter property

Properties

DatabasePath

Gets or sets the database path.

Declaration

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

Property Value

TYPE	DESCRIPTION
System.String	

SamplingFrequency

Sampling Frequency of this database

Declaration

```
public override int SamplingFrequency { get; }
```

Property Value

TYPE	DESCRIPTION
System.Int32	

Overrides

DataSetLoader.SamplingFrequency

SignerFilter

Ignores any signers during the loading, that do not match the predicate

Declaration

```
public Predicate<Signer> SignerFilter { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Predicate<Signer>	

StandardFeatures

Gets or sets a value indicating whether features are also loaded as Features

Declaration

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

Property Value

TYPE	DESCRIPTION
System.Boolean	

Methods

EnumerateSigners(Predicate<Signer>)

Declaration

```
public override IEnumerable<Signer> EnumerateSigners(Predicate<Signer> signerFilter)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Predicate<Signer>	signerFilter	

Returns

TYPE	DESCRIPTION
System.Collections.Generic.IEnumerable<Signer>	

Overrides

DataSetLoader.EnumerateSigners(Predicate<Signer>)

LoadSignature(Signature, MemoryStream, Boolean)

Loads one signature from specified stream.

Declaration

```
public static void LoadSignature(Signature signature, MemoryStream stream, bool standardFeatures)
```

Parameters

TYPE	NAME	DESCRIPTION
Signature	signature	Signature to write features to.
System.IO.MemoryStream	stream	Stream to read MCYT data from.

TYPE	NAME	DESCRIPTION
System.Boolean	standardFeatures	Convert loaded data to standard Features.

Remarks

Based on Mohammad's MCYT reader.

Implements

IDataSetLoader
ILoggerObject

Extension Methods

ILoggerObjectExtensions.LogError(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogError(ILoggerObject, Exception, String, Object[])
ILoggerObjectExtensions.LogInformation(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogWarning(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogWarning(ILoggerObject, Exception, String, Object[])
ILoggerObjectExtensions.LogTrace(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogTrace<TState>(ILoggerObject, TState, EventId, Exception, Func<TState, Exception, String>)
ILoggerObjectExtensions.LogCritical(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogDebug(ILoggerObject, String, Object[])

Class SigComp11ChineseLoader.SigComp11Ch

Set of features containing raw data loaded from SigComp11Chinese-format database.

Inheritance

System.Object
SigComp11ChineseLoader.SigComp11Ch

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: SigStat.Common.Loaders

Assembly: SigStat.Common.dll

Syntax

```
public static class SigComp11Ch
```

Fields

P

Z cooridnates from the online signature imported from the SigComp11Chinese database

Declaration

```
public static readonly FeatureDescriptor<List<int>> P
```

Field Value

TYPE	DESCRIPTION
FeatureDescriptor<System.Collections.Generic.List<System.Int32>>	

T

T values from the online signature imported from the SigComp11Chinese database

Declaration

```
public static readonly FeatureDescriptor<List<int>> T
```

Field Value

TYPE	DESCRIPTION
FeatureDescriptor<System.Collections.Generic.List<System.Int32>>	

X

X cooridnates from the online signature imported from the SigComp11Chinese database

Declaration

```
public static readonly FeatureDescriptor<List<int>> X
```

Field Value

TYPE	DESCRIPTION
FeatureDescriptor<System.Collections.Generic.List<System.Int32>>	

Y

Y cooridnates from the online signature imported from the SigComp11Chinese database

Declaration

```
public static readonly FeatureDescriptor<List<int>> Y
```

Field Value

TYPE	DESCRIPTION
FeatureDescriptor<System.Collections.Generic.List<System.Int32>>	

Class SigComp11DutchLoader

DataSetLoader for the SigComp11Dutch dataset

Inheritance

System.Object
DataSetLoader
SigComp11DutchLoader

Implements

IDataSetLoader
ILoggerObject

Inherited Members

DataSetLoader.Logger
DataSetLoader.EnumerateSigners()
DataSetLoader.IDataSetLoader.SignerFilter
System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: SigStat.Common.Loaders

Assembly: SigStat.Common.dll

Syntax

```
public class SigComp11DutchLoader : DataSetLoader, IDataSetLoader, ILoggerObject
```

Constructors

SigComp11DutchLoader(String, Boolean)

Initializes a new instance of the SigComp11DutchLoader class.

Declaration

```
public SigComp11DutchLoader(string databasePath, bool standardFeatures)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	databasePath	The database path.
System.Boolean	standardFeatures	if set to `true` features will be also stored in Features.

SigComp11DutchLoader(String, Boolean, Predicate<Signer>)

Initializes a new instance of the SigComp11DutchLoader class with specified database.

Declaration

```
public SigComp11DutchLoader(string databasePath, bool standardFeatures, Predicate<Signer> signerFilter = null)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	databasePath	Represents the path, to load the signatures from. It supports two basic approaches: • •
System.Boolean	standardFeatures	Convert loaded data to standard Features.
System.Predicate<Signer>	signerFilter	Sets the SignerFilter property

Properties

DatabasePath

Gets or sets the database path.

Declaration

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

Property Value

TYPE	DESCRIPTION
System.String	

SamplingFrequency

Sampling Frequency of this database

Declaration

```
public override int SamplingFrequency { get; }
```

Property Value

TYPE	DESCRIPTION
System.Int32	

Overrides

DataSetLoader.SamplingFrequency

SignerFilter

Ignores any signers during the loading, that do not match the predicate

Declaration

```
public Predicate<Signer> SignerFilter { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Predicate<Signer>	

StandardFeatures

Gets or sets a value indicating whether features are also loaded as Features

Declaration

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

Property Value

TYPE	DESCRIPTION
System.Boolean	

Methods

EnumerateSigners(Predicate<Signer>)

Declaration

```
public override IEnumerable<Signer> EnumerateSigners(Predicate<Signer> signerFilter)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Predicate<Signer>	signerFilter	

Returns

TYPE	DESCRIPTION
System.Collections.Generic.IEnumerable<Signer>	

Overrides

DataSetLoader.EnumerateSigners(Predicate<Signer>)

LoadSignature(Signature, MemoryStream, Boolean)

Loads one signature from specified stream.

Declaration

```
public static void LoadSignature(Signature signature, MemoryStream stream, bool standardFeatures)
```

Parameters

TYPE	NAME	DESCRIPTION
Signature	signature	Signature to write features to.
System.IO.MemoryStream	stream	Stream to read MCYT data from.

TYPE	NAME	DESCRIPTION
System.Boolean	standardFeatures	Convert loaded data to standard Features.

Remarks

Based on Mohammad's MCYT reader.

Implements

IDataSetLoader
ILoggerObject

Extension Methods

ILoggerObjectExtensions.LogError(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogError(ILoggerObject, Exception, String, Object[])
ILoggerObjectExtensions.LogInformation(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogWarning(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogWarning(ILoggerObject, Exception, String, Object[])
ILoggerObjectExtensions.LogTrace(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogTrace<TState>(ILoggerObject, TState, EventId, Exception, Func<TState, Exception, String>)
ILoggerObjectExtensions.LogCritical(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogDebug(ILoggerObject, String, Object[])

Class SigComp11DutchLoader.SigComp11

Set of features containing raw data loaded from MCYT-format database.

Inheritance

System.Object
SigComp11DutchLoader.SigComp11

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: SigStat.Common.Loaders

Assembly: SigStat.Common.dll

Syntax

```
public static class SigComp11
```

Fields

T

T values from the online signature imported from the SVC2004 database

Declaration

```
public static readonly FeatureDescriptor<List<int>> T
```

Field Value

TYPE	DESCRIPTION
FeatureDescriptor<System.Collections.Generic.List<System.Int32>>	

X

X cooridnates from the online signature imported from the MCYT database

Declaration

```
public static readonly FeatureDescriptor<List<int>> X
```

Field Value

TYPE	DESCRIPTION
FeatureDescriptor<System.Collections.Generic.List<System.Int32>>	

Y

Y cooridnates from the online signature imported from the MCYT database

Declaration

```
public static readonly FeatureDescriptor<List<int>> Y
```

Field Value

TYPE	DESCRIPTION
FeatureDescriptor<System.Collections.Generic.List<System.Int32>>	

Z

Z cooridnates from the online signature imported from the MCYT database

Declaration

```
public static readonly FeatureDescriptor<List<int>> Z
```

Field Value

TYPE	DESCRIPTION
FeatureDescriptor<System.Collections.Generic.List<System.Int32>>	

Class SigComp13JapaneseLoader

DataSetLoader for the SigComp13Japanese dataset

Inheritance

System.Object
DataSetLoader
SigComp13JapaneseLoader

Implements

IDataSetLoader
ILoggerObject

Inherited Members

DataSetLoader.Logger
DataSetLoader.EnumerateSigners()
DataSetLoader.IDataSetLoader.SignerFilter
System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: SigStat.Common.Loaders

Assembly: SigStat.Common.dll

Syntax

```
public class SigComp13JapaneseLoader : DataSetLoader, IDataSetLoader, ILoggerObject
```

Constructors

SigComp13JapaneseLoader(String, Boolean)

Initializes a new instance of the SigComp13JapaneseLoader class.

Declaration

```
public SigComp13JapaneseLoader(string databasePath, bool standardFeatures)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	databasePath	The database path.
System.Boolean	standardFeatures	if set to `true` features will be also stored in Features.

SigComp13JapaneseLoader(String, Boolean, Predicate<Signer>)

Initializes a new instance of the SigComp13JapaneseLoader class with specified database.

Declaration

```
public SigComp13JapaneseLoader(string databasePath, bool standardFeatures, Predicate<Signer> signerFilter =
null)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	databasePath	Represents the path, to load the signatures from. It supports two basic approaches: • •
System.Boolean	standardFeatures	Convert loaded data to standard Features.
System.Predicate<Signer>	signerFilter	Sets the SignerFilter property

Properties

DatabasePath

Gets or sets the database path.

Declaration

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

Property Value

TYPE	DESCRIPTION
System.String	

SamplingFrequency

Sampling Frequency of this database

Declaration

```
public override int SamplingFrequency { get; }
```

Property Value

TYPE	DESCRIPTION
System.Int32	

Overrides

DataSetLoader.SamplingFrequency

SignerFilter

Ignores any signers during the loading, that do not match the predicate

Declaration

```
public Predicate<Signer> SignerFilter { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Predicate<Signer>	

StandardFeatures

Gets or sets a value indicating whether features are also loaded as Features

Declaration

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

Property Value

TYPE	DESCRIPTION
System.Boolean	

Methods

EnumerateSigners(Predicate<Signer>)

Declaration

```
public override IEnumerable<Signer> EnumerateSigners(Predicate<Signer> signerFilter)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Predicate<Signer>	signerFilter	

Returns

TYPE	DESCRIPTION
System.Collections.Generic.IEnumerable<Signer>	

Overrides

DataSetLoader.EnumerateSigners(Predicate<Signer>)

LoadSignature(Signature, MemoryStream, Boolean)

Loads one signature from specified stream.

Declaration

```
public static void LoadSignature(Signature signature, MemoryStream stream, bool standardFeatures)
```

Parameters

TYPE	NAME	DESCRIPTION
Signature	signature	Signature to write features to.
System.IO.MemoryStream	stream	Stream to read SigComp13Japanese data from.

TYPE	NAME	DESCRIPTION
System.Boolean	standardFeatures	Convert loaded data to standard Features.

Remarks

Based on Mohammad's MCYT reader.

Implements

IDataSetLoader

ILoggerObject

Extension Methods

ILoggerObjectExtensions.LogError(ILoggerObject, String, Object[])

ILoggerObjectExtensions.LogError(ILoggerObject, Exception, String, Object[])

ILoggerObjectExtensions.LogInformation(ILoggerObject, String, Object[])

ILoggerObjectExtensions.LogWarning(ILoggerObject, String, Object[])

ILoggerObjectExtensions.LogWarning(ILoggerObject, Exception, String, Object[])

ILoggerObjectExtensions.LogTrace(ILoggerObject, String, Object[])

ILoggerObjectExtensions.LogTrace<TState>(ILoggerObject, TState, EventId, Exception, Func<TState, Exception, String>)

ILoggerObjectExtensions.LogCritical(ILoggerObject, String, Object[])

ILoggerObjectExtensions.LogDebug(ILoggerObject, String, Object[])

Class SigComp13JapaneseLoader.SigComp13Japanese

Set of features containing raw data loaded from SigComp13Japanese-format database.

Inheritance

System.Object
SigComp13JapaneseLoader.SigComp13Japanese

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: SigStat.Common.Loaders

Assembly: SigStat.Common.dll

Syntax

```
public static class SigComp13Japanese
```

Fields

P

Z cooridnates from the online signature imported from the SigComp13Japanese database (100 - pen down, 0 - pen up)

Declaration

```
public static readonly FeatureDescriptor<List<int>> P
```

Field Value

TYPE	DESCRIPTION
FeatureDescriptor<System.Collections.Generic.List<System.Int32>>	

T

Generated T values from the online signature imported from the SigComp13Japanese database

Declaration

```
public static readonly FeatureDescriptor<List<int>> T
```

Field Value

TYPE	DESCRIPTION
FeatureDescriptor<System.Collections.Generic.List<System.Int32>>	

X

X cooridnates from the online signature imported from the SigComp13Japanese database

Declaration

```
public static readonly FeatureDescriptor<List<int>> X
```

Field Value

TYPE	DESCRIPTION
FeatureDescriptor<System.Collections.Generic.List<System.Int32>>	

Y

Y cooridnates from the online signature imported from the SigComp13Japanese database

Declaration

```
public static readonly FeatureDescriptor<List<int>> Y
```

Field Value

TYPE	DESCRIPTION
FeatureDescriptor<System.Collections.Generic.List<System.Int32>>	

Class SigComp15GermanLoader

DataSetLoader for the SigComp15German dataset

Inheritance

System.Object
DataSetLoader
SigComp15GermanLoader

Implements

IDataSetLoader
ILoggerObject

Inherited Members

DataSetLoader.Logger
DataSetLoader.EnumerateSigners()
DataSetLoader.IDataSetLoader.SignerFilter
System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: SigStat.Common.Loaders

Assembly: SigStat.Common.dll

Syntax

```
public class SigComp15GermanLoader : DataSetLoader, IDataSetLoader, ILoggerObject
```

Constructors

SigComp15GermanLoader(String, Boolean)

Initializes a new instance of the SigComp15GermanLoader class.

Declaration

```
public SigComp15GermanLoader(string databasePath, bool standardFeatures)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	databasePath	The database path.
System.Boolean	standardFeatures	if set to `true` features will be also stored in Features.

SigComp15GermanLoader(String, Boolean, Predicate<Signer>)

Initializes a new instance of the SigComp15GermanLoader class with specified database.

Declaration

```
public SigComp15GermanLoader(string databasePath, bool standardFeatures, Predicate<Signer> signerFilter =
null)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	databasePath	Represents the path, to load the signatures from. It supports two basic approaches: ● ●
System.Boolean	standardFeatures	Convert loaded data to standard Features.
System.Predicate<Signer>	signerFilter	Sets the SignerFilter property

Properties

DatabasePath

Gets or sets the database path.

Declaration

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

Property Value

TYPE	DESCRIPTION
System.String	

SamplingFrequency

Sampling Frequency of this database

Declaration

```
public override int SamplingFrequency { get; }
```

Property Value

TYPE	DESCRIPTION
System.Int32	

Overrides

DataSetLoader.SamplingFrequency

SignerFilter

Ignores any signers during the loading, that do not match the predicate

Declaration

```
public Predicate<Signer> SignerFilter { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Predicate<Signer>	

StandardFeatures

Gets or sets a value indicating whether features are also loaded as Features

Declaration

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

Property Value

TYPE	DESCRIPTION
System.Boolean	

Methods

EnumerateSigners(Predicate<Signer>)

Declaration

```
public override IEnumerable<Signer> EnumerateSigners(Predicate<Signer> signerFilter)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Predicate<Signer>	signerFilter	

Returns

TYPE	DESCRIPTION
System.Collections.Generic.IEnumerable<Signer>	

Overrides

DataSetLoader.EnumerateSigners(Predicate<Signer>)

LoadSignature(Signature, MemoryStream, Boolean)

Loads one signature from specified stream.

Declaration

```
public static void LoadSignature(Signature signature, MemoryStream stream, bool standardFeatures)
```

Parameters

TYPE	NAME	DESCRIPTION
Signature	signature	Signature to write features to.
System.IO.MemoryStream	stream	Stream to read MCYT data from.

TYPE	NAME	DESCRIPTION
System.Boolean	standardFeatures	Convert loaded data to standard Features.

Remarks

Based on Mohammad's MCYT reader.

Implements

IDataSetLoader
ILoggerObject

Extension Methods

ILoggerObjectExtensions.LogError(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogError(ILoggerObject, Exception, String, Object[])
ILoggerObjectExtensions.LogInformation(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogWarning(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogWarning(ILoggerObject, Exception, String, Object[])
ILoggerObjectExtensions.LogTrace(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogTrace<TState>(ILoggerObject, TState, EventId, Exception, Func<TState, Exception, String>)
ILoggerObjectExtensions.LogCritical(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogDebug(ILoggerObject, String, Object[])

Class SigComp15GermanLoader.SigComp15

Set of features containing raw data loaded from SigComp15German-format database.

Inheritance

System.Object
SigComp15GermanLoader.SigComp15

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: SigStat.Common.Loaders

Assembly: SigStat.Common.dll

Syntax

```
public static class SigComp15
```

Fields

P

Z cooridnates from the online signature imported from the SigComp15German database

Declaration

```
public static readonly FeatureDescriptor<List<int>> P
```

Field Value

TYPE	DESCRIPTION
FeatureDescriptor<System.Collections.Generic.List<System.Int32>>	

T

T values from the online signature imported from the SigComp15German database

Declaration

```
public static readonly FeatureDescriptor<List<int>> T
```

Field Value

TYPE	DESCRIPTION
FeatureDescriptor<System.Collections.Generic.List<System.Int32>>	

X

X cooridnates from the online signature imported from the SigComp15German database

Declaration

```
public static readonly FeatureDescriptor<List<int>> X
```

Field Value

TYPE	DESCRIPTION
FeatureDescriptor<System.Collections.Generic.List<System.Int32>>	

Y

Y cooridnates from the online signature imported from the SigComp15German database

Declaration

```
public static readonly FeatureDescriptor<List<int>> Y
```

Field Value

TYPE	DESCRIPTION
FeatureDescriptor<System.Collections.Generic.List<System.Int32>>	

Class SigComp19OnlineLoader

DataSetLoader for the SigComp19 dataset

Inheritance

System.Object
DataSetLoader
SigComp19OnlineLoader

Implements

IDataSetLoader
ILoggerObject

Inherited Members

DataSetLoader.Logger
DataSetLoader.EnumerateSigners()
DataSetLoader.IDataSetLoader.SignerFilter
System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: SigStat.Common.Loaders

Assembly: SigStat.Common.dll

Syntax

```
public class SigComp19OnlineLoader : DataSetLoader, IDataSetLoader, ILoggerObject
```

Constructors

SigComp19OnlineLoader(String, Boolean)

Initializes a new instance of the SigComp19OnlineLoader class.

Declaration

```
public SigComp19OnlineLoader(string databasePath, bool standardFeatures)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	databasePath	The database path.
System.Boolean	standardFeatures	if set to `true` features will be also stored in Features.

Properties

DatabasePath

Gets or sets the database path.

Declaration

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

Property Value

TYPE	DESCRIPTION
System.String	

SamplingFrequency

sampling frequency for this database

Declaration

```
public override int SamplingFrequency { get; }
```

Property Value

TYPE	DESCRIPTION
System.Int32	

Overrides

DataSetLoader.SamplingFrequency

SignerFilter

Ignores any signers during the loading, that do not match the predicate

Declaration

```
public Predicate<Signer> SignerFilter { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Predicate<Signer>	

StandardFeatures

Gets or sets a value indicating whether features are also loaded as Features

Declaration

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

Property Value

TYPE	DESCRIPTION
System.Boolean	

Methods

EnumerateSigners(Predicate<Signer>)

Declaration

```
public override IEnumerable<Signer> EnumerateSigners(Predicate<Signer> signerFilter)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Predicate<Signer>	signerFilter	

Returns

TYPE	DESCRIPTION
System.Collections.Generic.IEnumerable<Signer>	

Overrides

DataSetLoader.EnumerateSigners(Predicate<Signer>)

LoadSignature(Signature, MemoryStream, Boolean)

Loads one signature from specified stream.

Declaration

```
public static void LoadSignature(Signature signature, MemoryStream stream, bool standardFeatures)
```

Parameters

TYPE	NAME	DESCRIPTION
Signature	signature	Signature to write features to.
System.IO.MemoryStream	stream	Stream to read SigComp19 data from.
System.Boolean	standardFeatures	Convert loaded data to standard Features.

Remarks

Based on Mohammad's MCYT reader.

Implements

IDataSetLoader
ILoggerObject

Extension Methods

ILoggerObjectExtensions.LogError(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogError(ILoggerObject, Exception, String, Object[])
ILoggerObjectExtensions.LogInformation(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogWarning(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogWarning(ILoggerObject, Exception, String, Object[])
ILoggerObjectExtensions.LogTrace(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogTrace<TState>(ILoggerObject, TState, EventId, Exception, Func<TState, Exception, String>)
ILoggerObjectExtensions.LogCritical(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogDebug(ILoggerObject, String, Object[])

Class SigComp19OnlineLoader.SigComp19

Set of features containing raw data loaded from SigComp19-format database.

Inheritance

System.Object
SigComp19OnlineLoader.SigComp19

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: SigStat.Common.Loaders

Assembly: SigStat.Common.dll

Syntax

```
public static class SigComp19
```

Fields

Altitude

Altitude from the online signature imported from the SigComp19 database

Declaration

```
public static readonly FeatureDescriptor<List<int>> Altitude
```

Field Value

TYPE	DESCRIPTION
FeatureDescriptor<System.Collections.Generic.List<System.Int32>>	

Azimuth

Azimuth from the online signature imported from the SigComp19 database

Declaration

```
public static readonly FeatureDescriptor<List<int>> Azimuth
```

Field Value

TYPE	DESCRIPTION
FeatureDescriptor<System.Collections.Generic.List<System.Int32>>	

Distance

Distance from the surface of the tablet from the online signature imported from the SigComp19 database

Declaration

```
public static readonly FeatureDescriptor<List<int>> Distance
```

Field Value

TYPE	DESCRIPTION
FeatureDescriptor<System.Collections.Generic.List<System.Int32>>	

EventType

EventType (pen up) values from the online signature imported from the SigComp19 database

Declaration

```
public static readonly FeatureDescriptor<List<int>> EventType
```

Field Value

TYPE	DESCRIPTION
FeatureDescriptor<System.Collections.Generic.List<System.Int32>>	

P

Pressure from the online signature imported from the SigComp19 database

Declaration

```
public static readonly FeatureDescriptor<List<int>> P
```

Field Value

TYPE	DESCRIPTION
FeatureDescriptor<System.Collections.Generic.List<System.Int32>>	

T

T values from the online signature imported from the SigComp19 database

Declaration

```
public static readonly FeatureDescriptor<List<int>> T
```

Field Value

TYPE	DESCRIPTION
FeatureDescriptor<System.Collections.Generic.List<System.Int32>>	

X

X cooridnates from the online signature imported from the SigComp19 database

Declaration

```
public static readonly FeatureDescriptor<List<int>> X
```

Field Value

TYPE	DESCRIPTION
FeatureDescriptor<System.Collections.Generic.List<System.Int32>>	

Y

Y cooridnates from the online signature imported from the SigComp19 database

Declaration

```
public static readonly FeatureDescriptor<List<int>> Y
```

Field Value

TYPE	DESCRIPTION
FeatureDescriptor<System.Collections.Generic.List<System.Int32>>	

Class Svc2004

Set of features containing raw data loaded from SVC2004-format database.

Inheritance

System.Object
Svc2004

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: SigStat.Common.Loaders

Assembly: SigStat.Common.dll

Syntax

```
public static class Svc2004
```

Fields

All

A list of all Svc2004 feature descriptors

Declaration

```
public static readonly FeatureDescriptor[] All
```

Field Value

TYPE	DESCRIPTION
FeatureDescriptor[]	

Altitude

Altitude values from the online signature imported from the SVC2004 database

Declaration

```
public static readonly FeatureDescriptor<List<int>> Altitude
```

Field Value

TYPE	DESCRIPTION
FeatureDescriptor<System.Collections.Generic.List<System.Int32>>	

Azimuth

Azimuth values from the online signature imported from the SVC2004 database

Declaration

```
public static readonly FeatureDescriptor<List<int>> Azimuth
```

Field Value

TYPE	DESCRIPTION
FeatureDescriptor<System.Collections.Generic.List<System.Int32>>	

Button

Button values from the online signature imported from the SVC2004 database

Declaration

```
public static readonly FeatureDescriptor<List<int>> Button
```

Field Value

TYPE	DESCRIPTION
FeatureDescriptor<System.Collections.Generic.List<System.Int32>>	

Pressure

Pressure values from the online signature imported from the SVC2004 database

Declaration

```
public static readonly FeatureDescriptor<List<int>> Pressure
```

Field Value

TYPE	DESCRIPTION
FeatureDescriptor<System.Collections.Generic.List<System.Int32>>	

T

T values from the online signature imported from the SVC2004 database

Declaration

```
public static readonly FeatureDescriptor<List<int>> T
```

Field Value

TYPE	DESCRIPTION
FeatureDescriptor<System.Collections.Generic.List<System.Int32>>	

X

X cooridnates from the online signature imported from the SVC2004 database

Declaration

```
public static readonly FeatureDescriptor<List<int>> X
```

Field Value

TYPE	DESCRIPTION
FeatureDescriptor<System.Collections.Generic.List<System.Int32>>	

Y

Y cooridnates from the online signature imported from the SVC2004 database

Declaration

```
public static readonly FeatureDescriptor<List<int>> Y
```

Field Value

TYPE	DESCRIPTION
FeatureDescriptor<System.Collections.Generic.List<System.Int32>>	

Class Svc2004Loader

Loads SVC2004-format database from .zip

Inheritance

System.Object
DataSetLoader
Svc2004Loader

Implements

IDataSetLoader
ILoggerObject

Inherited Members

DataSetLoader.Logger
DataSetLoader.EnumerateSigners()
DataSetLoader.IDataSetLoader.SignerFilter
System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: SigStat.Common.Loaders

Assembly: SigStat.Common.dll

Syntax

```
[JsonObject(MemberSerialization.OptOut)]
public class Svc2004Loader : DataSetLoader, IDataSetLoader, ILoggerObject
```

Constructors

Svc2004Loader(String, Boolean)

Initializes a new instance of the Svc2004Loader class with specified database.

Declaration

```
[JsonConstructor]
public Svc2004Loader(string databasePath, bool standardFeatures)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	databasePath	Represents the path, to load the signatures from. It supports two basic approaches: • •
System.Boolean	standardFeatures	Convert loaded data (Svc2004) to standard Features.

Svc2004Loader(String, Boolean, Predicate<Signer>)

Initializes a new instance of the Svc2004Loader class with specified database.

Declaration

```
public Svc2004Loader(string databasePath, bool standardFeatures, Predicate<Signer> signerFilter = null)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	databasePath	Represents the path, to load the signatures from. It supports two basic approaches: • •
System.Boolean	standardFeatures	Convert loaded data (Svc2004) to standard Features.
System.Predicate<Signer>	signerFilter	Sets the SignerFilter property

Properties

DatabasePath

Gets or sets the database path.

Declaration

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

Property Value

TYPE	DESCRIPTION
System.String	

SamplingFrequency

Sampling Frequency of the SVC database

Declaration

```
public override int SamplingFrequency { get; }
```

Property Value

TYPE	DESCRIPTION
System.Int32	

Overrides

DataSetLoader.SamplingFrequency

SignerFilter

Ignores any signers during the loading, that do not match the predicate

Declaration

```
public Predicate<Signer> SignerFilter { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Predicate<Signer>	

StandardFeatures

Gets or sets a value indicating whether features are also loaded as Features

Declaration

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

Property Value

TYPE	DESCRIPTION
System.Boolean	

Methods

EnumerateSigners(Predicate<Signer>)

Declaration

```
public override IEnumerable<Signer> EnumerateSigners(Predicate<Signer> signerFilter)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Predicate<Signer>	signerFilter	

Returns

TYPE	DESCRIPTION
System.Collections.Generic.IEnumerable<Signer>	

Overrides

DataSetLoader.EnumerateSigners(Predicate<Signer>)

LoadSignature(Signature, Stream, Boolean)

Loads one signature from specified stream.

Declaration

```
public static void LoadSignature(Signature signature, Stream stream, bool standardFeatures)
```

Parameters

TYPE	NAME	DESCRIPTION
Signature	signature	Signature to write features to.

TYPE	NAME	DESCRIPTION
System.IO.Stream	stream	Stream to read svc2004 data from.
System.Boolean	standardFeatures	Convert loaded data to standard Features.

LoadSignature(Signature, String, Boolean)

Loads one signature from specified file path.

Declaration

```
public void LoadSignature(Signature signature, string path, bool standardFeatures)
```

Parameters

TYPE	NAME	DESCRIPTION
Signature	signature	Signature to write features to.
System.String	path	Path to a file of format "US.txt"
System.Boolean	standardFeatures	Convert loaded data to standard Features.

Implements

IDataSetLoader

ILoggerObject

Extension Methods

ILoggerObjectExtensions.LogError(ILoggerObject, String, Object[])

ILoggerObjectExtensions.LogError(ILoggerObject, Exception, String, Object[])

ILoggerObjectExtensions.LogInformation(ILoggerObject, String, Object[])

ILoggerObjectExtensions.LogWarning(ILoggerObject, String, Object[])

ILoggerObjectExtensions.LogWarning(ILoggerObject, Exception, String, Object[])

ILoggerObjectExtensions.LogTrace(ILoggerObject, String, Object[])

ILoggerObjectExtensions.LogTrace<TState>(ILoggerObject, TState, EventId, Exception, Func<TState, Exception, String>)

ILoggerObjectExtensions.LogCritical(ILoggerObject, String, Object[])

ILoggerObjectExtensions.LogDebug(ILoggerObject, String, Object[])

Namespace SigStat.Common.Logging

Classes

BenchmarkKeyValueLogState

Specific state used for Benchmarks key-value information transiting

BenchmarkLogModel

Represents the results of a benchmark

BenchmarkResultsLogState

Specific state used for Benchmark result transiting

ClassifierDistanceLogState

Specific state for signature distance information transiting

CompositeLogger

Forwards messages to Microsoft.Extensions.Logging.ILogger components.

ExcelReportGenerator

This class is used to generate a report in Excel file format, form a Benchmark model.

KeyValueGroup

A group of key-value pairs

LogAnalyzer

Analizes logs and creates a model from the gained information

ReportInformationLogger

Logger for logging report informations.

SignatureLogState

Specific state used for signature information transiting

SignerLogState

Specific state used for signer information transiting

SignerResults

Informations of a signer

SignerResultsLogState

Specific state used for Signer result transiting

SigStatLogState

Base state used in report information logging.

SimpleConsoleLogger

Logs messages to System.Console. The font color is determined by the severity level.

Delegates

CompositeLogger.ErrorEventHandler

The event is raised whenever an error is logged.

ReportInformationLogger.LogStateLoggedEventHandler

The event is raised whenever a SigStatLogState is logged.

SimpleConsoleLogger.ConsoleMessageLoggedEventHandler

The event is raised whenever a console message is logged

Class BenchmarkKeyValueLogState

Specific state used for Benchmarks key-value information transiting

Inheritance

System.Object
SigStatLogState
BenchmarkKeyValueLogState

Inherited Members

SigStatLogState.Source
System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)

Namespace: SigStat.Common.Logging

Assembly: SigStat.Common.dll

Syntax

```
public class BenchmarkKeyValueLogState : SigStatLogState
```

Constructors

BenchmarkKeyValueLogState(String, String, Object)

Creates a BenchmarkKeyValueLogState

Declaration

```
public BenchmarkKeyValueLogState(string group, string key, object value)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	group	Group
System.String	key	Key
System.Object	value	Value

Properties

Group

Group of the key-value pair

Declaration

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

Property Value

TYPE	DESCRIPTION
System.String	

Key

Key

Declaration

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

Property Value

TYPE	DESCRIPTION
System.String	

Value

Value

Declaration

```
public object Value { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Object	

Methods

ToString()

Declaration

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

Returns

TYPE	DESCRIPTION
System.String	

Overrides

System.Object.ToString()

Class BenchmarkLogModel

Represents the results of a benchmark

Inheritance

System.Object

BenchmarkLogModel

Inherited Members

System.Object.Equals(System.Object)

System.Object.Equals(System.Object, System.Object)

System.Object.GetHashCode()

System.Object.GetType()

System.Object.MemberwiseClone()

System.Object.ReferenceEquals(System.Object, System.Object)

System.Object.ToString()

Namespace: SigStat.Common.Logging

Assembly: SigStat.Common.dll

Syntax

```
public class BenchmarkLogModel
```

Constructors

BenchmarkLogModel()

Default constructor creating a blank model.

Declaration

```
public BenchmarkLogModel()
```

Fields

BenchmarkResultsGroupName

Name of the "BenchmarkResults" group

Declaration

```
[JsonIgnore]
public const string BenchmarkResultsGroupName = "BenchmarkResults"
```

Field Value

TYPE	DESCRIPTION
System.String	

ExecutionGroupName

Name of the "Excecution" group

Declaration

```
[JsonIgnore]
public const string ExecutionGroupName = "Execution"
```

Field Value

TYPE	DESCRIPTION
System.String	

ParametersGroupName

Name of the "Parameters" group

Declaration

```
[JsonIgnore]
public const string ParametersGroupName = "Parameters"
```

Field Value

TYPE	DESCRIPTION
System.String	

Properties

BenchmarkResults

Benchmark results group

Declaration

```
[JsonIgnore]
public KeyValueGroup BenchmarkResults { get; }
```

Property Value

TYPE	DESCRIPTION
KeyValueGroup	

Excecution

Excecution group

Declaration

```
[JsonIgnore]
public KeyValueGroup Excecution { get; }
```

Property Value

TYPE	DESCRIPTION
KeyValueGroup	

KeyValueGroups

Benchmark results stored in Key-Value groups

Declaration

```
public Dictionary<string, KeyValueGroup> KeyValueGroups { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Collections.Generic.Dictionary<System.String, KeyValueGroup>	

Parameters

Parameters group

Declaration

```
[JsonIgnore]
public KeyValueGroup Parameters { get; }
```

Property Value

TYPE	DESCRIPTION
KeyValueGroup	

SignerResults

Results belonging to signers

Declaration

```
public SortedDictionary<string, SignerResults> SignerResults { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Collections.Generic.SortedDictionary<System.String, SignerResults>	

Class BenchmarkResultsLogState

Specific state used for Benchmark result transiting

Inheritance

System.Object
SigStatLogState
BenchmarkResultsLogState

Inherited Members

SigStatLogState.Source
System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)

Namespace: SigStat.Common.Logging

Assembly: SigStat.Common.dll

Syntax

```
public class BenchmarkResultsLogState : SigStatLogState
```

Constructors

BenchmarkResultsLogState(Double, Double, Double)

Creates a BenchmarkResultsLogState

Declaration

```
public BenchmarkResultsLogState(double aer, double far, double frr)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Double	aer	Aer
System.Double	far	Far
System.Double	frr	Frr

Properties

Aer

Average error rate

Declaration

```
public double Aer { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Double	

Far

False accaptance rate

Declaration

```
public double Far { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Double	

Frr

False rejection rate

Declaration

```
public double Frr { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Double	

Methods

ToString()

Declaration

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

Returns

TYPE	DESCRIPTION
System.String	

Overrides

System.Object.ToString()

Class ClassifierDistanceLogState

Specific state for signature distance information transiting

Inheritance

System.Object
SigStatLogState
ClassifierDistanceLogState

Inherited Members

SigStatLogState.Source
System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)

Namespace: SigStat.Common.Logging

Assembly: SigStat.Common.dll

Syntax

```
public class ClassifierDistanceLogState : SigStatLogState
```

Constructors

ClassifierDistanceLogState(String, String, String, String, Double)

Creates a ClassifierDistanceLogState

Declaration

```
public ClassifierDistanceLogState(string signer1Id, string signer2Id, string signature1Id, string
signature2Id, double distance)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	signer1Id	Id of the first signature's signer
System.String	signer2Id	Id of the second signature's signer
System.String	signature1Id	Id of the first signature
System.String	signature2Id	Id of the second signature
System.Double	distance	Distance values between the signatures

Properties

Distance

Distance values between the signatures

Declaration

```
public double Distance { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Double	

Signature1Id

Id of the first signature

Declaration

```
public string Signature1Id { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.String	

Signature2Id

Id of the second signature

Declaration

```
public string Signature2Id { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.String	

Signer1Id

Id of the first signature's signer

Declaration

```
public string Signer1Id { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.String	

Signer2Id

/// Id of the second signature's signer

Declaration

```
public string Signer2Id { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.String	

Methods

ToString()

Declaration

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

Returns

TYPE	DESCRIPTION
System.String	

Overrides

System.Object.ToString()

Class CompositeLogger

Forwards messages to Microsoft.Extensions.Logging.ILogger components.

Inheritance

System.Object
CompositeLogger

Implements

Microsoft.Extensions.Logging.ILogger

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: SigStat.Common.Logging

Assembly: SigStat.Common.dll

Syntax

```
public class CompositeLogger : ILogger
```

Properties

Loggers

The list of Microsoft.Extensions.Logging.ILogger components that messages are forwarded to. Empty by default.

Declaration

```
public List<ILogger> Loggers { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Collections.Generic.List<Microsoft.Extensions.Logging.ILogger>	

Methods

BeginScope<TState>(TState)

Calls Microsoft.Extensions.Logging.ILogger.BeginScope<TState>(TState) on each component.

Declaration

```
public IDisposable BeginScope<TState>(TState state)
```

Parameters

TYPE	NAME	DESCRIPTION
TState	state	

Returns

TYPE	DESCRIPTION
System.IDisposable	

Type Parameters

NAME	DESCRIPTION
TState	

IsEnabled(LogLevel)

Returns true if any of the Microsoft.Extensions.Logging.ILogger components are enabled on the specified Microsoft.Extensions.Logging.LogLevel.

Declaration

```
public bool IsEnabled(LogLevel logLevel)
```

Parameters

TYPE	NAME	DESCRIPTION
Microsoft.Extensions.Logging.LogLevel	logLevel	

Returns

TYPE	DESCRIPTION
System.Boolean	

Log<TState>(LogLevel, EventId, TState, Exception, Func<TState, Exception, String>)

Forwards the message to each Microsoft.Extensions.Logging.ILogger component.

Declaration

```
public void Log<TState>(LogLevel logLevel, EventId eventId, TState state, Exception exception, Func<TState,
Exception, string> formatter)
```

Parameters

TYPE	NAME	DESCRIPTION
Microsoft.Extensions.Logging.LogLevel	logLevel	
Microsoft.Extensions.Logging.EventId	eventId	
TState	state	
System.Exception	exception	
System.Func<TState, System.Exception, System.String>	formatter	

Type Parameters

NAME	DESCRIPTION
TState	

Events

Logged

Occurs when an error is logged.

Declaration

```
public event CompositeLogger.ErrorEventHandler Logged
```

Event Type

TYPE	DESCRIPTION
CompositeLogger.ErrorEventHandler	

Implements

Microsoft.Extensions.Logging.ILogger

Delegate CompositeLogger.ErrorEventHandler

The event is raised whenever an error is logged.

Namespace: SigStat.Common.Logging

Assembly: SigStat.Common.dll

Syntax

```
public delegate void ErrorEventHandler(string message, Exception exception, LogLevel level);
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	message	The message.
System.Exception	exception	The exception.
Microsoft.Extensions.Logging.LogLevel	level	The level.

Class ExcelReportGenerator

This class is used to generate a report in Excel file format, form a Benchmark model.

Inheritance

System.Object
ExcelReportGenerator

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: SigStat.Common.Logging

Assembly: SigStat.Common.dll

Syntax

```
public class ExcelReportGenerator
```

Methods

GenerateReport(BenchmarkLogModel, String)

Generates an Excel file that contains the report.

Declaration

```
public static void GenerateReport(BenchmarkLogModel model, string fileName = null)
```

Parameters

TYPE	NAME	DESCRIPTION
BenchmarkLogModel	model	The model of the report
System.String	fileName	The name of the generated excel file

Class KeyValueGroup

A group of key-value pairs

Inheritance

System.Object
KeyValueGroup

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: SigStat.Common.Logging

Assembly: SigStat.Common.dll

Syntax

```
[JsonConverter(typeof(KeyValueGroupConverter))]
public class KeyValueGroup
```

Constructors

KeyValueGroup(String)

Creates an emty key-value group

Declaration

```
public KeyValueGroup(string name)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	name	Name if the new group

Properties

Items

Key-Value pairs in the group

Declaration

```
public List<KeyValuePair<string, object>> Items { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Collections.Generic.List<System.Collections.Generic.KeyValuePair<System.String, System.Object>>	

Name

Name of the group

Declaration

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

Property Value

TYPE	DESCRIPTION
System.String	

Class LogAnalyzer

Analizes logs and creates a model from the gained information

Inheritance

System.Object
LogAnalyzer

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: SigStat.Common.Logging

Assembly: SigStat.Common.dll

Syntax

```
public class LogAnalyzer
```

Methods

GetBenchmarkLogModel(IEnumerable<SigStatLogState>)

Creates a BenchmarkLogModel from previous logs

Declaration

```
public static BenchmarkLogModel GetBenchmarkLogModel(IEnumerable<SigStatLogState> logs)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Collections.Generic.IEnumerable<SigStatLogState>	logs	The collection of logs, that contains the required information for a BenchmarkLogModel

Returns

TYPE	DESCRIPTION
BenchmarkLogModel	The Benchmark model filled with information according to the logs

Class ReportInformationLogger

Logger for logging report informations.

Inheritance

System.Object
ReportInformationLogger

Implements

Microsoft.Extensions.Logging.ILogger

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: SigStat.Common.Logging

Assembly: SigStat.Common.dll

Syntax

```
public class ReportInformationLogger : ILogger
```

Remarks

The class is thread safe

Constructors

ReportInformationLogger()

Initializes a new instance of ReportInformationLogger.

Declaration

```
public ReportInformationLogger()
```

Methods

BeginScope<TState>(TState)

Declaration

```
public IDisposable BeginScope<TState>(TState state)
```

Parameters

TYPE	NAME	DESCRIPTION
TState	state	

Returns

TYPE	DESCRIPTION
System.IDisposable	

Type Parameters

NAME	DESCRIPTION
TState	

GetReportLogs()

Enumerates the log entries

Declaration

```
public IEnumerable<SigStatLogState> GetReportLogs()
```

Returns

TYPE	DESCRIPTION
System.Collections.Generic.IEnumerable<SigStatLogState>	

IsEnabled(LogLevel)

Declaration

```
public bool IsEnabled(LogLevel logLevel)
```

Parameters

TYPE	NAME	DESCRIPTION
Microsoft.Extensions.Logging.LogLevel	logLevel	

Returns

TYPE	DESCRIPTION
System.Boolean	

Log<TState>(LogLevel, EventId, TState, Exception, Func<TState, Exception, String>)

Declaration

```
public void Log<TState>(LogLevel logLevel, EventId eventId, TState state, Exception exception, Func<TState,
Exception, string> formatter)
```

Parameters

TYPE	NAME	DESCRIPTION
Microsoft.Extensions.Logging.LogLevel	logLevel	
Microsoft.Extensions.Logging.EventId	eventId	
TState	state	
System.Exception	exception	
System.Func<TState, System.Exception, System.String>	formatter	

Type Parameters

NAME	DESCRIPTION
TState	

Events

Logged

Occurs when an error is logged.

Declaration

```
public event ReportInformationLogger.LogStateLoggedEventHandler Logged
```

Event Type

TYPE	DESCRIPTION
ReportInformationLogger.LogStateLoggedEventHandler	

Implements

Microsoft.Extensions.Logging.ILogger

Delegate ReportInformationLogger.LogStateLoggedEventHandler

The event is raised whenever a SigStatLogState is logged.

Namespace: SigStat.Common.Logging

Assembly: SigStat.Common.dll

Syntax

```
public delegate void LogStateLoggedEventHandler(SigStatLogState logState);
```

Parameters

TYPE	NAME	DESCRIPTION
SigStatLogState	logState	

Class SignatureLogState

Specific state used for signature information transiting

Inheritance

System.Object

SigStatLogState

SignatureLogState

Inherited Members

SigStatLogState.Source

System.Object.Equals(System.Object)

System.Object.Equals(System.Object, System.Object)

System.Object.GetHashCode()

System.Object.GetType()

System.Object.MemberwiseClone()

System.Object.ReferenceEquals(System.Object, System.Object)

System.Object.ToString()

Namespace: SigStat.Common.Logging

Assembly: SigStat.Common.dll

Syntax

```
public class SignatureLogState : SigStatLogState
```

Properties

SignatureID

Id of the signature

Declaration

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

Property Value

TYPE	DESCRIPTION
System.String	

SignerID

Id of the owning signer

Declaration

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

Property Value

TYPE	DESCRIPTION
System.String	

Class SignerLogState

Specific state used for signer information transiting

Inheritance

System.Object
SigStatLogState
SignerLogState
SignerResultsLogState

Inherited Members

SigStatLogState.Source
System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: SigStat.Common.Logging

Assembly: SigStat.Common.dll

Syntax

```
public class SignerLogState : SigStatLogState
```

Properties

SignerID

Id of the signer

Declaration

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

Property Value

TYPE	DESCRIPTION
System.String	

Class SignerResults

Informations of a signer

Inheritance

System.Object
SignerResults

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: SigStat.Common.Logging

Assembly: SigStat.Common.dll

Syntax

```
public class SignerResults
```

Constructors

SignerResults(String)

Creates a signer result with emty result values

Declaration

```
public SignerResults(string signerId)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	signerId	The id of the signer

Fields

Aer

Average Error Rate of the signer

Declaration

```
public object Aer
```

Field Value

TYPE	DESCRIPTION
System.Object	

Far

False Acceptance Rate of the signer

Declaration

```
public object Far
```

Field Value

TYPE	DESCRIPTION
System.Object	

Frr

False Rejection Rate of the signer

Declaration

```
public object Frr
```

Field Value

TYPE	DESCRIPTION
System.Object	

Properties

DistanceMatrix

Distacne matrix of the signers signatures

Declaration

```
[JsonConverter(typeof(DistanceMatrixConverter))]
public DistanceMatrix<string, string, double> DistanceMatrix { get; set; }
```

Property Value

TYPE	DESCRIPTION
DistanceMatrix<System.String, System.String, System.Double>	

SignerID

The ID of the signer

Declaration

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

Property Value

TYPE	DESCRIPTION
System.String	

Class SignerResultsLogState

Specific state used for Signer result transiting

Inheritance

System.Object
SigStatLogState
SignerLogState
SignerResultsLogState

Inherited Members

SignerLogState.SignerID
SigStatLogState.Source
System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)

Namespace: SigStat.Common.Logging

Assembly: SigStat.Common.dll

Syntax

```
public class SignerResultsLogState : SignerLogState
```

Constructors

SignerResultsLogState(String, Double, Double, Double)

Creates a SignerResultsLogState

Declaration

```
public SignerResultsLogState(string signerId, double aer, double far, double frr)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	signerId	Id of the signer
System.Double	aer	Aer
System.Double	far	Far
System.Double	frr	Frr

Properties

Aer

Average error rate

Declaration

```
public double Aer { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Double	

Far

False accaptance rate

Declaration

```
public double Far { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Double	

Frr

False rejection rate

Declaration

```
public double Frr { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Double	

Methods

ToString()

Declaration

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

Returns

TYPE	DESCRIPTION
System.String	

Overrides

System.Object.ToString()

Class SigStatLogState

Base state used in report information logging.

Inheritance

System.Object

SigStatLogState

BenchmarkKeyValueLogState

BenchmarkResultsLogState

ClassifierDistanceLogState

SignatureLogState

SignerLogState

Inherited Members

System.Object.Equals(System.Object)

System.Object.Equals(System.Object, System.Object)

System.Object.GetHashCode()

System.Object.GetType()

System.Object.MemberwiseClone()

System.Object.ReferenceEquals(System.Object, System.Object)

System.Object.ToString()

Namespace: SigStat.Common.Logging

Assembly: SigStat.Common.dll

Syntax

```
public class SigStatLogState
```

Properties

Source

Object from which the state originates.

Declaration

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

Property Value

TYPE	DESCRIPTION
System.String	

Class SimpleConsoleLogger

Logs messages to System.Console. The font color is determined by the severity level.

Inheritance

System.Object
SimpleConsoleLogger

Implements

Microsoft.Extensions.Logging.ILogger

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: SigStat.Common.Logging

Assembly: SigStat.Common.dll

Syntax

```
public class SimpleConsoleLogger : ILogger
```

Constructors

SimpleConsoleLogger()

Initializes a new instance of SimpleConsoleLogger with LogLevel set to Microsoft.Extensions.Logging.LogLevel.Information.

Declaration

```
public SimpleConsoleLogger()
```

SimpleConsoleLogger(LogLevel)

Initializes a new instance of SimpleConsoleLogger with a custom Microsoft.Extensions.Logging.LogLevel.

Declaration

```
public SimpleConsoleLogger(LogLevel logLevel)
```

Parameters

TYPE	NAME	DESCRIPTION
Microsoft.Extensions.Logging.LogLevel	logLevel	Initial value for LogLevel.

Properties

LogLevel

All events below this level will be filtered

Declaration

```
public LogLevel LogLevel { get; set; }
```

Property Value

TYPE	DESCRIPTION
Microsoft.Extensions.Logging.LogLevel	

Methods

BeginScope<TState>(TState)

Declaration

```
public IDisposable BeginScope<TState>(TState state)
```

Parameters

TYPE	NAME	DESCRIPTION
TState	state	

Returns

TYPE	DESCRIPTION
System.IDisposable	

Type Parameters

NAME	DESCRIPTION
TState	

IsEnabled(LogLevel)

Declaration

```
public bool IsEnabled(LogLevel logLevel)
```

Parameters

TYPE	NAME	DESCRIPTION
Microsoft.Extensions.Logging.LogLevel	logLevel	

Returns

TYPE	DESCRIPTION
System.Boolean	

Log<TState>(LogLevel, EventId, TState, Exception, Func<TState, Exception, String>)

Declaration

```
public void Log<TState>(LogLevel logLevel, EventId eventId, TState state, Exception exception, Func<TState,
Exception, string> formatter)
```

Parameters

TYPE	NAME	DESCRIPTION
Microsoft.Extensions.Logging.LogLevel	logLevel	
Microsoft.Extensions.Logging.EventId	eventId	
TState	state	
System.Exception	exception	
System.Func<TState, System.Exception, System.String>	formatter	

Type Parameters

NAME	DESCRIPTION
TState	

Events

Logged

Occurs when a console message is logged

Declaration

```
public event SimpleConsoleLogger.ConsoleMessageLoggedEventHandler Logged
```

Event Type

TYPE	DESCRIPTION
SimpleConsoleLogger.ConsoleMessageLoggedEventHandler	

Implements

Microsoft.Extensions.Logging.ILogger

Delegate SimpleConsoleLogger.ConsoleMessageLoggedEventHandler

The event is raised whenever a console message is logged

Namespace: SigStat.Common.Logging

Assembly: SigStat.Common.dll

Syntax

```
public delegate void ConsoleMessageLoggedEventHandler(string consoleMessage);
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	consoleMessage	

Namespace SigStat.Common.Model

Classes

SampleRateResults

used to store results for testing different sampling frequencies

Verifier

Uses pipelines to transform, train on, and classify Signature objects.

