



# Hoot.Standard Reference

## Table of Contents

Hoot.Standard Reference.....	6
RaptorDB Namespace .....	7
Global Class .....	7
CompressBitmapBytes Field.....	7
CompressDocumentOverKiloBytes Field.....	7
DefaultStringKeySize Field.....	8
FlushStorageFileImmediately Field .....	8
FreeBitmapMemoryOnSave Field .....	8
HighFrequencyKVDiskBlockSize Field .....	8
PageItemCount Field .....	9
SaveAsBinaryJSON Field .....	9
SaveIndexToDiskTimerSeconds Field .....	9
SplitStorageFilesMegaBytes Field.....	9
UseLessMemoryStructures Field.....	10
Hoot Class.....	10
Hoot(String, String, Boolean) Constructor .....	10
Hoot(String, String, Boolean, ITokenizer) Constructor.....	11
Hoot(IHootConfig) Constructor .....	11
Hoot(IHootConfig, ITokenizer) Constructor .....	12
Hoot.DocumentCount Property .....	12
Hoot.HootConfOptions Property.....	12
Hoot.WordCount Property .....	13
Hoot.Words Property .....	13
Hoot.Fetch<T> Method .....	13
Hoot.FindDocumentFileNames Method .....	14
Hoot.FindDocuments<T> Method.....	14
Hoot.FindRows Method .....	14
Hoot.FreeMemory Method .....	15
Hoot.Index (Int32, String, IHootFilter) Method.....	15
Hoot.Index (Document, Boolean, IHootFilter) Method .....	15
Hoot.Index (Int32, String) Method .....	16
Hoot.Index (Document, Boolean) Method.....	16

Hoot.IsIndexed Method .....	17
Hoot.OptimizeIndex Method .....	17
Hoot.Query Method .....	17
Hoot.RemoveDocument (Int32) Method .....	18
Hoot.RemoveDocument (String) Method .....	18
Hoot.Save Method .....	18
Hoot.Shutdown Method .....	19
HootConfig Class.....	19
HootConfig Constructor .....	19
HootConfig.DocMode Property.....	19
HootConfig.FileName Property .....	20
HootConfig.IgnoreNumerics Property.....	20
HootConfig.IndexPath Property .....	20
HootConfig.UseStopList Property.....	21
tokenizer Class.....	21
tokenizer.GenerateWordFreq Method .....	21
tokenizer.InitializeStopList Method .....	22
IHootConfig Interface .....	22
IHootConfig.DocMode Property.....	22
IHootConfig.FileName Property .....	23
IHootConfig.IgnoreNumerics Property.....	23
IHootConfig.IndexPath Property .....	23
IHootConfig.UseStopList Property.....	23
IHootFilter Interface .....	24
IHootFilter.FilterText Method .....	24
IHootFilter.InitializeFilter Method.....	24
ITokenizer Interface.....	25
ITokenizer.GenerateWordFreq Method.....	25
ITokenizer.InitializeStopList Method.....	25
RaptorDB.Common Namespace.....	27
FastDateTime Class.....	27
FastDateTime.Now Property .....	27
LocalUtcOffset Field .....	27
Helper Class .....	28