Class SampleRateResults

used to store results for testing different sampling frequencies

Inheritance

System.Object
SampleRateResults

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: SigStat.Common.Model

Assembly: SigStat.Common.dll

Syntax

```
public class SampleRateResults
```

Properties

AER

AER for current sampling frequency test

Declaration

```
[Input(AutoSetMode.IfNull)]
public double AER { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Double	

pointsAvg

average points of the signer

Declaration

```
[Input(AutoSetMode.IfNull)]
public double pointsAvg { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Double	

samplerate

current samplerate tested

Declaration

```
[Input(AutoSetMode.IfNull)]
public int samplerate { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Int32	

step

number of skipped points

Declaration

```
[Input(AutoSetMode.IfNull)]
public int step { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Int32	

Class Verifier

Uses pipelines to transform, train on, and classify Signature objects.

Inheritance

System.Object
Verifier

Implements

ILoggerObject

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: SigStat.Common.Model

Assembly: SigStat.Common.dll

Syntax

```
[JsonObject(MemberSerialization.OptOut)]
public class Verifier : ILoggerObject
```

Constructors

Verifier()

Initializes a new instance of the Verifier class.

Declaration

```
public Verifier()
```

Verifier(ILogger)

Initializes a new instance of the Verifier class

Declaration

```
[JsonConstructor]
public Verifier(ILogger logger = null)
```

Parameters

TYPE	NAME	DESCRIPTION
Microsoft.Extensions.Logging.ILogger	logger	Initializes the Logger property of the Verifier

Verifier(Verifier)

Initializes a new instance of the Verifier class based on another Verifier instance

Declaration

```
public Verifier(Verifier baseVerifier)
```

Parameters

TYPE	NAME	DESCRIPTION
Verifier	baseVerifier	The reference verifier

Properties

AllFeatures

This property is used by the Serializer to access a list of all FeatureDescriptors

Declaration

```
public Dictionary<string, FeatureDescriptor> AllFeatures { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Collections.Generic.Dictionary<System.String, FeatureDescriptor>	

Classifier

Gets or sets the classifier pipeline. Hands over the Logger object.

Declaration

```
public IClassifier Classifier { get; set; }
```

Property Value

TYPE	DESCRIPTION
IClassifier	

Logger

Gets or sets the class responsible for logging

Declaration

```
public ILogger Logger { get; set; }
```

Property Value

TYPE	DESCRIPTION
Microsoft.Extensions.Logging.ILogger	

Pipeline

Gets or sets the transform pipeline. Hands over the Logger object.

Declaration

```
public SequentialTransformPipeline Pipeline { get; set; }
```

Property Value

TYPE	DESCRIPTION
SequentialTransformPipeline	

SignerModel

Gets or sets the signer model.

Declaration

```
public ISignerModel SignerModel { get; set; }
```

Property Value

TYPE	DESCRIPTION
ISignerModel	The signer model.

Methods

Test(Signature)

Verifies the genuinity of `signature`.

Declaration

```
public virtual double Test(Signature signature)
```

Parameters

TYPE	NAME	DESCRIPTION
Signature	signature	

Returns

TYPE	DESCRIPTION
System.Double	True if `signature` passes the verification test.

Train(List<Signature>)

Trains the verifier with a list of signatures. Uses the Pipeline to extract features, and Classifier to find an optimized limit.

Declaration

```
public virtual void Train(List<Signature> signatures)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Collections.Generic.List<Signature>	signatures	The list of signatures to train on.

Remarks

Note that `signatures` may contain both genuine and forged signatures. It's up to the classifier, whether it takes advantage of both classes

Implements

ILoggerObject

Extension Methods

ILoggerObjectExtensions.LogError(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogError(ILoggerObject, Exception, String, Object[])
ILoggerObjectExtensions.LogInformation(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogWarning(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogWarning(ILoggerObject, Exception, String, Object[])
ILoggerObjectExtensions.LogTrace(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogTrace<TState>(ILoggerObject, TState, EventId, Exception, Func<TState, Exception, String>)
ILoggerObjectExtensions.LogCritical(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogDebug(ILoggerObject, String, Object[])

Namespace SigStat.Common.Pipeline

Classes

Input

Annotates an input FeatureDescriptor in a transformation pipeline

Output

Annotates an output FeatureDescriptor in a transformation pipeline

ParallelTransformPipeline

Runs pipeline items in parallel.

Default Pipeline Output: Range of all the Item outputs.

PipelineInput

Represents an input for a SigStat.Common.Pipeline.PipelineInput.PipelineItem

PipelineOutput

Represents an output for a SigStat.Common.Pipeline.PipelineOutput.PipelineItem

SequentialTransformPipeline

Runs pipeline items in a sequence.

Default Pipeline Output: Output of the last Item in the sequence.

Interfaces

IClassifier

Trains classification models based on reference signatures

IDistanceClassifier

Trains classification models based on reference signatures, by calculating the distances between signature pairs

IPipelineIO

Supports the definition of PipelineInput and PipelineOutput

ISignerModel

Analyzes signatures based on their similiarity to the trained model

Enums

AutoSetMode

Default strategy to set the value of a property

Enum AutoSetMode

Default strategy to set the value of a property

Namespace: SigStat.Common.Pipeline

Assembly: SigStat.Common.dll

Syntax

```
public enum AutoSetMode
```

Fields

NAME	DESCRIPTION
Always	Always set the value
IfNull	Set the value if it is null
Never	Never set the value

Interface IClassifier

Trains classification models based on reference signatures

Namespace: SigStat.Common.Pipeline

Assembly: SigStat.Common.dll

Syntax

```
public interface IClassifier
```

Methods

Test(ISignerModel, Signature)

Returns a double value in the range [0..1], representing the probability of the given signature belonging to the trained model.

-
-
-

Declaration

```
double Test(ISignerModel model, Signature signature)
```

Parameters

TYPE	NAME	DESCRIPTION
ISignerModel	model	The model aquired from the Train(List<Signature>) method
Signature	signature	The signature to test

Returns

TYPE	DESCRIPTION
System.Double	

Train(List<Signature>)

Trains a model based on the signatures and returns the trained model

Declaration

```
ISignerModel Train(List<Signature> signatures)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Collections.Generic.List<Signature>	signatures	

Returns

TYPE	DESCRIPTION
ISignerModel	

Interface IDistanceClassifier

Trains classification models based on reference signatures, by calculating the distances between signature pairs

Inherited Members

IClassifier.Train(List<Signature>)

IClassifier.Test(ISignerModel, Signature)

Namespace: SigStat.Common.Pipeline

Assembly: SigStat.Common.dll

Syntax

```
public interface IDistanceClassifier : IClassifier
```

Properties

DistanceFunction

A function to calculate the distance between two online signature points

Declaration

```
Func<double[], double[], double> DistanceFunction { get; }
```

Property Value

TYPE	DESCRIPTION
System.Func<System.Double[], System.Double[], System.Double>	

Class Input

Annotates an input FeatureDescriptor in a transformation pipeline

Inheritance

System.Object
System.Attribute
Input

Inherited Members

System.Attribute.Equals(System.Object)
System.Attribute.GetCustomAttribute(System.Reflection.Assembly, System.Type)
System.Attribute.GetCustomAttribute(System.Reflection.Assembly, System.Type, System.Boolean)
System.Attribute.GetCustomAttribute(System.Reflection.MemberInfo, System.Type)
System.Attribute.GetCustomAttribute(System.Reflection.MemberInfo, System.Type, System.Boolean)
System.Attribute.GetCustomAttribute(System.Reflection.Module, System.Type)
System.Attribute.GetCustomAttribute(System.Reflection.Module, System.Type, System.Boolean)
System.Attribute.GetCustomAttribute(System.Reflection.ParameterInfo, System.Type)
System.Attribute.GetCustomAttribute(System.Reflection.ParameterInfo, System.Type, System.Boolean)
System.Attribute.GetCustomAttributes(System.Reflection.Assembly)
System.Attribute.GetCustomAttributes(System.Reflection.Assembly, System.Boolean)
System.Attribute.GetCustomAttributes(System.Reflection.Assembly, System.Type)
System.Attribute.GetCustomAttributes(System.Reflection.Assembly, System.Type, System.Boolean)
System.Attribute.GetCustomAttributes(System.Reflection.MemberInfo)
System.Attribute.GetCustomAttributes(System.Reflection.MemberInfo, System.Boolean)
System.Attribute.GetCustomAttributes(System.Reflection.MemberInfo, System.Type)
System.Attribute.GetCustomAttributes(System.Reflection.MemberInfo, System.Type, System.Boolean)
System.Attribute.GetCustomAttributes(System.Reflection.Module)
System.Attribute.GetCustomAttributes(System.Reflection.Module, System.Boolean)
System.Attribute.GetCustomAttributes(System.Reflection.Module, System.Type)
System.Attribute.GetCustomAttributes(System.Reflection.Module, System.Type, System.Boolean)
System.Attribute.GetCustomAttributes(System.Reflection.ParameterInfo)
System.Attribute.GetCustomAttributes(System.Reflection.ParameterInfo, System.Boolean)
System.Attribute.GetCustomAttributes(System.Reflection.ParameterInfo, System.Type)
System.Attribute.GetCustomAttributes(System.Reflection.ParameterInfo, System.Type, System.Boolean)
System.Attribute.GetHashCode()
System.Attribute.IsDefaultAttribute()
System.Attribute.IsDefined(System.Reflection.Assembly, System.Type)
System.Attribute.IsDefined(System.Reflection.Assembly, System.Type, System.Boolean)
System.Attribute.IsDefined(System.Reflection.MemberInfo, System.Type)
System.Attribute.IsDefined(System.Reflection.MemberInfo, System.Type, System.Boolean)
System.Attribute.IsDefined(System.Reflection.Module, System.Type)
System.Attribute.IsDefined(System.Reflection.Module, System.Type, System.Boolean)
System.Attribute.IsDefined(System.Reflection.ParameterInfo, System.Type)
System.Attribute.IsDefined(System.Reflection.ParameterInfo, System.Type, System.Boolean)
System.Attribute.Match(System.Object)
System.Attribute.TypeId
System.Object.Equals(System.Object, System.Object)
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: SigStat.Common.Pipeline

Assembly: SigStat.Common.dll

Syntax

```
[AttributeUsage(AttributeTargets.Property, AllowMultiple = false)]
public class Input : Attribute
```

Constructors

Input(AutoSetMode)

Initializes a new instance of the Input class.

Declaration

```
public Input(AutoSetMode AutoSetMode = AutoSetMode.IfNull)
```

Parameters

TYPE	NAME	DESCRIPTION
AutoSetMode	AutoSetMode	The automatic set mode.

Fields

AutoSetMode

The automatic set mode

Declaration

```
public AutoSetMode AutoSetMode
```

Field Value

TYPE	DESCRIPTION
AutoSetMode	

See Also

System.Attribute

Interface IPipelineIO

Supports the definition of PipelineInput and PipelineOutput

Namespace: SigStat.Common.Pipeline

Assembly: SigStat.Common.dll

Syntax

```
public interface IPipelineIO
```

Properties

PipelineInputs

A collection of inputs for the pipeline elements

Declaration

```
List<PipelineInput> PipelineInputs { get; }
```

Property Value

TYPE	DESCRIPTION
System.Collections.Generic.List<PipelineInput>	

PipelineOutputs

A collection of outputs for the pipeline elements

Declaration

```
List<PipelineOutput> PipelineOutputs { get; }
```

Property Value

TYPE	DESCRIPTION
System.Collections.Generic.List<PipelineOutput>	

Interface ISignerModel

Analyzes signatures based on their similiarity to the trained model

Namespace: SigStat.Common.Pipeline

Assembly: SigStat.Common.dll

Syntax

```
public interface ISignerModel
```

Properties

SignerID

Identifies the signer, to whom this model belongs to

Declaration

```
string SignerID { get; }
```

Property Value

TYPE	DESCRIPTION
System.String	The signer identifier.

Class Output

Annotates an output FeatureDescriptor in a transformation pipeline

Inheritance

System.Object
System.Attribute
Output

Inherited Members

System.Attribute.Equals(System.Object)
System.Attribute.GetCustomAttribute(System.Reflection.Assembly, System.Type)
System.Attribute.GetCustomAttribute(System.Reflection.Assembly, System.Type, System.Boolean)
System.Attribute.GetCustomAttribute(System.Reflection.MemberInfo, System.Type)
System.Attribute.GetCustomAttribute(System.Reflection.MemberInfo, System.Type, System.Boolean)
System.Attribute.GetCustomAttribute(System.Reflection.Module, System.Type)
System.Attribute.GetCustomAttribute(System.Reflection.Module, System.Type, System.Boolean)
System.Attribute.GetCustomAttribute(System.Reflection.ParameterInfo, System.Type)
System.Attribute.GetCustomAttribute(System.Reflection.ParameterInfo, System.Type, System.Boolean)
System.Attribute.GetCustomAttributes(System.Reflection.Assembly)
System.Attribute.GetCustomAttributes(System.Reflection.Assembly, System.Boolean)
System.Attribute.GetCustomAttributes(System.Reflection.Assembly, System.Type)
System.Attribute.GetCustomAttributes(System.Reflection.Assembly, System.Type, System.Boolean)
System.Attribute.GetCustomAttributes(System.Reflection.MemberInfo)
System.Attribute.GetCustomAttributes(System.Reflection.MemberInfo, System.Boolean)
System.Attribute.GetCustomAttributes(System.Reflection.MemberInfo, System.Type)
System.Attribute.GetCustomAttributes(System.Reflection.MemberInfo, System.Type, System.Boolean)
System.Attribute.GetCustomAttributes(System.Reflection.Module)
System.Attribute.GetCustomAttributes(System.Reflection.Module, System.Boolean)
System.Attribute.GetCustomAttributes(System.Reflection.Module, System.Type)
System.Attribute.GetCustomAttributes(System.Reflection.Module, System.Type, System.Boolean)
System.Attribute.GetCustomAttributes(System.Reflection.ParameterInfo)
System.Attribute.GetCustomAttributes(System.Reflection.ParameterInfo, System.Boolean)
System.Attribute.GetCustomAttributes(System.Reflection.ParameterInfo, System.Type)
System.Attribute.GetCustomAttributes(System.Reflection.ParameterInfo, System.Type, System.Boolean)
System.Attribute.GetHashCode()
System.Attribute.IsDefaultAttribute()
System.Attribute.IsDefined(System.Reflection.Assembly, System.Type)
System.Attribute.IsDefined(System.Reflection.Assembly, System.Type, System.Boolean)
System.Attribute.IsDefined(System.Reflection.MemberInfo, System.Type)
System.Attribute.IsDefined(System.Reflection.MemberInfo, System.Type, System.Boolean)
System.Attribute.IsDefined(System.Reflection.Module, System.Type)
System.Attribute.IsDefined(System.Reflection.Module, System.Type, System.Boolean)
System.Attribute.IsDefined(System.Reflection.ParameterInfo, System.Type)
System.Attribute.IsDefined(System.Reflection.ParameterInfo, System.Type, System.Boolean)
System.Attribute.Match(System.Object)
System.Attribute.TypeId
System.Object.Equals(System.Object, System.Object)
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: SigStat.Common.Pipeline

Assembly: SigStat.Common.dll

Syntax

```
[AttributeUsage(AttributeTargets.Property, AllowMultiple = false)]
public class Output : Attribute
```

Constructors

Output()

Initializes a new instance of the Output class.

Declaration

```
public Output()
```

Output(String)

Initializes a new instance of the Output class.

Declaration

```
public Output(string Default)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	Default	The default.

Fields

Default

The default value for the property

Declaration

```
public string Default
```

Field Value

TYPE	DESCRIPTION
System.String	

Class ParallelTransformPipeline

Runs pipeline items in parallel.

Default Pipeline Output: Range of all the Item outputs.

Inheritance

System.Object

PipelineBase

ParallelTransformPipeline

Implements

ILoggerObject

IProgress

System.Collections.IEnumerable

ITransformation

IPipelineIO

Inherited Members

PipelineBase.Logger

PipelineBase.Progress

PipelineBase.ProgressChanged

PipelineBase.OnProgressChanged()

System.Object.Equals(System.Object)

System.Object.Equals(System.Object, System.Object)

System.Object.GetHashCode()

System.Object.GetType()

System.Object.MemberwiseClone()

System.Object.ReferenceEquals(System.Object, System.Object)

System.Object.ToString()

Namespace: SigStat.Common.Pipeline

Assembly: SigStat.Common.dll

Syntax

```
[JsonObject(MemberSerialization.OptOut)]
public class ParallelTransformPipeline : PipelineBase, ILoggerObject, IProgress, IEnumerable, ITransformation,
IPipelineIO
```

Fields

Items

List of transforms to be run parallel.

Declaration

```
public List<ITransformation> Items
```

Field Value

TYPE	DESCRIPTION
System.Collections.Generic.List<ITransformation>	

Properties

PipelineInputs

Gets the pipeline inputs.

Declaration

```
public override List<PipelineInput> PipelineInputs { get; }
```

Property Value

TYPE	DESCRIPTION
System.Collections.Generic.List<PipelineInput>	

Overrides

PipelineBase.PipelineInputs

PipelineOutputs

Gets the pipeline outputs.

Declaration

```
public override List<PipelineOutput> PipelineOutputs { get; }
```

Property Value

TYPE	DESCRIPTION
System.Collections.Generic.List<PipelineOutput>	

Overrides

PipelineBase.PipelineOutputs

Methods

Add(ITransformation)

Add new transform to the list.

Declaration

```
public void Add(ITransformation newItem)
```

Parameters

TYPE	NAME	DESCRIPTION
ITransformation	newItem	

GetEnumerator()

Declaration

```
public IEnumerator GetEnumerator()
```

Returns

TYPE	DESCRIPTION
System.Collections.IEnumerator	

Transform(Signature)

Executes transform Items parallel. Passes input features for each. Output is a range of all the Item outputs.

Declaration

```
public void Transform(Signature signature)
```

Parameters

TYPE	NAME	DESCRIPTION
Signature	signature	Signature to execute transform on.

Implements

ILoggerObject

IProgress

System.Collections.IEnumerable

ITransformation

IPipelineIO

Extension Methods

ILoggerObjectExtensions.LogError(ILoggerObject, String, Object[])

ILoggerObjectExtensions.LogError(ILoggerObject, Exception, String, Object[])

ILoggerObjectExtensions.LogInformation(ILoggerObject, String, Object[])

ILoggerObjectExtensions.LogWarning(ILoggerObject, String, Object[])

ILoggerObjectExtensions.LogWarning(ILoggerObject, Exception, String, Object[])

ILoggerObjectExtensions.LogTrace(ILoggerObject, String, Object[])

ILoggerObjectExtensions.LogTrace<TState>(ILoggerObject, TState, EventId, Exception, Func<TState, Exception, String>)

ILoggerObjectExtensions.LogCritical(ILoggerObject, String, Object[])

ILoggerObjectExtensions.LogDebug(ILoggerObject, String, Object[])

Class PipelineInput

Represents an input for a SigStat.Common.Pipeline.PipelineInput.PipelineItem

Inheritance

System.Object
PipelineInput

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: SigStat.Common.Pipeline

Assembly: SigStat.Common.dll

Syntax

```
public class PipelineInput
```

Constructors

PipelineInput(Object, PropertyInfo)

Initializes a new instance of the PipelineInput class.

Declaration

```
public PipelineInput(object PipelineItem, PropertyInfo PI)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Object	PipelineItem	The pipeline item.
System.Reflection.PropertyInfo	PI	The pi.

Exceptions

TYPE	CONDITION
System.Exception	Pipeline Input '{PropName}' of '{PipelineItem.ToString()}' not public

Properties

AutoSetMode

Gets the AutoSetMode

Declaration

```
public AutoSetMode AutoSetMode { get; }
```

Property Value

TYPE	DESCRIPTION
AutoSetMode	

FD

Gets or sets the fd.

Declaration

```
public object FD { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Object	

IsCollectionOfFeatureDescriptors

Gets a value indicating whether this instance is collection of feature descriptors.

Declaration

```
public bool IsCollectionOfFeatureDescriptors { get; }
```

Property Value

TYPE	DESCRIPTION
System.Boolean	`true` if this instance is collection of feature descriptors; otherwise, `false`.

PropName

Gets the name of the property.

Declaration

```
public string PropName { get; }
```

Property Value

TYPE	DESCRIPTION
System.String	

Type

Gets the type of the property

Declaration

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

Property Value

TYPE	DESCRIPTION
System.Type	

Class PipelineOutput

Represents an output for a SigStat.Common.Pipeline.PipelineOutput.PipelineItem

Inheritance

System.Object
PipelineOutput

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: SigStat.Common.Pipeline

Assembly: SigStat.Common.dll

Syntax

```
public class PipelineOutput
```

Constructors

PipelineOutput(Object, PropertyInfo)

Initializes a new instance of the PipelineOutput class.

Declaration

```
public PipelineOutput(object PipelineItem, PropertyInfo PI)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Object	PipelineItem	The pipeline item.
System.Reflection.PropertyInfo	PI	The pi.

Exceptions

TYPE	CONDITION
System.Exception	Pipeline Output '{PropName}' of '{PipelineItem.ToString()}' not public

Properties

Default

Gets the default value

Declaration

```
public string Default { get; }
```

Property Value

TYPE	DESCRIPTION
System.String	

FD

Gets or sets the fd.

Declaration

```
public object FD { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Object	

IsCollectionOfFeatureDescriptors

Gets a value indicating whether this instance is collection of feature descriptors.

Declaration

```
public bool IsCollectionOfFeatureDescriptors { get; }
```

Property Value

TYPE	DESCRIPTION
System.Boolean	`true` if this instance is collection of feature descriptors; otherwise, `false`.

IsTemporary

Gets a value indicating whether this instance is temporary.

Declaration

```
public bool IsTemporary { get; }
```

Property Value

TYPE	DESCRIPTION
System.Boolean	`true` if this instance is temporary; otherwise, `false`.

PropName

Gets the name of the property.

Declaration

```
public string PropName { get; }
```

Property Value

TYPE	DESCRIPTION
System.String	

Type

Gets the type of the property

Declaration

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

Property Value

TYPE	DESCRIPTION
System.Type	

Class SequentialTransformPipeline

Runs pipeline items in a sequence.

Default Pipeline Output: Output of the last Item in the sequence.

Inheritance

System.Object

PipelineBase

SequentialTransformPipeline

CentroidTranslate

TimeReset

Translate

Implements

ILoggerObject

IProgress

System.Collections.IEnumerable

ITransformation

IPipelineIO

Inherited Members

PipelineBase.Logger

PipelineBase.Progress

PipelineBase.ProgressChanged

PipelineBase.OnProgressChanged()

System.Object.Equals(System.Object)

System.Object.Equals(System.Object, System.Object)

System.Object.GetHashCode()

System.Object.GetType()

System.Object.MemberwiseClone()

System.Object.ReferenceEquals(System.Object, System.Object)

System.Object.ToString()

Namespace: SigStat.Common.Pipeline

Assembly: SigStat.Common.dll

Syntax

```
[JsonObject(MemberSerialization.OptOut)]
public class SequentialTransformPipeline : PipelineBase, ILoggerObject, IProgress, IEnumerable,
ITransformation, IPipelineIO
```

Fields

Items

List of transforms to be run in sequence.

Declaration

```
public List<ITransformation> Items
```

Field Value

TYPE	DESCRIPTION
System.Collections.Generic.List<ITransformation>	

Properties

PipelineInputs

Gets the pipeline inputs.

Declaration

```
public override List<PipelineInput> PipelineInputs { get; }
```

Property Value

TYPE	DESCRIPTION
System.Collections.Generic.List<PipelineInput>	

Overrides

PipelineBase.PipelineInputs

PipelineOutputs

Gets the pipeline outputs.

Declaration

```
public override List<PipelineOutput> PipelineOutputs { get; }
```

Property Value

TYPE	DESCRIPTION
System.Collections.Generic.List<PipelineOutput>	

Overrides

PipelineBase.PipelineOutputs

Methods

Add(ITransformation)

Add new transform to the list.

Declaration

```
public void Add(ITransformation newItem)
```

Parameters

TYPE	NAME	DESCRIPTION
ITransformation	newItem	

GetEnumerator()

Declaration

```
public IEnumerator GetEnumerator()
```

Returns

TYPE	DESCRIPTION
System.Collections.IEnumerator	

Transform(Signature)

Executes transform Items in sequence. Passes input features for each. Output is the output of the last Item in the sequence.

Declaration

```
public void Transform(Signature signature)
```

Parameters

TYPE	NAME	DESCRIPTION
Signature	signature	Signature to execute transform on.

Implements

ILoggerObject
IProgress
System.Collections.IEnumerable
ITransformation
IPipelineIO

Extension Methods

ILoggerObjectExtensions.LogError(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogError(ILoggerObject, Exception, String, Object[])
ILoggerObjectExtensions.LogInformation(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogWarning(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogWarning(ILoggerObject, Exception, String, Object[])
ILoggerObjectExtensions.LogTrace(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogTrace<TState>(ILoggerObject, TState, EventId, Exception, Func<TState, Exception, String>)
ILoggerObjectExtensions.LogCritical(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogDebug(ILoggerObject, String, Object[])

Namespace SigStat.Common.PipelineItems.Classifiers

Classes

DtwClassifier

Classifies Signatures with the algorithm.

DtwSignerModel

Represents a trained model for DtwClassifier

NearestNeighborEerClassifier

This IDistanceClassifier implementation will consider both test and training samples and claculate the threshold to separate the original and forged signatures to approximate EER. Note that this classifier is not applicable for real world scenarios. It was developed to test the theoratical boundaries of threshold based classification

NearestNeighborEerClassifier.SignerModel

Represents a trained model for NearestNeighborEerClassifier

OneClassNearestNeighborClassifier

This IDistanceClassifier implementation will consider both test and training samples and claculate the threshold to separate the original and forged signatures to approximate EER. Note that this classifier is not applicable for real world scenarios. It was developed to test the theoratical boundaries of threshold based classification

OneClassNearestNeighborClassifier.SignerModel

Represents a trained model for OneClassNearestNeighborClassifier

OptimalDtwClassifier

This IDistanceClassifier implementation will consider both test and training samples and claculate the threshold to separate the original and forged signatures to approximate EER. Note that this classifier is not applicable for real world scenarios. It was developed to test the theoratical boundaries of threshold based classification

OptimalDtwClassifier.OptimalDtwSignerModel

Represents a trained model for OptimalDtwClassifier

WeightedClassifier

Classifies Signatures by weighing other Classifier results.

Class DtwClassifier

Classifies Signatures with the algorithm.

Inheritance

System.Object
PipelineBase
DtwClassifier

Implements

ILoggerObject
IProgress
IPipelineIO
IDistanceClassifier
IClassifier

Inherited Members

PipelineBase.PipelineInputs
PipelineBase.PipelineOutputs
PipelineBase.Logger
PipelineBase.Progress
PipelineBase.ProgressChanged
PipelineBase.OnProgressChanged()
System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: SigStat.Common.PipelineItems.Classifiers

Assembly: SigStat.Common.dll

Syntax

```
[JsonObject(MemberSerialization.OptOut)]
public class DtwClassifier : PipelineBase, ILoggerObject, IProgress, IPipelineIO, IDistanceClassifier,
IClassifier
```

Constructors

DtwClassifier()

Initializes a new instance of the DtwClassifier class with the default Manhattan distance method.

Declaration

```
public DtwClassifier()
```

DtwClassifier(Func<Double[], Double[], Double>)

Initializes a new instance of the DtwClassifier class with a specified distance method.

Declaration

```
public DtwClassifier(Func<double[], double[], double> distanceMethod)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Func<System.Double[], System.Double[], System.Double>	distanceMethod	Accord.Math.Distance.*

Properties

DistanceFunction

The function used to calculate the distance between two data points during DTW calculation

Declaration

```
[JsonConverter(typeof(DistanceFunctionJsonConverter))]
public Func<double[], double[], double> DistanceFunction { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Func<System.Double[], System.Double[], System.Double>	

Features

Gets or sets the features to consider during distance calculation

Declaration

```
[Input(AutoSetMode.IfNull)]
public List<FeatureDescriptor> Features { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Collections.Generic.List<FeatureDescriptor>	

MultiplicationFactor

Gets or sets the multiplication factor to be used during threshold calculation

Declaration

```
public double MultiplicationFactor { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Double	

Methods

Test(ISignerModel, Signature)

Declaration

```
public double Test(ISignerModel model, Signature signature)
```

Parameters

TYPE	NAME	DESCRIPTION
ISignerModel	model	
Signature	signature	

Returns

TYPE	DESCRIPTION
System.Double	

Train(List<Signature>)

Declaration

```
public ISignerModel Train(List<Signature> signatures)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Collections.Generic.List<Signature>	signatures	

Returns

TYPE	DESCRIPTION
ISignerModel	

Implements

ILoggerObject
IProgress
IPipelineIO
IDistanceClassifier
IClassifier

Extension Methods

ILoggerObjectExtensions.LogError(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogError(ILoggerObject, Exception, String, Object[])
ILoggerObjectExtensions.LogInformation(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogWarning(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogWarning(ILoggerObject, Exception, String, Object[])
ILoggerObjectExtensions.LogTrace(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogTrace<TState>(ILoggerObject, TState, EventId, Exception, Func<TState, Exception, String>)
ILoggerObjectExtensions.LogCritical(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogDebug(ILoggerObject, String, Object[])

Class DtwSignerModel

Represents a trained model for DtwClassifier

Inheritance

System.Object
DtwSignerModel

Implements

ISignerModel

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: SigStat.Common.PipelineItems.Classifiers

Assembly: SigStat.Common.dll

Syntax

```
public class DtwSignerModel : ISignerModel
```

Fields

DistanceMatrix

DTW distance matrix of the genuine signatures

Declaration

```
public DistanceMatrix<string, string, double> DistanceMatrix
```

Field Value

TYPE	DESCRIPTION
DistanceMatrix<System.String, System.String, System.Double>	

Threshold

A threshold, that will be used for classification. Signatures with an average DTW distance from the genuines above this threshold will be classified as forgeries

Declaration

```
public double Threshold
```

Field Value

TYPE	DESCRIPTION
System.Double	

Properties

GenuineSignatures

A list a of genuine signatures used for training

Declaration

```
public List<KeyValuePair<string, double[][]>> GenuineSignatures { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Collections.Generic.List<System.Collections.Generic.KeyValuePair<System.String, System.Double[][]>>	

SignerID

Declaration

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

Property Value

TYPE	DESCRIPTION
System.String	

Implements

ISignerModel

Class NearestNeighborEerClassifier

This IDistanceClassifier implementation will consider both test and training samples and claculate the threshold to separate the original and forged signatures to approximate EER. Note that this classifier is not applicable for real world scenarios. It was developed to test the theoratical boundaries of threshold based classification

Inheritance

System.Object
PipelineBase
NearestNeighborEerClassifier

Implements

ILoggerObject
IProgress
IPipelineIO

Inherited Members

PipelineBase.PipelineInputs
PipelineBase.PipelineOutputs
PipelineBase.Logger
PipelineBase.Progress
PipelineBase.ProgressChanged
PipelineBase.OnProgressChanged()
System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: SigStat.Common.PipelineItems.Classifiers

Assembly: SigStat.Common.dll

Syntax

```
[JsonObject(MemberSerialization.OptOut)]
public class NearestNeighborEerClassifier : PipelineBase, ILoggerObject, IProgress, IPipelineIO
```

Constructors

NearestNeighborEerClassifier(Nullable<Int32>, IDistance<Double[][]>)

Initializes a new instance of the OptimalDtwClassifier class.

Declaration

```
public NearestNeighborEerClassifier(int? nearestNeighborCount, IDistance<double[][]> distanceFunction = null)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Nullable<System.Int32>	nearestNeighborCount	The number of nearest neighbours to consider during classification
IDistance<System.Double[][]>	distanceFunction	The distance function.

Properties

DistanceFunction

The function used to calculate the distance between two data sequences

Declaration

```
public IDistance<double[][]> DistanceFunction { get; set; }
```

Property Value

TYPE	DESCRIPTION
IDistance<System.Double[][]>	

Features

FeatureDescriptors to consider during classification

Declaration

```
[Input(AutoSetMode.IfNull)]
public List<FeatureDescriptor> Features { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Collections.Generic.List<FeatureDescriptor>	

NearestNeighborCount

Declaration

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

Property Value

TYPE	DESCRIPTION
System.Nullable<System.Int32>	

Sampler

Sampler used for selecting training and test sets during a benchmark

Declaration

```
public Sampler Sampler { get; set; }
```

Property Value

TYPE	DESCRIPTION
Sampler	

Methods

Test(ISignerModel, Signature)

Declaration

```
public double Test(ISignerModel signerModel, Signature signature)
```

Parameters

TYPE	NAME	DESCRIPTION
ISignerModel	signerModel	
Signature	signature	

Returns

TYPE	DESCRIPTION
System.Double	

Train(List<Signature>)

Declaration

```
public ISignerModel Train(List<Signature> signatures)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Collections.Generic.List<Signature>	signatures	

Returns

TYPE	DESCRIPTION
ISignerModel	

Train(List<Signature>, DistanceMatrix<String, String, Double>)

Trains the specified signatures based on a precalculated distance matrix

Declaration

```
public ISignerModel Train(List<Signature> signatures, DistanceMatrix<string, string, double> distanceMatrix)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Collections.Generic.List<Signature>	signatures	The signatures.
DistanceMatrix<System.String, System.String, System.Double>	distanceMatrix	The distance matrix may contain all the distance pairs for the signatures. If you ommit this parameter, distances will be calculated automatically using DistanceFunction.

Returns

TYPE	DESCRIPTION
ISignerModel	

Implements

ILoggerObject
IProgress
IPipelineIO

Extension Methods

ILoggerObjectExtensions.LogError(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogError(ILoggerObject, Exception, String, Object[])
ILoggerObjectExtensions.LogInformation(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogWarning(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogWarning(ILoggerObject, Exception, String, Object[])
ILoggerObjectExtensions.LogTrace(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogTrace<TState>(ILoggerObject, TState, EventId, Exception, Func<TState, Exception, String>)
ILoggerObjectExtensions.LogCritical(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogDebug(ILoggerObject, String, Object[])

See Also

PipelineBase
IDistanceClassifier

Class NearestNeighborEerClassifier.SignerModel

Represents a trained model for NearestNeighborEerClassifier

Inheritance

System.Object
NearestNeighborEerClassifier.SignerModel

Implements

ISignerModel

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: SigStat.Common.PipelineItems.Classifiers

Assembly: SigStat.Common.dll

Syntax

```
public class SignerModel : ISignerModel
```

Properties

DistanceMatrix

DTW distance matrix of the signatures

Declaration

```
public DistanceMatrix<string, string, double> DistanceMatrix { get; set; }
```

Property Value

TYPE	DESCRIPTION
DistanceMatrix<System.String, System.String, System.Double>	

ErrorRates

Gets or sets the error rates corresponding to specific thresholds

Declaration

```
public List<KeyValuePair<double, ErrorRate>> ErrorRates { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Collections.Generic.List<System.Collections.Generic.KeyValuePair<System.Double, ErrorRate>>	

SignatureDistanceFromTraining

Gets or sets the signature distance from training.

Declaration

```
public Dictionary<string, double> SignatureDistanceFromTraining { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Collections.Generic.Dictionary<System.String, System.Double>	

SignerID

Declaration

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

Property Value

TYPE	DESCRIPTION
System.String	

Threshold

A threshold, that will be used for classification. Signatures with an average DTW distance from the genuines above this threshold will be classified as forgeries

Declaration

```
public double Threshold { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Double	

Implements

ISignerModel

Class OneClassNearestNeighborClassifier

This IDistanceClassifier implementation will consider both test and training samples and claculate the threshold to separate the original and forged signatures to approximate EER. Note that this classifier is not applicable for real world scenarios. It was developed to test the theoratical boundaries of threshold based classification

Inheritance

System.Object
PipelineBase
OneClassNearestNeighborClassifier

Implements

ILoggerObject
IProgress
IPipelineIO
IClassifier

Inherited Members

PipelineBase.PipelineInputs
PipelineBase.PipelineOutputs
PipelineBase.Logger
PipelineBase.Progress
PipelineBase.ProgressChanged
PipelineBase.OnProgressChanged()
System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: SigStat.Common.PipelineItems.Classifiers

Assembly: SigStat.Common.dll

Syntax

```
[JsonObject(MemberSerialization.OptOut)]
public class OneClassNearestNeighborClassifier : PipelineBase, ILoggerObject, IProgress, IPipelineIO,
IClassifier
```

Constructors

OneClassNearestNeighborClassifier(Int32, Int32, Double, IDistance<Double[][]>)

Initializes a new instance of the OneClassNearestNeighborClassifier class.

Declaration

```
public OneClassNearestNeighborClassifier(int j = 1, int k = 1, double threshold = 1, IDistance<double[][]>
distanceFunction = null)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	j	The J parameter of the Ocjknn classifier

TYPE	NAME	DESCRIPTION
System.Int32	k	The K parameter of the Ocjknn classifier
System.Double	threshold	The K parameter of the Ocjknn classifier
IDistance<System.Double[][]>	distanceFunction	The distance function.

Properties

DistanceFunction

The function used to calculate the distance between two data sequences

Declaration

```
public IDistance<double[][]> DistanceFunction { get; set; }
```

Property Value

TYPE	DESCRIPTION
IDistance<System.Double[][]>	

Features

FeatureDescriptors to consider during classification

Declaration

```
[Input(AutoSetMode.IfNull)]
public List<FeatureDescriptor> Features { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Collections.Generic.List<FeatureDescriptor>	

J

The J parameter of the Ocjknn classifier

Declaration

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

Property Value

TYPE	DESCRIPTION
System.Int32	

K

The K parameter of the Ocjknn classifier

Declaration

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

Property Value

TYPE	DESCRIPTION
System.Int32	

Threshold

The Threshold parameter of the Ocjknn classifier

Declaration

```
public double Threshold { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Double	

Methods

Test(ISignerModel, Signature)

Declaration

```
public double Test(ISignerModel signerModel, Signature testSignature)
```

Parameters

TYPE	NAME	DESCRIPTION
ISignerModel	signerModel	
Signature	testSignature	

Returns

TYPE	DESCRIPTION
System.Double	

Train(List<Signature>)

Declaration

```
public ISignerModel Train(List<Signature> signatures)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Collections.Generic.List<Signature>	signatures	

Returns

TYPE	DESCRIPTION
ISignerModel	

Train(List<Signature>, DistanceMatrix<String, String, Double>)

Trains the specified signatures based on a precalculated distance matrix

Declaration

```
public ISignerModel Train(List<Signature> signatures, DistanceMatrix<string, string, double> distanceMatrix)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Collections.Generic.List<Signature>	signatures	The signatures.
DistanceMatrix<System.String, System.String, System.Double>	distanceMatrix	The distance matrix may contain all the distance pairs for the signatures. If you ommit this parameter, distances will be calculated automatically using DistanceFunction.

Returns

TYPE	DESCRIPTION
ISignerModel	

Implements

ILoggerObject
IProgress
IPipelineIO
IClassifier

Extension Methods

ILoggerObjectExtensions.LogError(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogError(ILoggerObject, Exception, String, Object[])
ILoggerObjectExtensions.LogInformation(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogWarning(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogWarning(ILoggerObject, Exception, String, Object[])
ILoggerObjectExtensions.LogTrace(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogTrace<TState>(ILoggerObject, TState, EventId, Exception, Func<TState, Exception, String>)
ILoggerObjectExtensions.LogCritical(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogDebug(ILoggerObject, String, Object[])

See Also

PipelineBase
IDistanceClassifier

Class OneClassNearestNeighborClassifier.SignerModel

Represents a trained model for OneClassNearestNeighborClassifier

Inheritance

System.Object
OneClassNearestNeighborClassifier.SignerModel

Implements

ISignerModel

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: SigStat.Common.PipelineItems.Classifiers

Assembly: SigStat.Common.dll

Syntax

```
public class SignerModel : ISignerModel
```

Properties

DistanceCache

Precalculated distances of known signatures

Declaration

```
public DistanceMatrix<string, string, double> DistanceCache { get; set; }
```

Property Value

TYPE	DESCRIPTION
DistanceMatrix<System.String, System.String, System.Double>	

SignerID

Declaration

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

Property Value

TYPE	DESCRIPTION
System.String	

TrainingSignatures

A list a of genuine signatures used for training

Declaration

```
public List<KeyValuePair<string, double[][]>> TrainingSignatures { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Collections.Generic.List<System.Collections.Generic.KeyValuePair<System.String, System.Double[][]>>	

Implements

ISignerModel

Class OptimalDtwClassifier

This IDistanceClassifier implementation will consider both test and training samples and claculate the threshold to separate the original and forged signatures to approximate EER. Note that this classifier is not applicable for real world scenarios. It was developed to test the theoratical boundaries of threshold based classification

Inheritance

System.Object
PipelineBase
OptimalDtwClassifier

Implements

ILoggerObject
IProgress
IPipelineIO
IDistanceClassifier
IClassifier

Inherited Members

PipelineBase.PipelineInputs
PipelineBase.PipelineOutputs
PipelineBase.Logger
PipelineBase.Progress
PipelineBase.ProgressChanged
PipelineBase.OnProgressChanged()
System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: SigStat.Common.PipelineItems.Classifiers

Assembly: SigStat.Common.dll

Syntax

```
[JsonObject(MemberSerialization.OptOut)]
public class OptimalDtwClassifier : PipelineBase, ILoggerObject, IProgress, IPipelineIO, IDistanceClassifier,
IClassifier
```

Constructors

OptimalDtwClassifier(Func<Double[], Double[], Double>)

Initializes a new instance of the OptimalDtwClassifier class.

Declaration

```
public OptimalDtwClassifier(Func<double[], double[], double> distanceFunction = null)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Func<System.Double[], System.Double[], System.Double>	distanceFunction	The distance function.

Properties

DistanceFunction

The function used to calculate the distance between two data points during DTW calculation

Declaration

```
[JsonConverter(typeof(DistanceFunctionJsonConverter))]
public Func<double[], double[], double> DistanceFunction { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Func<System.Double[], System.Double[], System.Double>	

Features

FeatureDescriptors to consider during classification

Declaration

```
[Input(AutoSetMode.IfNull)]
public List<FeatureDescriptor> Features { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Collections.Generic.List<FeatureDescriptor>	

Sampler

Sampler used for selecting training and test sets during a benchmark

Declaration

```
public Sampler Sampler { get; set; }
```

Property Value

TYPE	DESCRIPTION
Sampler	

WarpingWindowLength

Length of the warping window to be used with DTW

Declaration

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

Property Value

TYPE	DESCRIPTION
System.Int32	

Methods

Test(ISignerModel, Signature)

Declaration

```
public double Test(ISignerModel signerModel, Signature signature)
```

Parameters

TYPE	NAME	DESCRIPTION
ISignerModel	signerModel	
Signature	signature	

Returns

TYPE	DESCRIPTION
System.Double	

Train(List<Signature>)

Declaration

```
public ISignerModel Train(List<Signature> signatures)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Collections.Generic.List<Signature>	signatures	

Returns

TYPE	DESCRIPTION
ISignerModel	

Implements

ILoggerObject
IProgress
IPipelineIO
IDistanceClassifier
IClassifier

Extension Methods

ILoggerObjectExtensions.LogError(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogError(ILoggerObject, Exception, String, Object[])
ILoggerObjectExtensions.LogInformation(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogWarning(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogWarning(ILoggerObject, Exception, String, Object[])
ILoggerObjectExtensions.LogTrace(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogTrace<TState>(ILoggerObject, TState, EventId, Exception, Func<TState, Exception, String>)
ILoggerObjectExtensions.LogCritical(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogDebug(ILoggerObject, String, Object[])

See Also

PipelineBase

IDistanceClassifier

Class OptimalDtwClassifier.OptimalDtwSignerModel

Represents a trained model for OptimalDtwClassifier

Inheritance

System.Object
OptimalDtwClassifier.OptimalDtwSignerModel

Implements

ISignerModel

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: SigStat.Common.PipelineItems.Classifiers

Assembly: SigStat.Common.dll

Syntax

```
public class OptimalDtwSignerModel : ISignerModel
```

Properties

DistanceMatrix

DTW distance matrix of the signatures

Declaration

```
public DistanceMatrix<string, string, double> DistanceMatrix { get; set; }
```

Property Value

TYPE	DESCRIPTION
DistanceMatrix<System.String, System.String, System.Double>	

ErrorRates

Gets or sets the error rates corresponding to specific thresholds

Declaration

```
public List<KeyValuePair<double, ErrorRate>> ErrorRates { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Collections.Generic.List<System.Collections.Generic.KeyValuePair<System.Double, ErrorRate>>	

SignatureDistanceFromTraining

Gets or sets the signature distance from training.

Declaration

```
public Dictionary<string, double> SignatureDistanceFromTraining { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Collections.Generic.Dictionary<System.String, System.Double>	

SignerID

Declaration

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

Property Value

TYPE	DESCRIPTION
System.String	

Threshold

A threshold, that will be used for classification. Signatures with an average DTW distance from the genuines above this threshold will be classified as forgeries

Declaration

```
public double Threshold { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Double	

Implements

ISignerModel

Class WeightedClassifier

Classifies Signatures by weighing other Classifier results.

Inheritance

System.Object
PipelineBase
WeightedClassifier

Implements

ILoggerObject
IProgress
IPipelineIO
System.Collections.IEnumerable
IClassifier

Inherited Members

PipelineBase.PipelineInputs
PipelineBase.PipelineOutputs
PipelineBase.Logger
PipelineBase.Progress
PipelineBase.ProgressChanged
PipelineBase.OnProgressChanged()
System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: SigStat.Common.PipelineItems.Classifiers

Assembly: SigStat.Common.dll

Syntax

```
[JsonObject(MemberSerialization.OptOut)]
public class WeightedClassifier : PipelineBase, ILoggerObject, IProgress, IPipelineIO, IEnumerable,
IClassifier
```

Fields

Items

List of classifiers and belonging weights.

Declaration

```
public List<(IClassifier classifier, double weight)> Items
```

Field Value

TYPE	DESCRIPTION
System.Collections.Generic.List<System.ValueTuple<IClassifier, System.Double>>	

Methods

Add((IClassifier classifier, Double weight))

Add a new classifier with given weight to the list of items.

Declaration

```
public void Add((IClassifier classifier, double weight) newItem)
```

Parameters

TYPE	NAME	DESCRIPTION
System.ValueTuple<IClassifier, System.Double>	newItem	Classifier with belonging weight.

GetEnumerator()

Declaration

```
public IEnumerator GetEnumerator()
```

Returns

TYPE	DESCRIPTION
System.Collections.IEnumerator	

Test(ISignerModel, Signature)

Declaration

```
public double Test(ISignerModel model, Signature signature)
```

Parameters

TYPE	NAME	DESCRIPTION
ISignerModel	model	
Signature	signature	

Returns

TYPE	DESCRIPTION
System.Double	

Train(List<Signature>)

Declaration

```
public ISignerModel Train(List<Signature> signatures)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Collections.Generic.List<Signature>	signatures	

Returns

TYPE	DESCRIPTION
ISignerModel	

Implements

ILoggerObject
IProgress
IPipelineIO
System.Collections.IEnumerable
IClassifier

Extension Methods

ILoggerObjectExtensions.LogError(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogError(ILoggerObject, Exception, String, Object[])
ILoggerObjectExtensions.LogInformation(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogWarning(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogWarning(ILoggerObject, Exception, String, Object[])
ILoggerObjectExtensions.LogTrace(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogTrace<TState>(ILoggerObject, TState, EventId, Exception, Func<TState, Exception, String>)
ILoggerObjectExtensions.LogCritical(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogDebug(ILoggerObject, String, Object[])

Namespace SigStat.Common.PipelineItems.Transforms.Preprocessing

Classes

CubicInterpolation

Cubic interpolation algorithm

FillPenUpDurations

This transformation fills gaps of online signature by interpolating the last known feature values. Gaps should be represented in the signature with two zero pressure border points.

FillPenUpDurations.TimeSlot

Helper class for FillPenUpDurations

FilterPoints

Removes samples based on a criteria from online signature time series

LinearInterpolation

Performs linear interpolation on the input

NormalizeRotation

Performs rotation normalization on the online signature

NormalizeRotation2

Performs rotation normalization on the online signature

NormalizeRotation3

Performs rotation normalization on the online signature

NormalizeRotationForX

Performs rotation normalization on the online signature

OrthognalRotation

Performs rotation normalization on the online signature

RelativeScale

Maps values of a feature to a specific range.

InputFeature: feature to be scaled.

OutputFeature: output feature for scaled InputFeature

ResampleSamplesCountBased

Resamples an online signature to a specific sample count using the specified IInterpolation algorithm

SampleRate

Performs rotation normalization on the online signature

Scale

Maps values of a feature to a specific range.

InputFeature: feature to be scaled.

OutputFeature: output feature for scaled InputFeature

TranslatePreproc

This transformations can be used to translate the coordinates of an online signature

UniformScale

Maps values of a feature to a specific range and another proportional.

BaseDimension: feature modelled the base dimension of the scaling.

ProportionalDimension: feature modelled the dimension scaled proportionally to the base dimension.

BaseDimensionOutput: output feature for scaled BaseDimension

ProportionalDimensionOutput: output feature for scaled ProportionalDimension

ZNormalization

Maps values of a feature to a specific range.

InputFeature: feature to be scaled.

OutputFeature: output feature for scaled InputFeature

Interfaces

IInterpolation

Represents an interploation algorithm

Enums

OriginType

Origin specification for TranslatePreproc

ScalingMode

Mode specification for Scale

Class CubicInterpolation

Cubic interpolation algorithm

Inheritance

System.Object
CubicInterpolation

Implements

IInterpolation

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: SigStat.Common.PipelineItems.Transforms.Preprocessing

Assembly: SigStat.Common.dll

Syntax

```
[JsonObject(MemberSerialization.OptOut)]
public class CubicInterpolation : IInterpolation
```

Properties

FeatureValues

FeatureValues

Declaration

```
public List<double> FeatureValues { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Collections.Generic.List<System.Double>	

TimeValues

TimeValues

Declaration

```
public List<double> TimeValues { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Collections.Generic.List<System.Double>	

Methods

GetValue(Double)

Gets the value.

Declaration

```
public double GetValue(double timestamp)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Double	timestamp	The timestamp.

Returns

TYPE	DESCRIPTION
System.Double	

Exceptions

TYPE	CONDITION
System.NullReferenceException	List of timestamps is null or List of feature values is null
System.ArgumentOutOfRangeException	The given timestamp is not in the range of TimeValues

Implements

IInterpolation

Class FillPenUpDurations

This transformation fills gaps of online signature by interpolating the last known feature values. Gaps should be represented in the signature with two zero pressure border points.

Inheritance

System.Object
PipelineBase
FillPenUpDurations

Implements

ILoggerObject
IProgress
ITransformation
IPipelineIO

Inherited Members

PipelineBase.PipelineInputs
PipelineBase.PipelineOutputs
PipelineBase.Logger
PipelineBase.Progress
PipelineBase.ProgressChanged
PipelineBase.OnProgressChanged()
System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: SigStat.Common.PipelineItems.Transforms.Preprocessing

Assembly: SigStat.Common.dll

Syntax

```
[JsonObject(MemberSerialization.OptOut)]
public class FillPenUpDurations : PipelineBase, ILoggerObject, IProgress, ITransformation, IPipelineIO
```

Properties

InputFeatures

Gets or sets the features of an online signature that need to be altered

Declaration

```
[Input(AutoSetMode.IfNull)]
public List<FeatureDescriptor<List<double>>> InputFeatures { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Collections.Generic.List<FeatureDescriptor<System.Collections.Generic.List<System.Double>>>	

InterpolationType

An implementation of IInterpolation

Declaration

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

Property Value

TYPE	DESCRIPTION
System.Type	

OutputFeatures

Gets or sets the features of an online signature that were altered

Declaration

```
[Output]
public List<FeatureDescriptor<List<double>>> OutputFeatures { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Collections.Generic.List<FeatureDescriptor<System.Collections.Generic.List<System.Double>>>	

PointTypeInputFeature

Gets or sets the feature representing the type of the points in an online signature

Declaration

```
[Input(AutoSetMode.IfNull)]
public FeatureDescriptor<List<double>> PointTypeInputFeature { get; set; }
```

Property Value

TYPE	DESCRIPTION
FeatureDescriptor<System.Collections.Generic.List<System.Double>>	

PointTypeOutputFeature

Gets or sets the feature representing the modified point type values in an online signature

Declaration

```
[Output]
public FeatureDescriptor<List<double>> PointTypeOutputFeature { get; set; }
```

Property Value

TYPE	DESCRIPTION
FeatureDescriptor<System.Collections.Generic.List<System.Double>>	

PressureInputFeature

Gets or sets the feature representing pressure in an online signature

Declaration

```
[Input(AutoSetMode.IfNull)]
public FeatureDescriptor<List<double>> PressureInputFeature { get; set; }
```

Property Value

TYPE	DESCRIPTION
FeatureDescriptor<System.Collections.Generic.List<System.Double>>	

PressureOutputFeature

Gets or sets the feature representing the modified pressure values of an online signature

Declaration

```
[Output]
public FeatureDescriptor<List<double>> PressureOutputFeature { get; set; }
```

Property Value

TYPE	DESCRIPTION
FeatureDescriptor<System.Collections.Generic.List<System.Double>>	

TimeInputFeature

Gets or sets the feature representing the timestamps of an online signature

Declaration

```
[Input(AutoSetMode.IfNull)]
public FeatureDescriptor<List<double>> TimeInputFeature { get; set; }
```

Property Value

TYPE	DESCRIPTION
FeatureDescriptor<System.Collections.Generic.List<System.Double>>	

TimeOutputFeature

Gets or sets the feature representing the modified timestamps of an online signature

Declaration

```
[Output("FilledTime")]
public FeatureDescriptor<List<double>> TimeOutputFeature { get; set; }
```

Property Value

TYPE	DESCRIPTION
FeatureDescriptor<System.Collections.Generic.List<System.Double>>	

Methods

Transform(Signature)

Declaration

```
public void Transform(Signature signature)
```

Parameters

TYPE	NAME	DESCRIPTION
Signature	signature	

Implements

ILoggerObject
IProgress
ITransformation
IPipelineIO

Extension Methods

ILoggerObjectExtensions.LogError(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogError(ILoggerObject, Exception, String, Object[])
ILoggerObjectExtensions.LogInformation(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogWarning(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogWarning(ILoggerObject, Exception, String, Object[])
ILoggerObjectExtensions.LogTrace(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogTrace<TState>(ILoggerObject, TState, EventId, Exception, Func<TState, Exception, String>)
ILoggerObjectExtensions.LogCritical(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogDebug(ILoggerObject, String, Object[])

See Also

PipelineBase
ITransformation

Class FillPenUpDurations.TimeSlot

Helper class for FillPenUpDurations

Inheritance

System.Object

FillPenUpDurations.TimeSlot

Inherited Members

System.Object.Equals(System.Object)

System.Object.Equals(System.Object, System.Object)

System.Object.GetHashCode()

System.Object.GetType()

System.Object.MemberwiseClone()

System.Object.ReferenceEquals(System.Object, System.Object)

System.Object.ToString()

Namespace: SigStat.Common.PipelineItems.Transforms.Preprocessing

Assembly: SigStat.Common.dll

Syntax

```
public class TimeSlot
```

Properties

EndTime

Gets or sets the end time of the slot

Declaration

```
public double EndTime { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Double	

Length

Gets the length of the slot

Declaration

```
public double Length { get; }
```

Property Value

TYPE	DESCRIPTION
System.Double	

PenDown

This indicates whether the pen touches the paper during the time slot

Declaration

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

Property Value

TYPE	DESCRIPTION
System.Boolean	

StartTime

Gets or sets the start time of the slot

Declaration

```
public double StartTime { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Double	

Class FilterPoints

Removes samples based on a criteria from online signature time series

Inheritance

System.Object

PipelineBase

FilterPoints

Implements

ILoggerObject

IProgress

ITransformation

IPipelineIO

Inherited Members

PipelineBase.PipelineInputs

PipelineBase.PipelineOutputs

PipelineBase.Logger

PipelineBase.Progress

PipelineBase.ProgressChanged

PipelineBase.OnProgressChanged()

System.Object.Equals(System.Object)

System.Object.Equals(System.Object, System.Object)

System.Object.GetHashCode()

System.Object.GetType()

System.Object.MemberwiseClone()

System.Object.ReferenceEquals(System.Object, System.Object)

System.Object.ToString()

Namespace: SigStat.Common.PipelineItems.Transforms.Preprocessing

Assembly: SigStat.Common.dll

Syntax

```
public class FilterPoints : PipelineBase, ILoggerObject, IProgress, ITransformation, IPipelineIO
```

Properties

InputFeatures

FeatureDescriptor list of all features to resample

Declaration

```
[Input(AutoSetMode.IfNull)]
public List<FeatureDescriptor<List<double>>> InputFeatures { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Collections.Generic.List<FeatureDescriptor<System.Collections.Generic.List<System.Double>>>	

KeyFeatureInput

FeatureDescriptor that controls the removal of samples (e.g. Pressure)

Declaration

```
[Input(AutoSetMode.IfNull)]
public FeatureDescriptor<List<double>> KeyFeatureInput { get; set; }
```

Property Value

TYPE	DESCRIPTION
FeatureDescriptor<System.Collections.Generic.List<System.Double>>	

KeyFeatureOutput

Resampled output for FeatureDescriptor that controls the removal of samples (e.g. Pressure)

Declaration

```
[Output("FilterKeyFeatureOutput")]
public FeatureDescriptor<List<double>> KeyFeatureOutput { get; set; }
```

Property Value

TYPE	DESCRIPTION
FeatureDescriptor<System.Collections.Generic.List<System.Double>>	

OutputFeatures

Resampled output for all input features

Declaration

```
[Output]
public List<FeatureDescriptor<List<double>>> OutputFeatures { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Collections.Generic.List<FeatureDescriptor<System.Collections.Generic.List<System.Double>>>	

Percentile

The lowes percentile of the KeyFeatureInput will be removed during filtering

Declaration

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

Property Value

TYPE	DESCRIPTION
System.Int32	

Methods

Transform(Signature)

Declaration

```
public void Transform(Signature signature)
```

Parameters

TYPE	NAME	DESCRIPTION
Signature	signature	

Implements

ILoggerObject
IProgress
ITransformation
IPipelineIO

Extension Methods

ILoggerObjectExtensions.LogError(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogError(ILoggerObject, Exception, String, Object[])
ILoggerObjectExtensions.LogInformation(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogWarning(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogWarning(ILoggerObject, Exception, String, Object[])
ILoggerObjectExtensions.LogTrace(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogTrace<TState>(ILoggerObject, TState, EventId, Exception, Func<TState, Exception, String>)
ILoggerObjectExtensions.LogCritical(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogDebug(ILoggerObject, String, Object[])

See Also

PipelineBase
ITransformation

Interface IInterpolation

Represents an interploation algorithm

Namespace: SigStat.Common.PipelineItems.Transforms.Preprocessing

Assembly: SigStat.Common.dll

Syntax

```
public interface IInterpolation
```

Properties

FeatureValues

Gets or sets the feature values.

Declaration

```
List<double> FeatureValues { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Collections.Generic.List<System.Double>	

TimeValues

Timestamps

Declaration

```
List<double> TimeValues { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Collections.Generic.List<System.Double>	

Methods

GetValue(Double)

Gets the interpolated value at a given timestamp

Declaration

```
double GetValue(double timestamp)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Double	timestamp	The timestamp.

Returns

TYPE	DESCRIPTION
System.Double	

Class LinearInterpolation

Performs linear interpolation on the input

Inheritance

System.Object
LinearInterpolation

Implements

IInterpolation

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: SigStat.Common.PipelineItems.Transforms.Preprocessing

Assembly: SigStat.Common.dll

Syntax

```
[JsonObject(MemberSerialization.OptOut)]
public class LinearInterpolation : IInterpolation
```

Properties

FeatureValues

Declaration

```
public List<double> FeatureValues { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Collections.Generic.List<System.Double>	

TimeValues

Declaration

```
public List<double> TimeValues { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Collections.Generic.List<System.Double>	

Methods

GetValue(Double)

Gets the interpolated value at a given timestamp

Declaration

```
public double GetValue(double timestamp)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Double	timestamp	The timestamp.

Returns

TYPE	DESCRIPTION
System.Double	

Exceptions

TYPE	CONDITION
System.InvalidOperationException	TimeValues is not initialized
System.NullReferenceException	FeatureValues is not initialized
System.ArgumentOutOfRangeException	The given timestamp is not in the range of TimeValues

Implements

IInterpolation

See Also

IInterpolation

Class NormalizeRotation

Performs rotation normalization on the online signature

Inheritance

System.Object

PipelineBase

NormalizeRotation

Implements

ILoggerObject

IProgress

ITransformation

IPipelineIO

Inherited Members

PipelineBase.PipelineInputs

PipelineBase.PipelineOutputs

PipelineBase.Logger

PipelineBase.Progress

PipelineBase.ProgressChanged

PipelineBase.OnProgressChanged()

System.Object.Equals(System.Object)

System.Object.Equals(System.Object, System.Object)

System.Object.GetHashCode()

System.Object.GetType()

System.Object.MemberwiseClone()

System.Object.ReferenceEquals(System.Object, System.Object)

System.Object.ToString()

Namespace: SigStat.Common.PipelineItems.Transforms.Preprocessing

Assembly: SigStat.Common.dll

Syntax

```
[JsonObject(MemberSerialization.OptOut)]
public class NormalizeRotation : PipelineBase, ILoggerObject, IProgress, ITransformation, IPipelineIO
```

Properties

InputT

Gets or sets the input feature representing the timestamps of an online signature

Declaration

```
[Input(AutoSetMode.IfNull)]
public FeatureDescriptor<List<double>> InputT { get; set; }
```

Property Value

TYPE	DESCRIPTION
FeatureDescriptor<System.Collections.Generic.List<System.Double>>	

InputX

Gets or sets the input feature representing the X coordinates of an online signature

Declaration

```
[Input(AutoSetMode.IfNull)]
public FeatureDescriptor<List<double>> InputX { get; set; }
```

Property Value

TYPE	DESCRIPTION
FeatureDescriptor<System.Collections.Generic.List<System.Double>>	

InputY

Gets or sets the input feature representing the Y coordinates of an online signature

Declaration

```
[Input(AutoSetMode.IfNull)]
public FeatureDescriptor<List<double>> InputY { get; set; }
```

Property Value

TYPE	DESCRIPTION
FeatureDescriptor<System.Collections.Generic.List<System.Double>>	

OutputX

Gets or sets the output feature representing the X coordinates of an online signature

Declaration

```
[Output]
public FeatureDescriptor<List<double>> OutputX { get; set; }
```

Property Value

TYPE	DESCRIPTION
FeatureDescriptor<System.Collections.Generic.List<System.Double>>	

OutputY

Gets or sets the input feature representing the Y coordinates of an online signature

Declaration

```
[Output]
public FeatureDescriptor<List<double>> OutputY { get; set; }
```

Property Value

TYPE	DESCRIPTION
FeatureDescriptor<System.Collections.Generic.List<System.Double>>	

Methods

Transform(Signature)

Declaration

```
public void Transform(Signature signature)
```

Parameters

TYPE	NAME	DESCRIPTION
Signature	signature	

Implements

ILoggerObject
IProgress
ITransformation
IPipelineIO

Extension Methods

ILoggerObjectExtensions.LogError(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogError(ILoggerObject, Exception, String, Object[])
ILoggerObjectExtensions.LogInformation(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogWarning(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogWarning(ILoggerObject, Exception, String, Object[])
ILoggerObjectExtensions.LogTrace(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogTrace<TState>(ILoggerObject, TState, EventId, Exception, Func<TState, Exception, String>)
ILoggerObjectExtensions.LogCritical(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogDebug(ILoggerObject, String, Object[])

See Also

PipelineBase
ITransformation

Class NormalizeRotation2

Performs rotation normalization on the online signature

Inheritance

System.Object
PipelineBase
NormalizeRotation2

Implements

ILoggerObject
IProgress
ITransformation
IPipelineIO

Inherited Members

PipelineBase.PipelineInputs
PipelineBase.PipelineOutputs
PipelineBase.Logger
PipelineBase.Progress
PipelineBase.ProgressChanged
PipelineBase.OnProgressChanged()
System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: SigStat.Common.PipelineItems.Transforms.Preprocessing

Assembly: SigStat.Common.dll

Syntax

```
[JsonObject(MemberSerialization.OptOut)]
public class NormalizeRotation2 : PipelineBase, ILoggerObject, IProgress, ITransformation, IPipelineIO
```

Properties

InputX

Gets or sets the input feature representing the X coordinates of an online signature

Declaration

```
[Input(AutoSetMode.IfNull)]
public FeatureDescriptor<List<double>> InputX { get; set; }
```

Property Value

TYPE	DESCRIPTION
FeatureDescriptor<System.Collections.Generic.List<System.Double>>	

InputY

Gets or sets the input feature representing the Y coordinates of an online signature

Declaration

```
[Input(AutoSetMode.IfNull)]
public FeatureDescriptor<List<double>> InputY { get; set; }
```

Property Value

TYPE	DESCRIPTION
FeatureDescriptor<System.Collections.Generic.List<System.Double>>	

OutputX

Gets or sets the output feature representing the X coordinates of an online signature

Declaration

```
[Output]
public FeatureDescriptor<List<double>> OutputX { get; set; }
```

Property Value

TYPE	DESCRIPTION
FeatureDescriptor<System.Collections.Generic.List<System.Double>>	

OutputY

Gets or sets the output feature representing the Y coordinates of an online signature

Declaration

```
[Output]
public FeatureDescriptor<List<double>> OutputY { get; set; }
```

Property Value

TYPE	DESCRIPTION
FeatureDescriptor<System.Collections.Generic.List<System.Double>>	

Methods

Transform(Signature)

Declaration

```
public void Transform(Signature signature)
```

Parameters

TYPE	NAME	DESCRIPTION
Signature	signature	

Implements

ILoggerObject

IProgress

ITransformation

IPipelineIO

Extension Methods

ILoggerObjectExtensions.LogError(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogError(ILoggerObject, Exception, String, Object[])
ILoggerObjectExtensions.LogInformation(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogWarning(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogWarning(ILoggerObject, Exception, String, Object[])
ILoggerObjectExtensions.LogTrace(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogTrace<TState>(ILoggerObject, TState, EventId, Exception, Func<TState, Exception, String>)
ILoggerObjectExtensions.LogCritical(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogDebug(ILoggerObject, String, Object[])

See Also

PipelineBase
ITransformation

Class NormalizeRotation3

Performs rotation normalization on the online signature

Inheritance

System.Object

PipelineBase

NormalizeRotation3

Implements

ILoggerObject

IProgress

ITransformation

IPipelineIO

Inherited Members

PipelineBase.PipelineInputs

PipelineBase.PipelineOutputs

PipelineBase.Logger

PipelineBase.Progress

PipelineBase.ProgressChanged

PipelineBase.OnProgressChanged()

System.Object.Equals(System.Object)

System.Object.Equals(System.Object, System.Object)

System.Object.GetHashCode()

System.Object.GetType()

System.Object.MemberwiseClone()

System.Object.ReferenceEquals(System.Object, System.Object)

System.Object.ToString()

Namespace: SigStat.Common.PipelineItems.Transforms.Preprocessing

Assembly: SigStat.Common.dll

Syntax

```
[JsonObject(MemberSerialization.OptOut)]
public class NormalizeRotation3 : PipelineBase, ILoggerObject, IProgress, ITransformation, IPipelineIO
```

Properties

InputX

Gets or sets the input feature representing the X coordinates of an online signature

Declaration

```
[Input(AutoSetMode.IfNull)]
public FeatureDescriptor<List<double>> InputX { get; set; }
```

Property Value

TYPE	DESCRIPTION
FeatureDescriptor<System.Collections.Generic.List<System.Double>>	

InputY

Gets or sets the input feature representing the Y coordinates of an online signature

Declaration

```
[Input(AutoSetMode.IfNull)]
public FeatureDescriptor<List<double>> InputY { get; set; }
```

Property Value

TYPE	DESCRIPTION
FeatureDescriptor<System.Collections.Generic.List<System.Double>>	

OutputX

Gets or sets the output feature representing the X coordinates of an online signature

Declaration

```
[Output]
public FeatureDescriptor<List<double>> OutputX { get; set; }
```

Property Value

TYPE	DESCRIPTION
FeatureDescriptor<System.Collections.Generic.List<System.Double>>	

OutputY

Gets or sets the output feature representing the Y coordinates of an online signature

Declaration

```
[Output]
public FeatureDescriptor<List<double>> OutputY { get; set; }
```

Property Value

TYPE	DESCRIPTION
FeatureDescriptor<System.Collections.Generic.List<System.Double>>	

Methods

Transform(Signature)

Declaration

```
public void Transform(Signature signature)
```

Parameters

TYPE	NAME	DESCRIPTION
Signature	signature	

Implements

ILoggerObject

IProgress

ITransformation

IPipelineIO

Extension Methods

ILoggerObjectExtensions.LogError(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogError(ILoggerObject, Exception, String, Object[])
ILoggerObjectExtensions.LogInformation(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogWarning(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogWarning(ILoggerObject, Exception, String, Object[])
ILoggerObjectExtensions.LogTrace(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogTrace<TState>(ILoggerObject, TState, EventId, Exception, Func<TState, Exception, String>)
ILoggerObjectExtensions.LogCritical(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogDebug(ILoggerObject, String, Object[])

See Also

PipelineBase
ITransformation

Class NormalizeRotationForX

Performs rotation normalization on the online signature

Inheritance

System.Object
PipelineBase
NormalizeRotationForX

Implements

ILoggerObject
IProgress
ITransformation
IPipelineIO

Inherited Members

PipelineBase.PipelineInputs
PipelineBase.PipelineOutputs
PipelineBase.Logger
PipelineBase.Progress
PipelineBase.ProgressChanged
PipelineBase.OnProgressChanged()
System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: SigStat.Common.PipelineItems.Transforms.Preprocessing

Assembly: SigStat.Common.dll

Syntax

```
[JsonObject(MemberSerialization.OptOut)]
public class NormalizeRotationForX : PipelineBase, ILoggerObject, IProgress, ITransformation, IPipelineIO
```

Properties

InputT

Gets or sets the input feature representing the timestamps of an online signature

Declaration

```
[Input(AutoSetMode.IfNull)]
public FeatureDescriptor<List<double>> InputT { get; set; }
```

Property Value

TYPE	DESCRIPTION
FeatureDescriptor<System.Collections.Generic.List<System.Double>>	

InputX

Gets or sets the input feature representing the X coordinates of an online signature

Declaration

```
[Input(AutoSetMode.IfNull)]
public FeatureDescriptor<List<double>> InputX { get; set; }
```

Property Value

TYPE	DESCRIPTION
FeatureDescriptor<System.Collections.Generic.List<System.Double>>	

InputY

Gets or sets the input feature representing the Y coordinates of an online signature

Declaration

```
[Input(AutoSetMode.IfNull)]
public FeatureDescriptor<List<double>> InputY { get; set; }
```

Property Value

TYPE	DESCRIPTION
FeatureDescriptor<System.Collections.Generic.List<System.Double>>	

OutputX

Gets or sets the output feature representing the X coordinates of an online signature

Declaration

```
[Output]
public FeatureDescriptor<List<double>> OutputX { get; set; }
```

Property Value

TYPE	DESCRIPTION
FeatureDescriptor<System.Collections.Generic.List<System.Double>>	

OutputY

Gets or sets the input feature representing the Y coordinates of an online signature

Declaration

```
[Output]
public FeatureDescriptor<List<double>> OutputY { get; set; }
```

Property Value

TYPE	DESCRIPTION
FeatureDescriptor<System.Collections.Generic.List<System.Double>>	

Methods

Transform(Signature)

Declaration

```
public void Transform(Signature signature)
```

Parameters

TYPE	NAME	DESCRIPTION
Signature	signature	

Implements

ILoggerObject
IProgress
ITransformation
IPipelineIO

Extension Methods

ILoggerObjectExtensions.LogError(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogError(ILoggerObject, Exception, String, Object[])
ILoggerObjectExtensions.LogInformation(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogWarning(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogWarning(ILoggerObject, Exception, String, Object[])
ILoggerObjectExtensions.LogTrace(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogTrace<TState>(ILoggerObject, TState, EventId, Exception, Func<TState, Exception, String>)
ILoggerObjectExtensions.LogCritical(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogDebug(ILoggerObject, String, Object[])

See Also

PipelineBase
ITransformation

Enum OriginType

Origin specification for TranslatePreproc

Namespace: SigStat.Common.PipelineItems.Transforms.Preprocessing

Assembly: SigStat.Common.dll

Syntax

```
public enum OriginType
```

Fields

NAME	DESCRIPTION
CenterOfGravity	Center of gravity
Maximum	Maximum
Minimum	Minimum
Predefined	Predefined

Class OrthognalRotation

Performs rotation normalization on the online signature

Inheritance

System.Object

PipelineBase

OrthognalRotation

Implements

ILoggerObject

IProgress

ITransformation

IPipelineIO

Inherited Members

PipelineBase.PipelineInputs

PipelineBase.PipelineOutputs

PipelineBase.Logger

PipelineBase.Progress

PipelineBase.ProgressChanged

PipelineBase.OnProgressChanged()

System.Object.Equals(System.Object)

System.Object.Equals(System.Object, System.Object)

System.Object.GetHashCode()

System.Object.GetType()

System.Object.MemberwiseClone()

System.Object.ReferenceEquals(System.Object, System.Object)

System.Object.ToString()

Namespace: SigStat.Common.PipelineItems.Transforms.Preprocessing

Assembly: SigStat.Common.dll

Syntax

```
[JsonObject(MemberSerialization.OptOut)]
public class OrthognalRotation : PipelineBase, ILoggerObject, IProgress, ITransformation, IPipelineIO
```

Properties

InputT

Gets or sets the input feature representing the timestamps of an online signature

Declaration

```
[Input(AutoSetMode.IfNull)]
public FeatureDescriptor<List<double>> InputT { get; set; }
```

Property Value

TYPE	DESCRIPTION
FeatureDescriptor<System.Collections.Generic.List<System.Double>>	

InputX

Gets or sets the input feature representing the X coordinates of an online signature

Declaration

```
[Input(AutoSetMode.IfNull)]
public FeatureDescriptor<List<double>> InputX { get; set; }
```

Property Value

TYPE	DESCRIPTION
FeatureDescriptor<System.Collections.Generic.List<System.Double>>	

InputY

Gets or sets the input feature representing the Y coordinates of an online signature

Declaration

```
[Input(AutoSetMode.IfNull)]
public FeatureDescriptor<List<double>> InputY { get; set; }
```

Property Value

TYPE	DESCRIPTION
FeatureDescriptor<System.Collections.Generic.List<System.Double>>	

OutputX

Gets or sets the output feature representing the X coordinates of an online signature

Declaration

```
[Output]
public FeatureDescriptor<List<double>> OutputX { get; set; }
```

Property Value

TYPE	DESCRIPTION
FeatureDescriptor<System.Collections.Generic.List<System.Double>>	

OutputY

Gets or sets the input feature representing the Y coordinates of an online signature

Declaration

```
[Output]
public FeatureDescriptor<List<double>> OutputY { get; set; }
```

Property Value

TYPE	DESCRIPTION
FeatureDescriptor<System.Collections.Generic.List<System.Double>>	

Methods

Transform(Signature)

Declaration

```
public void Transform(Signature signature)
```

Parameters

TYPE	NAME	DESCRIPTION
Signature	signature	

Implements

ILoggerObject
IProgress
ITransformation
IPipelineIO

Extension Methods

ILoggerObjectExtensions.LogError(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogError(ILoggerObject, Exception, String, Object[])
ILoggerObjectExtensions.LogInformation(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogWarning(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogWarning(ILoggerObject, Exception, String, Object[])
ILoggerObjectExtensions.LogTrace(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogTrace<TState>(ILoggerObject, TState, EventId, Exception, Func<TState, Exception, String>)
ILoggerObjectExtensions.LogCritical(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogDebug(ILoggerObject, String, Object[])

See Also

PipelineBase
ITransformation

Class RelativeScale

Maps values of a feature to a specific range.

InputFeature: feature to be scaled.

OutputFeature: output feature for scaled InputFeature

Inheritance

System.Object

PipelineBase

RelativeScale

Implements

ILoggerObject

IProgress

ITransformation

IPipelineIO

Inherited Members

PipelineBase.PipelineInputs

PipelineBase.PipelineOutputs

PipelineBase.Logger

PipelineBase.Progress

PipelineBase.ProgressChanged

PipelineBase.OnProgressChanged()

System.Object.Equals(System.Object)

System.Object.Equals(System.Object, System.Object)

System.Object.GetHashCode()

System.Object.GetType()

System.Object.MemberwiseClone()

System.Object.ReferenceEquals(System.Object, System.Object)

System.Object.ToString()

Namespace: SigStat.Common.PipelineItems.Transforms.Preprocessing

Assembly: SigStat.Common.dll

Syntax

```
[JsonObject(MemberSerialization.OptOut)]
public class RelativeScale : PipelineBase, ILoggerObject, IProgress, ITransformation, IPipelineIO
```

Properties

InputFeature

Gets or sets the input feature.

Declaration

```
[Input(AutoSetMode.IfNull)]
public FeatureDescriptor<List<double>> InputFeature { get; set; }
```

Property Value

TYPE	DESCRIPTION
FeatureDescriptor<System.Collections.Generic.List<System.Double>>	

OutputFeature

Gets or sets the output feature.

Declaration

```
[Output("ScaledFeature")]
public FeatureDescriptor<List<double>> OutputFeature { get; set; }
```

Property Value

TYPE	DESCRIPTION
FeatureDescriptor<System.Collections.Generic.List<System.Double>>	

ReferenceFeature

Gets or sets the reference feature.

Declaration

```
[Input(AutoSetMode.IfNull)]
public FeatureDescriptor<List<double>> ReferenceFeature { get; set; }
```

Property Value

TYPE	DESCRIPTION
FeatureDescriptor<System.Collections.Generic.List<System.Double>>	

Methods

Transform(Signature)

Declaration

```
public void Transform(Signature signature)
```

Parameters

TYPE	NAME	DESCRIPTION
Signature	signature	

Implements

ILoggerObject
IProgress
ITransformation
IPipelineIO

Extension Methods

ILoggerObjectExtensions.LogError(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogError(ILoggerObject, Exception, String, Object[])
ILoggerObjectExtensions.LogInformation(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogWarning(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogWarning(ILoggerObject, Exception, String, Object[])
ILoggerObjectExtensions.LogTrace(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogTrace<TState>(ILoggerObject, TState, EventId, Exception, Func<TState, Exception, String>)
ILoggerObjectExtensions.LogCritical(ILoggerObject, String, Object[])

ILoggerObjectExtensions.LogDebug(ILoggerObject, String, Object[])

Class ResampleSamplesCountBased

Resamples an online signature to a specific sample count using the specified IInterpolation algorithm

Inheritance

System.Object
PipelineBase
ResampleSamplesCountBased

Implements

ILoggerObject
IProgress
ITransformation
IPipelineIO

Inherited Members

PipelineBase.PipelineInputs
PipelineBase.PipelineOutputs
PipelineBase.Logger
PipelineBase.Progress
PipelineBase.ProgressChanged
PipelineBase.OnProgressChanged()
System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: SigStat.Common.PipelineItems.Transforms.Preprocessing

Assembly: SigStat.Common.dll

Syntax

```
[JsonObject(MemberSerialization.OptOut)]
public class ResampleSamplesCountBased : PipelineBase, ILoggerObject, IProgress, ITransformation, IPipelineIO
```

Properties

InputFeatures

Gets or sets the input features.

Declaration

```
[Input(AutoSetMode.IfNull)]
public List<FeatureDescriptor<List<double>>> InputFeatures { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Collections.Generic.List<FeatureDescriptor<System.Collections.Generic.List<System.Double>>>	

InterpolationType

Gets or sets the type of the interpolation. IInterpolation

Declaration

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

Property Value

TYPE	DESCRIPTION
System.Type	

NumOfSamples

Gets or sets the number of samples.

Declaration

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

Property Value

TYPE	DESCRIPTION
System.Int32	

OriginalTFeature

Gets or sets the input timestamp feature.

Declaration

```
[Input(AutoSetMode.IfNull)]
public FeatureDescriptor<List<double>> OriginalTFeature { get; set; }
```

Property Value

TYPE	DESCRIPTION
FeatureDescriptor<System.Collections.Generic.List<System.Double>>	

OutputFeatures

Gets or sets the resampled features.

Declaration

```
[Output]
public List<FeatureDescriptor<List<double>>> OutputFeatures { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Collections.Generic.List<FeatureDescriptor<System.Collections.Generic.List<System.Double>>>	

ResampledTFeature

Gets or sets the resampled timestamp feature.

Declaration

```
[Output("ResampledTimestamps")]
public FeatureDescriptor<List<double>> ResampledTFeature { get; set; }
```

Property Value

TYPE	DESCRIPTION
FeatureDescriptor<System.Collections.Generic.List<System.Double>>	

Methods

Transform(Signature)

Declaration

```
public void Transform(Signature signature)
```

Parameters

TYPE	NAME	DESCRIPTION
Signature	signature	

Implements

ILoggerObject
IProgress
ITransformation
IPipelineIO

Extension Methods

ILoggerObjectExtensions.LogError(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogError(ILoggerObject, Exception, String, Object[])
ILoggerObjectExtensions.LogInformation(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogWarning(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogWarning(ILoggerObject, Exception, String, Object[])
ILoggerObjectExtensions.LogTrace(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogTrace<TState>(ILoggerObject, TState, EventId, Exception, Func<TState, Exception, String>)
ILoggerObjectExtensions.LogCritical(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogDebug(ILoggerObject, String, Object[])

See Also

PipelineBase
ITransformation

Class SampleRate

Performs rotation normalization on the online signature

Inheritance

System.Object
PipelineBase
SampleRate

Implements

ILoggerObject
IProgress
ITransformation
IPipelineIO

Inherited Members

PipelineBase.PipelineInputs
PipelineBase.PipelineOutputs
PipelineBase.Logger
PipelineBase.Progress
PipelineBase.ProgressChanged
PipelineBase.OnProgressChanged()
System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: SigStat.Common.PipelineItems.Transforms.Preprocessing

Assembly: SigStat.Common.dll

Syntax

```
[JsonObject(MemberSerialization.OptOut)]
public class SampleRate : PipelineBase, ILoggerObject, IProgress, ITransformation, IPipelineIO
```

Properties

InputP

Gets or sets the input feature representing the timestamps of an online signature

Declaration

```
[Input(AutoSetMode.IfNull)]
public FeatureDescriptor<List<double>> InputP { get; set; }
```

Property Value

TYPE	DESCRIPTION
FeatureDescriptor<System.Collections.Generic.List<System.Double>>	

InputX

Gets or sets the input feature representing the X coordinates of an online signature

Declaration

```
[Input(AutoSetMode.IfNull)]
public FeatureDescriptor<List<double>> InputX { get; set; }
```

Property Value

TYPE	DESCRIPTION
FeatureDescriptor<System.Collections.Generic.List<System.Double>>	

InputY

Gets or sets the input feature representing the Y coordinates of an online signature

Declaration

```
[Input(AutoSetMode.IfNull)]
public FeatureDescriptor<List<double>> InputY { get; set; }
```

Property Value

TYPE	DESCRIPTION
FeatureDescriptor<System.Collections.Generic.List<System.Double>>	

OutputP

Declaration

```
[Output]
public FeatureDescriptor<List<double>> OutputP { get; set; }
```

Property Value

TYPE	DESCRIPTION
FeatureDescriptor<System.Collections.Generic.List<System.Double>>	

OutputX

Gets or sets the output feature representing the X coordinates of an online signature

Declaration

```
[Output]
public FeatureDescriptor<List<double>> OutputX { get; set; }
```

Property Value

TYPE	DESCRIPTION
FeatureDescriptor<System.Collections.Generic.List<System.Double>>	

OutputY

Gets or sets the input feature representing the Y coordinates of an online signature

Declaration

```
[Output]
public FeatureDescriptor<List<double>> OutputY { get; set; }
```

Property Value

TYPE	DESCRIPTION
FeatureDescriptor<System.Collections.Generic.List<System.Double>>	

samplerate

Gets or sets the input feature representing the X coordinates of an online signature

Declaration

```
[Input(AutoSetMode.IfNull)]
public int samplerate { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Int32	

Methods

Transform(Signature)

Declaration

```
public void Transform(Signature signature)
```

Parameters

TYPE	NAME	DESCRIPTION
Signature	signature	

Implements

ILoggerObject
IProgress
ITransformation
IPipelineIO

Extension Methods

ILoggerObjectExtensions.LogError(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogError(ILoggerObject, Exception, String, Object[])
ILoggerObjectExtensions.LogInformation(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogWarning(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogWarning(ILoggerObject, Exception, String, Object[])
ILoggerObjectExtensions.LogTrace(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogTrace<TState>(ILoggerObject, TState, EventId, Exception, Func<TState, Exception, String>)
ILoggerObjectExtensions.LogCritical(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogDebug(ILoggerObject, String, Object[])

See Also

PipelineBase
ITransformation

Class Scale

Maps values of a feature to a specific range.

InputFeature: feature to be scaled.

OutputFeature: output feature for scaled InputFeature

Inheritance

System.Object

PipelineBase

Scale

Implements

ILoggerObject

IProgress

ITransformation

IPipelineIO

Inherited Members

PipelineBase.PipelineInputs

PipelineBase.PipelineOutputs

PipelineBase.Logger

PipelineBase.Progress

PipelineBase.ProgressChanged

PipelineBase.OnProgressChanged()

System.Object.Equals(System.Object)

System.Object.Equals(System.Object, System.Object)

System.Object.GetHashCode()

System.Object.GetType()

System.Object.MemberwiseClone()

System.Object.ReferenceEquals(System.Object, System.Object)

System.Object.ToString()

Namespace: SigStat.Common.PipelineItems.Transforms.Preprocessing

Assembly: SigStat.Common.dll

Syntax

```
[JsonObject(MemberSerialization.OptOut)]
public class Scale : PipelineBase, ILoggerObject, IProgress, ITransformation, IPipelineIO
```

Properties

InputFeature

Gets or sets the input feature.

Declaration

```
[Input(AutoSetMode.IfNull)]
public FeatureDescriptor<List<double>> InputFeature { get; set; }
```

Property Value

TYPE	DESCRIPTION
FeatureDescriptor<System.Collections.Generic.List<System.Double>>	

Mode

Type of the scaling which defines the scaling behavior

Declaration

```
public ScalingMode Mode { get; set; }
```

Property Value

TYPE	DESCRIPTION
ScalingMode	

OutputFeature

Gets or sets the output feature.

Declaration

```
[Output("ScaledFeature")]
public FeatureDescriptor<List<double>> OutputFeature { get; set; }
```

Property Value

TYPE	DESCRIPTION
FeatureDescriptor<System.Collections.Generic.List<System.Double>>	

Methods

Transform(Signature)

Declaration

```
public void Transform(Signature signature)
```

Parameters

TYPE	NAME	DESCRIPTION
Signature	signature	

Implements

ILoggerObject
IProgress
ITransformation
IPipelineIO

Extension Methods

ILoggerObjectExtensions.LogError(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogError(ILoggerObject, Exception, String, Object[])
ILoggerObjectExtensions.LogInformation(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogWarning(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogWarning(ILoggerObject, Exception, String, Object[])
ILoggerObjectExtensions.LogTrace(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogTrace<TState>(ILoggerObject, TState, EventId, Exception, Func<TState, Exception, String>)
ILoggerObjectExtensions.LogCritical(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogDebug(ILoggerObject, String, Object[])

Enum ScalingMode

Mode specification for Scale

Namespace: SigStat.Common.PipelineItems.Transforms.Preprocessing

Assembly: SigStat.Common.dll

Syntax

```
public enum ScalingMode
```

Fields

NAME	DESCRIPTION
Scaling1	Values are scaled into an interval, where the difference between the lower and upper bounds is 1
ScalingS	Values are scaled based on their standard deviation

Class TranslatePreproc

This transformations can be used to translate the coordinates of an online signature

Inheritance

System.Object
PipelineBase
TranslatePreproc

Implements

ILoggerObject
IProgress
ITransformation
IPipelineIO

Inherited Members

PipelineBase.PipelineInputs
PipelineBase.PipelineOutputs
PipelineBase.Logger
PipelineBase.Progress
PipelineBase.ProgressChanged
PipelineBase.OnProgressChanged()
System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: SigStat.Common.PipelineItems.Transforms.Preprocessing

Assembly: SigStat.Common.dll

Syntax

```
[JsonObject(MemberSerialization.OptOut)]
public class TranslatePreproc : PipelineBase, ILoggerObject, IProgress, ITransformation, IPipelineIO
```

Constructors

TranslatePreproc()

Initializes a new instance of the TranslatePreproc class.

Declaration

```
public TranslatePreproc()
```

TranslatePreproc(OriginType)

Initializes a new instance of the TranslatePreproc class.

Declaration

```
public TranslatePreproc(OriginType goalOrigin)
```

Parameters

TYPE	NAME	DESCRIPTION
OriginType	goalOrigin	The goal origin.

Properties

GoalOrigin

Goal origin of the translation

Declaration

```
public OriginType GoalOrigin { get; set; }
```

Property Value

TYPE	DESCRIPTION
OriginType	

InputFeature

Input FeatureDescriptor (e.g. X)

Declaration

```
[Input(AutoSetMode.IfNull)]
public FeatureDescriptor<List<double>> InputFeature { get; set; }
```

Property Value

TYPE	DESCRIPTION
FeatureDescriptor<System.Collections.Generic.List<System.Double>>	

NewOrigin

New origin after the translation

Declaration

```
public double NewOrigin { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Double	

OutputFeature

Output FeatureDescriptor (e.g. X)

Declaration

```
[Output("TranslatedFeature")]
public FeatureDescriptor<List<double>> OutputFeature { get; set; }
```

Property Value

TYPE	DESCRIPTION
FeatureDescriptor<System.Collections.Generic.List<System.Double>>	

Methods

Transform(Signature)

Declaration

```
public void Transform(Signature signature)
```

Parameters

TYPE	NAME	DESCRIPTION
Signature	signature	

Implements

ILoggerObject
IProgress
ITransformation
IPipelineIO

Extension Methods

ILoggerObjectExtensions.LogError(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogError(ILoggerObject, Exception, String, Object[])
ILoggerObjectExtensions.LogInformation(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogWarning(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogWarning(ILoggerObject, Exception, String, Object[])
ILoggerObjectExtensions.LogTrace(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogTrace<TState>(ILoggerObject, TState, EventId, Exception, Func<TState, Exception, String>)
ILoggerObjectExtensions.LogCritical(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogDebug(ILoggerObject, String, Object[])

See Also

PipelineBase
ITransformation

Class UniformScale

Maps values of a feature to a specific range and another proportional.

BaseDimension: feature modelled the base dimension of the scaling.

ProportionalDimension: feature modelled the dimension scaled proportionally to the base dimension.

BaseDimensionOutput: output feature for scaled BaseDimension

ProportionalDimensionOutput: output feature for scaled ProportionalDimension

Inheritance

System.Object
PipelineBase
UniformScale

Implements

ILoggerObject
IProgress
ITransformation
IPipelineIO

Inherited Members

PipelineBase.PipelineInputs
PipelineBase.PipelineOutputs
PipelineBase.Logger
PipelineBase.Progress
PipelineBase.ProgressChanged
PipelineBase.OnProgressChanged()
System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: SigStat.Common.PipelineItems.Transforms.Preprocessing

Assembly: SigStat.Common.dll

Syntax

```
[JsonObject(MemberSerialization.OptOut)]
public class UniformScale : PipelineBase, ILoggerObject, IProgress, ITransformation, IPipelineIO
```

Properties

BaseDimension

Gets or sets the base dimension.

Declaration

```
[Input(AutoSetMode.IfNull)]
public FeatureDescriptor<List<double>> BaseDimension { get; set; }
```

Property Value

TYPE	DESCRIPTION
FeatureDescriptor<System.Collections.Generic.List<System.Double>>	

BaseDimensionOutput

Gets or sets the output base dimension output.

Declaration

```
[Output("UniformScaledBaseDimension")]
public FeatureDescriptor<List<double>> BaseDimensionOutput { get; set; }
```

Property Value

TYPE	DESCRIPTION
FeatureDescriptor<System.Collections.Generic.List<System.Double>>	

NewMaxBaseValue

Upper bound of the interval, in which the base dimension will be scaled

Declaration

```
public double NewMaxBaseValue { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Double	

NewMinBaseValue

Lower bound of the interval, in which the base dimension will be scaled

Declaration

```
public double NewMinBaseValue { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Double	

NewMinProportionalValue

Lower bound of the interval, in which the proportional dimension will be scaled

Declaration

```
public double NewMinProportionalValue { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Double	

ProportionalDimension

Gets or sets the ProportionalDimension.

Declaration

```
[Input(AutoSetMode.IfNull)]
public FeatureDescriptor<List<double>> ProportionalDimension { get; set; }
```

Property Value

TYPE	DESCRIPTION
FeatureDescriptor<System.Collections.Generic.List<System.Double>>	

ProportionalDimensionOutput

Gets or sets the output proportional dimension output.

Declaration

```
[Output("UniformScaledProportionalDimension")]
public FeatureDescriptor<List<double>> ProportionalDimensionOutput { get; set; }
```

Property Value

TYPE	DESCRIPTION
FeatureDescriptor<System.Collections.Generic.List<System.Double>>	

Methods

Transform(Signature)

Declaration

```
public void Transform(Signature signature)
```

Parameters

TYPE	NAME	DESCRIPTION
Signature	signature	

Implements

ILoggerObject
IProgress
ITransformation
IPipelineIO

Extension Methods

ILoggerObjectExtensions.LogError(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogError(ILoggerObject, Exception, String, Object[])
ILoggerObjectExtensions.LogInformation(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogWarning(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogWarning(ILoggerObject, Exception, String, Object[])
ILoggerObjectExtensions.LogTrace(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogTrace<TState>(ILoggerObject, TState, EventId, Exception, Func<TState, Exception, String>)
ILoggerObjectExtensions.LogCritical(ILoggerObject, String, Object[])

ILoggerObjectExtensions.LogDebug(ILoggerObject, String, Object[])

Class ZNormalization

Maps values of a feature to a specific range.

InputFeature: feature to be scaled.

OutputFeature: output feature for scaled InputFeature

Inheritance

System.Object
PipelineBase
ZNormalization

Implements

ILoggerObject
IProgress
ITransformation
IPipelineIO

Inherited Members

PipelineBase.PipelineInputs
PipelineBase.PipelineOutputs
PipelineBase.Logger
PipelineBase.Progress
PipelineBase.ProgressChanged
PipelineBase.OnProgressChanged()
System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: SigStat.Common.PipelineItems.Transforms.Preprocessing

Assembly: SigStat.Common.dll

Syntax

```
[JsonObject(MemberSerialization.OptOut)]
public class ZNormalization : PipelineBase, ILoggerObject, IProgress, ITransformation, IPipelineIO
```

Properties

InputFeature

Gets or sets the input feature.

Declaration

```
[Input(AutoSetMode.IfNull)]
public FeatureDescriptor<List<double>> InputFeature { get; set; }
```

Property Value

TYPE	DESCRIPTION
FeatureDescriptor<System.Collections.Generic.List<System.Double>>	

OutputFeature

Gets or sets the output feature.

Declaration

```
[Output("Z-Normalized Feature")]
public FeatureDescriptor<List<double>> OutputFeature { get; set; }
```

Property Value

TYPE	DESCRIPTION
FeatureDescriptor<System.Collections.Generic.List<System.Double>>	

Methods

Transform(Signature)

Declaration

```
public void Transform(Signature signature)
```

Parameters

TYPE	NAME	DESCRIPTION
Signature	signature	

Implements

ILoggerObject
IProgress
ITransformation
IPipelineIO

Extension Methods

ILoggerObjectExtensions.LogError(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogError(ILoggerObject, Exception, String, Object[])
ILoggerObjectExtensions.LogInformation(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogWarning(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogWarning(ILoggerObject, Exception, String, Object[])
ILoggerObjectExtensions.LogTrace(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogTrace<TState>(ILoggerObject, TState, EventId, Exception, Func<TState, Exception, String>)
ILoggerObjectExtensions.LogCritical(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogDebug(ILoggerObject, String, Object[])

Namespace SigStat.Common.Transforms

Classes

AddConst

Adds a constant value to a feature. Works with collection features too.

Default Pipeline Output: Pipeline Input

AddVector

Adds a vector feature's elements to other features.

Default Pipeline Output: Pipeline Input

ApproximateOnlineFeatures

init Pressure, Altitude, Azimuth features with default values.

Default Pipeline Output: Features.Pressure, Features.Altitude, Features.Azimuth

Binarization

Generates a binary raster version of the input image with the iterative threshold method.

Pipeline Input type: Image{Rgba32}

Default Pipeline Output: (bool[,]) Binarized

BinaryRasterizer

Converts standard features to a binary raster.

Default Pipeline Input: Standard Features

Default Pipeline Output: (bool[,]) Binarized

CentroidExtraction

Extracts the Centroid (aka. Center Of Gravity) of the input features.

Default Pipeline Output: (List{double}) Centroid.

CentroidTranslate

Sequential pipeline to translate X and Y Features to Centroid. The following Transforms are called: CentroidExtraction, Multiply(-1), Translate

Default Pipeline Input: X, Y

Default Pipeline Output: (List{double}) Centroid

ComponentExtraction

Extracts unsorted components by tracing through the binary Skeleton raster.

Default Pipeline Input: (bool[,]) Skeleton, (List{Point}) EndPoints, (List{Point}) CrossingPoints

Default Pipeline Output: (List{List{PointF}}) Components

ComponentSorter

Sorts Component order by comparing each starting X value, and finding nearest components.

Default Pipeline Input: (bool[,]) Components

Default Pipeline Output: (bool[,]) Components

ComponentsToFeatures

Extracts standard Features from sorted Components.

Default Pipeline Input: (List{List{PointF}}) Components

Default Pipeline Output: X, Y, Button Features

EndpointExtraction

Extracts EndPoints and CrossingPoints from Skeleton.

Default Pipeline Input: (bool[,]) Skeleton

Default Pipeline Output: (List{Point}) EndPoints, (List{Point}) CrossingPoints

Extrema

Extracts minimum and maximum values of given feature.

Default Pipeline Output: (List{double}) Min, (List{double}) Max

HSCPThinning

Iteratively thins the input binary raster with the HSCPThinningStep algorithm.

Pipeline Input type: bool[,]

Default Pipeline Output: (bool[,]) HSCPThinningResult

ImageGenerator

Generates an image feature out of a binary raster. Optionally, saves the image to a png file. Useful for debugging pipeline steps.

Pipeline Input type: bool[,]

Default Pipeline Output: (bool[,]) Input, (Image{Rgba32}) InputImage

Map

Maps values of a feature to a specified range.

Pipeline Input type: List{double}

Default Pipeline Output: (List{double}) MapResult

Multiply

Multiplies the values of a feature with a given constant.

Pipeline Input type: List{double}

Default Pipeline Output: (List{double}) Input

Normalize

Maps values of a feature to 0.0 - 1.0 range.

Pipeline Input type: List{double}

Default Pipeline Output: (List{double}) NormalizationResult

OnePixelThinning

Iteratively thins the input binary raster with the SigStat.Common.Algorithms.OnePixelThinningStep algorithm.

Pipeline Input type: bool[,]

Default Pipeline Output: (bool[,]) OnePixelThinningResult

RealisticImageGenerator

Generates a realistic looking image of the Signature based on standard features. Uses blue ink and white paper. It does NOT save the image to file.

Default Pipeline Input: X, Y, Button, Pressure, Azimuth, Altitude Features

Default Pipeline Output: Image

Resize

Resizes the image to a specified width and height

TangentExtraction

Extracts tangent values of the standard X, Y Features

Default Pipeline Input: X, Y Features

Default Pipeline Output: (List{double}) Tangent

TimeReset

Sequential pipeline to reset time values to begin at 0. The following Transforms are called: Extrema, Multiply, AddVector.

Default Pipeline Input: T

Default Pipeline Output: T

Translate

Sequential pipeline to translate X and Y Features by specified vector (constant or feature). The following Transforms are called: AddConst twice, or AddVector.

Default Pipeline Input: X, Y

Default Pipeline Output: X, Y

Trim

Trims unnecessary empty space from a binary raster.

Pipeline Input type: bool[,]

Default Pipeline Output: (bool[,]) Trimmed

Enums

Binarization.ForegroundType

Represents the type of the input image.

Class AddConst

Adds a constant value to a feature. Works with collection features too.

Default Pipeline Output: Pipeline Input

Inheritance

System.Object

PipelineBase

AddConst

Implements

ILoggerObject

IProgress

ITransformation

IPipelineIO

Inherited Members

PipelineBase.PipelineInputs

PipelineBase.PipelineOutputs

PipelineBase.Logger

PipelineBase.Progress

PipelineBase.ProgressChanged

PipelineBase.OnProgressChanged()

System.Object.Equals(System.Object)

System.Object.Equals(System.Object, System.Object)

System.Object.GetHashCode()

System.Object.GetType()

System.Object.MemberwiseClone()

System.Object.ReferenceEquals(System.Object, System.Object)

System.Object.ToString()

Namespace: SigStat.Common.Transforms

Assembly: SigStat.Common.dll

Syntax

```
[JsonObject(MemberSerialization.OptOut)]
public class AddConst : PipelineBase, ILoggerObject, IProgress, ITransformation, IPipelineIO
```

Constructors

AddConst(Double)

Initializes a new instance of the AddConst class with specified settings.

Declaration

```
public AddConst(double value)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Double	value	The value to be added to the input feature.