Helper.CompareMemCmp Method .....	28
Helper.GetBytes (String) Method.....	28
Helper.GetBytes (Int16, Boolean) Method .....	29
Helper.GetBytes (Int32, Boolean) Method .....	29
Helper.GetBytes (Int64, Boolean) Method .....	29
Helper.GetString Method.....	30
Helper.ToInt16 (Byte(), Int32) Method .....	30
Helper.ToInt16 (Byte(), Int32, Boolean) Method .....	31
Helper.ToInt32 (Byte(), Int32, Boolean) Method .....	31
Helper.ToInt32 (Byte(), Int32) Method .....	32
Helper.ToInt64 (Byte(), Int32, Boolean) Method .....	32
Helper.ToInt64 (Byte(), Int32) Method .....	32
MurMur Field .....	33
SafeDictionary<TKey, TValue> Class.....	33
SafeDictionary<TKey, TValue> Constructor .....	34
SafeDictionary<TKey, TValue>(Int32) Constructor.....	34
SafeDictionary<TKey, TValue>.Item Property .....	34
SafeDictionary<TKey, TValue>.Add Method .....	34
SafeDictionary<TKey, TValue>.Clear Method .....	35
SafeDictionary<TKey, TValue>.Count Method .....	35
SafeDictionary<TKey, TValue>.GetEnumerator Method .....	35
SafeDictionary<TKey, TValue>.GetValue Method.....	36
SafeDictionary<TKey, TValue>.Keys Method .....	36
SafeDictionary<TKey, TValue>.Remove Method .....	36
SafeDictionary<TKey, TValue>.TryGetValue Method.....	37
SafeSortedList<T, V> Class.....	37
SafeSortedList<T, V>.Item Property .....	38
SafeSortedList<T, V>.Add Method .....	38
SafeSortedList<T, V>.Clear Method .....	38
SafeSortedList<T, V>.Count Method .....	39
SafeSortedList<T, V>.GetEnumerator Method.....	39
SafeSortedList<T, V>.GetKey Method .....	39
SafeSortedList<T, V>.GetValue (Int32) Method .....	40
SafeSortedList<T, V>.GetValue (T) Method.....	40

SafeSortedList<T, V>.Keys Method .....	40
SafeSortedList<T, V>.Remove Method.....	41
SafeSortedList<T, V>.TryGetValue Method.....	41
IKV<T, V> Interface .....	41
IKV<T, V>.Add Method .....	42
IKV<T, V>.Clear Method .....	42
IKV<T, V>.Count Method.....	42
IKV<T, V>.GetEnumerator Method .....	43
IKV<T, V>.GetValue Method.....	43
IKV<T, V>.Keys Method .....	43
IKV<T, V>.Remove Method .....	43
IKV<T, V>.TryGetValue Method .....	44
RaptorDB.Filters Namespace.....	45
HtmlFilter Class.....	45
HtmlFilter.FilterText Method .....	45
HtmlFilter.InitializeFilter Method.....	45
NoFilter Class .....	46
NoFilter.FilterText Method.....	46
NoFilter.InitializeFilter Method .....	47
hOOt Namespace .....	48
Document Class .....	48
Document(FileInfo, String) Constructor .....	48
Document Constructor.....	49
Document.DocNumber Property .....	49
Document.FileName Property.....	49
Document.ModifiedDate Property.....	49
Document.Text Property .....	49
Document.ToString Method.....	50
FileSize Field .....	50
Index .....	51

## Hoot.Standard Reference

### Namespaces

[RaptorDB](#)<sup>7</sup>, [RaptorDB.Common](#)<sup>27</sup>, [RaptorDB.Filters](#)<sup>45</sup>, [hOOT](#)<sup>48</sup>

## RaptorDB Namespace

### Classes

[Global](#)<sub>7</sub>, [Hoot](#)<sub>10</sub>, [HootConfig](#)<sub>19</sub>, [tokenizer](#)<sub>21</sub>

### Interfaces

[IHootConfig](#)<sub>22</sub>, [IHootFilter](#)<sub>24</sub>, [ITokenizer](#)<sub>25</sub>

## Global Class

**C#**

```
public class Global
```

### Requirements

**Namespace:** [RaptorDB](#)<sub>7</sub>

**Assembly:** Hoot.Standard (in Hoot.Standard.dll)

### Fields

[CompressBitmapBytes](#)<sub>7</sub>, [CompressDocumentOverKiloBytes](#)<sub>7</sub>, [DefaultStringKeySize](#)<sub>8</sub>,  
[FlushStorageFileImmediately](#)<sub>8</sub>, [FreeBitmapMemoryOnSave](#)<sub>8</sub>, [HighFrequencyKVDiskBlockSize](#)<sub>8</sub>,  
[PageItemCount](#)<sub>9</sub>, [SaveAsBinaryJSON](#)<sub>9</sub>, [SaveIndexToDiskTimerSeconds](#)<sub>9</sub>, [SplitStorageFilesMegaBytes](#)<sub>9</sub>,  
[UseLessMemoryStructures](#)<sub>10</sub>