Properties

Input

Input values for trasformation

Declaration

```
[Input(AutoSetMode.IfNull)]
public FeatureDescriptor<List<double>> Input { get; set; }
```

Property Value

TYPE	DESCRIPTION
FeatureDescriptor<System.Collections.Generic.List<System.Double>>	

Output

Output feature to store results

Declaration

```
[Output]
public FeatureDescriptor<List<double>> Output { get; set; }
```

Property Value

TYPE	DESCRIPTION
FeatureDescriptor<System.Collections.Generic.List<System.Double>>	

Methods

Transform(Signature)

Declaration

```
public void Transform(Signature signature)
```

Parameters

TYPE	NAME	DESCRIPTION
Signature	signature	

Implements

ILoggerObject
IProgress
ITransformation
IPipelineIO

Extension Methods

ILoggerObjectExtensions.LogError(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogError(ILoggerObject, Exception, String, Object[])
ILoggerObjectExtensions.LogInformation(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogWarning(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogWarning(ILoggerObject, Exception, String, Object[])
ILoggerObjectExtensions.LogTrace(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogTrace<TState>(ILoggerObject, TState, EventId, Exception, Func<TState, Exception, String>)
ILoggerObjectExtensions.LogCritical(ILoggerObject, String, Object[])

ILoggerObjectExtensions.LogDebug(ILoggerObject, String, Object[])

Class AddVector

Adds a vector feature's elements to other features.

Default Pipeline Output: Pipeline Input

Inheritance

System.Object

PipelineBase

AddVector

Implements

ILoggerObject

IProgress

ITransformation

IPipelineIO

Inherited Members

PipelineBase.PipelineInputs

PipelineBase.PipelineOutputs

PipelineBase.Logger

PipelineBase.Progress

PipelineBase.ProgressChanged

PipelineBase.OnProgressChanged()

System.Object.Equals(System.Object)

System.Object.Equals(System.Object, System.Object)

System.Object.GetHashCode()

System.Object.GetType()

System.Object.MemberwiseClone()

System.Object.ReferenceEquals(System.Object, System.Object)

System.Object.ToString()

Namespace: SigStat.Common.Transforms

Assembly: SigStat.Common.dll

Syntax

```
[JsonObject(MemberSerialization.OptOut)]
public class AddVector : PipelineBase, ILoggerObject, IProgress, ITransformation, IPipelineIO
```

Examples

Inputs are: Centroid.xy, X, Y . Adds Centroid.x to each element of X. Adds Centroid.y to each element of Y.

Constructors

AddVector(FeatureDescriptor<List<Double>>)

Initializes a new instance of the AddVector class with a vector feature. Don't forget to add as many Inputs as the vector's dimension.

Declaration

```
public AddVector(FeatureDescriptor<List<double>> vectorFeature)
```

Parameters

TYPE	NAME	DESCRIPTION
FeatureDescriptor<System.Collections.Generic.List<System.Double>>	vectorFeature	A collection-type feature where each element represents a dimension of the vector.

Properties

Inputs

Inputs

Declaration

```
[Input(AutoSetMode.IfNull)]
public List<FeatureDescriptor<List<double>>> Inputs { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Collections.Generic.List<FeatureDescriptor<System.Collections.Generic.List<System.Double>>>	

Outputs

Outputs

Declaration

```
[Output]
public List<FeatureDescriptor<List<double>>> Outputs { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Collections.Generic.List<FeatureDescriptor<System.Collections.Generic.List<System.Double>>>	

Methods

Transform(Signature)

Declaration

```
public void Transform(Signature signature)
```

Parameters

TYPE	NAME	DESCRIPTION
Signature	signature	

Implements

ILoggerObject
IProgress
ITransformation
IPipelineIO

Extension Methods

ILoggerObjectExtensions.LogError(ILoggerObject, String, Object[])

ILoggerObjectExtensions.LogError(ILoggerObject, Exception, String, Object[])

ILoggerObjectExtensions.LogInformation(ILoggerObject, String, Object[])

ILoggerObjectExtensions.LogWarning(ILoggerObject, String, Object[])

ILoggerObjectExtensions.LogWarning(ILoggerObject, Exception, String, Object[])

ILoggerObjectExtensions.LogTrace(ILoggerObject, String, Object[])

ILoggerObjectExtensions.LogTrace<TState>(ILoggerObject, TState, EventId, Exception, Func<TState, Exception, String>)

ILoggerObjectExtensions.LogCritical(ILoggerObject, String, Object[])

ILoggerObjectExtensions.LogDebug(ILoggerObject, String, Object[])

Class ApproximateOnlineFeatures

init Pressure, Altitude, Azimuth features with default values.

Default Pipeline Output: Features.Pressure, Features.Altitude, Features.Azimuth

Inheritance

System.Object
PipelineBase
ApproximateOnlineFeatures

Implements

ILoggerObject
IProgress
ITransformation
IPipelineIO

Inherited Members

PipelineBase.PipelineInputs
PipelineBase.PipelineOutputs
PipelineBase.Logger
PipelineBase.Progress
PipelineBase.ProgressChanged
PipelineBase.OnProgressChanged()
System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: SigStat.Common.Transforms

Assembly: SigStat.Common.dll

Syntax

```
[JsonObject(MemberSerialization.OptOut)]
public class ApproximateOnlineFeatures : PipelineBase, ILoggerObject, IProgress, ITransformation, IPipelineIO
```

Methods

Transform(Signature)

Declaration

```
public void Transform(Signature signature)
```

Parameters

TYPE	NAME	DESCRIPTION
Signature	signature	

Implements

ILoggerObject
IProgress
ITransformation

IPipelineIO

Extension Methods

ILoggerObjectExtensions.LogError(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogError(ILoggerObject, Exception, String, Object[])
ILoggerObjectExtensions.LogInformation(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogWarning(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogWarning(ILoggerObject, Exception, String, Object[])
ILoggerObjectExtensions.LogTrace(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogTrace<TState>(ILoggerObject, TState, EventId, Exception, Func<TState, Exception, String>)
ILoggerObjectExtensions.LogCritical(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogDebug(ILoggerObject, String, Object[])

Class Binarization

Generates a binary raster version of the input image with the iterative threshold method.

Pipeline Input type: Image{Rgba32}

Default Pipeline Output: (bool[,]) Binarized

Inheritance

System.Object
PipelineBase
Binarization

Implements

ILoggerObject
IProgress
ITransformation
IPipelineIO

Inherited Members

PipelineBase.PipelineInputs
PipelineBase.PipelineOutputs
PipelineBase.Logger
PipelineBase.Progress
PipelineBase.ProgressChanged
PipelineBase.OnProgressChanged()
System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: SigStat.Common.Transforms

Assembly: SigStat.Common.dll

Syntax

```
[JsonObject(MemberSerialization.OptOut)]
public class Binarization : PipelineBase, ILoggerObject, IProgress, ITransformation, IPipelineIO
```

Constructors

Binarization()

Initializes a new instance of the Binarization class with default settings: Iterative threshold and Dark.

Declaration

```
public Binarization()
```

Binarization(Binarization.ForegroundType, Nullable<Double>)

Initializes a new instance of the Binarization class with specified settings.

Declaration

```
public Binarization(Binarization.ForegroundType foregroundType, double? binThreshold)
```

Parameters

TYPE	NAME	DESCRIPTION
Binarization.ForegroundType	foregroundType	
System.Nullable<System.Double>	binThreshold	Use this threshold value instead of iteratively calculating it. Range from 0 to 1

Properties

InputImage

Gets or sets the featuredescriptor of the input image.

Declaration

```
[Input(AutoSetMode.IfNull)]
[JsonProperty]
public FeatureDescriptor<Image<Rgba32>> InputImage { get; set; }
```

Property Value

TYPE	DESCRIPTION
FeatureDescriptor<SixLabors.ImageSharp.Image<SixLabors.ImageSharp.PixelFormats.Rgba32>>	

OutputMask

Gets or sets the featuredescriptor of a the binarized image.

Declaration

```
[Output("Binarized")]
public FeatureDescriptor<bool[, ]> OutputMask { get; set; }
```

Property Value

TYPE	DESCRIPTION
FeatureDescriptor<System.Boolean[,]>	

Methods

Transform(Signature)

Declaration

```
public void Transform(Signature signature)
```

Parameters

TYPE	NAME	DESCRIPTION
Signature	signature	

Implements

ILoggerObject
IProgress
ITransformation

IPipelineIO

Extension Methods

ILoggerObjectExtensions.LogError(ILoggerObject, String, Object[])

ILoggerObjectExtensions.LogError(ILoggerObject, Exception, String, Object[])

ILoggerObjectExtensions.LogInformation(ILoggerObject, String, Object[])

ILoggerObjectExtensions.LogWarning(ILoggerObject, String, Object[])

ILoggerObjectExtensions.LogWarning(ILoggerObject, Exception, String, Object[])

ILoggerObjectExtensions.LogTrace(ILoggerObject, String, Object[])

ILoggerObjectExtensions.LogTrace<TState>(ILoggerObject, TState, EventId, Exception, Func<TState, Exception, String>)

ILoggerObjectExtensions.LogCritical(ILoggerObject, String, Object[])

ILoggerObjectExtensions.LogDebug(ILoggerObject, String, Object[])

Enum Binarization.ForegroundType

Represents the type of the input image.

Namespace: SigStat.Common.Transforms

Assembly: SigStat.Common.dll

Syntax

```
public enum ForegroundType
```

Fields

NAME	DESCRIPTION
Bright	Foreground is brighter than background. (for non-signature images)
Dark	(default) Foreground is darker than background. (eg. ink on paper)

Class BinaryRasterizer

Converts standard features to a binary raster.

Default Pipeline Input: Standard Features

Default Pipeline Output: (bool[,]) Binarized

Inheritance

System.Object
PipelineBase
BinaryRasterizer

Implements

ILoggerObject
IProgress
ITransformation
IPipelineIO

Inherited Members

PipelineBase.PipelineInputs
PipelineBase.PipelineOutputs
PipelineBase.Logger
PipelineBase.Progress
PipelineBase.ProgressChanged
PipelineBase.OnProgressChanged()
System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: SigStat.Common.Transforms

Assembly: SigStat.Common.dll

Syntax

```
[JsonObject(MemberSerialization.OptOut)]
public class BinaryRasterizer : PipelineBase, ILoggerObject, IProgress, ITransformation, IPipelineIO
```

Constructors

BinaryRasterizer(Int32, Int32, Single)

Initializes a new instance of the BinaryRasterizer class with specified raster size and pen width.

Declaration

```
public BinaryRasterizer(int resolutionX, int resolutionY, float penWidth)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	resolutionX	Raster width.

TYPE	NAME	DESCRIPTION
System.Int32	resolutionY	Raster height.
System.Single	penWidth	

Properties

InputButton

Gets or sets the FeatureDescriptor representing the stroke endings of an online signature

Declaration

```
[Input(AutoSetMode.IfNull)]
public FeatureDescriptor<List<bool>> InputButton { get; set; }
```

Property Value

TYPE	DESCRIPTION
FeatureDescriptor<System.Collections.Generic.List<System.Boolean>>	

InputX

Gets or sets the FeatureDescriptor representing the X coordinates of an online signature

Declaration

```
[Input(AutoSetMode.IfNull)]
public FeatureDescriptor<List<double>> InputX { get; set; }
```

Property Value

TYPE	DESCRIPTION
FeatureDescriptor<System.Collections.Generic.List<System.Double>>	

InputY

Gets or sets the FeatureDescriptor representing the Y coordinates of an online signature

Declaration

```
[Input(AutoSetMode.IfNull)]
public FeatureDescriptor<List<double>> InputY { get; set; }
```

Property Value

TYPE	DESCRIPTION
FeatureDescriptor<System.Collections.Generic.List<System.Double>>	

Output

Gets or sets the FeatureDescriptor representing the output of the transformation

Declaration

```
[Output("Binarized")]
public FeatureDescriptor<bool[, ]> Output { get; set; }
```

Property Value

TYPE	DESCRIPTION
FeatureDescriptor<System.Boolean[,]>	

Methods

Transform(Signature)

Declaration

```
public void Transform(Signature signature)
```

Parameters

TYPE	NAME	DESCRIPTION
Signature	signature	

Implements

ILoggerObject
IProgress
ITransformation
IPipelineIO

Extension Methods

ILoggerObjectExtensions.LogError(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogError(ILoggerObject, Exception, String, Object[])
ILoggerObjectExtensions.LogInformation(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogWarning(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogWarning(ILoggerObject, Exception, String, Object[])
ILoggerObjectExtensions.LogTrace(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogTrace<TState>(ILoggerObject, TState, EventId, Exception, Func<TState, Exception, String>)
ILoggerObjectExtensions.LogCritical(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogDebug(ILoggerObject, String, Object[])

Class CentroidExtraction

Extracts the Centroid (aka. Center Of Gravity) of the input features.

Default Pipeline Output: (List{double}) Centroid.

Inheritance

System.Object
PipelineBase
CentroidExtraction

Implements

ILoggerObject
IProgress
ITransformation
IPipelineIO

Inherited Members

PipelineBase.PipelineInputs
PipelineBase.PipelineOutputs
PipelineBase.Logger
PipelineBase.Progress
PipelineBase.ProgressChanged
PipelineBase.OnProgressChanged()
System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: SigStat.Common.Transforms

Assembly: SigStat.Common.dll

Syntax

```
[JsonObject(MemberSerialization.OptOut)]
public class CentroidExtraction : PipelineBase, ILoggerObject, IProgress, ITransformation, IPipelineIO
```

Properties

Inputs

List of features to process

Declaration

```
[Input(AutoSetMode.IfNull)]
public List<FeatureDescriptor<List<double>>> Inputs { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Collections.Generic.List<FeatureDescriptor<System.Collections.Generic.List<System.Double>>>	

OutputCentroid

List of centroid values

Declaration

```
[Output("Centroid")]
public FeatureDescriptor<List<double>> OutputCentroid { get; set; }
```

Property Value

TYPE	DESCRIPTION
FeatureDescriptor<System.Collections.Generic.List<System.Double>>	

Methods

Transform(Signature)

Declaration

```
public void Transform(Signature signature)
```

Parameters

TYPE	NAME	DESCRIPTION
Signature	signature	

Implements

ILoggerObject

IProgress

ITransformation

IPipelineIO

Extension Methods

ILoggerObjectExtensions.LogError(ILoggerObject, String, Object[])

ILoggerObjectExtensions.LogError(ILoggerObject, Exception, String, Object[])

ILoggerObjectExtensions.LogInformation(ILoggerObject, String, Object[])

ILoggerObjectExtensions.LogWarning(ILoggerObject, String, Object[])

ILoggerObjectExtensions.LogWarning(ILoggerObject, Exception, String, Object[])

ILoggerObjectExtensions.LogTrace(ILoggerObject, String, Object[])

ILoggerObjectExtensions.LogTrace<TState>(ILoggerObject, TState, EventId, Exception, Func<TState, Exception, String>)

ILoggerObjectExtensions.LogCritical(ILoggerObject, String, Object[])

ILoggerObjectExtensions.LogDebug(ILoggerObject, String, Object[])

Class CentroidTranslate

Sequential pipeline to translate X and Y Features to Centroid. The following Transforms are called: CentroidExtraction, Multiply(-1), Translate

Default Pipeline Input: X, Y

Default Pipeline Output: (List{double}) Centroid

Inheritance

System.Object
PipelineBase
SequentialTransformPipeline
CentroidTranslate

Implements

ILoggerObject
IProgress
System.Collections.IEnumerable
ITransformation
IPipelineIO

Inherited Members

SequentialTransformPipeline.Items
SequentialTransformPipeline.PipelineInputs
SequentialTransformPipeline.PipelineOutputs
SequentialTransformPipeline.GetEnumerator()
SequentialTransformPipeline.Add(ITransformation)
SequentialTransformPipeline.Transform(Signature)
PipelineBase.Logger
PipelineBase.Progress
PipelineBase.ProgressChanged
PipelineBase.OnProgressChanged()
System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: SigStat.Common.Transforms

Assembly: SigStat.Common.dll

Syntax

```
[JsonObject(MemberSerialization.OptOut)]
public class CentroidTranslate : SequentialTransformPipeline, ILoggerObject, IProgress, IEnumerable,
ITransformation, IPipelineIO
```

Remarks

This is a special case of Translate

Constructors

CentroidTranslate()

Initializes a new instance of the CentroidTranslate class.

Declaration

```
public CentroidTranslate()
```

Properties

InputX

Gets or sets the input feature representing the X coordinates of an online signature

Declaration

```
[Input(AutoSetMode.IfNull)]
public FeatureDescriptor<List<double>> InputX { get; set; }
```

Property Value

TYPE	DESCRIPTION
FeatureDescriptor<System.Collections.Generic.List<System.Double>>	

InputY

Gets or sets the input feature representing the Y coordinates of an online signature

Declaration

```
[Input(AutoSetMode.IfNull)]
public FeatureDescriptor<List<double>> InputY { get; set; }
```

Property Value

TYPE	DESCRIPTION
FeatureDescriptor<System.Collections.Generic.List<System.Double>>	

OutputX

Gets or sets the output feature representing the X coordinates of an online signature

Declaration

```
[Output("X")]
public FeatureDescriptor<List<double>> OutputX { get; set; }
```

Property Value

TYPE	DESCRIPTION
FeatureDescriptor<System.Collections.Generic.List<System.Double>>	

OutputY

Gets or sets the output feature representing the X coordinates of an online signature

Declaration

```
[Output("Y")]
public FeatureDescriptor<List<double>> OutputY { get; set; }
```

Property Value

TYPE	DESCRIPTION
FeatureDescriptor<System.Collections.Generic.List<System.Double>>	

Implements

ILoggerObject
IProgress
System.Collections.IEnumerable
ITransformation
IPipelineIO

Extension Methods

ILoggerObjectExtensions.LogError(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogError(ILoggerObject, Exception, String, Object[])
ILoggerObjectExtensions.LogInformation(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogWarning(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogWarning(ILoggerObject, Exception, String, Object[])
ILoggerObjectExtensions.LogTrace(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogTrace<TState>(ILoggerObject, TState, EventId, Exception, Func<TState, Exception, String>)
ILoggerObjectExtensions.LogCritical(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogDebug(ILoggerObject, String, Object[])

Class ComponentExtraction

Extracts unsorted components by tracing through the binary Skeleton raster.

Default Pipeline Input: (bool[,]) Skeleton, (List{Point}) EndPoints, (List{Point}) CrossingPoints

Default Pipeline Output: (List{List{PointF}}) Components

Inheritance

System.Object
PipelineBase
ComponentExtraction

Implements

ILoggerObject
IProgress
ITransformation
IPipelineIO

Inherited Members

PipelineBase.PipelineInputs
PipelineBase.PipelineOutputs
PipelineBase.Logger
PipelineBase.Progress
PipelineBase.ProgressChanged
PipelineBase.OnProgressChanged()
System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: SigStat.Common.Transforms

Assembly: SigStat.Common.dll

Syntax

```
[JsonObject(MemberSerialization.OptOut)]
public class ComponentExtraction : PipelineBase, ILoggerObject, IProgress, ITransformation, IPipelineIO
```

Constructors

ComponentExtraction(Int32)

Initializes a new instance of the ComponentExtraction class with specified sampling resolution.

Declaration

```
public ComponentExtraction(int samplingResolution)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	samplingResolution	Steps to trace before a new point is sampled. Smaller values result in a more precise tracing. Provide a positive value.

Properties

CrossingPoints

crossing points

Declaration

```
[Input(AutoSetMode.IfNull)]
public FeatureDescriptor<List<Point>> CrossingPoints { get; set; }
```

Property Value

TYPE	DESCRIPTION
FeatureDescriptor<System.Collections.Generic.List<System.Drawing.Point>>	

EndPoints

endpoints

Declaration

```
[Input(AutoSetMode.IfNull)]
public FeatureDescriptor<List<Point>> EndPoints { get; set; }
```

Property Value

TYPE	DESCRIPTION
FeatureDescriptor<System.Collections.Generic.List<System.Drawing.Point>>	

OutputComponents

Output components

Declaration

```
[Output("Components")]
public FeatureDescriptor<List<List<PointF>>> OutputComponents { get; set; }
```

Property Value

TYPE	DESCRIPTION
FeatureDescriptor<System.Collections.Generic.List<System.Collections.Generic.List<System.Drawing.PointF>>>	

Skeleton

binary representation of a signature image

Declaration

```
[Input(AutoSetMode.IfNull)]
public FeatureDescriptor<bool[, ]> Skeleton { get; set; }
```

Property Value

TYPE	DESCRIPTION
FeatureDescriptor<System.Boolean[,]>	

Methods

Transform(Signature)

Declaration

```
public void Transform(Signature signature)
```

Parameters

TYPE	NAME	DESCRIPTION
Signature	signature	

Implements

ILoggerObject
IProgress
ITransformation
IPipelineIO

Extension Methods

ILoggerObjectExtensions.LogError(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogError(ILoggerObject, Exception, String, Object[])
ILoggerObjectExtensions.LogInformation(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogWarning(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogWarning(ILoggerObject, Exception, String, Object[])
ILoggerObjectExtensions.LogTrace(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogTrace<TState>(ILoggerObject, TState, EventId, Exception, Func<TState, Exception, String>)
ILoggerObjectExtensions.LogCritical(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogDebug(ILoggerObject, String, Object[])

Class ComponentSorter

Sorts Component order by comparing each starting X value, and finding nearest components.

Default Pipeline Input: (bool[,]) Components

Default Pipeline Output: (bool[,]) Components

Inheritance

System.Object
PipelineBase
ComponentSorter

Implements

ILoggerObject
IProgress
ITransformation
IPipelineIO

Inherited Members

PipelineBase.PipelineInputs
PipelineBase.PipelineOutputs
PipelineBase.Logger
PipelineBase.Progress
PipelineBase.ProgressChanged
PipelineBase.OnProgressChanged()
System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: SigStat.Common.Transforms

Assembly: SigStat.Common.dll

Syntax

```
[JsonObject(MemberSerialization.OptOut)]
public class ComponentSorter : PipelineBase, ILoggerObject, IProgress, ITransformation, IPipelineIO
```

Properties

Input

Gets or sets the input.

Declaration

```
[Input(AutoSetMode.IfNull)]
public FeatureDescriptor<List<List<PointF>>> Input { get; set; }
```

Property Value

TYPE	DESCRIPTION
FeatureDescriptor<System.Collections.Generic.List<System.Collections.Generic.List<System.Drawing.PointF>>>	

Output

Gets or sets the output.

Declaration

```
[Output("Components")]
public FeatureDescriptor<List<List<PointF>>> Output { get; set; }
```

Property Value

TYPE	DESCRIPTION
FeatureDescriptor<System.Collections.Generic.List<System.Collections.Generic.List<System.Drawing.PointF>>>	

Methods

Transform(Signature)

Declaration

```
public void Transform(Signature signature)
```

Parameters

TYPE	NAME	DESCRIPTION
Signature	signature	

Implements

ILoggerObject
IProgress
ITransformation
IPipelineIO

Extension Methods

ILoggerObjectExtensions.LogError(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogError(ILoggerObject, Exception, String, Object[])
ILoggerObjectExtensions.LogInformation(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogWarning(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogWarning(ILoggerObject, Exception, String, Object[])
ILoggerObjectExtensions.LogTrace(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogTrace<TState>(ILoggerObject, TState, EventId, Exception, Func<TState, Exception, String>)
ILoggerObjectExtensions.LogCritical(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogDebug(ILoggerObject, String, Object[])

Class ComponentsToFeatures

Extracts standard Features from sorted Components.

Default Pipeline Input: (List{List{PointF}}) Components

Default Pipeline Output: X, Y, Button Features

Inheritance

System.Object
PipelineBase
ComponentsToFeatures

Implements

ILoggerObject
IProgress
ITransformation
IPipelineIO

Inherited Members

PipelineBase.PipelineInputs
PipelineBase.PipelineOutputs
PipelineBase.Logger
PipelineBase.Progress
PipelineBase.ProgressChanged
PipelineBase.OnProgressChanged()
System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: SigStat.Common.Transforms

Assembly: SigStat.Common.dll

Syntax

```
[JsonObject(MemberSerialization.OptOut)]
public class ComponentsToFeatures : PipelineBase, ILoggerObject, IProgress, ITransformation, IPipelineIO
```

Properties

Button

Button

Declaration

```
[Output("Button")]
public FeatureDescriptor<List<bool>> Button { get; set; }
```

Property Value

TYPE	DESCRIPTION
FeatureDescriptor<System.Collections.Generic.List<System.Boolean>>	

InputComponents

Components

Declaration

```
[Input(AutoSetMode.IfNull)]
public FeatureDescriptor<List<List<PointF>>> InputComponents { get; set; }
```

Property Value

TYPE	DESCRIPTION
FeatureDescriptor<System.Collections.Generic.List<System.Collections.Generic.List<System.Drawing.PointF>>>	

X

X

Declaration

```
[Output("X")]
public FeatureDescriptor<List<double>> X { get; set; }
```

Property Value

TYPE	DESCRIPTION
FeatureDescriptor<System.Collections.Generic.List<System.Double>>	

Y

Y

Declaration

```
[Output("Y")]
public FeatureDescriptor<List<double>> Y { get; set; }
```

Property Value

TYPE	DESCRIPTION
FeatureDescriptor<System.Collections.Generic.List<System.Double>>	

Methods

Transform(Signature)

Declaration

```
public void Transform(Signature signature)
```

Parameters

TYPE	NAME	DESCRIPTION
Signature	signature	

Implements

ILoggerObject

IProgress

ITransformation

IPipelineIO

Extension Methods

ILoggerObjectExtensions.LogError(ILoggerObject, String, Object[])

ILoggerObjectExtensions.LogError(ILoggerObject, Exception, String, Object[])

ILoggerObjectExtensions.LogInformation(ILoggerObject, String, Object[])

ILoggerObjectExtensions.LogWarning(ILoggerObject, String, Object[])

ILoggerObjectExtensions.LogWarning(ILoggerObject, Exception, String, Object[])

ILoggerObjectExtensions.LogTrace(ILoggerObject, String, Object[])

ILoggerObjectExtensions.LogTrace<TState>(ILoggerObject, TState, EventId, Exception, Func<TState, Exception, String>)

ILoggerObjectExtensions.LogCritical(ILoggerObject, String, Object[])

ILoggerObjectExtensions.LogDebug(ILoggerObject, String, Object[])

Class EndpointExtraction

Extracts EndPoints and CrossingPoints from Skeleton.

Default Pipeline Input: (bool[,]) Skeleton

Default Pipeline Output: (List{Point}) EndPoints, (List{Point}) CrossingPoints

Inheritance

System.Object
PipelineBase
EndpointExtraction

Implements

ILoggerObject
IProgress
ITransformation
IPipelineIO

Inherited Members

PipelineBase.PipelineInputs
PipelineBase.PipelineOutputs
PipelineBase.Logger
PipelineBase.Progress
PipelineBase.ProgressChanged
PipelineBase.OnProgressChanged()
System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: SigStat.Common.Transforms

Assembly: SigStat.Common.dll

Syntax

```
[JsonObject(MemberSerialization.OptOut)]
public class EndpointExtraction : PipelineBase, ILoggerObject, IProgress, ITransformation, IPipelineIO
```

Properties

OutputCrossingPoints

OutputCrossingPoints

Declaration

```
[Output("CrossingPoints")]
public FeatureDescriptor<List<Point>> OutputCrossingPoints { get; set; }
```

Property Value

TYPE	DESCRIPTION
FeatureDescriptor<System.Collections.Generic.List<System.Drawing.Point>>	

OutputEndpoints

OutputEndpoints

Declaration

```
[Output("EndPoints")]
public FeatureDescriptor<List<Point>> OutputEndpoints { get; set; }
```

Property Value

TYPE	DESCRIPTION
FeatureDescriptor<System.Collections.Generic.List<System.Drawing.Point>>	

Skeleton

Binary representation of an image

Declaration

```
[Input(AutoSetMode.IfNull)]
public FeatureDescriptor<bool[, ]> Skeleton { get; set; }
```

Property Value

TYPE	DESCRIPTION
FeatureDescriptor<System.Boolean[,]>	

Methods

Transform(Signature)

Declaration

```
public void Transform(Signature signature)
```

Parameters

TYPE	NAME	DESCRIPTION
Signature	signature	

Implements

ILoggerObject
IProgress
ITransformation
IPipelineIO

Extension Methods

ILoggerObjectExtensions.LogError(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogError(ILoggerObject, Exception, String, Object[])
ILoggerObjectExtensions.LogInformation(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogWarning(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogWarning(ILoggerObject, Exception, String, Object[])
ILoggerObjectExtensions.LogTrace(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogTrace<TState>(ILoggerObject, TState, EventId, Exception, Func<TState, Exception, String>)
ILoggerObjectExtensions.LogCritical(ILoggerObject, String, Object[])

ILoggerObjectExtensions.LogDebug(ILoggerObject, String, Object[])

Class Extrema

Extracts minimum and maximum values of given feature.

Default Pipeline Output: (List{double}) Min, (List{double}) Max

Inheritance

System.Object
PipelineBase
Extrema

Implements

ILoggerObject
IProgress
ITransformation
IPipelineIO

Inherited Members

PipelineBase.PipelineInputs
PipelineBase.PipelineOutputs
PipelineBase.Logger
PipelineBase.Progress
PipelineBase.ProgressChanged
PipelineBase.OnProgressChanged()
System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: SigStat.Common.Transforms

Assembly: SigStat.Common.dll

Syntax

```
[JsonObject(MemberSerialization.OptOut)]
public class Extrema : PipelineBase, ILoggerObject, IProgress, ITransformation, IPipelineIO
```

Remarks

Output features are lists, containing only one value each.

Methods

Transform(Signature)

Declaration

```
public void Transform(Signature signature)
```

Parameters

TYPE	NAME	DESCRIPTION
Signature	signature	

Implements

ILoggerObject

IProgress

ITransformation

IPipelineIO

Extension Methods

ILoggerObjectExtensions.LogError(ILoggerObject, String, Object[])

ILoggerObjectExtensions.LogError(ILoggerObject, Exception, String, Object[])

ILoggerObjectExtensions.LogInformation(ILoggerObject, String, Object[])

ILoggerObjectExtensions.LogWarning(ILoggerObject, String, Object[])

ILoggerObjectExtensions.LogWarning(ILoggerObject, Exception, String, Object[])

ILoggerObjectExtensions.LogTrace(ILoggerObject, String, Object[])

ILoggerObjectExtensions.LogTrace<TState>(ILoggerObject, TState, EventId, Exception, Func<TState, Exception, String>)

ILoggerObjectExtensions.LogCritical(ILoggerObject, String, Object[])

ILoggerObjectExtensions.LogDebug(ILoggerObject, String, Object[])

Class HSCPThinning

Iteratively thins the input binary raster with the HSCPThinningStep algorithm.

Pipeline Input type: bool[,]

Default Pipeline Output: (bool[,]) HSCPThinningResult

Inheritance

System.Object

PipelineBase

HSCPThinning

Implements

ILoggerObject

IProgress

ITransformation

IPipelineIO

Inherited Members

PipelineBase.PipelineInputs

PipelineBase.PipelineOutputs

PipelineBase.Logger

PipelineBase.Progress

PipelineBase.ProgressChanged

PipelineBase.OnProgressChanged()

System.Object.Equals(System.Object)

System.Object.Equals(System.Object, System.Object)

System.Object.GetHashCode()

System.Object.GetType()

System.Object.MemberwiseClone()

System.Object.ReferenceEquals(System.Object, System.Object)

System.Object.ToString()

Namespace: SigStat.Common.Transforms

Assembly: SigStat.Common.dll

Syntax

```
[JsonObject(MemberSerialization.OptOut)]
public class HSCPThinning : PipelineBase, ILoggerObject, IProgress, ITransformation, IPipelineIO
```

Properties

Input

Input FeatureDescriptor for the binary image of the signature

Declaration

```
[Input(AutoSetMode.IfNull)]
public FeatureDescriptor<bool[, ]> Input { get; set; }
```

Property Value

TYPE	DESCRIPTION
FeatureDescriptor<System.Boolean[,]>	

Output

Output FeatureDescriptor for the binary image of the signature

Declaration

```
[Output("HSCPThinningResult")]
public FeatureDescriptor<bool[, ]> Output { get; set; }
```

Property Value

TYPE	DESCRIPTION
FeatureDescriptor<System.Boolean[,]>	

Methods

Transform(Signature)

Declaration

```
public void Transform(Signature signature)
```

Parameters

TYPE	NAME	DESCRIPTION
Signature	signature	

Implements

ILoggerObject
IProgress
ITransformation
IPipelineIO

Extension Methods

ILoggerObjectExtensions.LogError(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogError(ILoggerObject, Exception, String, Object[])
ILoggerObjectExtensions.LogInformation(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogWarning(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogWarning(ILoggerObject, Exception, String, Object[])
ILoggerObjectExtensions.LogTrace(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogTrace<TState>(ILoggerObject, TState, EventId, Exception, Func<TState, Exception, String>)
ILoggerObjectExtensions.LogCritical(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogDebug(ILoggerObject, String, Object[])

Class ImageGenerator

Generates an image feature out of a binary raster. Optionally, saves the image to a png file. Useful for debugging pipeline steps.

Pipeline Input type: bool[,]

Default Pipeline Output: (bool[,]) Input, (Image{Rgba32}) InputImage

Inheritance

System.Object
PipelineBase
ImageGenerator

Implements

ILoggerObject
IProgress
ITransformation
IPipelineIO

Inherited Members

PipelineBase.PipelineInputs
PipelineBase.PipelineOutputs
PipelineBase.Logger
PipelineBase.Progress
PipelineBase.ProgressChanged
PipelineBase.OnProgressChanged()
System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: SigStat.Common.Transforms

Assembly: SigStat.Common.dll

Syntax

```
[JsonObject(MemberSerialization.OptOut)]
public class ImageGenerator : PipelineBase, ILoggerObject, IProgress, ITransformation, IPipelineIO
```

Constructors

ImageGenerator()

Initializes a new instance of the ImageGenerator class with default settings: skip file writing, Blue ink on white paper.

Declaration

```
public ImageGenerator()
```

ImageGenerator(Boolean)

Initializes a new instance of the ImageGenerator class with default settings.

Declaration

```
public ImageGenerator(bool writeToFile)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Boolean	writeToFile	Whether to save the generated image into a png file.

ImageGenerator(Boolean, Rgba32, Rgba32)

Initializes a new instance of the ImageGenerator class with specified settings.

Declaration

```
public ImageGenerator(bool writeToFile, Rgba32 foregroundColor, Rgba32 backgroundColor)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Boolean	writeToFile	Whether to save the generated image into a png file.
SixLabors.ImageSharp.PixelFormats.Rgba32	foregroundColor	Ink color.
SixLabors.ImageSharp.PixelFormats.Rgba32	backgroundColor	Paper color.

Properties

BackgroundColor

Gets or sets the color of the backgroung used to render the signature

Declaration

```
public Rgba32 BackgroundColor { get; set; }
```

Property Value

TYPE	DESCRIPTION
SixLabors.ImageSharp.PixelFormats.Rgba32	

ForegroundColor

Gets or sets the color of the foreground used to render the signature

Declaration

```
public Rgba32 ForegroundColor { get; set; }
```

Property Value

TYPE	DESCRIPTION
SixLabors.ImageSharp.PixelFormats.Rgba32	

Input

Input FeatureDescriptor for the binary image of a signature

Declaration

```
[Input(AutoSetMode.IfNull)]
public FeatureDescriptor<bool[, ]> Input { get; set; }
```

Property Value

TYPE	DESCRIPTION
FeatureDescriptor<System.Boolean[,]>	

OutputImage

Input FeatureDescriptor for the binary image of a signature

Declaration

```
[Output("Image")]
public FeatureDescriptor<Image<Rgba32>> OutputImage { get; set; }
```

Property Value

TYPE	DESCRIPTION
FeatureDescriptor<SixLabors.ImageSharp.Image<SixLabors.ImageSharp.PixelFormats.Rgba32>>	

WriteToFile

Gets or sets a value indicating whether the results should be saved to a file or not.

Declaration

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

Property Value

TYPE	DESCRIPTION
System.Boolean	`true` if results should be saved to a file otherwise, `false`.

Methods

Transform(Signature)

Declaration

```
public void Transform(Signature signature)
```

Parameters

TYPE	NAME	DESCRIPTION
Signature	signature	

Implements

ILoggerObject
IProgress

ITransformation

IPipelineIO

Extension Methods

ILoggerObjectExtensions.LogError(ILoggerObject, String, Object[])

ILoggerObjectExtensions.LogError(ILoggerObject, Exception, String, Object[])

ILoggerObjectExtensions.LogInformation(ILoggerObject, String, Object[])

ILoggerObjectExtensions.LogWarning(ILoggerObject, String, Object[])

ILoggerObjectExtensions.LogWarning(ILoggerObject, Exception, String, Object[])

ILoggerObjectExtensions.LogTrace(ILoggerObject, String, Object[])

ILoggerObjectExtensions.LogTrace<TState>(ILoggerObject, TState, EventId, Exception, Func<TState, Exception, String>)

ILoggerObjectExtensions.LogCritical(ILoggerObject, String, Object[])

ILoggerObjectExtensions.LogDebug(ILoggerObject, String, Object[])

Class Map

Maps values of a feature to a specified range.

Pipeline Input type: List{double}

Default Pipeline Output: (List{double}) MapResult

Inheritance

System.Object
PipelineBase
Map

Implements

ILoggerObject
IProgress
ITransformation
IPipelineIO

Inherited Members

PipelineBase.PipelineInputs
PipelineBase.PipelineOutputs
PipelineBase.Logger
PipelineBase.Progress
PipelineBase.ProgressChanged
PipelineBase.OnProgressChanged()
System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: SigStat.Common.Transforms

Assembly: SigStat.Common.dll

Syntax

```
[JsonObject(MemberSerialization.OptOut)]
public class Map : PipelineBase, ILoggerObject, IProgress, ITransformation, IPipelineIO
```

Constructors

Map(Double, Double)

Initializes a new instance of the Map class with specified settings.

Declaration

```
public Map(double minVal, double maxVal)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Double	minVal	New minimum value.

TYPE	NAME	DESCRIPTION
System.Double	maxVal	New maximum value.

Properties

Input

Input

Declaration

```
[Input(AutoSetMode.IfNull)]
public FeatureDescriptor<List<double>> Input { get; set; }
```

Property Value

TYPE	DESCRIPTION
FeatureDescriptor<System.Collections.Generic.List<System.Double>>	

Output

Output

Declaration

```
[Output("MapResult")]
public FeatureDescriptor<List<double>> Output { get; set; }
```

Property Value

TYPE	DESCRIPTION
FeatureDescriptor<System.Collections.Generic.List<System.Double>>	

Methods

Transform(Signature)

Declaration

```
public void Transform(Signature signature)
```

Parameters

TYPE	NAME	DESCRIPTION
Signature	signature	

Implements

ILoggerObject
IProgress
ITransformation
IPipelineIO

Extension Methods

ILoggerObjectExtensions.LogError(ILoggerObject, String, Object[])

ILoggerObjectExtensions.LogError(ILoggerObject, Exception, String, Object[])

ILoggerObjectExtensions.LogInformation(ILoggerObject, String, Object[])

ILoggerObjectExtensions.LogWarning(ILoggerObject, String, Object[])

ILoggerObjectExtensions.LogWarning(ILoggerObject, Exception, String, Object[])

ILoggerObjectExtensions.LogTrace(ILoggerObject, String, Object[])

ILoggerObjectExtensions.LogTrace<TState>(ILoggerObject, TState, EventId, Exception, Func<TState, Exception, String>)

ILoggerObjectExtensions.LogCritical(ILoggerObject, String, Object[])

ILoggerObjectExtensions.LogDebug(ILoggerObject, String, Object[])

Class Multiply

Multiplies the values of a feature with a given constant.

Pipeline Input type: List{double}

Default Pipeline Output: (List{double}) Input

Inheritance

System.Object
PipelineBase
Multiply

Implements

ILoggerObject
IProgress
ITransformation
IPipelineIO

Inherited Members

PipelineBase.PipelineInputs
PipelineBase.PipelineOutputs
PipelineBase.Logger
PipelineBase.Progress
PipelineBase.ProgressChanged
PipelineBase.OnProgressChanged()
System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: SigStat.Common.Transforms

Assembly: SigStat.Common.dll

Syntax

```
[JsonObject(MemberSerialization.OptOut)]
public class Multiply : PipelineBase, ILoggerObject, IProgress, ITransformation, IPipelineIO
```

Constructors

Multiply(Double)

Initializes a new instance of the Multiply class with specified settings.

Declaration

```
public Multiply(double byConst)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Double	byConst	The value to multiply the input feature by.

Properties

InputList

Input

Declaration

```
[Input(AutoSetMode.IfNull)]
public FeatureDescriptor<List<double>> InputList { get; set; }
```

Property Value

TYPE	DESCRIPTION
FeatureDescriptor<System.Collections.Generic.List<System.Double>>	

Output

Output

Declaration

```
[Output]
public FeatureDescriptor<List<double>> Output { get; set; }
```

Property Value

TYPE	DESCRIPTION
FeatureDescriptor<System.Collections.Generic.List<System.Double>>	

Methods

Transform(Signature)

Declaration

```
public void Transform(Signature signature)
```

Parameters

TYPE	NAME	DESCRIPTION
Signature	signature	

Implements

ILoggerObject
IProgress
ITransformation
IPipelineIO

Extension Methods

ILoggerObjectExtensions.LogError(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogError(ILoggerObject, Exception, String, Object[])
ILoggerObjectExtensions.LogInformation(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogWarning(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogWarning(ILoggerObject, Exception, String, Object[])
ILoggerObjectExtensions.LogTrace(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogTrace<TState>(ILoggerObject, TState, EventId, Exception, Func<TState, Exception, String>)

ILoggerObjectExtensions.LogCritical(ILoggerObject, String, Object[])

ILoggerObjectExtensions.LogDebug(ILoggerObject, String, Object[])

Class Normalize

Maps values of a feature to 0.0 - 1.0 range.

Pipeline Input type: List{double}

Default Pipeline Output: (List{double}) NormalizationResult

Inheritance

System.Object
PipelineBase
Normalize

Implements

ILoggerObject
IProgress
ITransformation
IPipelineIO

Inherited Members

PipelineBase.PipelineInputs
PipelineBase.PipelineOutputs
PipelineBase.Logger
PipelineBase.Progress
PipelineBase.ProgressChanged
PipelineBase.OnProgressChanged()
System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: SigStat.Common.Transforms

Assembly: SigStat.Common.dll

Syntax

```
[JsonObject(MemberSerialization.OptOut)]
public class Normalize : PipelineBase, ILoggerObject, IProgress, ITransformation, IPipelineIO
```

Remarks

This is a specific case of the Map transform.

Properties

Input

Input

Declaration

```
[Input(AutoSetMode.IfNull)]
public FeatureDescriptor<List<double>> Input { get; set; }
```

Property Value

TYPE	DESCRIPTION
FeatureDescriptor<System.Collections.Generic.List<System.Double>>	

Output

Output

Declaration

```
[Output]
public FeatureDescriptor<List<double>> Output { get; set; }
```

Property Value

TYPE	DESCRIPTION
FeatureDescriptor<System.Collections.Generic.List<System.Double>>	

Methods

Transform(Signature)

Declaration

```
public void Transform(Signature signature)
```

Parameters

TYPE	NAME	DESCRIPTION
Signature	signature	

Implements

ILoggerObject
IProgress
ITransformation
IPipelineIO

Extension Methods

ILoggerObjectExtensions.LogError(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogError(ILoggerObject, Exception, String, Object[])
ILoggerObjectExtensions.LogInformation(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogWarning(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogWarning(ILoggerObject, Exception, String, Object[])
ILoggerObjectExtensions.LogTrace(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogTrace<TState>(ILoggerObject, TState, EventId, Exception, Func<TState, Exception, String>)
ILoggerObjectExtensions.LogCritical(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogDebug(ILoggerObject, String, Object[])

Class OnePixelThinning

Iteratively thins the input binary raster with the SigStat.Common.Algorithms.OnePixelThinningStep algorithm.

Pipeline Input type: bool[,]

Default Pipeline Output: (bool[,]) OnePixelThinningResult

Inheritance

System.Object
PipelineBase
OnePixelThinning

Implements

ILoggerObject
IProgress
ITransformation
IPipelineIO

Inherited Members

PipelineBase.PipelineInputs
PipelineBase.PipelineOutputs
PipelineBase.Logger
PipelineBase.Progress
PipelineBase.ProgressChanged
PipelineBase.OnProgressChanged()
System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: SigStat.Common.Transforms

Assembly: SigStat.Common.dll

Syntax

```
[JsonObject(MemberSerialization.OptOut)]
public class OnePixelThinning : PipelineBase, ILoggerObject, IProgress, ITransformation, IPipelineIO
```

Properties

Input

Input FeatureDescriptor for the binary image of the signature

Declaration

```
[Input(AutoSetMode.IfNull)]
public FeatureDescriptor<bool[, ]> Input { get; set; }
```

Property Value

TYPE	DESCRIPTION
FeatureDescriptor<System.Boolean[,]>	

Output

Output FeatureDescriptor for the binary image of the signature

Declaration

```
[Output("OnePixelThinningResult")]
public FeatureDescriptor<bool[, ]> Output { get; set; }
```

Property Value

TYPE	DESCRIPTION
FeatureDescriptor<System.Boolean[,]>	

Methods

Transform(Signature)

Declaration

```
public void Transform(Signature signature)
```

Parameters

TYPE	NAME	DESCRIPTION
Signature	signature	

Implements

ILoggerObject
IProgress
ITransformation
IPipelineIO

Extension Methods

ILoggerObjectExtensions.LogError(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogError(ILoggerObject, Exception, String, Object[])
ILoggerObjectExtensions.LogInformation(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogWarning(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogWarning(ILoggerObject, Exception, String, Object[])
ILoggerObjectExtensions.LogTrace(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogTrace<TState>(ILoggerObject, TState, EventId, Exception, Func<TState, Exception, String>)
ILoggerObjectExtensions.LogCritical(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogDebug(ILoggerObject, String, Object[])

Class RealisticImageGenerator

Generates a realistic looking image of the Signature based on standard features. Uses blue ink and white paper. It does NOT save the image to file.

Default Pipeline Input: X, Y, Button, Pressure, Azimuth, Altitude Features

Default Pipeline Output: Image

Inheritance

System.Object
PipelineBase
RealisticImageGenerator

Implements

ILoggerObject
IProgress
ITransformation
IPipelineIO

Inherited Members

PipelineBase.PipelineInputs
PipelineBase.PipelineOutputs
PipelineBase.Logger
PipelineBase.Progress
PipelineBase.ProgressChanged
PipelineBase.OnProgressChanged()
System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: SigStat.Common.Transforms

Assembly: SigStat.Common.dll

Syntax

```
[JsonObject(MemberSerialization.OptOut)]
public class RealisticImageGenerator : PipelineBase, ILoggerObject, IProgress, ITransformation, IPipelineIO
```

Constructors

RealisticImageGenerator(Int32, Int32)

Initializes a new instance of the RealisticImageGenerator class with specified settings.

Declaration

```
public RealisticImageGenerator(int resolutionX, int resolutionY)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	resolutionX	Output image width.