## CompressBitmapBytes Field

**C#**

```
public static bool CompressBitmapBytes
```

### See Also

Applies to: [Global](#)<sub>7</sub>

## CompressDocumentOverKiloBytes Field

Compress the documents in the storage file if it is over this size (default = 100 Kilobytes)

- You will be trading CPU for disk IO

**C#**

```
public static UInt16 CompressDocumentOverKiloBytes
```

### See Also

Applies to: [Global](#)<sub>7</sub>

## DefaultStringKeySize Field

Default maximum string key size for indexes

**C#**

```
public static byte DefaultStringKeySize
```

### See Also

Applies to: [Global](#)<sub>7</sub>

## FlushStorageFileImmediately Field

Flush the StorageFile stream immediately

**C#**

```
public static bool FlushStorageFileImmediately
```

### See Also

Applies to: [Global](#)<sub>7</sub>

## FreeBitmapMemoryOnSave Field

Free bitmap index memory on save

**C#**

```
public static bool FreeBitmapMemoryOnSave
```

### See Also

Applies to: [Global](#)<sub>7</sub>

## HighFrequencyKVDiskBlockSize Field

Disk block size for high frequency KV storage file (default = 2048)

\* Do not use anything under 512 with large string keys

**C#**

```
public static UInt16 HighFrequencyKVDiskBlockSize
```

### See Also

Applies to: [Global](#)<sub>7</sub>



## PageItemCount Field

Number of items in each index page (default = 10000) [Expert only, do not change]

**C#**

```
public static UInt16 PageItemCount
```

### See Also

Applies to: [Global](#)<sub>7</sub>

## SaveAsBinaryJSON Field

Save doc as binary json

**C#**

```
public static bool SaveAsBinaryJSON
```

### See Also

Applies to: [Global](#)<sub>7</sub>

## SaveIndexToDiskTimerSeconds Field

KeyStore save to disk timer

**C#**

```
public static int SaveIndexToDiskTimerSeconds
```

### See Also

Applies to: [Global](#)<sub>7</sub>

## SplitStorageFilesMegaBytes Field

Split the data storage files in MegaBytes (default 0 = off) [500 = 500mb]

- You can set and unset this value anytime and it will operate from that point on.
- If you unset (0) the value previous split files will remain and all the data will go to the last file.

**C#**

```
public static UInt16 SplitStorageFilesMegaBytes
```

### See Also

Applies to: [Global](#)<sub>7</sub>

## UseLessMemoryStructures Field

**C#**

```
public static bool UseLessMemoryStructures
```

### See Also

Applies to: [Global](#)<sub>7</sub>

## Hoot Class

**C#**

```
public class Hoot
```

### Requirements

Namespace: [RaptorDB](#)<sub>7</sub>

Assembly: Hoot.Standard (in Hoot.Standard.dll)

### Constructors

[Hoot](#)<sub>10</sub>

### Properties

[DocumentCount](#)<sub>12</sub>, [HootConfOptions](#)<sub>12</sub>, [WordCount](#)<sub>13</sub>, [Words](#)<sub>13</sub>

### Methods

[Fetch<T>](#)<sub>13</sub>, [FindDocumentFileNames](#)<sub>14</sub>, [FindDocuments<T>](#)<sub>14</sub>, [FindRows](#)<sub>14</sub>, [FreeMemory](#)<sub>15</sub>, [Index](#)<sub>16</sub>, [IsIndexed](#)<sub>17</sub>, [OptimizeIndex](#)<sub>17</sub>, [Query](#)<sub>17</sub>, [RemoveDocument](#)<sub>18</sub>, [Save](#)<sub>18</sub>, [Shutdown](#)<sub>19</sub>

## Hoot(String, String, Boolean) Constructor

**NOTE:** This member is now obsolete.

Construct a new Hoot Index

**C#**

```
[Obsolete()]
public Hoot(
    string IndexPath,
    string FileName,
    bool DocMode
)
```

### Parameters

*IndexPath*

Path to Index File Storage

*FileName*

Filename prefix for indexes

*DocMode*

Use Document Mode

### See Also

Applies to: [Hoot<sub>10</sub>](#)

## Hoot(String, String, Boolean, ITokenizer) Constructor

**NOTE:** This member is now obsolete.

Construct a new Hoot Index

```
C#  
  
[Obsolete()]  
public Hoot(  
    string indexPath,  
    string fileName,  
    bool docMode,  
    ITokenizer tokenizer  
)
```

### Parameters

*indexPath*

*fileName*

*docMode*

*tokenizer*

Custom Tokenizer to parse text

### See Also

Applies to: [Hoot<sub>10</sub>](#)

## Hoot(IHootConfig) Constructor

Initialize with the Configuration file

```
C#  
  
public Hoot(  
    IHootConfig config  
)
```

### Parameters

*config*

Configuration File

### See Also

Applies to: [Hoot<sub>10</sub>](#)

## Hoot(IHootConfig, ITokenizer) Constructor

Construct a new object using configuration file and custom tokenizer

**C#**

```
public Hoot(  
    IHootConfig config,  
    ITokenizer tokenizer  
)
```

### Parameters

*config*

Configuration file

*tokenizer*

Custom Tokenizer to parse text

### See Also

Applies to: [Hoot<sub>10</sub>](#)

## Hoot.DocumentCount Property

Get Document Count

**C#**

```
public int DocumentCount {get;}
```

### See Also

Applies to: [Hoot<sub>10</sub>](#)

## Hoot.HootConfOptions Property

Configuration file

**C#**

```
public IHootConfig HootConfOptions {get; set;}
```

### See Also

Applies to: [Hoot<sub>10</sub>](#)

## Hoot.WordCount Property

Get Word Count

**C#**

```
public int WordCount {get;}
```

### See Also

Applies to: [Hoot<sub>10</sub>](#)

## Hoot.Words Property

Get List of Words

**C#**

```
public string[] Words {get;}
```

### See Also

Applies to: [Hoot<sub>10</sub>](#)

## Hoot.Fetch<T> Method

Fetch a Document

**C#**

```
public T Fetch<T>(  
    int docnum  
)
```

### Type Parameters

*T*

Type of Document

### Parameters

*docnum*

Document Number

### See Also

Applies to: [Hoot<sub>10</sub>](#)

## Hoot.FindDocumentFileNames Method

Find Documents File Names

**C#**

```
public IEnumerable<string> FindDocumentFileNames(  
    string filter  
)
```

### Parameters

*filter*

### See Also

Applies to: [Hoot<sub>10</sub>](#)

## Hoot.FindDocuments<T> Method

Find Documents

**C#**

```
public IEnumerable<T> FindDocuments<T>(  
    string filter  
)
```

### Type Parameters

*T*

### Parameters

*filter*

### See Also

Applies to: [Hoot<sub>10</sub>](#)

## Hoot.FindRows Method

Find Rows

**C#**

```
public IEnumerable<int> FindRows(  
    string filter  
)
```

### Parameters

*filter*

## See Also

Applies to: [Hoot<sub>10</sub>](#)

## Hoot.FreeMemory Method

Free Memory

**C#**

```
public void FreeMemory()
```

## See Also

Applies to: [Hoot<sub>10</sub>](#)

## Hoot.Index (Int32, String, IHootFilter) Method

Index a Text String using a filter

**C#**

```
public void Index(  
    int recordnumber,  
    string text,  
    IHootFilter filter  
)
```

## Parameters

*recordnumber*

*text*

*filter*

## See Also

Applies to: [Hoot<sub>10</sub>](#)

## Hoot.Index (Document, Boolean, IHootFilter) Method

Index a Document

**C#**

```
public int Index(  
    Document doc,
```

```
bool deleteold,  
IHootFilter filter  
)
```

### Parameters

*doc*

*deleteold*

*filter*

### See Also

Applies to: [Hoot<sub>10</sub>](#)