TYPE	NAME	DESCRIPTION
System.Int32	resolutionY	Output image height.

Properties

Altitude

Input FeatureDescriptor describing the altitude values of an online signature

Declaration

```
[Input(AutoSetMode.IfNull)]
public FeatureDescriptor<List<double>> Altitude { get; set; }
```

Property Value

TYPE	DESCRIPTION
FeatureDescriptor<System.Collections.Generic.List<System.Double>>	

Azimuth

Input FeatureDescriptor describing the azimuth values of an online signature

Declaration

```
[Input(AutoSetMode.IfNull)]
public FeatureDescriptor<List<double>> Azimuth { get; set; }
```

Property Value

TYPE	DESCRIPTION
FeatureDescriptor<System.Collections.Generic.List<System.Double>>	

Button

Input FeatureDescriptor describing the stroke endings of an online signature

Declaration

```
[Input(AutoSetMode.IfNull)]
public FeatureDescriptor<List<bool>> Button { get; set; }
```

Property Value

TYPE	DESCRIPTION
FeatureDescriptor<System.Collections.Generic.List<System.Boolean>>	

OutputImage

Output FeatureDescriptor describing the generated image of the signature

Declaration

```
[Output("RealisticImage")]
public FeatureDescriptor<Image<Rgba32>> OutputImage { get; set; }
```

Property Value

TYPE	DESCRIPTION
FeatureDescriptor<SixLabors.ImageSharp.Image<SixLabors.ImageSharp.PixelFormats.Rgba32>>	

Pressure

Input FeatureDescriptor describing the pressure values of an online signature

Declaration

```
[Input(AutoSetMode.IfNull)]
public FeatureDescriptor<List<double>> Pressure { get; set; }
```

Property Value

TYPE	DESCRIPTION
FeatureDescriptor<System.Collections.Generic.List<System.Double>>	

X

Input FeatureDescriptor describing the X coordinates of an online signature

Declaration

```
[Input(AutoSetMode.IfNull)]
public FeatureDescriptor<List<double>> X { get; set; }
```

Property Value

TYPE	DESCRIPTION
FeatureDescriptor<System.Collections.Generic.List<System.Double>>	

Y

Input FeatureDescriptor describing the Y coordinates of an online signature

Declaration

```
[Input(AutoSetMode.IfNull)]
public FeatureDescriptor<List<double>> Y { get; set; }
```

Property Value

TYPE	DESCRIPTION
FeatureDescriptor<System.Collections.Generic.List<System.Double>>	

Methods

Transform(Signature)

Declaration

```
public void Transform(Signature signature)
```

Parameters

TYPE	NAME	DESCRIPTION
Signature	signature	

Implements

ILoggerObject
IProgress
ITransformation
IPipelineIO

Extension Methods

ILoggerObjectExtensions.LogError(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogError(ILoggerObject, Exception, String, Object[])
ILoggerObjectExtensions.LogInformation(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogWarning(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogWarning(ILoggerObject, Exception, String, Object[])
ILoggerObjectExtensions.LogTrace(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogTrace<TState>(ILoggerObject, TState, EventId, Exception, Func<TState, Exception, String>)
ILoggerObjectExtensions.LogCritical(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogDebug(ILoggerObject, String, Object[])

Class Resize

Resizes the image to a specified width and height

Inheritance

System.Object

PipelineBase

Resize

Implements

ILoggerObject

IProgress

ITransformation

IPipelineIO

Inherited Members

PipelineBase.PipelineInputs

PipelineBase.PipelineOutputs

PipelineBase.Logger

PipelineBase.Progress

PipelineBase.ProgressChanged

PipelineBase.OnProgressChanged()

System.Object.Equals(System.Object)

System.Object.Equals(System.Object, System.Object)

System.Object.GetHashCode()

System.Object.GetType()

System.Object.MemberwiseClone()

System.Object.ReferenceEquals(System.Object, System.Object)

System.Object.ToString()

Namespace: SigStat.Common.Transforms

Assembly: SigStat.Common.dll

Syntax

```
[JsonObject(MemberSerialization.OptOut)]
public class Resize : PipelineBase, ILoggerObject, IProgress, ITransformation, IPipelineIO
```

Properties

Height

The new height. Leave it as null, if you do not want to explicitly specify a given height

Declaration

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

Property Value

TYPE	DESCRIPTION
System.Nullable<System.Int32>	

InputImage

Input FeatureDescriptor describing the image of the signature

Declaration

```
[Input(AutoSetMode.IfNull)]
public FeatureDescriptor<Image<Rgba32>> InputImage { get; set; }
```

Property Value

TYPE	DESCRIPTION
FeatureDescriptor<SixLabors.ImageSharp.Image<SixLabors.ImageSharp.PixelFormats.Rgba32>>	

OutputImage

Output FeatureDescriptor describing the resized image of the signature

Declaration

```
[Output("Resized")]
public FeatureDescriptor<Image<Rgba32>> OutputImage { get; set; }
```

Property Value

TYPE	DESCRIPTION
FeatureDescriptor<SixLabors.ImageSharp.Image<SixLabors.ImageSharp.PixelFormats.Rgba32>>	

ResizeFunction

Set a resize function if you want to dynamically calculate the new width and height of the image

Declaration

```
public Func<Image<Rgba32>, Size> ResizeFunction { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Func<SixLabors.ImageSharp.Image<SixLabors.ImageSharp.PixelFormats.Rgba32>, SixLabors.Primitives.Size>	

Width

The new width. Leave it as null, if you do not want to explicitly specify a given width

Declaration

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

Property Value

TYPE	DESCRIPTION
System.Nullable<System.Int32>	

Methods

Transform(Signature)

Declaration

```
public void Transform(Signature signature)
```

Parameters

TYPE	NAME	DESCRIPTION
Signature	signature	

Implements

ILoggerObject
IProgress
ITransformation
IPipelineIO

Extension Methods

ILoggerObjectExtensions.LogError(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogError(ILoggerObject, Exception, String, Object[])
ILoggerObjectExtensions.LogInformation(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogWarning(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogWarning(ILoggerObject, Exception, String, Object[])
ILoggerObjectExtensions.LogTrace(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogTrace<TState>(ILoggerObject, TState, EventId, Exception, Func<TState, Exception, String>)
ILoggerObjectExtensions.LogCritical(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogDebug(ILoggerObject, String, Object[])

Class TangentExtraction

Extracts tangent values of the standard X, Y Features

Default Pipeline Input: X, Y Features

Default Pipeline Output: (List{double}) Tangent

Inheritance

System.Object
PipelineBase
TangentExtraction

Implements

ILoggerObject
IProgress
ITransformation
IPipelineIO

Inherited Members

PipelineBase.PipelineInputs
PipelineBase.PipelineOutputs
PipelineBase.Logger
PipelineBase.Progress
PipelineBase.ProgressChanged
PipelineBase.OnProgressChanged()
System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: SigStat.Common.Transforms

Assembly: SigStat.Common.dll

Syntax

```
[JsonObject(MemberSerialization.OptOut)]
public class TangentExtraction : PipelineBase, ILoggerObject, IProgress, ITransformation, IPipelineIO
```

Properties

OutputTangent

Gets or sets the output feature representing the tangent angles of an online signature

Declaration

```
[Output("Tangent")]
public FeatureDescriptor<List<double>> OutputTangent { get; set; }
```

Property Value

TYPE	DESCRIPTION
FeatureDescriptor<System.Collections.Generic.List<System.Double>>	

X

Gets or sets the input feature representing the X coordinates of an online signature

Declaration

```
[Input(AutoSetMode.IfNull)]
public FeatureDescriptor<List<double>> X { get; set; }
```

Property Value

TYPE	DESCRIPTION
FeatureDescriptor<System.Collections.Generic.List<System.Double>>	

Y

Gets or sets the input feature representing the Y coordinates of an online signature

Declaration

```
[Input(AutoSetMode.IfNull)]
public FeatureDescriptor<List<double>> Y { get; set; }
```

Property Value

TYPE	DESCRIPTION
FeatureDescriptor<System.Collections.Generic.List<System.Double>>	

Methods

Transform(Signature)

Declaration

```
public void Transform(Signature signature)
```

Parameters

TYPE	NAME	DESCRIPTION
Signature	signature	

Implements

ILoggerObject
IProgress
ITransformation
IPipelineIO

Extension Methods

ILoggerObjectExtensions.LogError(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogError(ILoggerObject, Exception, String, Object[])
ILoggerObjectExtensions.LogInformation(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogWarning(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogWarning(ILoggerObject, Exception, String, Object[])
ILoggerObjectExtensions.LogTrace(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogTrace<TState>(ILoggerObject, TState, EventId, Exception, Func<TState, Exception, String>)
ILoggerObjectExtensions.LogCritical(ILoggerObject, String, Object[])

ILoggerObjectExtensions.LogDebug(ILoggerObject, String, Object[])

Class TimeReset

Sequential pipeline to reset time values to begin at 0. The following Transforms are called: Extrema, Multiply, AddVector.

Default Pipeline Input: T

Default Pipeline Output: T

Inheritance

System.Object
PipelineBase
SequentialTransformPipeline
TimeReset

Implements

ILoggerObject
IProgress
System.Collections.IEnumerable
ITransformation
IPipelineIO

Inherited Members

SequentialTransformPipeline.Items
SequentialTransformPipeline.PipelineInputs
SequentialTransformPipeline.PipelineOutputs
SequentialTransformPipeline.GetEnumerator()
SequentialTransformPipeline.Add(ITransformation)
SequentialTransformPipeline.Transform(Signature)
PipelineBase.Logger
PipelineBase.Progress
PipelineBase.ProgressChanged
PipelineBase.OnProgressChanged()
System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: SigStat.Common.Transforms

Assembly: SigStat.Common.dll

Syntax

```
[JsonObject(MemberSerialization.OptOut)]
public class TimeReset : SequentialTransformPipeline, ILoggerObject, IProgress, IEnumerable, ITransformation,
IPipelineIO
```

Constructors

TimeReset()

Initializes a new instance of the TimeReset class.

Declaration

```
public TimeReset()
```

Implements

ILoggerObject

IProgress

System.Collections.IEnumerable

ITransformation

IPipelineIO

Extension Methods

ILoggerObjectExtensions.LogError(ILoggerObject, String, Object[])

ILoggerObjectExtensions.LogError(ILoggerObject, Exception, String, Object[])

ILoggerObjectExtensions.LogInformation(ILoggerObject, String, Object[])

ILoggerObjectExtensions.LogWarning(ILoggerObject, String, Object[])

ILoggerObjectExtensions.LogWarning(ILoggerObject, Exception, String, Object[])

ILoggerObjectExtensions.LogTrace(ILoggerObject, String, Object[])

ILoggerObjectExtensions.LogTrace<TState>(ILoggerObject, TState, EventId, Exception, Func<TState, Exception, String>)

ILoggerObjectExtensions.LogCritical(ILoggerObject, String, Object[])

ILoggerObjectExtensions.LogDebug(ILoggerObject, String, Object[])

Class Translate

Sequential pipeline to translate X and Y Features by specified vector (constant or feature). The following Transforms are called: AddConst twice, or AddVector.

Default Pipeline Input: X, Y

Default Pipeline Output: X, Y

Inheritance

System.Object

PipelineBase

SequentialTransformPipeline

Translate

Implements

ILoggerObject

IProgress

System.Collections.IEnumerable

ITransformation

IPipelineIO

Inherited Members

SequentialTransformPipeline.Items

SequentialTransformPipeline.PipelineInputs

SequentialTransformPipeline.PipelineOutputs

SequentialTransformPipeline.GetEnumerator()

SequentialTransformPipeline.Add(ITransformation)

SequentialTransformPipeline.Transform(Signature)

PipelineBase.Logger

PipelineBase.Progress

PipelineBase.ProgressChanged

PipelineBase.OnProgressChanged()

System.Object.Equals(System.Object)

System.Object.Equals(System.Object, System.Object)

System.Object.GetHashCode()

System.Object.GetType()

System.Object.MemberwiseClone()

System.Object.ReferenceEquals(System.Object, System.Object)

System.Object.ToString()

Namespace: SigStat.Common.Transforms

Assembly: SigStat.Common.dll

Syntax

```
[JsonObject(MemberSerialization.OptOut)]
public class Translate : SequentialTransformPipeline, ILoggerObject, IProgress, IEnumerable, ITransformation,
IPipelineIO
```

Constructors

Translate(FeatureDescriptor<List<Double>>)

Declaration

```
public Translate(FeatureDescriptor<List<double>> vectorFeature)
```

Parameters

TYPE	NAME	DESCRIPTION
FeatureDescriptor<System.Collections.Generic.List<System.Double>>	vectorFeature	Feature to translate X and Y by.

Translate(Double, Double)

Declaration

```
public Translate(double xAdd, double yAdd)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Double	xAdd	Value to translate X by.
System.Double	yAdd	Value to translate Y by.

Properties

InputX

The feature representing the horizontal coordinates of an online signature

Declaration

```
[Input(AutoSetMode.IfNull)]
public FeatureDescriptor<List<double>> InputX { get; set; }
```

Property Value

TYPE	DESCRIPTION
FeatureDescriptor<System.Collections.Generic.List<System.Double>>	

InputY

The feature representing the vertical coordinates of an online signature

Declaration

```
[Input(AutoSetMode.IfNull)]
public FeatureDescriptor<List<double>> InputY { get; set; }
```

Property Value

TYPE	DESCRIPTION
FeatureDescriptor<System.Collections.Generic.List<System.Double>>	

OutputX

Target feature for storing the transformed horizontal coordinates

Declaration

```
[Output("X")]
public FeatureDescriptor<List<double>> OutputX { get; set; }
```

Property Value

TYPE	DESCRIPTION
FeatureDescriptor<System.Collections.Generic.List<System.Double>>	

OutputY

Target feature for storing the transformed vertical coordinates

Declaration

```
[Output("Y")]
public FeatureDescriptor<List<double>> OutputY { get; set; }
```

Property Value

TYPE	DESCRIPTION
FeatureDescriptor<System.Collections.Generic.List<System.Double>>	

Implements

ILoggerObject
IProgress
System.Collections.IEnumerable
ITransformation
IPipelineIO

Extension Methods

ILoggerObjectExtensions.LogError(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogError(ILoggerObject, Exception, String, Object[])
ILoggerObjectExtensions.LogInformation(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogWarning(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogWarning(ILoggerObject, Exception, String, Object[])
ILoggerObjectExtensions.LogTrace(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogTrace<TState>(ILoggerObject, TState, EventId, Exception, Func<TState, Exception, String>)
ILoggerObjectExtensions.LogCritical(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogDebug(ILoggerObject, String, Object[])

Class Trim

Trims unnecessary empty space from a binary raster.

Pipeline Input type: bool[,]

Default Pipeline Output: (bool[,]) Trimmed

Inheritance

System.Object
PipelineBase
Trim

Implements

ILoggerObject
IProgress
ITransformation
IPipelineIO

Inherited Members

PipelineBase.PipelineInputs
PipelineBase.PipelineOutputs
PipelineBase.Logger
PipelineBase.Progress
PipelineBase.ProgressChanged
PipelineBase.OnProgressChanged()
System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: SigStat.Common.Transforms

Assembly: SigStat.Common.dll

Syntax

```
[JsonObject(MemberSerialization.OptOut)]
public class Trim : PipelineBase, ILoggerObject, IProgress, ITransformation, IPipelineIO
```

Constructors

Trim(Int32)

Declaration

```
public Trim(int framewidth)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	framewidth	Leave a border around the trimmed area. framewidth > 0

Properties

Input

Input FeatureDescriptor describing the image of the signature

Declaration

```
[Input(AutoSetMode.IfNull)]
public FeatureDescriptor<bool[, ]> Input { get; set; }
```

Property Value

TYPE	DESCRIPTION
FeatureDescriptor<System.Boolean[,]>	

Output

Output FeatureDescriptor describing the trimed image of the signature

Declaration

```
[Output("Trimmed")]
public FeatureDescriptor<bool[, ]> Output { get; set; }
```

Property Value

TYPE	DESCRIPTION
FeatureDescriptor<System.Boolean[,]>	

Methods

Transform(Signature)

Declaration

```
public void Transform(Signature signature)
```

Parameters

TYPE	NAME	DESCRIPTION
Signature	signature	

Implements

ILoggerObject
IProgress
ITransformation
IPipelineIO

Extension Methods

ILoggerObjectExtensions.LogError(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogError(ILoggerObject, Exception, String, Object[])
ILoggerObjectExtensions.LogInformation(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogWarning(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogWarning(ILoggerObject, Exception, String, Object[])
ILoggerObjectExtensions.LogTrace(ILoggerObject, String, Object[])
ILoggerObjectExtensions.LogTrace<TState>(ILoggerObject, TState, EventId, Exception, Func<TState, Exception, String>)
ILoggerObjectExtensions.LogCritical(ILoggerObject, String, Object[])

ILoggerObjectExtensions.LogDebug(ILoggerObject, String, Object[])

Namespace SigStat.FusionBenchmark

Classes

FusionPipelines

Class FusionPipelines

Inheritance

System.Object
FusionPipelines

Namespace: SigStat.FusionBenchmark

Assembly: SigStat.FusionBenchmark.dll

Syntax

```
public class FusionPipelines : object
```

Fields

DOSConst

Declaration

```
public static double DOSConst
```

Field Value

TYPE	DESCRIPTION
System.Double	

DtwpairingJump

Declaration

```
public static int DtwpairingJump
```

Field Value

TYPE	DESCRIPTION
System.Int32	

DtwPairingScaling

Declaration

```
public static double DtwPairingScaling
```

Field Value

TYPE	DESCRIPTION
System.Double	

MyClassifierRange

Declaration

```
public static Tuple<double, double> MyClassifierRange
```

Field Value

TYPE	DESCRIPTION
Tuple<System.Double, System.Double>	

MyFeatures

Declaration

```
public static List<FeatureDescriptor> MyFeatures
```

Field Value

TYPE	DESCRIPTION
List<FeatureDescriptor>	

MyFunc

Declaration

```
public static Func<double[], double[], double> MyFunc
```

Field Value

TYPE	DESCRIPTION
Func<System.Double[], System.Double[], System.Double>	

MyPairingRange

Declaration

```
public static Tuple<double, double> MyPairingRange
```

Field Value

TYPE	DESCRIPTION
Tuple<System.Double, System.Double>	

NumOfRef

Declaration

```
public static int NumOfRef
```

Field Value

TYPE	DESCRIPTION
System.Int32	

OffToOnScaling

Declaration

```
public static double OffToOnScaling
```

Field Value

TYPE	DESCRIPTION
System.Double	

OffToOnVMax

Declaration

```
public static double OffToOnVMax
```

Field Value

TYPE	DESCRIPTION
System.Double	

Methods

GetBenchmark(List<Signer>, Boolean)

Declaration

```
public static VerifierBenchmark GetBenchmark(List<Signer> signers, bool isOptimal)
```

Parameters

TYPE	NAME	DESCRIPTION
List<Signer>	signers	
System.Boolean	isOptimal	

Returns

TYPE	DESCRIPTION
VerifierBenchmark	

GetBenchmarkWithOnlySigner(Signer, Boolean)

Declaration

```
public static VerifierBenchmark GetBenchmarkWithOnlySigner(Signer signer, bool isOptimal)
```

Parameters

TYPE	NAME	DESCRIPTION
Signer	signer	
System.Boolean	isOptimal	

Returns

TYPE	DESCRIPTION
VerifierBenchmark	

GetBiosecureIDOfflineLoader(String)

Declaration

```
public static BiosecureIDOfflineLoader GetBiosecureIDOfflineLoader(string path = "")
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	path	

Returns

TYPE	DESCRIPTION
BiosecureIDOfflineLoader	

GetBiosecureIDOnlineLoader(String)

Declaration

```
public static BiosecureIDOnlineLoader GetBiosecureIDOnlineLoader(string path = "")
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	path	

Returns

TYPE	DESCRIPTION
BiosecureIDOnlineLoader	

GetDistanceMatrixViewer(List<Signature>, List<Signature>)

Declaration

```
public static DistanceMatrixViewer GetDistanceMatrixViewer(List<Signature> onlineSignatures, List<Signature>
offlineSignatures)
```

Parameters

TYPE	NAME	DESCRIPTION
List<Signature>	onlineSignatures	
List<Signature>	offlineSignatures	

Returns

TYPE	DESCRIPTION
DistanceMatrixViewer	

GetFusionPipeline(List<Signer>, Boolean, Int32)

Declaration

```
public static SequentialTransformPipeline GetFusionPipeline(List<Signer> onlineSigners, bool isParallel, int
baseSigInputCntID)
```

Parameters

TYPE	NAME	DESCRIPTION
List<Signer>	onlineSigners	
System.Boolean	isParallel	
System.Int32	baseSigInputCntID	

Returns

TYPE	DESCRIPTION
SequentialTransformPipeline	

GetHackedOfflinePipeline()

Declaration

```
public static SequentialTransformPipeline GetHackedOfflinePipeline()
```

Returns

TYPE	DESCRIPTION
SequentialTransformPipeline	

GetHackedOnToOnPipeline(RectangleF)

Declaration

```
public static SequentialTransformPipeline GetHackedOnToOnPipeline(RectangleF goalBounds)
```

Parameters

TYPE	NAME	DESCRIPTION
RectangleF	goalBounds	

Returns

TYPE	DESCRIPTION
SequentialTransformPipeline	

GetMarosPipeline()

Declaration

```
public static SequentialTransformPipeline GetMarosPipeline()
```

Returns

TYPE	DESCRIPTION
SequentialTransformPipeline	

GetMixedSigner(Signer, Signer)

Declaration

```
public static Signer GetMixedSigner(Signer refSigner, Signer testSigner)
```

Parameters

TYPE	NAME	DESCRIPTION
Signer	refSigner	
Signer	testSigner	

Returns

TYPE	DESCRIPTION
Signer	

GetOfflinePipeline()

Declaration

```
public static SequentialTransformPipeline GetOfflinePipeline()
```

Returns

TYPE	DESCRIPTION
SequentialTransformPipeline	

GetOffToOnTransform()

Declaration

```
public static SequentialTransformPipeline GetOffToOnTransform()
```

Returns

TYPE	DESCRIPTION
SequentialTransformPipeline	

GetOnlinePipeline()

Declaration

```
public static SequentialTransformPipeline GetOnlinePipeline()
```

Returns

TYPE	DESCRIPTION
SequentialTransformPipeline	

GetOnlineToOfflinePipeline(RectangleF)

Declaration

```
public static SequentialTransformPipeline GetOnlineToOfflinePipeline(RectangleF goalBounds)
```

Parameters

TYPE	NAME	DESCRIPTION
RectangleF	goalBounds	

Returns

TYPE	DESCRIPTION
SequentialTransformPipeline	

GetOptimalBenchmark(List<Signer>)

Declaration

```
public static VerifierBenchmark GetOptimalBenchmark(List<Signer> signers)
```

Parameters

TYPE	NAME	DESCRIPTION
List<Signer>	signers	

Returns

TYPE	DESCRIPTION
VerifierBenchmark	

GetPureBenchmark(List<Signer>)

Declaration

```
public static VerifierBenchmark GetPureBenchmark(List<Signer> signers)
```

Parameters

TYPE	NAME	DESCRIPTION
List<Signer>	signers	

Returns

TYPE	DESCRIPTION
VerifierBenchmark	

GetSVCOfflineLoader(String)

Declaration

```
public static Svc2004OfflineLoader GetSVCOfflineLoader(string path = "")
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	path	

Returns

TYPE	DESCRIPTION
Svc2004OfflineLoader	

GetSVCOnlineLoader(String)

Declaration

```
public static Svc2004OnlineLoader GetSVCOnlineLoader(string path = "")
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	path	

Returns

TYPE	DESCRIPTION
Svc2004OnlineLoader	

GetXYSaver()

Declaration

```
public static XYSaver GetXYSaver()
```

Returns

TYPE	DESCRIPTION
XYSaver	

Namespace SigStat.FusionBenchmark.FusionDemos

Classes

DistanceViewing

FusionBenchmarkResults

OnlineOnlineBenchmark

StrokePairingExam

Strokepairingmatrix

Class DistanceViewing

Inheritance

System.Object

DistanceViewing

Namespace: SigStat.FusionBenchmark.FusionDemos

Assembly: SigStat.FusionBenchmark.dll

Syntax

```
public static class DistanceViewing : object
```

Methods

Calculate(String[], DataSetLoader, DataSetLoader)

Declaration

```
public static void Calculate(string[] ids, DataSetLoader offlineLoader, DataSetLoader onlineLoader)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String[]	ids	
DataSetLoader	offlineLoader	
DataSetLoader	onlineLoader	

Class FusionBenchmarkResults

Inheritance

System.Object
FusionBenchmarkResults

Namespace: SigStat.FusionBenchmark.FusionDemos

Assembly: SigStat.FusionBenchmark.dll

Syntax

```
public class FusionBenchmarkResults : object
```

Properties

OffOffResults

Declaration

```
public BenchmarkResults OffOffResults { get; set; }
```

Property Value

TYPE	DESCRIPTION
BenchmarkResults	

OffOnResults

Declaration

```
public BenchmarkResults OffOnResults { get; set; }
```

Property Value

TYPE	DESCRIPTION
BenchmarkResults	

OnOffResults

Declaration

```
public BenchmarkResults OnOffResults { get; set; }
```

Property Value

TYPE	DESCRIPTION
BenchmarkResults	

OnOnResults

Declaration

```
public BenchmarkResults OnOnResults { get; set; }
```

Property Value

TYPE	DESCRIPTION
BenchmarkResults	

Class OnlineOnlineBenchmark

Inheritance

System.Object
OnlineOnlineBenchmark

Namespace: SigStat.FusionBenchmark.FusionDemos

Assembly: SigStat.FusionBenchmark.dll

Syntax

```
public static class OnlineOnlineBenchmark : object
```

Methods

BenchMarkWithAllSigners(Boolean, DataSetLoader)

Declaration

```
public static BenchmarkResults BenchMarkWithAllSigners(bool isoptimal, DataSetLoader onlineLoader)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Boolean	isoptimal	
DataSetLoader	onlineLoader	

Returns

TYPE	DESCRIPTION
BenchmarkResults	

Class StrokePairingExam

Inheritance

System.Object
StrokePairingExam

Namespace: SigStat.FusionBenchmark.FusionDemos

Assembly: SigStat.FusionBenchmark.dll

Syntax

```
public static class StrokePairingExam : object
```

Methods

CalculateForID(String[], DataSetLoader, DataSetLoader)

Declaration

```
public static void CalculateForID(string[] ids, DataSetLoader offlineLoader, DataSetLoader onlineLoader)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String[]	ids	
DataSetLoader	offlineLoader	
DataSetLoader	onlineLoader	

Class Strokepairingmatrix

Inheritance

System.Object

Strokepairingmatrix

Namespace: SigStat.FusionBenchmark.FusionDemos

Assembly: SigStat.FusionBenchmark.dll

Syntax

```
public static class Strokepairingmatrix : object
```

Methods

Calculate(String[], DataSetLoader, DataSetLoader)

Declaration

```
public static void Calculate(string[] ids, DataSetLoader offlineLoader, DataSetLoader onlineLoader)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String[]	ids	
DataSetLoader	offlineLoader	
DataSetLoader	onlineLoader	

Namespace SigStat.FusionBenchmark.FusionDemos.FinalPipelines

Classes

FinalFusionPipelines

OnlySignerBenchmark

Class FinalFusionPipelines

Inheritance

System.Object
FinalFusionPipelines

Namespace: SigStat.FusionBenchmark.FusionDemos.FinalPipelines

Assembly: SigStat.FusionBenchmark.dll

Syntax

```
public static class FinalFusionPipelines : object
```

Fields

DtwPairingScaling

Declaration

```
public static double DtwPairingScaling
```

Field Value

TYPE	DESCRIPTION
System.Double	

MyClassifierRange

Declaration

```
public static Tuple<double, double> MyClassifierRange
```

Field Value

TYPE	DESCRIPTION
Tuple<System.Double, System.Double>	

MyFeatures

Declaration

```
public static List<FeatureDescriptor> MyFeatures
```

Field Value

TYPE	DESCRIPTION
List<FeatureDescriptor>	

MyFunc

Declaration

```
public static Func<double[], double[], double> MyFunc
```

Field Value

TYPE	DESCRIPTION
Func<System.Double[], System.Double[], System.Double>	

MyPairingRange

Declaration

```
public static Tuple<double, double> MyPairingRange
```

Field Value

TYPE	DESCRIPTION
Tuple<System.Double, System.Double>	

MySampler

Declaration

```
public static Sampler MySampler
```

Field Value

TYPE	DESCRIPTION
Sampler	

NumOfRef

Declaration

```
public static int NumOfRef
```

Field Value

TYPE	DESCRIPTION
System.Int32	

OffToOnScaling

Declaration

```
public static double OffToOnScaling
```

Field Value

TYPE	DESCRIPTION
System.Double	

Methods

EmptyPipeline()

Declaration

```
public static SequentialTransformPipeline EmptyPipeline()
```

Returns

TYPE	DESCRIPTION
SequentialTransformPipeline	

GetFusionPipeline(List<Signer>, Boolean, Int32)

Declaration

```
public static SequentialTransformPipeline GetFusionPipeline(List<Signer> onlineSigners, bool isParallel, int
baseSigInputCntID)
```

Parameters

TYPE	NAME	DESCRIPTION
List<Signer>	onlineSigners	
System.Boolean	isParallel	
System.Int32	baseSigInputCntID	

Returns

TYPE	DESCRIPTION
SequentialTransformPipeline	

GetHackedOnToOnPipeline(RectangleF)

Declaration

```
public static SequentialTransformPipeline GetHackedOnToOnPipeline(RectangleF goalBounds)
```

Parameters

TYPE	NAME	DESCRIPTION
RectangleF	goalBounds	

Returns

TYPE	DESCRIPTION
SequentialTransformPipeline	

GetOfflinePipelineAlap()

Declaration

```
public static SequentialTransformPipeline GetOfflinePipelineAlap()
```

Returns

TYPE	DESCRIPTION
SequentialTransformPipeline	

GetOfflinePipelineMaros()

Declaration

```
public static SequentialTransformPipeline GetOfflinePipelineMaros()
```

Returns

TYPE	DESCRIPTION
SequentialTransformPipeline	

GetOfflinePipelineMerging()

Declaration

```
public static SequentialTransformPipeline GetOfflinePipelineMerging()
```

Returns

TYPE	DESCRIPTION
SequentialTransformPipeline	

GetOfflineSavers()

Declaration

```
public static SequentialTransformPipeline GetOfflineSavers()
```

Returns

TYPE	DESCRIPTION
SequentialTransformPipeline	

GetOnlinePipeline1()

Declaration

```
public static SequentialTransformPipeline GetOnlinePipeline1()
```

Returns

TYPE	DESCRIPTION
SequentialTransformPipeline	

GetOnlinePipeline2()

Declaration

```
public static SequentialTransformPipeline GetOnlinePipeline2()
```

Returns

TYPE	DESCRIPTION
SequentialTransformPipeline	

Class OnlySignerBenchmark

Inheritance

System.Object
OnlySignerBenchmark

Namespace: SigStat.FusionBenchmark.FusionDemos.FinalPipelines

Assembly: SigStat.FusionBenchmark.dll

Syntax

```
public static class OnlySignerBenchmark : object
```

Methods

BenchmarkWithSigner(Boolean, Signer)

Declaration

```
public static BenchmarkResults BenchmarkWithSigner(bool isOptimal, Signer signer)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Boolean	isOptimal	
Signer	signer	

Returns

TYPE	DESCRIPTION
BenchmarkResults	

Namespace SigStat.FusionBenchmark.FusionDemos.PipelineBenchmarks

Classes

FusionVerifierBenchmark

MarosBenchmark

MarosDtwPairing

Class FusionVerifierBenchmark

Inheritance

System.Object

FusionVerifierBenchmark

Namespace: SigStat.FusionBenchmark.FusionDemos.PipelineBenchmarks

Assembly: SigStat.FusionBenchmark.dll

Syntax

```
public static class FusionVerifierBenchmark : object
```

Methods

BenchMarkingWithAllSigners(Boolean, DataSetLoader, DataSetLoader)

Declaration

```
public static FusionBenchmarkResults BenchMarkingWithAllSigners(bool isoptimal, DataSetLoader offlineLoader,
DataSetLoader onlineLoader)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Boolean	isoptimal	
DataSetLoader	offlineLoader	
DataSetLoader	onlineLoader	

Returns

TYPE	DESCRIPTION
FusionBenchmarkResults	

Class MarosBenchmark

Inheritance

System.Object

MarosBenchmark

Namespace: SigStat.FusionBenchmark.FusionDemos.PipelineBenchmarks

Assembly: SigStat.FusionBenchmark.dll

Syntax

```
public static class MarosBenchmark : object
```

Methods

BenchmarkingWithAllSigners(Boolean, DataSetLoader)

Declaration

```
public static BenchmarkResults BenchmarkingWithAllSigners(bool isoptimal, DataSetLoader offlineLoader)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Boolean	isoptimal	
DataSetLoader	offlineLoader	

Returns

TYPE	DESCRIPTION
BenchmarkResults	

Class MarosDtwPairing

Inheritance

System.Object

MarosDtwPairing

Namespace: SigStat.FusionBenchmark.FusionDemos.PipelineBenchmarks

Assembly: SigStat.FusionBenchmark.dll

Syntax

```
public static class MarosDtwPairing : object
```

Methods

BenchmarkingWithAllSigners(Boolean, DataSetLoader, DataSetLoader)

Declaration

```
public static BenchmarkResults BenchmarkingWithAllSigners(bool isoptimal, DataSetLoader offlineLoader,
DataSetLoader onlineLoader)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Boolean	isoptimal	
DataSetLoader	offlineLoader	
DataSetLoader	onlineLoader	

Returns

TYPE	DESCRIPTION
BenchmarkResults	

Namespace SigStat.FusionBenchmark.FusionDemos.ReSamplingBenchmarks

Classes

ReSamplingExtractions

Class ReSamplingExtractions

Inheritance

System.Object
ReSamplingExtractions

Namespace: SigStat.FusionBenchmark.FusionDemos.ReSamplingBenchmarks

Assembly: SigStat.FusionBenchmark.dll

Syntax

```
public static class ReSamplingExtractions : object
```

Methods

Calculate(DataSetLoader)

Declaration

```
public static Tuple<List<double>, List<List<double>>> Calculate(DataSetLoader loader)
```

Parameters

TYPE	NAME	DESCRIPTION
DataSetLoader	loader	

Returns

TYPE	DESCRIPTION
Tuple<List<System.Double>, List<List<System.Double>>>	

Namespace SigStat.FusionBenchmark.FusionFeatureExtraction

Classes

OnlineToOfflineFeature

Class OnlineToOfflineFeature

Inheritance

System.Object
PipelineBase
OnlineToOfflineFeature

Implements

ILoggerObject
IProgress
ITransformation
IPipelineIO

Inherited Members

PipelineBase.PipelineInputs
PipelineBase.PipelineOutputs
PipelineBase.Logger
PipelineBase.Progress
PipelineBase.ProgressChanged
PipelineBase.OnProgressChanged()

Namespace: SigStat.FusionBenchmark.FusionFeatureExtraction

Assembly: SigStat.FusionBenchmark.dll

Syntax

```
public class OnlineToOfflineFeature : PipelineBase, ILoggerObject, IProgress, ITransformation, IPipelineIO
```

Properties

InputButton

Declaration

```
public FeatureDescriptor<List<bool>> InputButton { get; set; }
```

Property Value

TYPE	DESCRIPTION
FeatureDescriptor<List<System.Boolean>>	

InputGoalBounds

Declaration

```
public RectangleF InputGoalBounds { get; set; }
```

Property Value

TYPE	DESCRIPTION
RectangleF	

InputScaleRate

Declaration

```
public double InputScaleRate { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Double	

InputT

Declaration

```
public FeatureDescriptor<List<double>> InputT { get; set; }
```

Property Value

TYPE	DESCRIPTION
FeatureDescriptor<List<System.Double>>	

InputX

Declaration

```
public FeatureDescriptor<List<double>> InputX { get; set; }
```

Property Value

TYPE	DESCRIPTION
FeatureDescriptor<List<System.Double>>	

InputY

Declaration

```
public FeatureDescriptor<List<double>> InputY { get; set; }
```

Property Value

TYPE	DESCRIPTION
FeatureDescriptor<List<System.Double>>	

OutputBaseTrajectory

Declaration

```
public FeatureDescriptor<List<Vertex>> OutputBaseTrajectory { get; set; }
```

Property Value

TYPE	DESCRIPTION
FeatureDescriptor<List<Vertex>>	

OutputVertices

Declaration

```
public FeatureDescriptor<List<Vertex>> OutputVertices { get; set; }
```

Property Value

TYPE	DESCRIPTION
FeatureDescriptor<List<Vertex>>	

Methods

MakeVertexLine(Vertex, Vertex)

Declaration

```
public static List<Vertex> MakeVertexLine(Vertex fromVertex, Vertex toVertex)
```

Parameters

TYPE	NAME	DESCRIPTION
Vertex	fromVertex	
Vertex	toVertex	

Returns

TYPE	DESCRIPTION
List<Vertex>	

Transform(Signature)

Declaration

```
public void Transform(Signature signature)
```

Parameters

TYPE	NAME	DESCRIPTION
Signature	signature	

Implements

ILoggerObject
IProgress
ITransformation
IPipelineIO

Namespace SigStat.FusionBenchmark.FusionMathHelper

Classes

Analizises

Geometry

PointFHelper

Translations

VectorFHelper

Structs

PointFSection

PointSection

StraightLineF

VectorF

Class Analizises

Inheritance

System.Object
Analizises

Namespace: SigStat.FusionBenchmark.FusionMathHelper

Assembly: SigStat.FusionBenchmark.dll

Syntax

```
public static class Analizises : object
```

Methods

Differentiate(List<Double>, Int32)

Declaration

```
public static List<double> Differentiate(this List<double> values, int diffIdx = 1)
```

Parameters

TYPE	NAME	DESCRIPTION
List<System.Double>	values	
System.Int32	diffIdx	

Returns

TYPE	DESCRIPTION
List<System.Double>	

Class Geometry

Inheritance

System.Object

Geometry

Namespace: SigStat.FusionBenchmark.FusionMathHelper

Assembly: SigStat.FusionBenchmark.dll

Syntax

```
public static class Geometry : object
```

Fields

Origo

Declaration

```
public static readonly Point Origo
```

Field Value

TYPE	DESCRIPTION
Point	

OrigoF

Declaration

```
public static readonly PointF OrigoF
```

Field Value

TYPE	DESCRIPTION
PointF	

Methods

DiffAngle(Double, Double)

Declaration

```
public static double DiffAngle(double angle1, double angle2)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Double	angle1	
System.Double	angle2	

Returns

TYPE	DESCRIPTION
System.Double	

DiffVectorAngle(Double[], Double[])

Declaration

```
public static double DiffVectorAngle(double[] aVec1, double[] aVec2)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Double[]	aVec1	
System.Double[]	aVec2	

Returns

TYPE	DESCRIPTION
System.Double	

DirectedDiffAngle(Double, Double)

Declaration

```
public static double DirectedDiffAngle(double angle1, double angle2)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Double	angle1	
System.Double	angle2	

Returns

TYPE	DESCRIPTION
System.Double	

Direction(PointSection)

Declaration

```
public static double Direction(this PointSection section)
```

Parameters

TYPE	NAME	DESCRIPTION
PointSection	section	

Returns

TYPE	DESCRIPTION
System.Double	

EndDirection(Stroke, Int32)

Declaration

```
public static double EndDirection(this Stroke stroke, int diffIdx)
```

Parameters

TYPE	NAME	DESCRIPTION
Stroke	stroke	
System.Int32	diffIdx	

Returns

TYPE	DESCRIPTION
System.Double	

Euclidean(Point, Point)

Declaration

```
public static double Euclidean(Point lhs, Point rhs)
```

Parameters

TYPE	NAME	DESCRIPTION
Point	lhs	
Point	rhs	

Returns

TYPE	DESCRIPTION
System.Double	

Euclidean(PointF, PointF)

Declaration

```
public static double Euclidean(PointF lhs, PointF rhs)
```

Parameters

TYPE	NAME	DESCRIPTION
PointF	lhs	
PointF	rhs	

Returns

TYPE	DESCRIPTION
System.Double	

Lefter(Point, Point)

Declaration

```
public static int Lefter(Point lhs, Point rhs)
```

Parameters

TYPE	NAME	DESCRIPTION
Point	lhs	
Point	rhs	

Returns

TYPE	DESCRIPTION
System.Int32	

StartDirection(Stroke, Int32)

Declaration

```
public static double StartDirection(this Stroke stroke, int diffIdx)
```

Parameters

TYPE	NAME	DESCRIPTION
Stroke	stroke	
System.Int32	diffIdx	

Returns

TYPE	DESCRIPTION
System.Double	

Class PointFHelper

Inheritance

System.Object

PointFHelper

Namespace: SigStat.FusionBenchmark.FusionMathHelper

Assembly: SigStat.FusionBenchmark.dll

Syntax

```
public static class PointFHelper : object
```

Methods

DistanceFrom(PointF, StraightLineF)

Declaration

```
public static double DistanceFrom(this PointF pointF, StraightLineF lineF)
```

Parameters

TYPE	NAME	DESCRIPTION
PointF	pointF	
StraightLineF	lineF	

Returns

TYPE	DESCRIPTION
System.Double	

ToPointF(Point)

Declaration

```
public static PointF ToPointF(this Point point)
```

Parameters

TYPE	NAME	DESCRIPTION
Point	point	

Returns

TYPE	DESCRIPTION
PointF	

ToPointFList(List<Point>)

Declaration

```
public static List<PointF> ToPointFList(this List<Point> list)
```

Parameters

TYPE	NAME	DESCRIPTION
List<Point>	list	

Returns

TYPE	DESCRIPTION
List<PointF>	

Struct PointFSection

Namespace: SigStat.FusionBenchmark.FusionMathHelper

Assembly: SigStat.FusionBenchmark.dll

Syntax

```
public struct PointFSection
```

Constructors

PointFSection(PointF, PointF)

Declaration

```
public PointFSection(PointF start, PointF end)
```

Parameters

TYPE	NAME	DESCRIPTION
PointF	start	
PointF	end	

Properties

Direction

Declaration

```
public double Direction { get; }
```

Property Value

TYPE	DESCRIPTION
System.Double	

End

Declaration

```
public PointF End { get; set; }
```

Property Value

TYPE	DESCRIPTION
PointF	

Length

Declaration

```
public double Length { get; }
```

Property Value

TYPE	DESCRIPTION
System.Double	

Start

Declaration

```
public PointF Start { get; set; }
```

Property Value

TYPE	DESCRIPTION
PointF	

Struct PointSection

Namespace: SigStat.FusionBenchmark.FusionMathHelper

Assembly: SigStat.FusionBenchmark.dll

Syntax

```
public struct PointSection
```

Constructors

PointSection(Point, Point)

Declaration

```
public PointSection(Point start, Point end)
```

Parameters

TYPE	NAME	DESCRIPTION
Point	start	
Point	end	

Properties

End

Declaration

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

Property Value

TYPE	DESCRIPTION
Point	

Length

Declaration

```
public double Length { get; }
```

Property Value

TYPE	DESCRIPTION
System.Double	

Start

Declaration

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

Property Value

TYPE	DESCRIPTION
Point	

Extension Methods

Geometry.Direction(PointSection)

Struct StraightLineF

Namespace: SigStat.FusionBenchmark.FusionMathHelper

Assembly: SigStat.FusionBenchmark.dll

Syntax

```
public struct StraightLineF
```

Constructors

StraightLineF(PointF, PointF)

Declaration

```
public StraightLineF(PointF from, PointF to)
```

Parameters

TYPE	NAME	DESCRIPTION
PointF	from	
PointF	to	

StraightLineF(PointF, VectorF)

Declaration

```
public StraightLineF(PointF pointF, VectorF normalVectorF)
```

Parameters

TYPE	NAME	DESCRIPTION
PointF	pointF	
VectorF	normalVectorF	

Properties

A

Declaration

```
public float A { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Single	

B

Declaration

```
public float B { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Single	

C

Declaration

```
public float C { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Single	

Direction

Declaration

```
public double Direction { get; }
```

Property Value

TYPE	DESCRIPTION
System.Double	

DirVector

Declaration

```
public VectorF DirVector { get; }
```

Property Value

TYPE	DESCRIPTION
VectorF	

M

Declaration

```
public float M { get; }
```

Property Value

TYPE	DESCRIPTION
System.Single	

NormalVector

Declaration

```
public VectorF NormalVector { get; }
```

Property Value

TYPE	DESCRIPTION
VectorF	

Class Translations

Inheritance

System.Object
Translations

Namespace: SigStat.FusionBenchmark.FusionMathHelper

Assembly: SigStat.FusionBenchmark.dll

Syntax

```
public static class Translations : object
```

Methods

GetDirections(List<Double>, List<Double>)

Declaration

```
public static List<double> GetDirections(List<double> xs, List<double> ys)
```

Parameters

TYPE	NAME	DESCRIPTION
List<System.Double>	xs	
List<System.Double>	ys	

Returns

TYPE	DESCRIPTION
List<System.Double>	

MergeLists(List<Double>[])

Declaration

```
public static List<double[]> MergeLists(List<double>[] lists)
```

Parameters

TYPE	NAME	DESCRIPTION
List<System.Double>[]	lists	

Returns

TYPE	DESCRIPTION
List<System.Double[]>	

Normalize(List<Double>)

Declaration

```
public static List<double> Normalize(List<double> values)
```

Parameters

TYPE	NAME	DESCRIPTION
List<System.Double>	values	

Returns

TYPE	DESCRIPTION
List<System.Double>	

Normalize(List<Double>, Double, Double)

Declaration

```
public static List<double> Normalize(List<double> values, double left, double right)
```

Parameters

TYPE	NAME	DESCRIPTION
List<System.Double>	values	
System.Double	left	
System.Double	right	

Returns

TYPE	DESCRIPTION
List<System.Double>	

Struct VectorF

Namespace: SigStat.FusionBenchmark.FusionMathHelper

Assembly: SigStat.FusionBenchmark.dll

Syntax

```
public struct VectorF
```

Constructors

VectorF(PointF, PointF)

Declaration

```
public VectorF(PointF from, PointF to)
```

Parameters

TYPE	NAME	DESCRIPTION
PointF	from	
PointF	to	

VectorF(Double, Double)

Declaration

```
public VectorF(double x, double y)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Double	x	
System.Double	y	

VectorF(Int32, Int32)

Declaration

```
public VectorF(int x, int y)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	x	
System.Int32	y	

VectorF(Single, Single)

Declaration

```
public VectorF(float x, float y)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Single	x	
System.Single	y	

Properties

X

Declaration

```
public float X { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Single	

Y

Declaration

```
public float Y { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Single	

Extension Methods

VectorFHelper.Length(VectorF)
VectorFHelper.UnitVector(VectorF)
VectorFHelper.Rotate(VectorF, Double)
VectorFHelper.Multiply(VectorF, Single)
VectorFHelper.Multiply(VectorF, Double)

Class VectorFHelper

Inheritance

System.Object

VectorFHelper

Namespace: SigStat.FusionBenchmark.FusionMathHelper

Assembly: SigStat.FusionBenchmark.dll

Syntax

```
public static class VectorFHelper : object
```

Methods

Length(VectorF)

Declaration

```
public static double Length(this VectorF v)
```

Parameters

TYPE	NAME	DESCRIPTION
VectorF	v	

Returns

TYPE	DESCRIPTION
System.Double	

Multiply(VectorF, Double)

Declaration

```
public static VectorF Multiply(this VectorF v, double lambda)
```

Parameters

TYPE	NAME	DESCRIPTION
VectorF	v	
System.Double	lambda	

Returns

TYPE	DESCRIPTION
VectorF	

Multiply(VectorF, Single)

Declaration

```
public static VectorF Multiply(this VectorF v, float lambda)
```

Parameters

TYPE	NAME	DESCRIPTION
VectorF	v	
System.Single	lambda	

Returns

TYPE	DESCRIPTION
VectorF	

Rotate(VectorF, Double)

Declaration

```
public static VectorF Rotate(this VectorF v, double phi)
```

Parameters

TYPE	NAME	DESCRIPTION
VectorF	v	
System.Double	phi	

Returns

TYPE	DESCRIPTION
VectorF	

UnitVector(VectorF)

Declaration

```
public static VectorF UnitVector(this VectorF v)
```

Parameters

TYPE	NAME	DESCRIPTION
VectorF	v	

Returns

TYPE	DESCRIPTION
VectorF	

Namespace SigStat.FusionBenchmark.GraphExtraction

Classes

ConnectionNode

ConnectionNodesHelper

ListHelper

SkeletonHelper

Stroke

StrokeComponent

StrokeComponentHelper

StrokeHelper

StrokeMerging

Vertex

VerticesHelper

Class ConnectionNode

Inheritance

System.Object

ConnectionNode

Namespace: SigStat.FusionBenchmark.GraphExtraction

Assembly: SigStat.FusionBenchmark.dll

Syntax

```
public class ConnectionNode : HashSet<Vertex>
```

Methods

Degree(List<Stroke>)

Declaration

```
public int Degree(List<Stroke> strokes)
```

Parameters

TYPE	NAME	DESCRIPTION
List<Stroke>	strokes	

Returns

TYPE	DESCRIPTION
System.Int32	

InStrokes(List<Stroke>)

Declaration

```
public List<Stroke> InStrokes(List<Stroke> strokes)
```

Parameters

TYPE	NAME	DESCRIPTION
List<Stroke>	strokes	

Returns

TYPE	DESCRIPTION
List<Stroke>	

OutStrokes(List<Stroke>)

Declaration

```
public List<Stroke> OutStrokes(List<Stroke> strokes)
```

Parameters

TYPE	NAME	DESCRIPTION
List<Stroke>	strokes	

Returns

TYPE	DESCRIPTION
List<Stroke>	

Class ConnectionNodesHelper

Inheritance

System.Object
ConnectionNodesHelper

Namespace: SigStat.FusionBenchmark.GraphExtraction

Assembly: SigStat.FusionBenchmark.dll

Syntax

```
public static class ConnectionNodesHelper : object
```

Methods

Add(List<ConnectionNode>, List<Vertex>)

Declaration

```
public static void Add(this List<ConnectionNode> connectionNodes, List<Vertex> vertices)
```

Parameters

TYPE	NAME	DESCRIPTION
List<ConnectionNode>	connectionNodes	
List<Vertex>	vertices	

Add(List<ConnectionNode>, StrokeComponent)

Declaration

```
public static void Add(this List<ConnectionNode> connectionNodes, StrokeComponent comp)
```

Parameters

TYPE	NAME	DESCRIPTION
List<ConnectionNode>	connectionNodes	
StrokeComponent	comp	

Add(List<ConnectionNode>, Vertex)

Declaration

```
public static void Add(this List<ConnectionNode> connectionNodes, Vertex vertex)
```

Parameters

TYPE	NAME	DESCRIPTION
List<ConnectionNode>	connectionNodes	
Vertex	vertex	

FindConnectionNode(List<ConnectionNode>, Vertex)

Declaration

```
public static ConnectionNode FindConnectionNode(this List<ConnectionNode> connectionNodes, Vertex vertex)
```

Parameters

TYPE	NAME	DESCRIPTION
List<ConnectionNode>	connectionNodes	
Vertex	vertex	

Returns

TYPE	DESCRIPTION
ConnectionNode	

Class ListHelper

Inheritance

System.Object
ListHelper

Namespace: SigStat.FusionBenchmark.GraphExtraction

Assembly: SigStat.FusionBenchmark.dll

Syntax

```
public static class ListHelper : object
```

Methods

IsIdxValid<T>(List<T>, Int32)

Declaration

```
public static bool IsIdxValid<T>(this List<T> list, int idx)
```

Parameters

TYPE	NAME	DESCRIPTION
List<T>	list	
System.Int32	idx	

Returns

TYPE	DESCRIPTION
System.Boolean	

Type Parameters

NAME	DESCRIPTION
T	

Class SkeletonHelper

Inheritance

System.Object
SkeletonHelper

Namespace: SigStat.FusionBenchmark.GraphExtraction

Assembly: SigStat.FusionBenchmark.dll

Syntax

```
public static class SkeletonHelper : object
```

Methods

GetRutovitz(Boolean[,], Point)

Declaration

```
public static int GetRutovitz(this bool[, ] skeleton, Point p)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Boolean[,]	skeleton	
Point	p	

Returns

TYPE	DESCRIPTION
System.Int32	

GetRutovitz(Boolean[,], Int32, Int32)

Declaration

```
public static int GetRutovitz(this bool[, ] skeleton, int x, int y)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Boolean[,]	skeleton	
System.Int32	x	
System.Int32	y	

Returns

TYPE	DESCRIPTION
System.Int32	

GetRutovitzNeighbourHood(Boolean[,], Int32, Int32)

Declaration

```
public static bool[] GetRutovitzNeighbourHood(this bool[, ] skeleton, int x, int y)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Boolean[,]	skeleton	
System.Int32	x	
System.Int32	y	

Returns

TYPE	DESCRIPTION
System.Boolean[]	

Valid(Boolean[,], Point)

Declaration

```
public static bool Valid(this bool[, ] skeleton, Point p)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Boolean[,]	skeleton	
Point	p	

Returns

TYPE	DESCRIPTION
System.Boolean	

Valid(Boolean[,], Int32, Int32)

Declaration

```
public static bool Valid(this bool[, ] skeleton, int x, int y)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Boolean[,]	skeleton	
System.Int32	x	
System.Int32	y	

Returns

TYPE	DESCRIPTION
System.Boolean	

Class Stroke

Inheritance

System.Object
Stroke

Namespace: SigStat.FusionBenchmark.GraphExtraction

Assembly: SigStat.FusionBenchmark.dll

Syntax

```
public class Stroke : List<Vertex>
```

Constructors

Stroke()

Declaration

```
public Stroke()
```

Stroke(List<Vertex>)

Declaration

```
public Stroke(List<Vertex> list)
```

Parameters

TYPE	NAME	DESCRIPTION
List<Vertex>	list	

Properties

Component

Declaration

```
public StrokeComponent Component { get; set; }
```

Property Value

TYPE	DESCRIPTION
StrokeComponent	

End

Declaration

```
public Vertex End { get; }
```

Property Value

TYPE	DESCRIPTION
Vertex	

Sibling

Declaration

```
public Stroke Sibling { get; }
```

Property Value

TYPE	DESCRIPTION
Stroke	

Start

Declaration

```
public Vertex Start { get; }
```

Property Value

TYPE	DESCRIPTION
Vertex	

Methods

CreateSibling(Stroke)

Declaration

```
public static Stroke CreateSibling(Stroke stroke)
```

Parameters

TYPE	NAME	DESCRIPTION
Stroke	stroke	

Returns

TYPE	DESCRIPTION
Stroke	

Equals(Object)

Declaration

```
public override bool Equals(object obj)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Object	obj	

Returns

TYPE	DESCRIPTION
System.Boolean	

Extension Methods

Geometry.StartDirection(Stroke, Int32)

Geometry.EndDirection(Stroke, Int32)

Class StrokeComponent

Inheritance

System.Object

StrokeComponent

Namespace: SigStat.FusionBenchmark.GraphExtraction

Assembly: SigStat.FusionBenchmark.dll

Syntax

```
public class StrokeComponent : object
```

Constructors

StrokeComponent(Stroke)

Declaration

```
public StrokeComponent(Stroke stroke)
```

Parameters

TYPE	NAME	DESCRIPTION
Stroke	stroke	

StrokeComponent(Stroke, Stroke)

Declaration

```
public StrokeComponent(Stroke stroke1, Stroke stroke2)
```

Parameters

TYPE	NAME	DESCRIPTION
Stroke	stroke1	
Stroke	stroke2	

Properties

Strokes

Declaration

```
public List<Stroke> Strokes { get; }
```

Property Value

TYPE	DESCRIPTION
List<Stroke>	

Methods

GetWithEnd(Vertex)

Declaration

```
public Stroke GetWithEnd(Vertex end)
```

Parameters

TYPE	NAME	DESCRIPTION
Vertex	end	

Returns

TYPE	DESCRIPTION
Stroke	

GetWithStart(Vertex)

Declaration

```
public Stroke GetWithStart(Vertex start)
```

Parameters

TYPE	NAME	DESCRIPTION
Vertex	start	

Returns

TYPE	DESCRIPTION
Stroke	

Class StrokeComponentHelper

Inheritance

System.Object

StrokeComponentHelper

Namespace: SigStat.FusionBenchmark.GraphExtraction

Assembly: SigStat.FusionBenchmark.dll

Syntax

```
public static class StrokeComponentHelper : object
```

Methods

GetAllStrokes(List<StrokeComponent>)

Declaration

```
public static List<Stroke> GetAllStrokes(this List<StrokeComponent> components)
```

Parameters

TYPE	NAME	DESCRIPTION
List<StrokeComponent>	components	

Returns

TYPE	DESCRIPTION
List<Stroke>	

Class StrokeHelper

Inheritance

System.Object

StrokeHelper

Namespace: SigStat.FusionBenchmark.GraphExtraction

Assembly: SigStat.FusionBenchmark.dll

Syntax

```
public static class StrokeHelper : object
```

Class StrokeMerging

Inheritance

System.Object
PipelineBase
StrokeMerging

Implements

ILoggerObject
IProgress
ITransformation
IPipelineIO

Inherited Members

PipelineBase.PipelineInputs
PipelineBase.PipelineOutputs
PipelineBase.Logger
PipelineBase.Progress
PipelineBase.ProgressChanged
PipelineBase.OnProgressChanged()

Namespace: SigStat.FusionBenchmark.GraphExtraction

Assembly: SigStat.FusionBenchmark.dll

Syntax

```
public class StrokeMerging : PipelineBase, ILoggerObject, IProgress, ITransformation, IPipelineIO
```

Properties

InputComponent

Declaration

```
public FeatureDescriptor<List<StrokeComponent>> InputComponent { get; set; }
```

Property Value

TYPE	DESCRIPTION
FeatureDescriptor<List<StrokeComponent>>	

InputConnectionNodes

Declaration

```
public FeatureDescriptor<List<ConnectionNode>> InputConnectionNodes { get; set; }
```

Property Value

TYPE	DESCRIPTION
FeatureDescriptor<List<ConnectionNode>>	

InputWidthOfPen

Declaration

```
public FeatureDescriptor<double> InputWidthOfPen { get; set; }
```

Property Value

TYPE	DESCRIPTION
FeatureDescriptor<System.Double>	

OutputComponents

Declaration

```
public FeatureDescriptor<List<StrokeComponent>> OutputComponents { get; set; }
```

Property Value

TYPE	DESCRIPTION
FeatureDescriptor<List<StrokeComponent>>	

Methods

Transform(Signature)

Declaration

```
public void Transform(Signature signature)
```

Parameters

TYPE	NAME	DESCRIPTION
Signature	signature	

Implements

ILoggerObject

IProgress

ITransformation

IPipelineIO

Class Vertex

Inheritance

System.Object
Vertex

Namespace: SigStat.FusionBenchmark.GraphExtraction

Assembly: SigStat.FusionBenchmark.dll

Syntax

```
public class Vertex : object
```

Constructors

Vertex(Point, Boolean)

Declaration

```
public Vertex(Point pos, bool on = true)
```

Parameters

TYPE	NAME	DESCRIPTION
Point	pos	
System.Boolean	on	

Properties

Degree

Declaration

```
public int Degree { get; }
```

Property Value

TYPE	DESCRIPTION
System.Int32	

Neighbours

Declaration

```
public List<Vertex> Neighbours { get; set; }
```

Property Value

TYPE	DESCRIPTION
List<Vertex>	

On

Declaration

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

Property Value

TYPE	DESCRIPTION
System.Boolean	

PointType

Declaration

```
public VertexType PointType { get; }
```

Property Value

TYPE	DESCRIPTION
VertexType	

Pos

Declaration

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

Property Value

TYPE	DESCRIPTION
Point	

PosF

Declaration

```
public PointF PosF { get; set; }
```

Property Value

TYPE	DESCRIPTION
PointF	

Rutovitz

Declaration

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

Property Value

TYPE	DESCRIPTION
System.Int32	

Methods

AreNeighbours(Object, Object)

Declaration

```
public static bool AreNeighbours(object objL, object objR)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Object	objL	
System.Object	objR	

Returns

TYPE	DESCRIPTION
System.Boolean	

Equals(Object)

Declaration

```
public override bool Equals(object obj)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Object	obj	

Returns

TYPE	DESCRIPTION
System.Boolean	

Class VerticesHelper

Inheritance

System.Object
VerticesHelper

Namespace: SigStat.FusionBenchmark.GraphExtraction

Assembly: SigStat.FusionBenchmark.dll

Syntax

```
public static class VerticesHelper : object
```

Methods

ConnectionPoints(List<Vertex>)

Declaration

```
public static List<Vertex> ConnectionPoints(this List<Vertex> vertices)
```

Parameters

TYPE	NAME	DESCRIPTION
List<Vertex>	vertices	

Returns

TYPE	DESCRIPTION
List<Vertex>	

CrossingPoints(List<Vertex>)

Declaration

```
public static List<Vertex> CrossingPoints(this List<Vertex> vertices)
```

Parameters

TYPE	NAME	DESCRIPTION
List<Vertex>	vertices	

Returns

TYPE	DESCRIPTION
List<Vertex>	

EndPoints(List<Vertex>)

Declaration

```
public static List<Vertex> EndPoints(this List<Vertex> vertices)
```

Parameters

TYPE	NAME	DESCRIPTION
List<Vertex>	vertices	

Returns

TYPE	DESCRIPTION
List<Vertex>	

GetDirections(List<Vertex>)

Declaration

```
public static List<double> GetDirections(this List<Vertex> list)
```

Parameters

TYPE	NAME	DESCRIPTION
List<Vertex>	list	

Returns

TYPE	DESCRIPTION
List<System.Double>	

GetDtwPairingFeature(List<Vertex>, Int32)

Declaration

```
public static List<double[]> GetDtwPairingFeature(this List<Vertex> list, int inputScale = 1)
```

Parameters

TYPE	NAME	DESCRIPTION
List<Vertex>	list	
System.Int32	inputScale	

Returns

TYPE	DESCRIPTION
List<System.Double[]>	

GetOriginalXYs(List<Vertex>)

Declaration

```
public static List<double[]> GetOriginalXYs(this List<Vertex> list)
```

Parameters

TYPE	NAME	DESCRIPTION
List<Vertex>	list	

Returns

TYPE	DESCRIPTION
List<System.Double[]>	

GetXs(List<Vertex>)

Declaration

```
public static List<double> GetXs(this List<Vertex> list)
```

Parameters

TYPE	NAME	DESCRIPTION
List<Vertex>	list	

Returns

TYPE	DESCRIPTION
List<System.Double>	

GetYs(List<Vertex>)

Declaration

```
public static List<double> GetYs(this List<Vertex> list)
```

Parameters

TYPE	NAME	DESCRIPTION
List<Vertex>	list	

Returns

TYPE	DESCRIPTION
List<System.Double>	

StrokeEnds(List<Vertex>)

Declaration

```
public static List<Vertex> StrokeEnds(this List<Vertex> vertices)
```

Parameters

TYPE	NAME	DESCRIPTION
List<Vertex>	vertices	

Returns

TYPE	DESCRIPTION
List<Vertex>	

Namespace SigStat.FusionBenchmark.LineTransforms

Classes

DOSBasedAlgorithm

EqualResampling

LineFittingAlgorithm

PseudoVelocityAlgorithm

Class DOSBasedAlgorithm

Inheritance

System.Object
DOSBasedAlgorithm

Namespace: SigStat.FusionBenchmark.LineTransforms

Assembly: SigStat.FusionBenchmark.dll

Syntax

```
public static class DOSBasedAlgorithm : object
```

Methods

Calculate(List<PointF>, Int32, Double)

Declaration

```
public static List<double> Calculate(List<PointF> pointFs, int diffIdx, double length)
```

Parameters

TYPE	NAME	DESCRIPTION
List<PointF>	pointFs	
System.Int32	diffIdx	
System.Double	length	

Returns

TYPE	DESCRIPTION
List<System.Double>	

MakeSection(List<PointF>, Int32, Int32, Double)

Declaration

```
public static PointFSection MakeSection(List<PointF> pointFs, int idx, int idxPlus, double length)
```

Parameters

TYPE	NAME	DESCRIPTION
List<PointF>	pointFs	
System.Int32	idx	
System.Int32	idxPlus	
System.Double	length	

Returns

TYPE	DESCRIPTION
PointFSection	

TYPE	DESCRIPTION

Class EqualResampling

Inheritance

System.Object
EqualResampling

Namespace: SigStat.FusionBenchmark.LineTransforms

Assembly: SigStat.FusionBenchmark.dll

Syntax

```
public static class EqualResampling : object
```

Methods

Calculate(List<PointF>, Int32)

Declaration

```
public static List<PointF> Calculate(List<PointF> list, int equalCnt)
```

Parameters

TYPE	NAME	DESCRIPTION
List<PointF>	list	
System.Int32	equalCnt	

Returns

TYPE	DESCRIPTION
List<PointF>	

Class LineFittingAlgorithm

Inheritance

System.Object
LineFittingAlgorithm

Namespace: SigStat.FusionBenchmark.LineTransforms

Assembly: SigStat.FusionBenchmark.dll

Syntax

```
public static class LineFittingAlgorithm : object
```

Methods

Calculate(List<PointF>)

Declaration

```
public static List<PointF> Calculate(List<PointF> list)
```

Parameters

TYPE	NAME	DESCRIPTION
List<PointF>	list	

Returns

TYPE	DESCRIPTION
List<PointF>	

Class PseudoVelocityAlgorithm

Inheritance

System.Object

PseudoVelocityAlgorithm

Namespace: SigStat.FusionBenchmark.LineTransforms

Assembly: SigStat.FusionBenchmark.dll

Syntax

```
public static class PseudoVelocityAlgorithm : object
```

Methods

Calculate(List<PointF>, List<Double>, Double, Double)

Declaration

```
public static List<PointF> Calculate(List<PointF> pointFs, List<double> dosList, double vMax, double deltaT =
1)
```

Parameters

TYPE	NAME	DESCRIPTION
List<PointF>	pointFs	
List<System.Double>	dosList	
System.Double	vMax	
System.Double	deltaT	

Returns

TYPE	DESCRIPTION
List<PointF>	

Namespace SigStat.FusionBenchmark.Loaders

Classes

BiosecureID

BiosecureIDOfflineLoader

BiosecureIDOnlineLoader

MemoryLoader

Svc2004

Set of features containing raw data loaded from SVC2004-format database. This part is different from SigStat.Common.Svc2004Loader

Svc2004OfflineLoader

Svc2004OnlineLoader

Class BiosecureID

Inheritance

System.Object

BiosecureID

Namespace: SigStat.FusionBenchmark.Loaders

Assembly: SigStat.FusionBenchmark.dll

Syntax

```
public static class BiosecureID : object
```

Fields

Altitude

Altitude values from the online signature imported from the SVC2004 database

Declaration

```
public static readonly FeatureDescriptor<List<double>> Altitude
```

Field Value

TYPE	DESCRIPTION
FeatureDescriptor<List<System.Double>>	

Azimuth

Azimuth values from the online signature imported from the SVC2004 database

Declaration

```
public static readonly FeatureDescriptor<List<double>> Azimuth
```

Field Value

TYPE	DESCRIPTION
FeatureDescriptor<List<System.Double>>	

Button

Button values from the online signature imported from the SVC2004 database

Declaration

```
public static readonly FeatureDescriptor<List<bool>> Button
```

Field Value

TYPE	DESCRIPTION
FeatureDescriptor<List<System.Boolean>>	

Pressure

Pressure values from the online signature imported from the SVC2004 database

Declaration

```
public static readonly FeatureDescriptor<List<double>> Pressure
```

Field Value

TYPE	DESCRIPTION
FeatureDescriptor<List<System.Double>>	

T

T values from the online signature imported from the SVC2004 database

Declaration

```
public static readonly FeatureDescriptor<List<double>> T
```

Field Value

TYPE	DESCRIPTION
FeatureDescriptor<List<System.Double>>	

X

Declaration

```
public static readonly FeatureDescriptor<List<double>> X
```

Field Value

TYPE	DESCRIPTION
FeatureDescriptor<List<System.Double>>	

Y

Y cooridnates from the online signature imported from the SVC2004 database

Declaration

```
public static readonly FeatureDescriptor<List<double>> Y
```

Field Value

TYPE	DESCRIPTION
FeatureDescriptor<List<System.Double>>	

Class BiosecureIDOfflineLoader

Inheritance

System.Object
DataSetLoader
BiosecureIDOfflineLoader

Implements

IDataSetLoader
ILoggerObject

Inherited Members

DataSetLoader.Logger
DataSetLoader.EnumerateSigners()
DataSetLoader.EnumerateSigners(Predicate<Signer>)
DataSetLoader.IDataSetLoader.SignerFilter

Namespace: SigStat.FusionBenchmark.Loaders

Assembly: SigStat.FusionBenchmark.dll

Syntax

```
public class BiosecureIDOfflineLoader : DataSetLoader, IDataSetLoader, ILoggerObject
```

Constructors

BiosecureIDOfflineLoader(String, Predicate<Signer>)

Declaration

```
public BiosecureIDOfflineLoader(string databasePath, Predicate<Signer> signerFilter = null)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	databasePath	
Predicate<Signer>	signerFilter	

Properties

DatabasePath

Declaration

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

Property Value

TYPE	DESCRIPTION
System.String	

SamplingFrequency

Declaration

```
public override int SamplingFrequency { get; }
```

Property Value

TYPE	DESCRIPTION
System.Int32	

Overrides

DataSetLoader.SamplingFrequency

SignerFilter

Declaration

```
public Predicate<Signer> SignerFilter { get; set; }
```

Property Value

TYPE	DESCRIPTION
Predicate<Signer>	

Methods

EnumerateSigners(Predicate<Signer>)

Declaration

```
public override IEnumerable<Signer> EnumerateSigners(Predicate<Signer> signerFilter)
```

Parameters

TYPE	NAME	DESCRIPTION
Predicate<Signer>	signerFilter	

Returns

TYPE	DESCRIPTION
IEnumerable<Signer>	

LoadOfflineSignature(Signature, String)

Declaration

```
public void LoadOfflineSignature(Signature signature, string file)
```

Parameters

TYPE	NAME	DESCRIPTION
Signature	signature	
System.String	file	

Implements

IDataSetLoader

ILoggerObject

Class BiosecureIDOnlineLoader

Inheritance

System.Object
DataSetLoader
BiosecureIDOnlineLoader

Implements

IDataSetLoader
ILoggerObject

Inherited Members

DataSetLoader.Logger
DataSetLoader.EnumerateSigners()
DataSetLoader.EnumerateSigners(Predicate<Signer>)
DataSetLoader.IDataSetLoader.SignerFilter

Namespace: SigStat.FusionBenchmark.Loaders

Assembly: SigStat.FusionBenchmark.dll

Syntax

```
public class BiosecureIDOnlineLoader : DataSetLoader, IDataSetLoader, ILoggerObject
```

Constructors

BiosecureIDOnlineLoader(String, Predicate<Signer>)

Declaration

```
public BiosecureIDOnlineLoader(string databasePath, Predicate<Signer> signerFilter = null)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	databasePath	
Predicate<Signer>	signerFilter	

Properties

DatabasePath

Declaration

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

Property Value

TYPE	DESCRIPTION
System.String	

SamplingFrequency

Declaration

```
public override int SamplingFrequency { get; }
```

Property Value

TYPE	DESCRIPTION
System.Int32	

Overrides

DataSetLoader.SamplingFrequency

SignerFilter

Declaration

```
public Predicate<Signer> SignerFilter { get; set; }
```

Property Value

TYPE	DESCRIPTION
Predicate<Signer>	

Methods

EnumerateSigners(Predicate<Signer>)

Declaration

```
public override IEnumerable<Signer> EnumerateSigners(Predicate<Signer> signerFilter)
```

Parameters

TYPE	NAME	DESCRIPTION
Predicate<Signer>	signerFilter	

Returns

TYPE	DESCRIPTION
IEnumerable<Signer>	

LoadOnlineSignature(Signature, String)

Declaration

```
public void LoadOnlineSignature(Signature signature, string file)
```

Parameters

TYPE	NAME	DESCRIPTION
Signature	signature	
System.String	file	

Implements

IDataSetLoader

ILoggerObject

Class MemoryLoader

Inheritance

System.Object
DataSetLoader
MemoryLoader

Implements

IDataSetLoader
ILoggerObject

Inherited Members

DataSetLoader.Logger
DataSetLoader.EnumerateSigners()
DataSetLoader.EnumerateSigners(Predicate<Signer>)
DataSetLoader.IDataSetLoader.SignerFilter

Namespace: SigStat.FusionBenchmark.Loaders

Assembly: SigStat.FusionBenchmark.dll

Syntax

```
public class MemoryLoader : DataSetLoader, IDataSetLoader, ILoggerObject
```

Properties

SamplingFrequency

Declaration

```
public override int SamplingFrequency { get; }
```

Property Value

TYPE	DESCRIPTION
System.Int32	

Overrides

DataSetLoader.SamplingFrequency

Signers

Declaration

```
public List<Signer> Signers { get; set; }
```

Property Value

TYPE	DESCRIPTION
List<Signer>	

Methods

EnumerateSigners(Predicate<Signer>)

Declaration

```
public override IEnumerable<Signer> EnumerateSigners(Predicate<Signer> signerFilter)
```

Parameters

TYPE	NAME	DESCRIPTION
Predicate<Signer>	signerFilter	

Returns

TYPE	DESCRIPTION
IEnumerable<Signer>	

Implements

IDataSetLoader

ILoggerObject

Class Svc2004

Set of features containing raw data loaded from SVC2004-format database. This part is different from SigStat.Common.Svc2004Loader

Inheritance

System.Object
Svc2004

Namespace: SigStat.FusionBenchmark.Loaders

Assembly: SigStat.FusionBenchmark.dll

Syntax

```
public static class Svc2004 : object
```

Fields

Altitude

Altitude values from the online signature imported from the SVC2004 database

Declaration

```
public static readonly FeatureDescriptor<List<double>> Altitude
```

Field Value

TYPE	DESCRIPTION
FeatureDescriptor<List<System.Double>>	

Azimuth

Azimuth values from the online signature imported from the SVC2004 database

Declaration

```
public static readonly FeatureDescriptor<List<double>> Azimuth
```

Field Value

TYPE	DESCRIPTION
FeatureDescriptor<List<System.Double>>	

Button

Button values from the online signature imported from the SVC2004 database

Declaration

```
public static readonly FeatureDescriptor<List<bool>> Button
```

Field Value

TYPE	DESCRIPTION
FeatureDescriptor<List<System.Boolean>>	

Pressure

Pressure values from the online signature imported from the SVC2004 database

Declaration

```
public static readonly FeatureDescriptor<List<double>> Pressure
```

Field Value

TYPE	DESCRIPTION
FeatureDescriptor<List<System.Double>>	

T

T values from the online signature imported from the SVC2004 database

Declaration

```
public static readonly FeatureDescriptor<List<double>> T
```

Field Value

TYPE	DESCRIPTION
FeatureDescriptor<List<System.Double>>	

X

Declaration

```
public static readonly FeatureDescriptor<List<double>> X
```

Field Value

TYPE	DESCRIPTION
FeatureDescriptor<List<System.Double>>	

Y

Y cooridnates from the online signature imported from the SVC2004 database

Declaration

```
public static readonly FeatureDescriptor<List<double>> Y
```

Field Value

TYPE	DESCRIPTION
FeatureDescriptor<List<System.Double>>	

Class Svc2004OfflineLoader

Inheritance

System.Object
DataSetLoader
Svc2004OfflineLoader

Implements

IDataSetLoader
ILoggerObject

Inherited Members

DataSetLoader.Logger
DataSetLoader.EnumerateSigners()
DataSetLoader.EnumerateSigners(Predicate<Signer>)
DataSetLoader.IDataSetLoader.SignerFilter

Namespace: SigStat.FusionBenchmark.Loaders

Assembly: SigStat.FusionBenchmark.dll

Syntax

```
public class Svc2004OfflineLoader : DataSetLoader, IDataSetLoader, ILoggerObject
```

Constructors

Svc2004OfflineLoader(String, Predicate<Signer>)

Declaration

```
public Svc2004OfflineLoader(string databasePath, Predicate<Signer> signerFilter = null)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	databasePath	
Predicate<Signer>	signerFilter	

Properties

DatabasePath

Declaration

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

Property Value

TYPE	DESCRIPTION
System.String	

SamplingFrequency

Declaration

```
public override int SamplingFrequency { get; }
```

Property Value

TYPE	DESCRIPTION
System.Int32	

Overrides

DataSetLoader.SamplingFrequency

SignerFilter

Declaration

```
public Predicate<Signer> SignerFilter { get; set; }
```

Property Value

TYPE	DESCRIPTION
Predicate<Signer>	

Methods

EnumerateSigners(Predicate<Signer>)

Declaration

```
public override IEnumerable<Signer> EnumerateSigners(Predicate<Signer> signerFilter)
```

Parameters

TYPE	NAME	DESCRIPTION
Predicate<Signer>	signerFilter	

Returns

TYPE	DESCRIPTION
IEnumerable<Signer>	

LoadOfflineSignature(Signature, String)

Declaration

```
public void LoadOfflineSignature(Signature signature, string file)
```

Parameters

TYPE	NAME	DESCRIPTION
Signature	signature	
System.String	file	

Implements

IDataSetLoader

ILoggerObject

Class Svc2004OnlineLoader

Inheritance

System.Object
DataSetLoader
Svc2004OnlineLoader

Implements

IDataSetLoader
ILoggerObject

Inherited Members

DataSetLoader.Logger
DataSetLoader.EnumerateSigners()
DataSetLoader.EnumerateSigners(Predicate<Signer>)
DataSetLoader.IDataSetLoader.SignerFilter

Namespace: SigStat.FusionBenchmark.Loaders

Assembly: SigStat.FusionBenchmark.dll

Syntax

```
public class Svc2004OnlineLoader : DataSetLoader, IDataSetLoader, ILoggerObject
```

Constructors

Svc2004OnlineLoader(String, Boolean)

Initializes a new instance of the Svc2004Loader class with specified database.

Declaration

```
public Svc2004OnlineLoader(string databasePath, bool standardFeatures)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	databasePath	Represents the path, to load the signatures from. It supports two basic approaches: • •
System.Boolean	standardFeatures	Convert loaded data (Svc2004) to standard Features.

Svc2004OnlineLoader(String, Boolean, Predicate<Signer>)

Initializes a new instance of the Svc2004Loader class with specified database.

Declaration

```
public Svc2004OnlineLoader(string databasePath, bool standardFeatures, Predicate<Signer> signerFilter = null)
```

Parameters

TYPE	NAME	DESCRIPTION

TYPE	NAME	DESCRIPTION
System.String	databasePath	Represents the path, to load the signatures from. It supports two basic approaches: • •
System.Boolean	standardFeatures	Convert loaded data (Svc2004) to standard Features.
Predicate<Signer>	signerFilter	Sets the SignerFilter property

Properties

DatabasePath

Gets or sets the database path.

Declaration

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

Property Value

TYPE	DESCRIPTION
System.String	

SamplingFrequency

Declaration

```
public override int SamplingFrequency { get; }
```

Property Value

TYPE	DESCRIPTION
System.Int32	

Overrides

DataSetLoader.SamplingFrequency

SignerFilter

Ignores any signers during the loading, that do not match the predicate

Declaration

```
public Predicate<Signer> SignerFilter { get; set; }
```

Property Value

TYPE	DESCRIPTION
Predicate<Signer>	

StandardFeatures

Gets or sets a value indicating whether features are also loaded as Features

Declaration

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

Property Value

TYPE	DESCRIPTION
System.Boolean	

Methods

EnumerateSigners(Predicate<Signer>)

Declaration

```
public override IEnumerable<Signer> EnumerateSigners(Predicate<Signer> signerFilter)
```

Parameters

TYPE	NAME	DESCRIPTION
Predicate<Signer>	signerFilter	

Returns

TYPE	DESCRIPTION
IEnumerable<Signer>	

LoadSignature(Signature, Stream, Boolean)

Loads one signature from specified stream.

Declaration

```
public static void LoadSignature(Signature signature, Stream stream, bool standardFeatures)
```

Parameters

TYPE	NAME	DESCRIPTION
Signature	signature	Signature to write features to.
Stream	stream	Stream to read svc2004 data from.
System.Boolean	standardFeatures	Convert loaded data to standard Features.

LoadSignature(Signature, String, Boolean)

Loads one signature from specified file path.

Declaration

```
public void LoadSignature(Signature signature, string path, bool standardFeatures)
```

Parameters

TYPE	NAME	DESCRIPTION
Signature	signature	Signature to write features to.
System.String	path	Path to a file of format "US.txt"
System.Boolean	standardFeatures	Convert loaded data to standard Features.

Implements

IDataSetLoader

ILoggerObject

Namespace SigStat.FusionBenchmark.OfflineVerifier

Classes

OfflineVerifier

Class OfflineVerifier

Inheritance

System.Object
OfflineVerifier

Implements

ILoggerObject

Namespace: SigStat.FusionBenchmark.OfflineVerifier

Assembly: SigStat.FusionBenchmark.dll

Syntax

```
public class OfflineVerifier : object, ILoggerObject
```

Constructors

OfflineVerifier()

Declaration

```
public OfflineVerifier()
```

OfflineVerifier(ILogger)

Declaration

```
public OfflineVerifier(ILogger logger = null)
```

Parameters

TYPE	NAME	DESCRIPTION
ILogger	logger	

OfflineVerifier(OfflineVerifier)

Declaration

```
public OfflineVerifier(OfflineVerifier v)
```

Parameters

TYPE	NAME	DESCRIPTION
OfflineVerifier	v	

Properties

AllFeatures

Declaration

```
public Dictionary<string, FeatureDescriptor> AllFeatures { get; }
```

Property Value

TYPE	DESCRIPTION
Dictionary<System.String, FeatureDescriptor>	

Classifier

Declaration

```
public IClassifier Classifier { get; set; }
```

Property Value

TYPE	DESCRIPTION
IClassifier	

FusionPipeline

Declaration

```
public SequentialTransformPipeline FusionPipeline { get; set; }
```

Property Value

TYPE	DESCRIPTION
SequentialTransformPipeline	

Logger

Declaration

```
public ILogger Logger { get; set; }
```

Property Value

TYPE	DESCRIPTION
ILogger	

OfflinePipeline

Declaration

```
public SequentialTransformPipeline OfflinePipeline { get; set; }
```

Property Value

TYPE	DESCRIPTION
SequentialTransformPipeline	

SignerModel

Declaration

```
public ISignerModel SignerModel { get; set; }
```

Property Value

TYPE	DESCRIPTION
ISignerModel	

Methods

Test(Signature)

Declaration

```
public virtual double Test(Signature signature)
```

Parameters

TYPE	NAME	DESCRIPTION
Signature	signature	

Returns

TYPE	DESCRIPTION
System.Double	

Train(List<Signature>)

Declaration

```
public virtual void Train(List<Signature> signatures)
```

Parameters

TYPE	NAME	DESCRIPTION
List<Signature>	signatures	

Implements

ILoggerObject

Namespace SigStat.FusionBenchmark.ReSamplingFeatures

Classes

ReSamplingFeatureExtraction

Class ReSamplingFeatureExtraction

Inheritance

System.Object
PipelineBase
ReSamplingFeatureExtraction

Implements

ILoggerObject
IProgress
IPipelineIO

Inherited Members

PipelineBase.PipelineInputs
PipelineBase.PipelineOutputs
PipelineBase.Logger
PipelineBase.Progress
PipelineBase.ProgressChanged
PipelineBase.OnProgressChanged()

Namespace: SigStat.FusionBenchmark.ReSamplingFeatures

Assembly: SigStat.FusionBenchmark.dll

Syntax

```
public class ReSamplingFeatureExtraction : PipelineBase, ILoggerObject, IProgress, IPipelineIO
```

Properties

InputButton

Declaration

```
public FeatureDescriptor<List<bool>> InputButton { get; set; }
```

Property Value

TYPE	DESCRIPTION
FeatureDescriptor<List<System.Boolean>>	

InputX

Declaration

```
public FeatureDescriptor<List<double>> InputX { get; set; }
```

Property Value

TYPE	DESCRIPTION
FeatureDescriptor<List<System.Double>>	

InputY

Declaration

```
public FeatureDescriptor<List<double>> InputY { get; set; }
```

Property Value

TYPE	DESCRIPTION
FeatureDescriptor<List<System.Double>>	

Methods

Calculate(List<Signer>)

Declaration

```
public Tuple<List<double>, List<List<double>>> Calculate(List<Signer> signers)
```

Parameters

TYPE	NAME	DESCRIPTION
List<Signer>	signers	

Returns

TYPE	DESCRIPTION
Tuple<List<System.Double>, List<List<System.Double>>>	

Calculate(Signature)

Declaration

```
public Tuple<List<double>, List<List<double>>> Calculate(Signature signature)
```

Parameters

TYPE	NAME	DESCRIPTION
Signature	signature	

Returns

TYPE	DESCRIPTION
Tuple<List<System.Double>, List<List<System.Double>>>	

Calculate(Signer)

Declaration

```
public Tuple<List<double>, List<List<double>>> Calculate(Signer signer)
```

Parameters

TYPE	NAME	DESCRIPTION
Signer	signer	

Returns

TYPE	DESCRIPTION
Tuple<List<System.Double>, List<List<System.Double>>>	

Implements

ILoggerObject

IProgress

IPipelineIO

Namespace SigStat.FusionBenchmark.ReSamplingFeatures.FeatureExtractAlgorithms

Classes

JustOnAlgorithm

Interfaces

ICalculate

Interface ICalculate

Namespace: SigStat.FusionBenchmark.ReSamplingFeatures.FeatureExtractAlgorithms

Assembly: SigStat.FusionBenchmark.dll

Syntax

```
public interface ICalculate
```

Methods

Calculate()

Declaration

```
List<double> Calculate()
```

Returns

TYPE	DESCRIPTION
List<System.Double>	

Class JustOnAlgorithm

Inheritance

System.Object
JustOnAlgorithm

Namespace: SigStat.FusionBenchmark.ReSamplingFeatures.FeatureExtractAlgorithms

Assembly: SigStat.FusionBenchmark.dll

Syntax

```
public static class JustOnAlgorithm : object
```

Methods

Calculate(List<List<Double>>, List<Boolean>)

Declaration

```
public static List<List<double>> Calculate(List<List<double>> lists, List<bool> bs)
```

Parameters

TYPE	NAME	DESCRIPTION
List<List<System.Double>>	lists	
List<System.Boolean>	bs	

Returns

TYPE	DESCRIPTION
List<List<System.Double>>	

Calculate(List<Double>, List<Boolean>)

Declaration

```
public static List<double> Calculate(List<double> list, List<bool> bs)
```

Parameters

TYPE	NAME	DESCRIPTION
List<System.Double>	list	
List<System.Boolean>	bs	

Returns

TYPE	DESCRIPTION
List<System.Double>	

Namespace SigStat.FusionBenchmark.ReSamplingFeatures.ReSamplingFuncs

Classes

Kivono

Class Kivono

Inheritance

System.Object

Kivono

Namespace: SigStat.FusionBenchmark.ReSamplingFeatures.ReSamplingFuncs

Assembly: SigStat.FusionBenchmark.dll

Syntax

```
public static class Kivono : object
```

Methods

Calculate(Double, Double)

Declaration

```
public static double Calculate(double p, double q)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Double	p	
System.Double	q	

Returns

TYPE	DESCRIPTION
System.Double	

Namespace SigStat.FusionBenchmark.TrajectoryRecovery

Classes

AlterDtwPairing

Class AlterDtwPairing

Inheritance

System.Object
PipelineBase
AlterDtwPairing

Implements

ILoggerObject
IProgress
IPipelineIO

Inherited Members

PipelineBase.PipelineInputs
PipelineBase.PipelineOutputs
PipelineBase.Logger
PipelineBase.Progress
PipelineBase.ProgressChanged
PipelineBase.OnProgressChanged()

Namespace: SigStat.FusionBenchmark.TrajectoryRecovery

Assembly: SigStat.FusionBenchmark.dll

Syntax

```
public class AlterDtwPairing : PipelineBase, ILoggerObject, IProgress, IPipelineIO
```

Properties

BaseSignature

Declaration

```
public Signature BaseSignature { get; set; }
```

Property Value

TYPE	DESCRIPTION
Signature	

InputBaseTrajectory

Declaration

```
public FeatureDescriptor<List<Vertex>> InputBaseTrajectory { get; set; }
```

Property Value

TYPE	DESCRIPTION
FeatureDescriptor<List<Vertex>>	

InputComponents

Declaration

```
public FeatureDescriptor<List<StrokeComponent>> InputComponents { get; set; }
```

Property Value

TYPE	DESCRIPTION
FeatureDescriptor<List<StrokeComponent>>	

InputIsParallel

Declaration

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

Property Value

TYPE	DESCRIPTION
System.Boolean	

InputJump

Declaration

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

Property Value

TYPE	DESCRIPTION
System.Int32	

InputScale

Declaration

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

Property Value

TYPE	DESCRIPTION
System.Int32	

InputWindowFrom

Declaration

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

Property Value

TYPE	DESCRIPTION
System.Int32	

InputWindowJump

Declaration

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

Property Value

TYPE	DESCRIPTION
System.Int32	

InputWindowTo

Declaration

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

Property Value

TYPE	DESCRIPTION
System.Int32	

Methods

Calculate(Signature)

Declaration

```
public double Calculate(Signature signature)
```

Parameters

TYPE	NAME	DESCRIPTION
Signature	signature	

Returns

TYPE	DESCRIPTION
System.Double	

DtwPairingDistance(Double[], Double[])

Declaration

```
public static double DtwPairingDistance(double[] vec1, double[] vec2)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Double[]	vec1	
System.Double[]	vec2	

Returns

TYPE	DESCRIPTION
System.Double	

Implements

ILoggerObject
IProgress

IPipelineIO

Namespace SigStat.FusionBenchmark.VisualHelpers

Classes

BenchmarkRsults

DistanceMatrixViewer

ImageHelper

StrokePairingDistances

StrokePairSaver

TxtHelper

XYSaver

Class BenchmarkRsults

Inheritance

System.Object

BenchmarkRsults

Namespace: SigStat.FusionBenchmark.VisualHelpers

Assembly: SigStat.FusionBenchmark.dll

Syntax

```
public class BenchmarkRsults : object
```

Class DistanceMatrixViewer

Inheritance

System.Object
PipelineBase
DistanceMatrixViewer

Implements

ILoggerObject
IProgress
IPipelineIO

Inherited Members

PipelineBase.PipelineInputs
PipelineBase.PipelineOutputs
PipelineBase.Logger
PipelineBase.Progress
PipelineBase.ProgressChanged
PipelineBase.OnProgressChanged()

Namespace: SigStat.FusionBenchmark.VisualHelpers

Assembly: SigStat.FusionBenchmark.dll

Syntax

```
public class DistanceMatrixViewer : PipelineBase, ILoggerObject, IProgress, IPipelineIO
```

Properties

InputFeatures

Declaration

```
public List<FeatureDescriptor> InputFeatures { get; set; }
```

Property Value

TYPE	DESCRIPTION
List<FeatureDescriptor>	

InputFunc

Declaration

```
public Func<double[], double[], double> InputFunc { get; set; }
```

Property Value

TYPE	DESCRIPTION
Func<System.Double[], System.Double[], System.Double>	

InputSignatures

Declaration

```
public List<Signature> InputSignatures { get; set; }
```

Property Value

TYPE	DESCRIPTION
List<Signature>	

Methods

Calculate()

Declaration

```
public double[, ] Calculate()
```

Returns

TYPE	DESCRIPTION
System.Double[,]	

Implements

ILoggerObject

IProgress

IPipelineIO

Class ImageHelper

Inheritance

System.Object
ImageHelper

Namespace: SigStat.FusionBenchmark.VisualHelpers

Assembly: SigStat.FusionBenchmark.dll

Syntax

```
public static class ImageHelper : object
```

Methods

Move(System.Drawing.Point, Int32, Int32)

Declaration

```
public static System.Drawing.Point Move(this System.Drawing.Point point, int dx, int dy)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Drawing.Point	point	
System.Int32	dx	
System.Int32	dy	

Returns

TYPE	DESCRIPTION
System.Drawing.Point	

ReColour(Image<Rgba32>, Vertex, Rgba32)

Declaration

```
public static void ReColour(this Image<Rgba32> img, Vertex vertex, Rgba32 newCol)
```

Parameters

TYPE	NAME	DESCRIPTION
Image<Rgba32>	img	
Vertex	vertex	
Rgba32	newCol	

ReColour(Image<Rgba32>, System.Drawing.Point, Rgba32)

Declaration

```
public static void ReColour(this Image<Rgba32> img, System.Drawing.Point pos, Rgba32 newCol)
```

Parameters

TYPE	NAME	DESCRIPTION
Image<Rgba32>	img	
System.Drawing.Point	pos	
Rgba32	newCol	

ValidCoord(Image<Rgba32>, System.Drawing.Point)

Declaration

```
public static bool ValidCoord(this Image<Rgba32> img, System.Drawing.Point pos)
```

Parameters

TYPE	NAME	DESCRIPTION
Image<Rgba32>	img	
System.Drawing.Point	pos	

Returns

TYPE	DESCRIPTION
System.Boolean	

VariableColour(Image<Rgba32>, System.Drawing.Point, Int32)

Declaration

```
public static void VariableColour(this Image<Rgba32> img, System.Drawing.Point pos, int cnt)
```

Parameters

TYPE	NAME	DESCRIPTION
Image<Rgba32>	img	
System.Drawing.Point	pos	
System.Int32	cnt	

VariableColourLine(Image<Rgba32>, System.Drawing.Point, System.Drawing.Point, Int32)

Declaration

```
public static void VariableColourLine(this Image<Rgba32> img, System.Drawing.Point pos, System.Drawing.Point
pos2, int cnt)
```

Parameters

TYPE	NAME	DESCRIPTION
Image<Rgba32>	img	
System.Drawing.Point	pos	

TYPE	NAME	DESCRIPTION
System.Drawing.Point	pos2	
System.Int32	cnt	

Class StrokePairingDistances

Inheritance

System.Object
PipelineBase
StrokePairingDistances

Implements

ILoggerObject
IProgress
IPipelineIO

Inherited Members

PipelineBase.PipelineInputs
PipelineBase.PipelineOutputs
PipelineBase.Logger
PipelineBase.Progress
PipelineBase.ProgressChanged
PipelineBase.OnProgressChanged()

Namespace: SigStat.FusionBenchmark.VisualHelpers

Assembly: SigStat.FusionBenchmark.dll

Syntax

```
public class StrokePairingDistances : PipelineBase, ILoggerObject, IProgress, IPipelineIO
```

Properties

InputOfflineTrajectory

Declaration

```
public FeatureDescriptor<List<Vertex>> InputOfflineTrajectory { get; set; }
```

Property Value

TYPE	DESCRIPTION
FeatureDescriptor<List<Vertex>>	

InputOnlineTrajectory

Declaration

```
public FeatureDescriptor<List<Vertex>> InputOnlineTrajectory { get; set; }
```

Property Value

TYPE	DESCRIPTION
FeatureDescriptor<List<Vertex>>	

OfflineSignatures

Declaration

```
public List<Signature> OfflineSignatures { get; set; }
```

Property Value

TYPE	DESCRIPTION
List<Signature>	

OnlineSignatures

Declaration

```
public List<Signature> OnlineSignatures { get; set; }
```

Property Value

TYPE	DESCRIPTION
List<Signature>	

Methods

Calculate()

Declaration

```
public List<Tuple<string, string, double>> Calculate()
```

Returns

TYPE	DESCRIPTION
List<Tuple<System.String, System.String, System.Double>>	

PenDistance(Double[], Double[])

Declaration

```
public static double PenDistance(double[] vec1, double[] vec2)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Double[]	vec1	
System.Double[]	vec2	

Returns

TYPE	DESCRIPTION
System.Double	

Implements

ILoggerObject
IProgress
IPipelineIO

Class StrokePairSaver

Inheritance

System.Object
PipelineBase
StrokePairSaver

Implements

ILoggerObject
IProgress
ITransformation
IPipelineIO

Inherited Members

PipelineBase.PipelineInputs
PipelineBase.PipelineOutputs
PipelineBase.Logger
PipelineBase.Progress
PipelineBase.ProgressChanged
PipelineBase.OnProgressChanged()

Namespace: SigStat.FusionBenchmark.VisualHelpers

Assembly: SigStat.FusionBenchmark.dll

Syntax

```
public class StrokePairSaver : PipelineBase, ILoggerObject, IProgress, ITransformation, IPipelineIO
```

Properties

InputBasePath

Declaration

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

Property Value

TYPE	DESCRIPTION
System.String	

InputFileName

Declaration

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

Property Value

TYPE	DESCRIPTION
System.String	

InputImage

Declaration

```
public FeatureDescriptor<Image<Rgba32>> InputImage { get; set; }
```

Property Value

TYPE	DESCRIPTION
FeatureDescriptor<Image<Rgba32>>	

InputStrokeMatches

Declaration

```
public FeatureDescriptor<List<Tuple<int, Stroke, double, int>>> InputStrokeMatches { get; set; }
```

Property Value

TYPE	DESCRIPTION
FeatureDescriptor<List<Tuple<System.Int32, Stroke, System.Double, System.Int32>>>	

InputTrajectory

Declaration

```
public FeatureDescriptor<List<Vertex>> InputTrajectory { get; set; }
```

Property Value

TYPE	DESCRIPTION
FeatureDescriptor<List<Vertex>>	

Methods

Transform(Signature)

Declaration

```
public void Transform(Signature signature)
```

Parameters

TYPE	NAME	DESCRIPTION
Signature	signature	

Implements

ILoggerObject

IProgress

ITransformation

IPipelineIO

Class TxtHelper

Inheritance

System.Object

TxtHelper

Namespace: SigStat.FusionBenchmark.VisualHelpers

Assembly: SigStat.FusionBenchmark.dll

Syntax