## Hoot.Index (Int32, String) Method

Index a text String

**C#**

```
public void Index(  
    int recordnumber,  
    string text  
)
```

### Parameters

*recordnumber*

*text*

### See Also

Applies to: [Hoot<sub>10</sub>](#)

## Hoot.Index (Document, Boolean) Method

Index a Document

**C#**

```
public int Index(  
    Document doc,  
    bool deleteold  
)
```

### Parameters



*doc*

*deleteold*

## See Also

Applies to: [Hoot<sub>10</sub>](#)

## Hoot.IsIndexed Method

Check if a File is Indexed

**C#**

```
public bool IsIndexed(  
    string filename  
)
```

## Parameters

*filename*

## See Also

Applies to: [Hoot<sub>10</sub>](#)

## Hoot.OptimizeIndex Method

Optimize the Index

**C#**

```
public void OptimizeIndex()
```

## See Also

Applies to: [Hoot<sub>10</sub>](#)

## Hoot.Query Method

Query the Index

**C#**

```
public MGRB Query(  
    string filter,  
    int maxsize  
)
```

## Parameters

*filter*

*maxsize*

## See Also

Applies to: [Hoot<sub>10</sub>](#)

## Hoot.RemoveDocument (Int32) Method

Remove a Document

**C#**

```
public void RemoveDocument(  
    int number  
)
```

## Parameters

*number*

## See Also

Applies to: [Hoot<sub>10</sub>](#)

## Hoot.RemoveDocument (String) Method

Remove a Document by File Name

**C#**

```
public bool RemoveDocument(  
    string filename  
)
```

## Parameters

*filename*

## See Also

Applies to: [Hoot<sub>10</sub>](#)

## Hoot.Save Method

Save the Index

**C#**

```
public void Save()
```

## See Also

Applies to: [Hoot](#)<sub>10</sub>

## Hoot.Shutdown Method

Shutdown the Process

**C#**

```
public void Shutdown()
```

## See Also

Applies to: [Hoot](#)<sub>10</sub>

## HootConfig Class

Hoot Configuration file that can be initialized and passed to hoot Constructor

**C#**

```
public class HootConfig : IHootConfig
```

## Requirements

Namespace: [RaptorDB](#)<sub>7</sub>

Assembly: Hoot.Standard (in Hoot.Standard.dll)

## Constructors

[HootConfig](#)<sub>19</sub>

## Properties

[DocMode](#)<sub>19</sub>, [FileName](#)<sub>20</sub>, [IgnoreNumerics](#)<sub>20</sub>, [IndexPath](#)<sub>20</sub>, [UseStopList](#)<sub>21</sub>