```
public static class TxtHelper : object
```

Fields

BasePath

Declaration

```
public static string BasePath
```

Field Value

TYPE	DESCRIPTION
System.String	

Methods

ArrayToLines(Double[,])

Declaration

```
public static string[] ArrayToLines(double[, ] dists)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Double[,]	dists	

Returns

TYPE	DESCRIPTION
System.String[]	

BenchmarkResToLines(BenchmarkResults)

Declaration

```
public static string[] BenchmarkResToLines(BenchmarkResults results)
```

Parameters

TYPE	NAME	DESCRIPTION
BenchmarkResults	results	

Returns

TYPE	DESCRIPTION
System.String[]	

ReSamplingResultsToLines(List<Double>, List<List<Double>>)

Declaration

```
public static string[] ReSamplingResultsToLines(List<double> resList, List<List<double>> dataLists)
```

Parameters

TYPE	NAME	DESCRIPTION
List<System.Double>	resList	
List<List<System.Double>>	dataLists	

Returns

TYPE	DESCRIPTION
System.String[]	

ReSamplingResultsToLines(Tuple<List<Double>, List<List<Double>>>)

Declaration

```
public static string[] ReSamplingResultsToLines(Tuple<List<double>, List<List<double>>> results)
```

Parameters

TYPE	NAME	DESCRIPTION
Tuple<List<System.Double>, List<List<System.Double>>>	results	

Returns

TYPE	DESCRIPTION
System.String[]	

Save(String[], String)

Declaration

```
public static void Save(string[] lines, string fileName)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String[]	lines	
System.String	fileName	

TuplesToLines(List<Tuple<String, String, Double>>)

Declaration

```
public static string[] TuplesToLines(List<Tuple<string, string, double>> tuples)
```

Parameters

TYPE	NAME	DESCRIPTION
List<Tuple<System.String, System.String, System.Double>>	tuples	

Returns

TYPE	DESCRIPTION
System.String[]	

Class XYSaver

Inheritance

System.Object
PipelineBase
XYSaver

Implements

ILoggerObject
IProgress
ITransformation
IPipelineIO

Inherited Members

PipelineBase.PipelineInputs
PipelineBase.PipelineOutputs
PipelineBase.Logger
PipelineBase.Progress
PipelineBase.ProgressChanged
PipelineBase.OnProgressChanged()

Namespace: SigStat.FusionBenchmark.VisualHelpers

Assembly: SigStat.FusionBenchmark.dll

Syntax

```
public class XYSaver : PipelineBase, ILoggerObject, IProgress, ITransformation, IPipelineIO
```

Properties

InputBasePath

Declaration

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

Property Value

TYPE	DESCRIPTION
System.String	

InputButton

Declaration

```
public FeatureDescriptor<List<bool>> InputButton { get; set; }
```

Property Value

TYPE	DESCRIPTION
FeatureDescriptor<List<System.Boolean>>	

InputFileName

Declaration

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

Property Value

TYPE	DESCRIPTION
System.String	

InputImage

Declaration

```
public FeatureDescriptor<Image<Rgba32>> InputImage { get; set; }
```

Property Value

TYPE	DESCRIPTION
FeatureDescriptor<Image<Rgba32>>	

InputX

Declaration

```
public FeatureDescriptor<List<double>> InputX { get; set; }
```

Property Value

TYPE	DESCRIPTION
FeatureDescriptor<List<System.Double>>	

InputY

Declaration

```
public FeatureDescriptor<List<double>> InputY { get; set; }
```

Property Value

TYPE	DESCRIPTION
FeatureDescriptor<List<System.Double>>	

Methods

Transform(Signature)

Declaration

```
public void Transform(Signature signature)
```

Parameters

TYPE	NAME	DESCRIPTION
Signature	signature	

Implements

ILoggerObject

IProgress

ITransformation

IPipelineIO

Namespace SigStat.UI

Classes

App

Interaction logic for App.xaml

MainViewModel

MainWindow

Interaction logic for MainWindow.xaml

SignatureVisualizer

Enums

DisplayMode

Class App

Interaction logic for App.xaml

Inheritance

System.Object
System.Windows.Threading.DispatcherObject
System.Windows.Application
App

Implements

System.Windows.Markup.IQueryAmbient

Inherited Members

System.Windows.Application.Run()
System.Windows.Application.Run(System.Windows.Window)
System.Windows.Application.Shutdown()
System.Windows.Application.Shutdown(System.Int32)
System.Windows.Application.FindResource(System.Object)
System.Windows.Application.TryFindResource(System.Object)
System.Windows.Application.LoadComponent(System.Object, System.Uri)
System.Windows.Application.LoadComponent(System.Uri)
System.Windows.Application.GetResourceStream(System.Uri)
System.Windows.Application.GetContentStream(System.Uri)
System.Windows.Application.GetRemoteStream(System.Uri)
System.Windows.Application.GetCookie(System.Uri)
System.Windows.Application.SetCookie(System.Uri, System.String)
System.Windows.Application.System.Windows.Markup.IQueryAmbient.IsAmbientPropertyAvailable(System.String)
System.Windows.Application.OnStartup(System.Windows.StartupEventArgs)
System.Windows.Application.OnExit(System.Windows.ExitEventArgs)
System.Windows.Application.OnActivated(System.EventArgs)
System.Windows.Application.OnDeactivated(System.EventArgs)
System.Windows.Application.OnSessionEnding(System.Windows.SessionEndingCancelEventArgs)
System.Windows.Application.OnNavigating(System.Windows.Navigation.NavigatingCancelEventArgs)
System.Windows.Application.OnNavigated(System.Windows.Navigation.NavigationEventArgs)
System.Windows.Application.OnNavigationProgress(System.Windows.Navigation.NavigationProgressEventArgs)
System.Windows.Application.OnNavigationFailed(System.Windows.Navigation.NavigationFailedEventArgs)
System.Windows.Application.OnLoadCompleted(System.Windows.Navigation.NavigationEventArgs)
System.Windows.Application.OnNavigationStopped(System.Windows.Navigation.NavigationEventArgs)
System.Windows.Application.OnFragmentNavigation(System.Windows.Navigation.FragmentNavigationEventArgs)
System.Windows.Application.Current
System.Windows.Application.Windows
System.Windows.Application.MainWindow
System.Windows.Application.ShutdownMode
System.Windows.Application.Resources
System.Windows.Application.StartupUri
System.Windows.Application.Properties
System.Windows.Application.ResourceAssembly
System.Windows.Application.Startup
System.Windows.Application.Exit
System.Windows.Application.Activated
System.Windows.Application.Deactivated
System.Windows.Application.SessionEnding

System.Windows.Application.DispatcherUnhandledException

System.Windows.Application.Navigating

System.Windows.Application.Navigated

System.Windows.Application.NavigationProgress

System.Windows.Application.NavigationFailed

System.Windows.Application.LoadCompleted

System.Windows.Application.NavigationStopped

System.Windows.Application.FragmentNavigation

System.Windows.Threading.DispatcherObject.Dispatcher

System.Object.ToString()

System.Object.Equals(System.Object)

System.Object.Equals(System.Object, System.Object)

System.Object.ReferenceEquals(System.Object, System.Object)

System.Object.GetHashCode()

System.Object.GetType()

System.Object.MemberwiseClone()

Namespace: SigStat.UI

Assembly: SigStat.UI.dll

Syntax

```
public class App : Application, IHaveResources, IQueryAmbient
```

Implements

System.Windows.Markup.IQueryAmbient

Enum DisplayMode

Namespace: SigStat.UI

Assembly: SigStat.UI.dll

Syntax

```
public enum DisplayMode
```

Fields

NAME	DESCRIPTION
Original	
Zoom	

Class MainViewModel

Inheritance

System.Object
GalaSoft.MvvmLight.ObservableObject
MainViewModel

Implements

System.ComponentModel.INotifyPropertyChanged

Inherited Members

GalaSoft.MvvmLight.ObservableObject.VerifyPropertyName(System.String)
GalaSoft.MvvmLight.ObservableObject.RaisePropertyChanged(System.String)
GalaSoft.MvvmLight.ObservableObject.RaisePropertyChanged<T>(System.Linq.Expressions.Expression<System.Func<T>>)
GalaSoft.MvvmLight.ObservableObject.GetPropertyName<T>(System.Linq.Expressions.Expression<System.Func<T>>)
GalaSoft.MvvmLight.ObservableObject.Set<T>(System.Linq.Expressions.Expression<System.Func<T>>, T, T)
GalaSoft.MvvmLight.ObservableObject.Set<T>(System.String, T, T)
GalaSoft.MvvmLight.ObservableObject.Set<T>(T, T, System.String)
GalaSoft.MvvmLight.ObservableObject.PropertyChangedHandler
GalaSoft.MvvmLight.ObservableObject.PropertyChanged
System.Object.ToString()
System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()

Namespace: SigStat.UI

Assembly: SigStat.UI.dll

Syntax

```
public class MainViewModel : ObservableObject, INotifyPropertyChanged
```

Constructors

MainViewModel()

Declaration

```
public MainViewModel()
```

Properties

DatasetLoaders

Declaration

```
public ObservableCollection<Type> DatasetLoaders { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Collections.ObjectModel.ObservableCollection<System.Type>	

SelectedDatasetLoader

Declaration

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

Property Value

TYPE	DESCRIPTION
System.Type	

SelectedSignature

Declaration

```
public Signature SelectedSignature { get; set; }
```

Property Value

TYPE	DESCRIPTION
Signature	

SelectedSigner

Declaration

```
public Signer SelectedSigner { get; set; }
```

Property Value

TYPE	DESCRIPTION
Signer	

Signers

Declaration

```
public ObservableCollection<Signer> Signers { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Collections.ObjectModel.ObservableCollection<Signer>	

Implements

System.ComponentModel.INotifyPropertyChanged

Class MainWindow

Interaction logic for MainWindow.xaml

Inheritance

System.Object
System.Windows.Threading.DispatcherObject
System.Windows.DependencyObject
System.Windows.Media.Visual
System.Windows.UIElement
System.Windows.FrameworkElement
System.Windows.Controls.Control
System.Windows.Controls.ContentControl
System.Windows.Window
MainWindow

Implements

System.Windows.Media.Animation.IAnimatable
System.Windows.IFrameworkInputElement
System.Windows.IInputElement
System.ComponentModel.ISupportInitialize
System.Windows.Markup.IQueryAmbient
System.Windows.Markup.IAddChild

Inherited Members

System.Windows.Window.TaskbarItemInfoProperty
System.Windows.Window.DpiChangedEvent
System.Windows.Window.AllowsTransparencyProperty
System.Windows.Window.TitleProperty
System.Windows.Window.IconProperty
System.Windows.Window.SizeToContentProperty
System.Windows.Window.TopProperty
System.Windows.Window.LeftProperty
System.Windows.Window.ShowInTaskbarProperty
System.Windows.Window.IsActiveProperty
System.Windows.Window.WindowStyleProperty
System.Windows.Window.WindowStateProperty
System.Windows.Window.ResizeModeProperty
System.Windows.Window.TopmostProperty
System.Windows.Window.ShowActivatedProperty
System.Windows.Window.Show()
System.Windows.Window.Hide()
System.Windows.Window.Close()
System.Windows.Window.DragMove()
System.Windows.Window.ShowDialog()
System.Windows.Window.Activate()
System.Windows.Window.GetWindow(System.Windows.DependencyObject)
System.Windows.Window.OnCreateAutomationPeer()
System.Windows.Window.OnDpiChanged(System.Windows.DpiScale, System.Windows.DpiScale)
System.Windows.Window.OnVisualParentChanged(System.Windows.DependencyObject)
System.Windows.Window.MeasureOverride(System.Windows.Size)
System.Windows.Window.ArrangeOverride(System.Windows.Size)
System.Windows.Window.OnContentChanged(System.Object, System.Object)

System.Windows.Window.OnSourceInitialized(System.EventArgs)

System.Windows.Window.OnActivated(System.EventArgs)

System.Windows.Window.OnDeactivated(System.EventArgs)

System.Windows.Window.OnStateChanged(System.EventArgs)

System.Windows.Window.OnLocationChanged(System.EventArgs)

System.Windows.Window.OnClosing(System.ComponentModel.CancelEventArgs)

System.Windows.Window.OnClosed(System.EventArgs)

System.Windows.Window.OnContentRendered(System.EventArgs)

System.Windows.Window.OnManipulationBoundaryFeedback(System.Windows.Input.ManipulationBoundaryFeedbackEventArgs)

System.Windows.Window.LogicalChildren

System.Windows.Window.TaskbarItemInfo

System.Windows.Window.AllowsTransparency

System.Windows.Window.Title

System.Windows.Window.Icon

System.Windows.Window.SizeToContent

System.Windows.Window.Top

System.Windows.Window.Left

System.Windows.Window.RestoreBounds

System.Windows.Window.WindowStartupLocation

System.Windows.Window.ShowInTaskbar

System.Windows.Window.IsActive

System.Windows.Window.Owner

System.Windows.Window.OwnedWindows

System.Windows.Window.DialogResult

System.Windows.Window.WindowStyle

System.Windows.Window.WindowState

System.Windows.Window.ResizeMode

System.Windows.Window.Topmost

System.Windows.Window.ShowActivated

System.Windows.Window.SourceInitialized

System.Windows.Window.DpiChanged

System.Windows.Window.Activated

System.Windows.Window.Deactivated

System.Windows.Window.StateChanged

System.Windows.Window.LocationChanged

System.Windows.Window.Closing

System.Windows.Window.Closed

System.Windows.Window.ContentRendered

System.Windows.Controls.ContentControl.ContentProperty

System.Windows.Controls.ContentControl.HasContentProperty

System.Windows.Controls.ContentControl.ContentTemplateProperty

System.Windows.Controls.ContentControl.ContentTemplateSelectorProperty

System.Windows.Controls.ContentControl.ContentStringFormatProperty

System.Windows.Controls.ContentControl.System.Windows.Markup.IAddChild.AddChild(System.Object)

System.Windows.Controls.ContentControl.AddChild(System.Object)

System.Windows.Controls.ContentControl.System.Windows.Markup.IAddChild.AddText(System.String)

System.Windows.Controls.ContentControl.AddText(System.String)

System.Windows.Controls.ContentControl.OnContentTemplateChanged(System.Windows.DataTemplate, System.Windows.DataTemplate)

System.Windows.Controls.ContentControl.OnContentTemplateSelectorChanged(System.Windows.Controls.DataTemplateSelector, System.Windows.Controls.DataTemplateSelector)

System.Windows.Controls.ContentControl.OnContentStringFormatChanged(System.String, System.String)

System.Windows.Controls.ContentControl.Content
System.Windows.Controls.ContentControl.HasContent
System.Windows.Controls.ContentControl.ContentTemplate
System.Windows.Controls.ContentControl.ContentTemplateSelector
System.Windows.Controls.ContentControl.ContentStringFormat
System.Windows.Controls.Control.BorderBrushProperty
System.Windows.Controls.Control.BorderThicknessProperty
System.Windows.Controls.Control.BackgroundProperty
System.Windows.Controls.Control.ForegroundProperty
System.Windows.Controls.Control.FontFamilyProperty
System.Windows.Controls.Control.FontSizeProperty
System.Windows.Controls.Control.FontStretchProperty
System.Windows.Controls.Control.FontStyleProperty
System.Windows.Controls.Control.FontWeightProperty
System.Windows.Controls.Control.HorizontalContentAlignmentProperty
System.Windows.Controls.Control.VerticalContentAlignmentProperty
System.Windows.Controls.Control.TabIndexProperty
System.Windows.Controls.Control.IsTabStopProperty
System.Windows.Controls.Control.PaddingProperty
System.Windows.Controls.Control.TemplateProperty
System.Windows.Controls.Control.PreviewMouseDoubleClickEvent
System.Windows.Controls.Control.MouseDoubleClickEvent
System.Windows.Controls.Control.OnTemplateChanged(System.Windows.Controls.ControlTemplate, System.Windows.Controls.ControlTemplate)
System.Windows.Controls.Control.ToString()
System.Windows.Controls.Control.OnPreviewMouseDoubleClick(System.Windows.Input.MouseButtonEventArgs)
System.Windows.Controls.Control.OnMouseDoubleClick(System.Windows.Input.MouseButtonEventArgs)
System.Windows.Controls.Control.BorderBrush
System.Windows.Controls.Control.BorderThickness
System.Windows.Controls.Control.Background
System.Windows.Controls.Control.Foreground
System.Windows.Controls.Control.FontFamily
System.Windows.Controls.Control.FontSize
System.Windows.Controls.Control.FontStretch
System.Windows.Controls.Control.FontStyle
System.Windows.Controls.Control.FontWeight
System.Windows.Controls.Control.HorizontalContentAlignment
System.Windows.Controls.Control.VerticalContentAlignment
System.Windows.Controls.Control.TabIndex
System.Windows.Controls.Control.IsTabStop
System.Windows.Controls.Control.Padding
System.Windows.Controls.Control.Template
System.Windows.Controls.Control.HandlesScrolling
System.Windows.Controls.Control.PreviewMouseDoubleClick
System.Windows.Controls.Control.MouseDoubleClick
System.Windows.FrameworkElement.StyleProperty
System.Windows.FrameworkElement.OverridesDefaultStyleProperty
System.Windows.FrameworkElement.UseLayoutRoundingProperty
System.Windows.FrameworkElement.DefaultStyleKeyProperty
System.Windows.FrameworkElement.DataContextProperty
System.Windows.FrameworkElement.BindingGroupProperty
System.Windows.FrameworkElement.LanguageProperty

System.Windows.FrameworkElement.NameProperty
System.Windows.FrameworkElement.TagProperty
System.Windows.FrameworkElement.InputScopeProperty
System.Windows.FrameworkElement.RequestBringIntoViewEvent
System.Windows.FrameworkElement.SizeChangedEvent
System.Windows.FrameworkElement.ActualWidthProperty
System.Windows.FrameworkElement.ActualHeightProperty
System.Windows.FrameworkElement.LayoutTransformProperty
System.Windows.FrameworkElement.WidthProperty
System.Windows.FrameworkElement.MinWidthProperty
System.Windows.FrameworkElement.MaxWidthProperty
System.Windows.FrameworkElement.HeightProperty
System.Windows.FrameworkElement.MinHeightProperty
System.Windows.FrameworkElement.MaxHeightProperty
System.Windows.FrameworkElement.FlowDirectionProperty
System.Windows.FrameworkElement.MarginProperty
System.Windows.FrameworkElement.HorizontalAlignmentProperty
System.Windows.FrameworkElement.VerticalAlignmentProperty
System.Windows.FrameworkElement.FocusVisualStyleProperty
System.Windows.FrameworkElement.CursorProperty
System.Windows.FrameworkElement.ForceCursorProperty
System.Windows.FrameworkElement.LoadedEvent
System.Windows.FrameworkElement.UnloadedEvent
System.Windows.FrameworkElement.ToolTipProperty
System.Windows.FrameworkElement.ContextMenuProperty
System.Windows.FrameworkElement.ToolTipOpeningEvent
System.Windows.FrameworkElement.ToolTipClosingEvent
System.Windows.FrameworkElement.ContextMenuOpeningEvent
System.Windows.FrameworkElement.ContextMenuClosingEvent
System.Windows.FrameworkElement.OnStyleChanged(System.Windows.Style, System.Windows.Style)
System.Windows.FrameworkElement.ParentLayoutInvalidated(System.Windows.UIElement)
System.Windows.FrameworkElement.ApplyTemplate()
System.Windows.FrameworkElement.OnApplyTemplate()
System.Windows.FrameworkElement.BeginStoryboard(System.Windows.Media.Animation.Storyboard)
System.Windows.FrameworkElement.BeginStoryboard(System.Windows.Media.Animation.Storyboard, System.Windows.Media.Animation.HandoffBehavior)
System.Windows.FrameworkElement.BeginStoryboard(System.Windows.Media.Animation.Storyboard, System.Windows.Media.Animation.HandoffBehavior, System.Boolean)
System.Windows.FrameworkElement.GetVisualChild(System.Int32)
System.Windows.FrameworkElement.System.Windows.Markup.IQueryAmbient.IsAmbientPropertyAvailable(System.String)
System.Windows.FrameworkElement.GetTemplateChild(System.String)
System.Windows.FrameworkElement.FindResource(System.Object)
System.Windows.FrameworkElement.TryFindResource(System.Object)
System.Windows.FrameworkElement.SetResourceReference(System.Windows.DependencyProperty, System.Object)
System.Windows.FrameworkElement.OnPropertyChanged(System.Windows.DependencyPropertyChangedEventArgs)
System.Windows.FrameworkElement.GetBindingExpression(System.Windows.DependencyProperty)
System.Windows.FrameworkElement.SetBinding(System.Windows.DependencyProperty, System.Windows.Data.BindingBase)
System.Windows.FrameworkElement.SetBinding(System.Windows.DependencyProperty, System.String)
System.Windows.FrameworkElement.GetUIParentCore()
System.Windows.FrameworkElement.BringIntoView()
System.Windows.FrameworkElement.BringIntoView(System.Windows.Rect)
System.Windows.FrameworkElement.GetFlowDirection(System.Windows.DependencyObject)

System.Windows.FrameworkElement.SetFlowDirection(System.Windows.DependencyObject, System.Windows.FlowDirection)

System.Windows.FrameworkElement.MeasureCore(System.Windows.Size)

System.Windows.FrameworkElement.ArrangeCore(System.Windows.Rect)

System.Windows.FrameworkElement.OnRenderSizeChanged(System.Windows.SizeChangedInfo)

System.Windows.FrameworkElement.GetLayoutClip(System.Windows.Size)

System.Windows.FrameworkElement.MoveFocus(System.Windows.Input.TraversalRequest)

System.Windows.FrameworkElement.PredictFocus(System.Windows.Input.FocusNavigationDirection)

System.Windows.FrameworkElement.OnGotFocus(System.Windows.RoutedEventArgs)

System.Windows.FrameworkElement.BeginInit()

System.Windows.FrameworkElement.EndInit()

System.Windows.FrameworkElement.OnInitialized(System.EventArgs)

System.Windows.FrameworkElement.OnToolTipOpening(System.Windows.Controls.ToolTipEventArgs)

System.Windows.FrameworkElement.OnToolTipClosing(System.Windows.Controls.ToolTipEventArgs)

System.Windows.FrameworkElement.OnContextMenuOpening(System.Windows.Controls.ContextMenuEventArgs)

System.Windows.FrameworkElement.OnContextMenuClosing(System.Windows.Controls.ContextMenuEventArgs)

System.Windows.FrameworkElement.RegisterName(System.String, System.Object)

System.Windows.FrameworkElement.UnregisterName(System.String)

System.Windows.FrameworkElement.FindName(System.String)

System.Windows.FrameworkElement.UpdateDefaultStyle()

System.Windows.FrameworkElement.AddLogicalChild(System.Object)

System.Windows.FrameworkElement.RemoveLogicalChild(System.Object)

System.Windows.FrameworkElement.Style

System.Windows.FrameworkElement.OverridesDefaultStyle

System.Windows.FrameworkElement.UseLayoutRounding

System.Windows.FrameworkElement.DefaultStyleKey

System.Windows.FrameworkElement.Triggers

System.Windows.FrameworkElement.TemplatedParent

System.Windows.FrameworkElement.VisualChildrenCount

System.Windows.FrameworkElement.Resources

System.Windows.FrameworkElement.InheritanceBehavior

System.Windows.FrameworkElement.DataContext

System.Windows.FrameworkElement.BindingGroup

System.Windows.FrameworkElement.Language

System.Windows.FrameworkElement.Name

System.Windows.FrameworkElement.Tag

System.Windows.FrameworkElement.InputScope

System.Windows.FrameworkElement.ActualWidth

System.Windows.FrameworkElement.ActualHeight

System.Windows.FrameworkElement.LayoutTransform

System.Windows.FrameworkElement.Width

System.Windows.FrameworkElement.MinWidth

System.Windows.FrameworkElement.MaxWidth

System.Windows.FrameworkElement.Height

System.Windows.FrameworkElement.MinHeight

System.Windows.FrameworkElement.MaxHeight

System.Windows.FrameworkElement.FlowDirection

System.Windows.FrameworkElement.Margin

System.Windows.FrameworkElement.HorizontalAlignment

System.Windows.FrameworkElement.VerticalAlignment

System.Windows.FrameworkElement.FocusVisualStyle

System.Windows.FrameworkElement.Cursor

System.Windows.FrameworkElement.ForceCursor

System.Windows.FrameworkElement.IsInitialized
System.Windows.FrameworkElement.IsLoaded
System.Windows.FrameworkElement.ToolTip
System.Windows.FrameworkElement.ContextMenu
System.Windows.FrameworkElement.Parent
System.Windows.FrameworkElement.TargetUpdated
System.Windows.FrameworkElement.SourceUpdated
System.Windows.FrameworkElement.DataContextChanged
System.Windows.FrameworkElement.RequestBringIntoView
System.Windows.FrameworkElement.SizeChanged
System.Windows.FrameworkElement.Initialized
System.Windows.FrameworkElement.Loaded
System.Windows.FrameworkElement.Unloaded
System.Windows.FrameworkElement.ToolTipOpening
System.Windows.FrameworkElement.ToolTipClosing
System.Windows.FrameworkElement.ContextMenuOpening
System.Windows.FrameworkElement.ContextMenuClosing
System.Windows.UIElement.PreviewMouseDownEvent
System.Windows.UIElement.MouseDownEvent
System.Windows.UIElement.PreviewMouseUpEvent
System.Windows.UIElement.MouseUpEvent
System.Windows.UIElement.PreviewMouseLeftButtonDownEvent
System.Windows.UIElement.MouseLeftButtonDownEvent
System.Windows.UIElement.PreviewMouseLeftButtonUpEvent
System.Windows.UIElement.MouseLeftButtonUpEvent
System.Windows.UIElement.PreviewMouseRightButtonDownEvent
System.Windows.UIElement.MouseRightButtonDownEvent
System.Windows.UIElement.PreviewMouseRightButtonUpEvent
System.Windows.UIElement.MouseRightButtonUpEvent
System.Windows.UIElement.PreviewMouseMoveEvent
System.Windows.UIElement.MouseMoveEvent
System.Windows.UIElement.PreviewMouseWheelEvent
System.Windows.UIElement.MouseWheelEvent
System.Windows.UIElement.MouseEnterEvent
System.Windows.UIElement.MouseLeaveEvent
System.Windows.UIElement.GotMouseCaptureEvent
System.Windows.UIElement.LostMouseCaptureEvent
System.Windows.UIElement.QueryCursorEvent
System.Windows.UIElement.PreviewStylusDownEvent
System.Windows.UIElement.StylusDownEvent
System.Windows.UIElement.PreviewStylusUpEvent
System.Windows.UIElement.StylusUpEvent
System.Windows.UIElement.PreviewStylusMoveEvent
System.Windows.UIElement.StylusMoveEvent
System.Windows.UIElement.PreviewStylusInAirMoveEvent
System.Windows.UIElement.StylusInAirMoveEvent
System.Windows.UIElement.StylusEnterEvent
System.Windows.UIElement.StylusLeaveEvent
System.Windows.UIElement.PreviewStylusInRangeEvent
System.Windows.UIElement.StylusInRangeEvent
System.Windows.UIElement.PreviewStylusOutOfRangeEvent
System.Windows.UIElement.StylusOutOfRangeEvent

System.Windows.UIElement.PreviewStylusSystemGestureEvent
System.Windows.UIElement.StylusSystemGestureEvent
System.Windows.UIElement.GotStylusCaptureEvent
System.Windows.UIElement.LostStylusCaptureEvent
System.Windows.UIElement.StylusButtonDownEvent
System.Windows.UIElement.StylusButtonUpEvent
System.Windows.UIElement.PreviewStylusButtonDownEvent
System.Windows.UIElement.PreviewStylusButtonUpEvent
System.Windows.UIElement.PreviewKeyDownEvent
System.Windows.UIElement.KeyDownEvent
System.Windows.UIElement.PreviewKeyUpEvent
System.Windows.UIElement.KeyUpEvent
System.Windows.UIElement.PreviewGotKeyboardFocusEvent
System.Windows.UIElement.GotKeyboardFocusEvent
System.Windows.UIElement.PreviewLostKeyboardFocusEvent
System.Windows.UIElement.LostKeyboardFocusEvent
System.Windows.UIElement.PreviewTextInputEvent
System.Windows.UIElement.TextInputEvent
System.Windows.UIElement.PreviewQueryContinueDragEvent
System.Windows.UIElement.QueryContinueDragEvent
System.Windows.UIElement.PreviewGiveFeedbackEvent
System.Windows.UIElement.GiveFeedbackEvent
System.Windows.UIElement.PreviewDragEnterEvent
System.Windows.UIElement.DragEnterEvent
System.Windows.UIElement.PreviewDragOverEvent
System.Windows.UIElement.DragOverEvent
System.Windows.UIElement.PreviewDragLeaveEvent
System.Windows.UIElement.DragLeaveEvent
System.Windows.UIElement.PreviewDropEvent
System.Windows.UIElement.DropEvent
System.Windows.UIElement.PreviewTouchDownEvent
System.Windows.UIElement.TouchDownEvent
System.Windows.UIElement.PreviewTouchMoveEvent
System.Windows.UIElement.TouchMoveEvent
System.Windows.UIElement.PreviewTouchUpEvent
System.Windows.UIElement.TouchUpEvent
System.Windows.UIElement.GotTouchCaptureEvent
System.Windows.UIElement.LostTouchCaptureEvent
System.Windows.UIElement.TouchEnterEvent
System.Windows.UIElement.TouchLeaveEvent
System.Windows.UIElement.IsMouseDirectlyOverProperty
System.Windows.UIElement.IsMouseOverProperty
System.Windows.UIElement.IsStylusOverProperty
System.Windows.UIElement.IsKeyboardFocusWithinProperty
System.Windows.UIElement.IsMouseCapturedProperty
System.Windows.UIElement.IsMouseCaptureWithinProperty
System.Windows.UIElement.IsStylusDirectlyOverProperty
System.Windows.UIElement.IsStylusCapturedProperty
System.Windows.UIElement.IsStylusCaptureWithinProperty
System.Windows.UIElement.IsKeyboardFocusedProperty
System.Windows.UIElement.AreAnyTouchesDirectlyOverProperty
System.Windows.UIElement.AreAnyTouchesOverProperty

System.Windows.UIElement.AreAnyTouchesCapturedProperty
System.Windows.UIElement.AreAnyTouchesCapturedWithinProperty
System.Windows.UIElement.AllowDropProperty
System.Windows.UIElement.RenderTransformProperty
System.Windows.UIElement.RenderTransformOriginProperty
System.Windows.UIElement.OpacityProperty
System.Windows.UIElement.OpacityMaskProperty
System.Windows.UIElement.BitmapEffectProperty
System.Windows.UIElement.EffectProperty
System.Windows.UIElement.BitmapEffectInputProperty
System.Windows.UIElement.CacheModeProperty
System.Windows.UIElement.UidProperty
System.Windows.UIElement.VisibilityProperty
System.Windows.UIElement.ClipToBoundsProperty
System.Windows.UIElement.ClipProperty
System.Windows.UIElement.SnapsToDevicePixelsProperty
System.Windows.UIElement.GotFocusEvent
System.Windows.UIElement.LostFocusEvent
System.Windows.UIElement.IsFocusedProperty
System.Windows.UIElement.IsEnabledProperty
System.Windows.UIElement.IsHitTestVisibleProperty
System.Windows.UIElement.IsVisibleProperty
System.Windows.UIElement.FocusableProperty
System.Windows.UIElement.IsManipulationEnabledProperty
System.Windows.UIElement.ManipulationStartingEvent
System.Windows.UIElement.ManipulationStartedEvent
System.Windows.UIElement.ManipulationDeltaEvent
System.Windows.UIElement.ManipulationInertiaStartingEvent
System.Windows.UIElement.ManipulationBoundaryFeedbackEvent
System.Windows.UIElement.ManipulationCompletedEvent
System.Windows.UIElement.ApplyAnimationClock(System.Windows.DependencyProperty,
System.Windows.Media.Animation.AnimationClock)
System.Windows.UIElement.ApplyAnimationClock(System.Windows.DependencyProperty,
System.Windows.Media.Animation.AnimationClock, System.Windows.Media.Animation.HandoffBehavior)
System.Windows.UIElement.BeginAnimation(System.Windows.DependencyProperty,
System.Windows.Media.Animation.AnimationTimeline)
System.Windows.UIElement.BeginAnimation(System.Windows.DependencyProperty,
System.Windows.Media.Animation.AnimationTimeline, System.Windows.Media.Animation.HandoffBehavior)
System.Windows.UIElement.GetAnimationBaseValue(System.Windows.DependencyProperty)
System.Windows.UIElement.RaiseEvent(System.Windows.RoutedEventArgs)
System.Windows.UIElement.AddHandler(System.Windows.RoutedEvent, System.Delegate)
System.Windows.UIElement.AddHandler(System.Windows.RoutedEvent, System.Delegate, System.Boolean)
System.Windows.UIElement.RemoveHandler(System.Windows.RoutedEvent, System.Delegate)
System.Windows.UIElement.AddToEventRoute(System.Windows.EventRoute, System.Windows.RoutedEventArgs)
System.Windows.UIElement.OnPreviewMouseDown(System.Windows.Input.MouseButtonEventArgs)
System.Windows.UIElement.OnMouseDown(System.Windows.Input.MouseButtonEventArgs)
System.Windows.UIElement.OnPreviewMouseUp(System.Windows.Input.MouseButtonEventArgs)
System.Windows.UIElement.OnMouseUp(System.Windows.Input.MouseButtonEventArgs)
System.Windows.UIElement.OnPreviewMouseLeftButtonDown(System.Windows.Input.MouseButtonEventArgs)
System.Windows.UIElement.OnMouseLeftButtonDown(System.Windows.Input.MouseButtonEventArgs)
System.Windows.UIElement.OnPreviewMouseLeftButtonUp(System.Windows.Input.MouseButtonEventArgs)
System.Windows.UIElement.OnMouseLeftButtonUp(System.Windows.Input.MouseButtonEventArgs)

System.Windows.UIElement.OnPreviewMouseRightButtonDown(System.Windows.Input.MouseButtonEventArgs)

System.Windows.UIElement.OnMouseRightButtonDown(System.Windows.Input.MouseButtonEventArgs)

System.Windows.UIElement.OnPreviewMouseRightButtonUp(System.Windows.Input.MouseButtonEventArgs)

System.Windows.UIElement.OnMouseRightButtonUp(System.Windows.Input.MouseButtonEventArgs)

System.Windows.UIElement.OnPreviewMouseMove(System.Windows.Input.MouseEventArgs)

System.Windows.UIElement.OnMouseMove(System.Windows.Input.MouseEventArgs)

System.Windows.UIElement.OnPreviewMouseWheel(System.Windows.Input.MouseWheelEventArgs)

System.Windows.UIElement.OnMouseWheel(System.Windows.Input.MouseWheelEventArgs)

System.Windows.UIElement.OnMouseEnter(System.Windows.Input.MouseEventArgs)

System.Windows.UIElement.OnMouseLeave(System.Windows.Input.MouseEventArgs)

System.Windows.UIElement.OnGotMouseCapture(System.Windows.Input.MouseEventArgs)

System.Windows.UIElement.OnLostMouseCapture(System.Windows.Input.MouseEventArgs)

System.Windows.UIElement.OnQueryCursor(System.Windows.Input.QueryCursorEventArgs)

System.Windows.UIElement.OnPreviewStylusDown(System.Windows.Input.StylusDownEventArgs)

System.Windows.UIElement.OnStylusDown(System.Windows.Input.StylusDownEventArgs)

System.Windows.UIElement.OnPreviewStylusUp(System.Windows.Input.StylusEventArgs)

System.Windows.UIElement.OnStylusUp(System.Windows.Input.StylusEventArgs)

System.Windows.UIElement.OnPreviewStylusMove(System.Windows.Input.StylusEventArgs)

System.Windows.UIElement.OnStylusMove(System.Windows.Input.StylusEventArgs)

System.Windows.UIElement.OnPreviewStylusInAirMove(System.Windows.Input.StylusEventArgs)

System.Windows.UIElement.OnStylusInAirMove(System.Windows.Input.StylusEventArgs)

System.Windows.UIElement.OnStylusEnter(System.Windows.Input.StylusEventArgs)

System.Windows.UIElement.OnStylusLeave(System.Windows.Input.StylusEventArgs)

System.Windows.UIElement.OnPreviewStylusInRange(System.Windows.Input.StylusEventArgs)

System.Windows.UIElement.OnStylusInRange(System.Windows.Input.StylusEventArgs)

System.Windows.UIElement.OnPreviewStylusOutOfRange(System.Windows.Input.StylusEventArgs)

System.Windows.UIElement.OnStylusOutOfRange(System.Windows.Input.StylusEventArgs)

System.Windows.UIElement.OnPreviewStylusSystemGesture(System.Windows.Input.StylusSystemGestureEventArgs)

System.Windows.UIElement.OnStylusSystemGesture(System.Windows.Input.StylusSystemGestureEventArgs)

System.Windows.UIElement.OnGotStylusCapture(System.Windows.Input.StylusEventArgs)

System.Windows.UIElement.OnLostStylusCapture(System.Windows.Input.StylusEventArgs)

System.Windows.UIElement.OnStylusButtonDown(System.Windows.Input.StylusButtonEventArgs)

System.Windows.UIElement.OnStylusButtonUp(System.Windows.Input.StylusButtonEventArgs)

System.Windows.UIElement.OnPreviewStylusButtonDown(System.Windows.Input.StylusButtonEventArgs)

System.Windows.UIElement.OnPreviewStylusButtonUp(System.Windows.Input.StylusButtonEventArgs)

System.Windows.UIElement.OnPreviewKeyDown(System.Windows.Input.KeyEventArgs)

System.Windows.UIElement.OnKeyDown(System.Windows.Input.KeyEventArgs)

System.Windows.UIElement.OnPreviewKeyUp(System.Windows.Input.KeyEventArgs)

System.Windows.UIElement.OnKeyUp(System.Windows.Input.KeyEventArgs)

System.Windows.UIElement.OnPreviewGotKeyboardFocus(System.Windows.Input.KeyboardFocusChangedEventArgs)

System.Windows.UIElement.OnGotKeyboardFocus(System.Windows.Input.KeyboardFocusChangedEventArgs)

System.Windows.UIElement.OnPreviewLostKeyboardFocus(System.Windows.Input.KeyboardFocusChangedEventArgs)

System.Windows.UIElement.OnLostKeyboardFocus(System.Windows.Input.KeyboardFocusChangedEventArgs)

System.Windows.UIElement.OnPreviewTextInput(System.Windows.Input.TextCompositionEventArgs)

System.Windows.UIElement.OnTextInput(System.Windows.Input.TextCompositionEventArgs)

System.Windows.UIElement.OnPreviewQueryContinueDrag(System.Windows.QueryContinueDragEventArgs)

System.Windows.UIElement.OnQueryContinueDrag(System.Windows.QueryContinueDragEventArgs)

System.Windows.UIElement.OnPreviewGiveFeedback(System.Windows.GiveFeedbackEventArgs)

System.Windows.UIElement.OnGiveFeedback(System.Windows.GiveFeedbackEventArgs)

System.Windows.UIElement.OnPreviewDragEnter(System.Windows.DragEventArgs)

System.Windows.UIElement.OnDragEnter(System.Windows.DragEventArgs)

System.Windows.UIElement.OnPreviewDragOver(System.Windows.DragEventArgs)

System.Windows.UIElement.OnDragOver(System.Windows.DragEventArgs)
System.Windows.UIElement.OnPreviewDragLeave(System.Windows.DragEventArgs)
System.Windows.UIElement.OnDragLeave(System.Windows.DragEventArgs)
System.Windows.UIElement.OnPreviewDrop(System.Windows.DragEventArgs)
System.Windows.UIElement.OnDrop(System.Windows.DragEventArgs)
System.Windows.UIElement.OnPreviewTouchDown(System.Windows.Input.TouchEventArgs)
System.Windows.UIElement.OnTouchDown(System.Windows.Input.TouchEventArgs)
System.Windows.UIElement.OnPreviewTouchMove(System.Windows.Input.TouchEventArgs)
System.Windows.UIElement.OnTouchMove(System.Windows.Input.TouchEventArgs)
System.Windows.UIElement.OnPreviewTouchUp(System.Windows.Input.TouchEventArgs)
System.Windows.UIElement.OnTouchUp(System.Windows.Input.TouchEventArgs)
System.Windows.UIElement.OnGotTouchCapture(System.Windows.Input.TouchEventArgs)
System.Windows.UIElement.OnLostTouchCapture(System.Windows.Input.TouchEventArgs)
System.Windows.UIElement.OnTouchEnter(System.Windows.Input.TouchEventArgs)
System.Windows.UIElement.OnTouchLeave(System.Windows.Input.TouchEventArgs)
System.Windows.UIElement.OnIsMouseDirectlyOverChanged(System.Windows.DependencyPropertyChangedEventArgs)
System.Windows.UIElement.OnIsKeyboardFocusWithinChanged(System.Windows.DependencyPropertyChangedEventArgs)
System.Windows.UIElement.OnIsMouseCapturedChanged(System.Windows.DependencyPropertyChangedEventArgs)
System.Windows.UIElement.OnIsMouseCaptureWithinChanged(System.Windows.DependencyPropertyChangedEventArgs)
System.Windows.UIElement.OnIsStylusDirectlyOverChanged(System.Windows.DependencyPropertyChangedEventArgs)
System.Windows.UIElement.OnIsStylusCapturedChanged(System.Windows.DependencyPropertyChangedEventArgs)
System.Windows.UIElement.OnIsStylusCaptureWithinChanged(System.Windows.DependencyPropertyChangedEventArgs)
System.Windows.UIElement.OnIsKeyboardFocusedChanged(System.Windows.DependencyPropertyChangedEventArgs)
System.Windows.UIElement.InvalidateMeasure()
System.Windows.UIElement.InvalidateArrange()
System.Windows.UIElement.InvalidateVisual()
System.Windows.UIElement.OnChildDesiredSizeChanged(System.Windows.UIElement)
System.Windows.UIElement.Measure(System.Windows.Size)
System.Windows.UIElement.Arrange(System.Windows.Rect)
System.Windows.UIElement.OnRender(System.Windows.Media.DrawingContext)
System.Windows.UIElement.UpdateLayout()
System.Windows.UIElement.TranslatePoint(System.Windows.Point, System.Windows.UIElement)
System.Windows.UIElement.InputHitTest(System.Windows.Point)
System.Windows.UIElement.CaptureMouse()
System.Windows.UIElement.ReleaseMouseCapture()
System.Windows.UIElement.CaptureStylus()
System.Windows.UIElement.ReleaseStylusCapture()
System.Windows.UIElement.Focus()
System.Windows.UIElement.OnAccessKey(System.Windows.Input.AccessKeyEventArgs)
System.Windows.UIElement.HitTestCore(System.Windows.Media.PointHitTestParameters)
System.Windows.UIElement.HitTestCore(System.Windows.Media.GeometryHitTestParameters)
System.Windows.UIElement.OnLostFocus(System.Windows.RoutedEventArgs)
System.Windows.UIElement.OnManipulationStarting(System.Windows.Input.ManipulationStartingEventArgs)
System.Windows.UIElement.OnManipulationStarted(System.Windows.Input.ManipulationStartedEventArgs)
System.Windows.UIElement.OnManipulationDelta(System.Windows.Input.ManipulationDeltaEventArgs)
System.Windows.UIElement.OnManipulationInertiaStarting(System.Windows.Input.ManipulationInertiaStartingEventArgs)
System.Windows.UIElement.OnManipulationCompleted(System.Windows.Input.ManipulationCompletedEventArgs)
System.Windows.UIElement.CaptureTouch(System.Windows.Input.TouchDevice)
System.Windows.UIElement.ReleaseTouchCapture(System.Windows.Input.TouchDevice)
System.Windows.UIElement.ReleaseAllTouchCaptures()
System.Windows.UIElement.HasAnimatedProperties
System.Windows.UIElement.InputBindings

System.Windows.UIElement.CommandBindings
System.Windows.UIElement.AllowDrop
System.Windows.UIElement.StylusPlugIns
System.Windows.UIElement.DesiredSize
System.Windows.UIElement.IsMeasureValid
System.Windows.UIElement.IsArrangeValid
System.Windows.UIElement.RenderSize
System.Windows.UIElement.RenderTransform
System.Windows.UIElement.RenderTransformOrigin
System.Windows.UIElement.IsMouseDirectlyOver
System.Windows.UIElement.IsMouseOver
System.Windows.UIElement.IsStylusOver
System.Windows.UIElement.IsKeyboardFocusWithin
System.Windows.UIElement.IsMouseCaptured
System.Windows.UIElement.IsMouseCaptureWithin
System.Windows.UIElement.IsStylusDirectlyOver
System.Windows.UIElement.IsStylusCaptured
System.Windows.UIElement.IsStylusCaptureWithin
System.Windows.UIElement.IsKeyboardFocused
System.Windows.UIElement.IsInputMethodEnabled
System.Windows.UIElement.Opacity
System.Windows.UIElement.OpacityMask
System.Windows.UIElement.BitmapEffect
System.Windows.UIElement.Effect
System.Windows.UIElement.BitmapEffectInput
System.Windows.UIElement.CacheMode
System.Windows.UIElement.Uid
System.Windows.UIElement.Visibility
System.Windows.UIElement.ClipToBounds
System.Windows.UIElement.Clip
System.Windows.UIElement.SnapsToDevicePixels
System.Windows.UIElement.HasEffectiveKeyboardFocus
System.Windows.UIElement.IsFocused
System.Windows.UIElement.IsEnabled
System.Windows.UIElement.IsEnabledCore
System.Windows.UIElement.IsHitTestVisible
System.Windows.UIElement.IsVisible
System.Windows.UIElement.Focusable
System.Windows.UIElement.PersistId
System.Windows.UIElement.IsManipulationEnabled
System.Windows.UIElement.AreAnyTouchesOver
System.Windows.UIElement.AreAnyTouchesDirectlyOver
System.Windows.UIElement.AreAnyTouchesCapturedWithin
System.Windows.UIElement.AreAnyTouchesCaptured
System.Windows.UIElement.TouchesCaptured
System.Windows.UIElement.TouchesCapturedWithin
System.Windows.UIElement.TouchesOver
System.Windows.UIElement.TouchesDirectlyOver
System.Windows.UIElement.PreviewMouseDown
System.Windows.UIElement.MouseDown
System.Windows.UIElement.PreviewMouseUp
System.Windows.UIElement.MouseUp

System.Windows.UIElement.PreviewMouseLeftButtonDown
System.Windows.UIElement.MouseLeftButtonDown
System.Windows.UIElement.PreviewMouseLeftButtonUp
System.Windows.UIElement.MouseLeftButtonUp
System.Windows.UIElement.PreviewMouseRightButtonDown
System.Windows.UIElement.MouseRightButtonDown
System.Windows.UIElement.PreviewMouseRightButtonUp
System.Windows.UIElement.MouseRightButtonUp
System.Windows.UIElement.PreviewMouseMove
System.Windows.UIElement.MouseMove
System.Windows.UIElement.PreviewMouseWheel
System.Windows.UIElement.MouseWheel
System.Windows.UIElement.MouseEnter
System.Windows.UIElement.MouseLeave
System.Windows.UIElement.GotMouseCapture
System.Windows.UIElement.LostMouseCapture
System.Windows.UIElement.QueryCursor
System.Windows.UIElement.PreviewStylusDown
System.Windows.UIElement.StylusDown
System.Windows.UIElement.PreviewStylusUp
System.Windows.UIElement.StylusUp
System.Windows.UIElement.PreviewStylusMove
System.Windows.UIElement.StylusMove
System.Windows.UIElement.PreviewStylusInAirMove
System.Windows.UIElement.StylusInAirMove
System.Windows.UIElement.StylusEnter
System.Windows.UIElement.StylusLeave
System.Windows.UIElement.PreviewStylusInRange
System.Windows.UIElement.StylusInRange
System.Windows.UIElement.PreviewStylusOutOfRange
System.Windows.UIElement.StylusOutOfRange
System.Windows.UIElement.PreviewStylusSystemGesture
System.Windows.UIElement.StylusSystemGesture
System.Windows.UIElement.GotStylusCapture
System.Windows.UIElement.LostStylusCapture
System.Windows.UIElement.StylusButtonDown
System.Windows.UIElement.StylusButtonUp
System.Windows.UIElement.PreviewStylusButtonDown
System.Windows.UIElement.PreviewStylusButtonUp
System.Windows.UIElement.PreviewKeyDown
System.Windows.UIElement.KeyDown
System.Windows.UIElement.PreviewKeyUp
System.Windows.UIElement.KeyUp
System.Windows.UIElement.PreviewGotKeyboardFocus
System.Windows.UIElement.GotKeyboardFocus
System.Windows.UIElement.PreviewLostKeyboardFocus
System.Windows.UIElement.LostKeyboardFocus
System.Windows.UIElement.PreviewTextInput
System.Windows.UIElement.TextInput
System.Windows.UIElement.PreviewQueryContinueDrag
System.Windows.UIElement.QueryContinueDrag
System.Windows.UIElement.PreviewGiveFeedback

System.Windows.UIElement.GiveFeedback
System.Windows.UIElement.PreviewDragEnter
System.Windows.UIElement.DragEnter
System.Windows.UIElement.PreviewDragOver
System.Windows.UIElement.DragOver
System.Windows.UIElement.PreviewDragLeave
System.Windows.UIElement.DragLeave
System.Windows.UIElement.PreviewDrop
System.Windows.UIElement.Drop
System.Windows.UIElement.PreviewTouchDown
System.Windows.UIElement.TouchDown
System.Windows.UIElement.PreviewTouchMove
System.Windows.UIElement.TouchMove
System.Windows.UIElement.PreviewTouchUp
System.Windows.UIElement.TouchUp
System.Windows.UIElement.GotTouchCapture
System.Windows.UIElement.LostTouchCapture
System.Windows.UIElement.TouchEnter
System.Windows.UIElement.TouchLeave
System.Windows.UIElement.IsMouseDirectlyOverChanged
System.Windows.UIElement.IsKeyboardFocusWithinChanged
System.Windows.UIElement.IsMouseCapturedChanged
System.Windows.UIElement.IsMouseCaptureWithinChanged
System.Windows.UIElement.IsStylusDirectlyOverChanged
System.Windows.UIElement.IsStylusCapturedChanged
System.Windows.UIElement.IsStylusCaptureWithinChanged
System.Windows.UIElement.IsKeyboardFocusedChanged
System.Windows.UIElement.LayoutUpdated
System.Windows.UIElement.GotFocus
System.Windows.UIElement.LostFocus
System.Windows.UIElement.IsEnabledChanged
System.Windows.UIElement.IsHitTestVisibleChanged
System.Windows.UIElement.IsVisibleChanged
System.Windows.UIElement.FocusableChanged
System.Windows.UIElement.ManipulationStarting
System.Windows.UIElement.ManipulationStarted
System.Windows.UIElement.ManipulationDelta
System.Windows.UIElement.ManipulationInertiaStarting
System.Windows.UIElement.ManipulationBoundaryFeedback
System.Windows.UIElement.ManipulationCompleted
System.Windows.Media.Visual.AddVisualChild(System.Windows.Media.Visual)
System.Windows.Media.Visual.RemoveVisualChild(System.Windows.Media.Visual)
System.Windows.Media.Visual.OnVisualChildrenChanged(System.Windows.DependencyObject,
System.Windows.DependencyObject)
System.Windows.Media.Visual.IsAncestorOf(System.Windows.DependencyObject)
System.Windows.Media.Visual.IsDescendantOf(System.Windows.DependencyObject)
System.Windows.Media.Visual.FindCommonVisualAncestor(System.Windows.DependencyObject)
System.Windows.Media.Visual.TransformToAncestor(System.Windows.Media.Visual)
System.Windows.Media.Visual.TransformToAncestor(System.Windows.Media.Media3D.Visual3D)
System.Windows.Media.Visual.TransformToDescendant(System.Windows.Media.Visual)
System.Windows.Media.Visual.TransformToVisual(System.Windows.Media.Visual)
System.Windows.Media.Visual.PointToScreen(System.Windows.Point)

System.Windows.Media.Visual.PointFromScreen(System.Windows.Point)

System.Windows.Media.Visual.VisualParent

System.Windows.Media.Visual.VisualTransform

System.Windows.Media.Visual.VisualEffect

System.Windows.Media.Visual.VisualBitmapEffect

System.Windows.Media.Visual.VisualBitmapEffectInput

System.Windows.Media.Visual.VisualCacheMode

System.Windows.Media.Visual.VisualScrollableAreaClip

System.Windows.Media.Visual.VisualClip

System.Windows.Media.Visual.VisualOffset

System.Windows.Media.Visual.VisualOpacity

System.Windows.Media.Visual.VisualEdgeMode

System.Windows.Media.Visual.VisualBitmapScalingMode

System.Windows.Media.Visual.VisualClearTypeHint

System.Windows.Media.Visual.VisualTextRenderingMode

System.Windows.Media.Visual.VisualTextHintingMode

System.Windows.Media.Visual.VisualOpacityMask

System.Windows.Media.Visual.VisualXSnappingGuidelines

System.Windows.Media.Visual.VisualYSnappingGuidelines

System.Windows.DependencyObject.Equals(System.Object)

System.Windows.DependencyObject.GetHashCode()

System.Windows.DependencyObject.GetValue(System.Windows.DependencyProperty)

System.Windows.DependencyObject.SetValue(System.Windows.DependencyProperty, System.Object)

System.Windows.DependencyObject.SetCurrentValue(System.Windows.DependencyProperty, System.Object)

System.Windows.DependencyObject.SetValue(System.Windows.DependencyPropertyKey, System.Object)

System.Windows.DependencyObject.ClearValue(System.Windows.DependencyProperty)

System.Windows.DependencyObject.ClearValue(System.Windows.DependencyPropertyKey)

System.Windows.DependencyObject.CoerceValue(System.Windows.DependencyProperty)

System.Windows.DependencyObject.InvalidateProperty(System.Windows.DependencyProperty)

System.Windows.DependencyObject.ShouldSerializeProperty(System.Windows.DependencyProperty)

System.Windows.DependencyObject.ReadLocalValue(System.Windows.DependencyProperty)

System.Windows.DependencyObject.GetLocalValueEnumerator()

System.Windows.DependencyObject.DependencyObjectType

System.Windows.DependencyObject.IsSealed

System.Windows.Threading.DispatcherObject.Dispatcher

System.Object.Equals(System.Object, System.Object)

System.Object.ReferenceEquals(System.Object, System.Object)

System.Object.GetType()

System.Object.MemberwiseClone()

Namespace: SigStat.UI

Assembly: SigStat.UI.dll

Syntax

```
public class MainWindow : Window, DUCE.IResource, IAnimatable, IFrameworkInputElement, IInputElement,
ISupportInitialize, IHaveResources, IQueryAmbient, IAddChild, IWindowService
```

Constructors

MainWindow()

Declaration

```
public MainWindow()
```

Properties

NormalizeRotaiton2

Declaration

```
public object NormalizeRotaiton2 { get; }
```

Property Value

TYPE	DESCRIPTION
System.Object	

VM

Declaration

```
public MainViewModel VM { get; set; }
```

Property Value

TYPE	DESCRIPTION
MainViewModel	

Implements

System.Windows.Media.Animation.IAnimatable
System.Windows.IFrameworkInputElement
System.Windows.IInputElement
System.ComponentModel.ISupportInitialize
System.Windows.Markup.IQueryAmbient
System.Windows.Markup.IAddChild

Class SignatureVisualizer

Inheritance

System.Object
System.Windows.Threading.DispatcherObject
System.Windows.DependencyObject
System.Windows.Media.Visual
System.Windows.UIElement
System.Windows.FrameworkElement
System.Windows.Controls.Panel
System.Windows.Controls.Grid
SignatureVisualizer

Implements

System.Windows.Media.Animation.IAnimatable
System.Windows.IFrameworkInputElement
System.Windows.IInputElement
System.ComponentModel.ISupportInitialize
System.Windows.Markup.IQueryAmbient
System.Windows.Markup.IAddChild

Inherited Members

System.Windows.Controls.Grid.ShowGridLinesProperty
System.Windows.Controls.Grid.ColumnProperty
System.Windows.Controls.Grid.RowProperty
System.Windows.Controls.Grid.ColumnSpanProperty
System.Windows.Controls.Grid.RowSpanProperty
System.Windows.Controls.Grid.IsSharedSizeScopeProperty
System.Windows.Controls.Grid.System.Windows.Markup.IAddChild.AddChild(System.Object)
System.Windows.Controls.Grid.System.Windows.Markup.IAddChild.AddText(System.String)
System.Windows.Controls.Grid.SetColumn(System.Windows.UIElement, System.Int32)
System.Windows.Controls.Grid.GetColumn(System.Windows.UIElement)
System.Windows.Controls.Grid.SetRow(System.Windows.UIElement, System.Int32)
System.Windows.Controls.Grid.GetRow(System.Windows.UIElement)
System.Windows.Controls.Grid.SetColumnSpan(System.Windows.UIElement, System.Int32)
System.Windows.Controls.Grid.GetColumnSpan(System.Windows.UIElement)
System.Windows.Controls.Grid.SetRowSpan(System.Windows.UIElement, System.Int32)
System.Windows.Controls.Grid.GetRowSpan(System.Windows.UIElement)
System.Windows.Controls.Grid.SetIsSharedSizeScope(System.Windows.UIElement, System.Boolean)
System.Windows.Controls.Grid.GetIsSharedSizeScope(System.Windows.UIElement)
System.Windows.Controls.Grid.GetVisualChild(System.Int32)
System.Windows.Controls.Grid.MeasureOverride(System.Windows.Size)
System.Windows.Controls.Grid.ArrangeOverride(System.Windows.Size)
System.Windows.Controls.Grid.OnVisualChildrenChanged(System.Windows.DependencyObject,
System.Windows.DependencyObject)
System.Windows.Controls.Grid.LogicalChildren
System.Windows.Controls.Grid.ShowGridLines
System.Windows.Controls.Grid.ColumnDefinitions
System.Windows.Controls.Grid.RowDefinitions
System.Windows.Controls.Grid.VisualChildrenCount
System.Windows.Controls.Panel.BackgroundProperty
System.Windows.Controls.Panel.IsItemsHostProperty
System.Windows.Controls.Panel.ZIndexProperty

System.Windows.Controls.Panel.OnRender(System.Windows.Media.DrawingContext)
System.Windows.Controls.Panel.OnIsItemsHostChanged(System.Boolean, System.Boolean)
System.Windows.Controls.Panel.CreateUIElementCollection(System.Windows.FrameworkElement)
System.Windows.Controls.Panel.SetZIndex(System.Windows.UIElement, System.Int32)
System.Windows.Controls.Panel.GetZIndex(System.Windows.UIElement)
System.Windows.Controls.Panel.Background
System.Windows.Controls.Panel.Children
System.Windows.Controls.Panel.IsItemsHost
System.Windows.Controls.Panel.LogicalOrientationPublic
System.Windows.Controls.Panel.LogicalOrientation
System.Windows.Controls.Panel.HasLogicalOrientationPublic
System.Windows.Controls.Panel.HasLogicalOrientation
System.Windows.Controls.Panel.InternalChildren
System.Windows.FrameworkElement.StyleProperty
System.Windows.FrameworkElement.OverridesDefaultStyleProperty
System.Windows.FrameworkElement.UseLayoutRoundingProperty
System.Windows.FrameworkElement.DefaultStyleKeyProperty
System.Windows.FrameworkElement.DataContextProperty
System.Windows.FrameworkElement.BindingGroupProperty
System.Windows.FrameworkElement.LanguageProperty
System.Windows.FrameworkElement.NameProperty
System.Windows.FrameworkElement.TagProperty
System.Windows.FrameworkElement.InputScopeProperty
System.Windows.FrameworkElement.RequestBringIntoViewEvent
System.Windows.FrameworkElement.SizeChangedEvent
System.Windows.FrameworkElement.ActualWidthProperty
System.Windows.FrameworkElement.ActualHeightProperty
System.Windows.FrameworkElement.LayoutTransformProperty
System.Windows.FrameworkElement.WidthProperty
System.Windows.FrameworkElement.MinWidthProperty
System.Windows.FrameworkElement.MaxWidthProperty
System.Windows.FrameworkElement.HeightProperty
System.Windows.FrameworkElement.MinHeightProperty
System.Windows.FrameworkElement.MaxHeightProperty
System.Windows.FrameworkElement.FlowDirectionProperty
System.Windows.FrameworkElement.MarginProperty
System.Windows.FrameworkElement.HorizontalAlignmentProperty
System.Windows.FrameworkElement.VerticalAlignmentProperty
System.Windows.FrameworkElement.FocusVisualStyleProperty
System.Windows.FrameworkElement.CursorProperty
System.Windows.FrameworkElement.ForceCursorProperty
System.Windows.FrameworkElement.LoadedEvent
System.Windows.FrameworkElement.UnloadedEvent
System.Windows.FrameworkElement.ToolTipProperty
System.Windows.FrameworkElement.ContextMenuProperty
System.Windows.FrameworkElement.ToolTipOpeningEvent
System.Windows.FrameworkElement.ToolTipClosingEvent
System.Windows.FrameworkElement.ContextMenuOpeningEvent
System.Windows.FrameworkElement.ContextMenuClosingEvent
System.Windows.FrameworkElement.OnStyleChanged(System.Windows.Style, System.Windows.Style)
System.Windows.FrameworkElement.ParentLayoutInvalidated(System.Windows.UIElement)
System.Windows.FrameworkElement.ApplyTemplate()

System.Windows.FrameworkElement.OnApplyTemplate()
System.Windows.FrameworkElement.BeginStoryboard(System.Windows.Media.Animation.Storyboard)
System.Windows.FrameworkElement.BeginStoryboard(System.Windows.Media.Animation.Storyboard, System.Windows.Media.Animation.HandoffBehavior)
System.Windows.FrameworkElement.BeginStoryboard(System.Windows.Media.Animation.Storyboard, System.Windows.Media.Animation.HandoffBehavior, System.Boolean)
System.Windows.FrameworkElement.System.Windows.Markup.IQueryAmbient.IsAmbientPropertyAvailable(System.String)
System.Windows.FrameworkElement.GetTemplateChild(System.String)
System.Windows.FrameworkElement.FindResource(System.Object)
System.Windows.FrameworkElement.TryFindResource(System.Object)
System.Windows.FrameworkElement.SetResourceReference(System.Windows.DependencyProperty, System.Object)
System.Windows.FrameworkElement.OnPropertyChanged(System.Windows.DependencyPropertyChangedEventArgs)
System.Windows.FrameworkElement.OnVisualParentChanged(System.Windows.DependencyObject)
System.Windows.FrameworkElement.GetBindingExpression(System.Windows.DependencyProperty)
System.Windows.FrameworkElement.SetBinding(System.Windows.DependencyProperty, System.Windows.Data.BindingBase)
System.Windows.FrameworkElement.SetBinding(System.Windows.DependencyProperty, System.String)
System.Windows.FrameworkElement.GetUIParentCore()
System.Windows.FrameworkElement.BringIntoView()
System.Windows.FrameworkElement.BringIntoView(System.Windows.Rect)
System.Windows.FrameworkElement.GetFlowDirection(System.Windows.DependencyObject)
System.Windows.FrameworkElement.SetFlowDirection(System.Windows.DependencyObject, System.Windows.FlowDirection)
System.Windows.FrameworkElement.MeasureCore(System.Windows.Size)
System.Windows.FrameworkElement.ArrangeCore(System.Windows.Rect)
System.Windows.FrameworkElement.OnRenderSizeChanged(System.Windows.SizeChangedInfo)
System.Windows.FrameworkElement.GetLayoutClip(System.Windows.Size)
System.Windows.FrameworkElement.MoveFocus(System.Windows.Input.TraversalRequest)
System.Windows.FrameworkElement.PredictFocus(System.Windows.Input.FocusNavigationDirection)
System.Windows.FrameworkElement.OnGotFocus(System.Windows.RoutedEventArgs)
System.Windows.FrameworkElement.BeginInit()
System.Windows.FrameworkElement.EndInit()
System.Windows.FrameworkElement.OnToolTipOpening(System.Windows.Controls.ToolTipEventArgs)
System.Windows.FrameworkElement.OnToolTipClosing(System.Windows.Controls.ToolTipEventArgs)
System.Windows.FrameworkElement.OnContextMenuOpening(System.Windows.Controls.ContextMenuEventArgs)
System.Windows.FrameworkElement.OnContextMenuClosing(System.Windows.Controls.ContextMenuEventArgs)
System.Windows.FrameworkElement.RegisterName(System.String, System.Object)
System.Windows.FrameworkElement.UnregisterName(System.String)
System.Windows.FrameworkElement.FindName(System.String)
System.Windows.FrameworkElement.UpdateDefaultStyle()
System.Windows.FrameworkElement.AddLogicalChild(System.Object)
System.Windows.FrameworkElement.RemoveLogicalChild(System.Object)
System.Windows.FrameworkElement.Style
System.Windows.FrameworkElement.OverridesDefaultStyle
System.Windows.FrameworkElement.UseLayoutRounding
System.Windows.FrameworkElement.DefaultStyleKey
System.Windows.FrameworkElement.Triggers
System.Windows.FrameworkElement.TemplatedParent
System.Windows.FrameworkElement.Resources
System.Windows.FrameworkElement.InheritanceBehavior
System.Windows.FrameworkElement.DataContext
System.Windows.FrameworkElement.BindingGroup
System.Windows.FrameworkElement.Language
System.Windows.FrameworkElement.Name

System.Windows.FrameworkElement.Tag
System.Windows.FrameworkElement.InputScope
System.Windows.FrameworkElement.ActualWidth
System.Windows.FrameworkElement.ActualHeight
System.Windows.FrameworkElement.LayoutTransform
System.Windows.FrameworkElement.Width
System.Windows.FrameworkElement.MinWidth
System.Windows.FrameworkElement.MaxWidth
System.Windows.FrameworkElement.Height
System.Windows.FrameworkElement.MinHeight
System.Windows.FrameworkElement.MaxHeight
System.Windows.FrameworkElement.FlowDirection
System.Windows.FrameworkElement.Margin
System.Windows.FrameworkElement.HorizontalAlignment
System.Windows.FrameworkElement.VerticalAlignment
System.Windows.FrameworkElement.FocusVisualStyle
System.Windows.FrameworkElement.Cursor
System.Windows.FrameworkElement.ForceCursor
System.Windows.FrameworkElement.IsInitialized
System.Windows.FrameworkElement.IsLoaded
System.Windows.FrameworkElement.ToolTip
System.Windows.FrameworkElement.ContextMenu
System.Windows.FrameworkElement.Parent
System.Windows.FrameworkElement.TargetUpdated
System.Windows.FrameworkElement.SourceUpdated
System.Windows.FrameworkElement.DataContextChanged
System.Windows.FrameworkElement.RequestBringIntoView
System.Windows.FrameworkElement.SizeChanged
System.Windows.FrameworkElement.Initialized
System.Windows.FrameworkElement.Loaded
System.Windows.FrameworkElement.Unloaded
System.Windows.FrameworkElement.ToolTipOpening
System.Windows.FrameworkElement.ToolTipClosing
System.Windows.FrameworkElement.ContextMenuOpening
System.Windows.FrameworkElement.ContextMenuClosing
System.Windows.UIElement.PreviewMouseDownEvent
System.Windows.UIElement.MouseDownEvent
System.Windows.UIElement.PreviewMouseUpEvent
System.Windows.UIElement.MouseUpEvent
System.Windows.UIElement.PreviewMouseLeftButtonDownEvent
System.Windows.UIElement.MouseLeftButtonDownEvent
System.Windows.UIElement.PreviewMouseLeftButtonUpEvent
System.Windows.UIElement.MouseLeftButtonUpEvent
System.Windows.UIElement.PreviewMouseRightButtonDownEvent
System.Windows.UIElement.MouseRightButtonDownEvent
System.Windows.UIElement.PreviewMouseRightButtonUpEvent
System.Windows.UIElement.MouseRightButtonUpEvent
System.Windows.UIElement.PreviewMouseMoveEvent
System.Windows.UIElement.MouseMoveEvent
System.Windows.UIElement.PreviewMouseWheelEvent
System.Windows.UIElement.MouseWheelEvent
System.Windows.UIElement.MouseEnterEvent

System.Windows.UIElement.MouseLeaveEvent
System.Windows.UIElement.GotMouseCaptureEvent
System.Windows.UIElement.LostMouseCaptureEvent
System.Windows.UIElement.QueryCursorEvent
System.Windows.UIElement.PreviewStylusDownEvent
System.Windows.UIElement.StylusDownEvent
System.Windows.UIElement.PreviewStylusUpEvent
System.Windows.UIElement.StylusUpEvent
System.Windows.UIElement.PreviewStylusMoveEvent
System.Windows.UIElement.StylusMoveEvent
System.Windows.UIElement.PreviewStylusInAirMoveEvent
System.Windows.UIElement.StylusInAirMoveEvent
System.Windows.UIElement.StylusEnterEvent
System.Windows.UIElement.StylusLeaveEvent
System.Windows.UIElement.PreviewStylusInRangeEvent
System.Windows.UIElement.StylusInRangeEvent
System.Windows.UIElement.PreviewStylusOutOfRangeEvent
System.Windows.UIElement.StylusOutOfRangeEvent
System.Windows.UIElement.PreviewStylusSystemGestureEvent
System.Windows.UIElement.StylusSystemGestureEvent
System.Windows.UIElement.GotStylusCaptureEvent
System.Windows.UIElement.LostStylusCaptureEvent
System.Windows.UIElement.StylusButtonDownEvent
System.Windows.UIElement.StylusButtonUpEvent
System.Windows.UIElement.PreviewStylusButtonDownEvent
System.Windows.UIElement.PreviewStylusButtonUpEvent
System.Windows.UIElement.PreviewKeyDownEvent
System.Windows.UIElement.KeyDownEvent
System.Windows.UIElement.PreviewKeyUpEvent
System.Windows.UIElement.KeyUpEvent
System.Windows.UIElement.PreviewGotKeyboardFocusEvent
System.Windows.UIElement.GotKeyboardFocusEvent
System.Windows.UIElement.PreviewLostKeyboardFocusEvent
System.Windows.UIElement.LostKeyboardFocusEvent
System.Windows.UIElement.PreviewTextInputEvent
System.Windows.UIElement.TextInputEvent
System.Windows.UIElement.PreviewQueryContinueDragEvent
System.Windows.UIElement.QueryContinueDragEvent
System.Windows.UIElement.PreviewGiveFeedbackEvent
System.Windows.UIElement.GiveFeedbackEvent
System.Windows.UIElement.PreviewDragEnterEvent
System.Windows.UIElement.DragEnterEvent
System.Windows.UIElement.PreviewDragOverEvent
System.Windows.UIElement.DragOverEvent
System.Windows.UIElement.PreviewDragLeaveEvent
System.Windows.UIElement.DragLeaveEvent
System.Windows.UIElement.PreviewDropEvent
System.Windows.UIElement.DropEvent
System.Windows.UIElement.PreviewTouchDownEvent
System.Windows.UIElement.TouchDownEvent
System.Windows.UIElement.PreviewTouchMoveEvent
System.Windows.UIElement.TouchMoveEvent

System.Windows.UIElement.PreviewTouchUpEvent
System.Windows.UIElement.TouchUpEvent
System.Windows.UIElement.GotTouchCaptureEvent
System.Windows.UIElement.LostTouchCaptureEvent
System.Windows.UIElement.TouchEnterEvent
System.Windows.UIElement.TouchLeaveEvent
System.Windows.UIElement.IsMouseDirectlyOverProperty
System.Windows.UIElement.IsMouseOverProperty
System.Windows.UIElement.IsStylusOverProperty
System.Windows.UIElement.IsKeyboardFocusWithinProperty
System.Windows.UIElement.IsMouseCapturedProperty
System.Windows.UIElement.IsMouseCaptureWithinProperty
System.Windows.UIElement.IsStylusDirectlyOverProperty
System.Windows.UIElement.IsStylusCapturedProperty
System.Windows.UIElement.IsStylusCaptureWithinProperty
System.Windows.UIElement.IsKeyboardFocusedProperty
System.Windows.UIElement.AreAnyTouchesDirectlyOverProperty
System.Windows.UIElement.AreAnyTouchesOverProperty
System.Windows.UIElement.AreAnyTouchesCapturedProperty
System.Windows.UIElement.AreAnyTouchesCapturedWithinProperty
System.Windows.UIElement.AllowDropProperty
System.Windows.UIElement.RenderTransformProperty
System.Windows.UIElement.RenderTransformOriginProperty
System.Windows.UIElement.OpacityProperty
System.Windows.UIElement.OpacityMaskProperty
System.Windows.UIElement.BitmapEffectProperty
System.Windows.UIElement.EffectProperty
System.Windows.UIElement.BitmapEffectInputProperty
System.Windows.UIElement.CacheModeProperty
System.Windows.UIElement.UidProperty
System.Windows.UIElement.VisibilityProperty
System.Windows.UIElement.ClipToBoundsProperty
System.Windows.UIElement.ClipProperty
System.Windows.UIElement.SnapsToDevicePixelsProperty
System.Windows.UIElement.GotFocusEvent
System.Windows.UIElement.LostFocusEvent
System.Windows.UIElement.IsFocusedProperty
System.Windows.UIElement.IsEnabledProperty
System.Windows.UIElement.IsHitTestVisibleProperty
System.Windows.UIElement.IsVisibleProperty
System.Windows.UIElement.FocusableProperty
System.Windows.UIElement.IsManipulationEnabledProperty
System.Windows.UIElement.ManipulationStartingEvent
System.Windows.UIElement.ManipulationStartedEvent
System.Windows.UIElement.ManipulationDeltaEvent
System.Windows.UIElement.ManipulationInertiaStartingEvent
System.Windows.UIElement.ManipulationBoundaryFeedbackEvent
System.Windows.UIElement.ManipulationCompletedEvent
System.Windows.UIElement.ApplyAnimationClock(System.Windows.DependencyProperty, System.Windows.Media.Animation.AnimationClock)
System.Windows.UIElement.ApplyAnimationClock(System.Windows.DependencyProperty, System.Windows.Media.Animation.AnimationClock, System.Windows.Media.Animation.HandoffBehavior)

System.Windows.UIElement.BeginAnimation(System.Windows.DependencyProperty, System.Windows.Media.Animation.AnimationTimeline)
System.Windows.UIElement.BeginAnimation(System.Windows.DependencyProperty, System.Windows.Media.Animation.AnimationTimeline, System.Windows.Media.Animation.HandoffBehavior)
System.Windows.UIElement.GetAnimationBaseValue(System.Windows.DependencyProperty)
System.Windows.UIElement.RaiseEvent(System.Windows.RoutedEventArgs)
System.Windows.UIElement.AddHandler(System.Windows.RoutedEvent, System.Delegate)
System.Windows.UIElement.AddHandler(System.Windows.RoutedEvent, System.Delegate, System.Boolean)
System.Windows.UIElement.RemoveHandler(System.Windows.RoutedEvent, System.Delegate)
System.Windows.UIElement.AddToEventRoute(System.Windows.EventRoute, System.Windows.RoutedEventArgs)
System.Windows.UIElement.OnPreviewMouseDown(System.Windows.Input.MouseButtonEventArgs)
System.Windows.UIElement.OnMouseDown(System.Windows.Input.MouseButtonEventArgs)
System.Windows.UIElement.OnPreviewMouseUp(System.Windows.Input.MouseButtonEventArgs)
System.Windows.UIElement.OnMouseUp(System.Windows.Input.MouseButtonEventArgs)
System.Windows.UIElement.OnPreviewMouseLeftButtonDown(System.Windows.Input.MouseButtonEventArgs)
System.Windows.UIElement.OnPreviewMouseLeftButtonUp(System.Windows.Input.MouseButtonEventArgs)
System.Windows.UIElement.OnPreviewMouseRightButtonDown(System.Windows.Input.MouseButtonEventArgs)
System.Windows.UIElement.OnMouseRightButtonDown(System.Windows.Input.MouseButtonEventArgs)
System.Windows.UIElement.OnPreviewMouseRightButtonUp(System.Windows.Input.MouseButtonEventArgs)
System.Windows.UIElement.OnMouseRightButtonUp(System.Windows.Input.MouseButtonEventArgs)
System.Windows.UIElement.OnPreviewMouseMove(System.Windows.Input.MouseEventArgs)
System.Windows.UIElement.OnMouseEnter(System.Windows.Input.MouseEventArgs)
System.Windows.UIElement.OnMouseLeave(System.Windows.Input.MouseEventArgs)
System.Windows.UIElement.OnGotMouseCapture(System.Windows.Input.MouseEventArgs)
System.Windows.UIElement.OnLostMouseCapture(System.Windows.Input.MouseEventArgs)
System.Windows.UIElement.OnQueryCursor(System.Windows.Input.QueryCursorEventArgs)
System.Windows.UIElement.OnPreviewStylusDown(System.Windows.Input.StylusDownEventArgs)
System.Windows.UIElement.OnStylusDown(System.Windows.Input.StylusDownEventArgs)
System.Windows.UIElement.OnPreviewStylusUp(System.Windows.Input.StylusEventArgs)
System.Windows.UIElement.OnStylusUp(System.Windows.Input.StylusEventArgs)
System.Windows.UIElement.OnPreviewStylusMove(System.Windows.Input.StylusEventArgs)
System.Windows.UIElement.OnStylusMove(System.Windows.Input.StylusEventArgs)
System.Windows.UIElement.OnPreviewStylusInAirMove(System.Windows.Input.StylusEventArgs)
System.Windows.UIElement.OnStylusInAirMove(System.Windows.Input.StylusEventArgs)
System.Windows.UIElement.OnStylusEnter(System.Windows.Input.StylusEventArgs)
System.Windows.UIElement.OnStylusLeave(System.Windows.Input.StylusEventArgs)
System.Windows.UIElement.OnPreviewStylusInRange(System.Windows.Input.StylusEventArgs)
System.Windows.UIElement.OnStylusInRange(System.Windows.Input.StylusEventArgs)
System.Windows.UIElement.OnPreviewStylusOutOfRange(System.Windows.Input.StylusEventArgs)
System.Windows.UIElement.OnStylusOutOfRange(System.Windows.Input.StylusEventArgs)
System.Windows.UIElement.OnPreviewStylusSystemGesture(System.Windows.Input.StylusSystemGestureEventArgs)
System.Windows.UIElement.OnStylusSystemGesture(System.Windows.Input.StylusSystemGestureEventArgs)
System.Windows.UIElement.OnGotStylusCapture(System.Windows.Input.StylusEventArgs)
System.Windows.UIElement.OnLostStylusCapture(System.Windows.Input.StylusEventArgs)
System.Windows.UIElement.OnStylusButtonDown(System.Windows.Input.StylusButtonEventArgs)
System.Windows.UIElement.OnStylusButtonUp(System.Windows.Input.StylusButtonEventArgs)
System.Windows.UIElement.OnPreviewStylusButtonDown(System.Windows.Input.StylusButtonEventArgs)
System.Windows.UIElement.OnPreviewStylusButtonUp(System.Windows.Input.StylusButtonEventArgs)
System.Windows.UIElement.OnPreviewKeyDown(System.Windows.Input.KeyEventArgs)
System.Windows.UIElement.OnKeyDown(System.Windows.Input.KeyEventArgs)
System.Windows.UIElement.OnPreviewKeyUp(System.Windows.Input.KeyEventArgs)
System.Windows.UIElement.OnKeyUp(System.Windows.Input.KeyEventArgs)

System.Windows.UIElement.OnPreviewGotKeyboardFocus(System.Windows.Input.KeyboardFocusChangedEventArgs)
System.Windows.UIElement.OnGotKeyboardFocus(System.Windows.Input.KeyboardFocusChangedEventArgs)
System.Windows.UIElement.OnPreviewLostKeyboardFocus(System.Windows.Input.KeyboardFocusChangedEventArgs)
System.Windows.UIElement.OnLostKeyboardFocus(System.Windows.Input.KeyboardFocusChangedEventArgs)
System.Windows.UIElement.OnPreviewTextInput(System.Windows.Input.TextCompositionEventArgs)
System.Windows.UIElement.OnTextInput(System.Windows.Input.TextCompositionEventArgs)
System.Windows.UIElement.OnPreviewQueryContinueDrag(System.Windows.QueryContinueDragEventArgs)
System.Windows.UIElement.OnQueryContinueDrag(System.Windows.QueryContinueDragEventArgs)
System.Windows.UIElement.OnPreviewGiveFeedback(System.Windows.GiveFeedbackEventArgs)
System.Windows.UIElement.OnGiveFeedback(System.Windows.GiveFeedbackEventArgs)
System.Windows.UIElement.OnPreviewDragEnter(System.Windows.DragEventArgs)
System.Windows.UIElement.OnDragEnter(System.Windows.DragEventArgs)
System.Windows.UIElement.OnPreviewDragOver(System.Windows.DragEventArgs)
System.Windows.UIElement.OnDragOver(System.Windows.DragEventArgs)
System.Windows.UIElement.OnPreviewDragLeave(System.Windows.DragEventArgs)
System.Windows.UIElement.OnDragLeave(System.Windows.DragEventArgs)
System.Windows.UIElement.OnPreviewDrop(System.Windows.DragEventArgs)
System.Windows.UIElement.OnDrop(System.Windows.DragEventArgs)
System.Windows.UIElement.OnPreviewTouchDown(System.Windows.Input.TouchEventArgs)
System.Windows.UIElement.OnTouchDown(System.Windows.Input.TouchEventArgs)
System.Windows.UIElement.OnPreviewTouchMove(System.Windows.Input.TouchEventArgs)
System.Windows.UIElement.OnTouchMove(System.Windows.Input.TouchEventArgs)
System.Windows.UIElement.OnPreviewTouchUp(System.Windows.Input.TouchEventArgs)
System.Windows.UIElement.OnTouchUp(System.Windows.Input.TouchEventArgs)
System.Windows.UIElement.OnGotTouchCapture(System.Windows.Input.TouchEventArgs)
System.Windows.UIElement.OnLostTouchCapture(System.Windows.Input.TouchEventArgs)
System.Windows.UIElement.OnTouchEnter(System.Windows.Input.TouchEventArgs)
System.Windows.UIElement.OnTouchLeave(System.Windows.Input.TouchEventArgs)
System.Windows.UIElement.OnIsMouseDirectlyOverChanged(System.Windows.DependencyPropertyChangedEventArgs)
System.Windows.UIElement.OnIsKeyboardFocusWithinChanged(System.Windows.DependencyPropertyChangedEventArgs)
System.Windows.UIElement.OnIsMouseCapturedChanged(System.Windows.DependencyPropertyChangedEventArgs)
System.Windows.UIElement.OnIsMouseCaptureWithinChanged(System.Windows.DependencyPropertyChangedEventArgs)
System.Windows.UIElement.OnIsStylusDirectlyOverChanged(System.Windows.DependencyPropertyChangedEventArgs)
System.Windows.UIElement.OnIsStylusCapturedChanged(System.Windows.DependencyPropertyChangedEventArgs)
System.Windows.UIElement.OnIsStylusCaptureWithinChanged(System.Windows.DependencyPropertyChangedEventArgs)
System.Windows.UIElement.OnIsKeyboardFocusedChanged(System.Windows.DependencyPropertyChangedEventArgs)
System.Windows.UIElement.InvalidateMeasure()
System.Windows.UIElement.InvalidateArrange()
System.Windows.UIElement.InvalidateVisual()
System.Windows.UIElement.OnChildDesiredSizeChanged(System.Windows.UIElement)
System.Windows.UIElement.Measure(System.Windows.Size)
System.Windows.UIElement.Arrange(System.Windows.Rect)
System.Windows.UIElement.UpdateLayout()
System.Windows.UIElement.TranslatePoint(System.Windows.Point, System.Windows.UIElement)
System.Windows.UIElement.InputHitTest(System.Windows.Point)
System.Windows.UIElement.CaptureMouse()
System.Windows.UIElement.ReleaseMouseCapture()
System.Windows.UIElement.CaptureStylus()
System.Windows.UIElement.ReleaseStylusCapture()
System.Windows.UIElement.Focus()
System.Windows.UIElement.OnAccessKey(System.Windows.Input.AccessKeyEventArgs)
System.Windows.UIElement.HitTestCore(System.Windows.Media.PointHitTestParameters)

System.Windows.UIElement.HitTestCore(System.Windows.Media.GeometryHitTestParameters)
System.Windows.UIElement.OnLostFocus(System.Windows.RoutedEventArgs)
System.Windows.UIElement.OnCreateAutomationPeer()
System.Windows.UIElement.OnManipulationStarting(System.Windows.Input.ManipulationStartingEventArgs)
System.Windows.UIElement.OnManipulationStarted(System.Windows.Input.ManipulationStartedEventArgs)
System.Windows.UIElement.OnManipulationDelta(System.Windows.Input.ManipulationDeltaEventArgs)
System.Windows.UIElement.OnManipulationInertiaStarting(System.Windows.Input.ManipulationInertiaStartingEventArgs)
System.Windows.UIElement.OnManipulationBoundaryFeedback(System.Windows.Input.ManipulationBoundaryFeedbackEventArgs)
System.Windows.UIElement.OnManipulationCompleted(System.Windows.Input.ManipulationCompletedEventArgs)
System.Windows.UIElement.CaptureTouch(System.Windows.Input.TouchDevice)
System.Windows.UIElement.ReleaseTouchCapture(System.Windows.Input.TouchDevice)
System.Windows.UIElement.ReleaseAllTouchCaptures()
System.Windows.UIElement.HasAnimatedProperties
System.Windows.UIElement.InputBindings
System.Windows.UIElement.CommandBindings
System.Windows.UIElement.AllowDrop
System.Windows.UIElement.StylusPlugIns
System.Windows.UIElement.DesiredSize
System.Windows.UIElement.IsMeasureValid
System.Windows.UIElement.IsArrangeValid
System.Windows.UIElement.RenderSize
System.Windows.UIElement.RenderTransform
System.Windows.UIElement.RenderTransformOrigin
System.Windows.UIElement.IsMouseDirectlyOver
System.Windows.UIElement.IsMouseOver
System.Windows.UIElement.IsStylusOver
System.Windows.UIElement.IsKeyboardFocusWithin
System.Windows.UIElement.IsMouseCaptured
System.Windows.UIElement.IsMouseCaptureWithin
System.Windows.UIElement.IsStylusDirectlyOver
System.Windows.UIElement.IsStylusCaptured
System.Windows.UIElement.IsStylusCaptureWithin
System.Windows.UIElement.IsKeyboardFocused
System.Windows.UIElement.IsInputMethodEnabled
System.Windows.UIElement.Opacity
System.Windows.UIElement.OpacityMask
System.Windows.UIElement.BitmapEffect
System.Windows.UIElement.Effect
System.Windows.UIElement.BitmapEffectInput
System.Windows.UIElement.CacheMode
System.Windows.UIElement.Uid
System.Windows.UIElement.Visibility
System.Windows.UIElement.ClipToBounds
System.Windows.UIElement.Clip
System.Windows.UIElement.SnapsToDevicePixels
System.Windows.UIElement.HasEffectiveKeyboardFocus
System.Windows.UIElement.IsFocused
System.Windows.UIElement.IsEnabled
System.Windows.UIElement.IsEnabledCore
System.Windows.UIElement.IsHitTestVisible
System.Windows.UIElement.IsVisible
System.Windows.UIElement.Focusable

System.Windows.UIElement.PersistId
System.Windows.UIElement.IsManipulationEnabled
System.Windows.UIElement.AreAnyTouchesOver
System.Windows.UIElement.AreAnyTouchesDirectlyOver
System.Windows.UIElement.AreAnyTouchesCapturedWithin
System.Windows.UIElement.AreAnyTouchesCaptured
System.Windows.UIElement.TouchesCaptured
System.Windows.UIElement.TouchesCapturedWithin
System.Windows.UIElement.TouchesOver
System.Windows.UIElement.TouchesDirectlyOver
System.Windows.UIElement.PreviewMouseDown
System.Windows.UIElement.MouseDown
System.Windows.UIElement.PreviewMouseUp
System.Windows.UIElement.MouseUp
System.Windows.UIElement.PreviewMouseLeftButtonDown
System.Windows.UIElement.MouseLeftButtonDown
System.Windows.UIElement.PreviewMouseLeftButtonUp
System.Windows.UIElement.MouseLeftButtonUp
System.Windows.UIElement.PreviewMouseRightButtonDown
System.Windows.UIElement.MouseRightButtonDown
System.Windows.UIElement.PreviewMouseRightButtonUp
System.Windows.UIElement.MouseRightButtonUp
System.Windows.UIElement.PreviewMouseMove
System.Windows.UIElement.MouseMove
System.Windows.UIElement.PreviewMouseWheel
System.Windows.UIElement.MouseWheel
System.Windows.UIElement.MouseEnter
System.Windows.UIElement.MouseLeave
System.Windows.UIElement.GotMouseCapture
System.Windows.UIElement.LostMouseCapture
System.Windows.UIElement.QueryCursor
System.Windows.UIElement.PreviewStylusDown
System.Windows.UIElement.StylusDown
System.Windows.UIElement.PreviewStylusUp
System.Windows.UIElement.StylusUp
System.Windows.UIElement.PreviewStylusMove
System.Windows.UIElement.StylusMove
System.Windows.UIElement.PreviewStylusInAirMove
System.Windows.UIElement.StylusInAirMove
System.Windows.UIElement.StylusEnter
System.Windows.UIElement.StylusLeave
System.Windows.UIElement.PreviewStylusInRange
System.Windows.UIElement.StylusInRange
System.Windows.UIElement.PreviewStylusOutOfRange
System.Windows.UIElement.StylusOutOfRange
System.Windows.UIElement.PreviewStylusSystemGesture
System.Windows.UIElement.StylusSystemGesture
System.Windows.UIElement.GotStylusCapture
System.Windows.UIElement.LostStylusCapture
System.Windows.UIElement.StylusButtonDown
System.Windows.UIElement.StylusButtonUp
System.Windows.UIElement.PreviewStylusButtonDown

System.Windows.UIElement.PreviewStylusButtonUp
System.Windows.UIElement.PreviewKeyDown
System.Windows.UIElement.KeyDown
System.Windows.UIElement.PreviewKeyUp
System.Windows.UIElement.KeyUp
System.Windows.UIElement.PreviewGotKeyboardFocus
System.Windows.UIElement.GotKeyboardFocus
System.Windows.UIElement.PreviewLostKeyboardFocus
System.Windows.UIElement.LostKeyboardFocus
System.Windows.UIElement.PreviewTextInput
System.Windows.UIElement.TextInput
System.Windows.UIElement.PreviewQueryContinueDrag
System.Windows.UIElement.QueryContinueDrag
System.Windows.UIElement.PreviewGiveFeedback
System.Windows.UIElement.GiveFeedback
System.Windows.UIElement.PreviewDragEnter
System.Windows.UIElement.DragEnter
System.Windows.UIElement.PreviewDragOver
System.Windows.UIElement.DragOver
System.Windows.UIElement.PreviewDragLeave
System.Windows.UIElement.DragLeave
System.Windows.UIElement.PreviewDrop
System.Windows.UIElement.Drop
System.Windows.UIElement.PreviewTouchDown
System.Windows.UIElement.TouchDown
System.Windows.UIElement.PreviewTouchMove
System.Windows.UIElement.TouchMove
System.Windows.UIElement.PreviewTouchUp
System.Windows.UIElement.TouchUp
System.Windows.UIElement.GotTouchCapture
System.Windows.UIElement.LostTouchCapture
System.Windows.UIElement.TouchEnter
System.Windows.UIElement.TouchLeave
System.Windows.UIElement.IsMouseDirectlyOverChanged
System.Windows.UIElement.IsKeyboardFocusWithinChanged
System.Windows.UIElement.IsMouseCapturedChanged
System.Windows.UIElement.IsMouseCaptureWithinChanged
System.Windows.UIElement.IsStylusDirectlyOverChanged
System.Windows.UIElement.IsStylusCapturedChanged
System.Windows.UIElement.IsStylusCaptureWithinChanged
System.Windows.UIElement.IsKeyboardFocusedChanged
System.Windows.UIElement.LayoutUpdated
System.Windows.UIElement.GotFocus
System.Windows.UIElement.LostFocus
System.Windows.UIElement.IsEnabledChanged
System.Windows.UIElement.IsHitTestVisibleChanged
System.Windows.UIElement.IsVisibleChanged
System.Windows.UIElement.FocusableChanged
System.Windows.UIElement.ManipulationStarting
System.Windows.UIElement.ManipulationStarted
System.Windows.UIElement.ManipulationDelta
System.Windows.UIElement.ManipulationInertiaStarting

System.Windows.UIElement.ManipulationBoundaryFeedback
System.Windows.UIElement.ManipulationCompleted
System.Windows.Media.Visual.AddVisualChild(System.Windows.Media.Visual)
System.Windows.Media.Visual.RemoveVisualChild(System.Windows.Media.Visual)
System.Windows.Media.Visual.OnDpiChanged(System.Windows.DpiScale, System.Windows.DpiScale)
System.Windows.Media.Visual.IsAncestorOf(System.Windows.DependencyObject)
System.Windows.Media.Visual.IsDescendantOf(System.Windows.DependencyObject)
System.Windows.Media.Visual.FindCommonVisualAncestor(System.Windows.DependencyObject)
System.Windows.Media.Visual.TransformToAncestor(System.Windows.Media.Visual)
System.Windows.Media.Visual.TransformToAncestor(System.Windows.Media.Media3D.Visual3D)
System.Windows.Media.Visual.TransformToDescendant(System.Windows.Media.Visual)
System.Windows.Media.Visual.TransformToVisual(System.Windows.Media.Visual)
System.Windows.Media.Visual.PointToScreen(System.Windows.Point)
System.Windows.Media.Visual.PointFromScreen(System.Windows.Point)
System.Windows.Media.Visual.VisualParent
System.Windows.Media.Visual.VisualTransform
System.Windows.Media.Visual.VisualEffect
System.Windows.Media.Visual.VisualBitmapEffect
System.Windows.Media.Visual.VisualBitmapEffectInput
System.Windows.Media.Visual.VisualCacheMode
System.Windows.Media.Visual.VisualScrollableAreaClip
System.Windows.Media.Visual.VisualClip
System.Windows.Media.Visual.VisualOffset
System.Windows.Media.Visual.VisualOpacity
System.Windows.Media.Visual.VisualEdgeMode
System.Windows.Media.Visual.VisualBitmapScalingMode
System.Windows.Media.Visual.VisualClearTypeHint
System.Windows.Media.Visual.VisualTextRenderingMode
System.Windows.Media.Visual.VisualTextHintingMode
System.Windows.Media.Visual.VisualOpacityMask
System.Windows.Media.Visual.VisualXSnappingGuidelines
System.Windows.Media.Visual.VisualYSnappingGuidelines
System.Windows.DependencyObject.Equals(System.Object)
System.Windows.DependencyObject.GetHashCode()
System.Windows.DependencyObject.GetValue(System.Windows.DependencyProperty)
System.Windows.DependencyObject.SetValue(System.Windows.DependencyProperty, System.Object)
System.Windows.DependencyObject.SetCurrentValue(System.Windows.DependencyProperty, System.Object)
System.Windows.DependencyObject.SetValue(System.Windows.DependencyPropertyKey, System.Object)
System.Windows.DependencyObject.ClearValue(System.Windows.DependencyProperty)
System.Windows.DependencyObject.ClearValue(System.Windows.DependencyPropertyKey)
System.Windows.DependencyObject.CoerceValue(System.Windows.DependencyProperty)
System.Windows.DependencyObject.InvalidateProperty(System.Windows.DependencyProperty)
System.Windows.DependencyObject.ShouldSerializeProperty(System.Windows.DependencyProperty)
System.Windows.DependencyObject.ReadLocalValue(System.Windows.DependencyProperty)
System.Windows.DependencyObject.GetLocalValueEnumerator()
System.Windows.DependencyObject.DependencyObjectType
System.Windows.DependencyObject.IsSealed
System.Windows.Threading.DispatcherObject.Dispatcher
System.Object.ToString()
System.Object.Equals(System.Object, System.Object)
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.GetType()

System.Object.MemberwiseClone()

Namespace: SigStat.UI

Assembly: SigStat.UI.dll

Syntax

```
public class SignatureVisualizer : Grid, DUCE.IResource, IAnimatable, IFrameworkInputElement, IInputElement,
ISupportInitialize, IHaveResources, IQueryAmbient, IAddChild
```

Constructors

SignatureVisualizer()

Declaration

```
public SignatureVisualizer()
```

Fields

DisplayModeProperty

Declaration

```
public static readonly DependencyProperty DisplayModeProperty
```

Field Value

TYPE	DESCRIPTION
System.Windows.DependencyProperty	

ShowAxesProperty

Declaration

```
public static readonly DependencyProperty ShowAxesProperty
```

Field Value

TYPE	DESCRIPTION
System.Windows.DependencyProperty	

SignatureProperty

Declaration

```
public static readonly DependencyProperty SignatureProperty
```

Field Value

TYPE	DESCRIPTION
System.Windows.DependencyProperty	

Properties

DisplayMode

Declaration

```
public DisplayMode DisplayMode { get; set; }
```

Property Value

TYPE	DESCRIPTION
DisplayMode	

ShowAxes

Declaration

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

Property Value

TYPE	DESCRIPTION
System.Boolean	

Signature

Declaration

```
public Signature Signature { get; set; }
```

Property Value

TYPE	DESCRIPTION
Signature	

Methods

OnInitialized(EventArgs)

Declaration

```
protected override void OnInitialized(EventArgs e)
```

Parameters

TYPE	NAME	DESCRIPTION
System.EventArgs	e	

Overrides

System.Windows.FrameworkElement.OnInitialized(System.EventArgs)

OnMouseLeftButtonDown(MouseButtonEventArgs)

Declaration

```
protected override void OnMouseLeftButtonDown(MouseButtonEventArgs e)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Windows.Input.MouseButtonEventArgs	e	

Overrides

System.Windows.UIElement.OnMouseLeftButtonDown(System.Windows.Input.MouseButtonEventArgs)

OnMouseLeftButtonUp(MouseButtonEventArgs)

Declaration

```
protected override void OnMouseLeftButtonUp(MouseButtonEventArgs e)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Windows.Input.MouseButtonEventArgs	e	

Overrides

System.Windows.UIElement.OnMouseLeftButtonUp(System.Windows.Input.MouseButtonEventArgs)

OnMouseMove(MouseEventArgs)

Declaration

```
protected override void OnMouseMove(MouseEventArgs e)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Windows.Input.MouseEventArgs	e	

Overrides

System.Windows.UIElement.OnMouseMove(System.Windows.Input.MouseEventArgs)

OnMouseWheel(MouseWheelEventArgs)

Declaration

```
protected override void OnMouseWheel(MouseWheelEventArgs e)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Windows.Input.MouseWheelEventArgs	e	

Overrides

System.Windows.UIElement.OnMouseWheel(System.Windows.Input.MouseWheelEventArgs)

OnPreviewMouseWheel(MouseWheelEventArgs)

Declaration

```
protected override void OnPreviewMouseWheel(MouseWheelEventArgs e)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Windows.Input.MouseWheelEventArgs	e	

Overrides

System.Windows.UIElement.OnPreviewMouseWheel(System.Windows.Input.MouseWheelEventArgs)

Implements

System.Windows.Media.Animation.IAnimatable
System.Windows.IFrameworkInputElement

System.Windows.IInputElement
System.ComponentModel.ISupportInitialize
System.Windows.Markup.IQueryAmbient
System.Windows.Markup.IAddChild