## HootConfig Constructor

**C#**

```
public HootConfig()
```

## See Also

Applies to: [HootConfig](#)<sub>19</sub>

## HootConfig.DocMode Property

Use Document Mode

**C#**

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

## Implements

[IHootConfig.DocMode](#)<sub>22</sub>

## See Also

Applies to: [HootConfig](#)<sub>19</sub>

## HootConfig.FileName Property

Filename prefix for index files. Defaults to indexx

**C#**

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

## Implements

[IHootConfig.FileName](#)<sub>23</sub>

## See Also

Applies to: [HootConfig](#)<sub>19</sub>

## HootConfig.IgnoreNumerics Property

Ignore numeric words, ie 123,555, etc

**C#**

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

## Implements

[IHootConfig.IgnoreNumerics](#)<sub>23</sub>

## See Also

Applies to: [HootConfig](#)<sub>19</sub>

## HootConfig.IndexPath Property

Path where index files are stored

**C#**

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

## Implements

[IHootConfig.IndexPath](#)<sub>23</sub>

## See Also

Applies to: [HootConfig](#)<sub>19</sub>

## HootConfig.UseStopList Property

Use Word Stop List

**C#**

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

## Implements

[IHootConfig.UseStopList](#)<sub>23</sub>

## See Also

Applies to: [HootConfig](#)<sub>19</sub>

## tokenizer Class

**C#**

```
public class tokenizer : ITokenizer
```

## Requirements

**Namespace:** [RaptorDB](#)<sub>7</sub>

**Assembly:** Hoot.Standard (in Hoot.Standard.dll)

## Methods

[GenerateWordFreq](#)<sub>21</sub>, [InitializeStopList](#)<sub>22</sub>

## tokenizer.GenerateWordFreq Method

**C#**

```
public Dictionary<string, int> GenerateWordFreq(  
    string text,  
    IHootConfig config  
)
```

## Parameters

*text*

*config*

## Implements

[ITokenizer.GenerateWordFreq](#)<sub>25</sub>

## See Also

Applies to: [tokenizer](#)<sub>21</sub>

## tokenizer.InitializeStopList Method

Initialize the Stop List

**C#**

```
public void InitializeStopList(  
    string indexFolder  
)
```

## Parameters

*indexFolder*

## Implements

[ITokenizer.InitializeStopList](#)<sub>25</sub>

## See Also

Applies to: [tokenizer](#)<sub>21</sub>

## IHootConfig Interface

**C#**

```
public interface IHootConfig
```

## Requirements

**Namespace:** [RaptorDB](#)<sub>7</sub>

**Assembly:** Hoot.Standard (in Hoot.Standard.dll)

## Properties

[DocMode](#)<sub>22</sub>, [FileName](#)<sub>23</sub>, [IgnoreNumerics](#)<sub>23</sub>, [IndexPath](#)<sub>23</sub>, [UseStopList](#)<sub>23</sub>

## IHootConfig.DocMode Property

**C#**

```
bool DocMode {get; set;}
```

## See Also

Applies to: [IHootConfig](#)<sub>22</sub>

## IHootConfig.FileName Property

**C#**

```
string FileName {get; set;}
```

## See Also

Applies to: [IHootConfig](#)<sub>22</sub>

## IHootConfig.IgnoreNumerics Property

**C#**

```
bool IgnoreNumerics {get; set;}
```

## See Also

Applies to: [IHootConfig](#)<sub>22</sub>

## IHootConfig.IndexPath Property

**C#**

```
string IndexPath {get; set;}
```

## See Also

Applies to: [IHootConfig](#)<sub>22</sub>

## IHootConfig.UseStopList Property

**C#**

```
bool UseStopList {get; set;}
```

## See Also

Applies to: [IHootConfig](#)<sub>22</sub>

## IHootFilter Interface

**C#**

```
public interface IHootFilter
```

### Requirements

**Namespace:** [RaptorDB](#)<sub>7</sub>**Assembly:** Hoot.Standard (in Hoot.Standard.dll)

### Methods

[FilterText](#)<sub>24</sub>, [InitializeFilter](#)<sub>24</sub>

## IHootFilter.FilterText Method

Filter the input test

**C#**

```
string FilterText(  
    string input  
)
```

### Parameters

*input*

Text to Filter

### Returns

Filtered Text

### See Also

Applies to: [IHootFilter](#)<sub>24</sub>

## IHootFilter.InitializeFilter Method

Perform Initialization of Filter

**C#**

```
void InitializeFilter(  
    string filterPath  
)
```

### Parameters

*filterPath*

Path of Filter Text to load

### See Also



Applies to: [IHootFilter](#)<sub>24</sub>

## ITokenizer Interface

**C#**

```
public interface ITokenizer
```

### Requirements

**Namespace:** [RaptorDB](#)<sub>7</sub>

**Assembly:** Hoot.Standard (in Hoot.Standard.dll)

### Methods

[GenerateWordFreq](#)<sub>25</sub>, [InitializeStopList](#)<sub>25</sub>

## ITokenizer.GenerateWordFreq Method

**C#**

```
Dictionary<string, int> GenerateWordFreq(  
    string text,  
    IHootConfig config  
)
```

### Parameters

*text*

*config*

### See Also

Applies to: [ITokenizer](#)<sub>25</sub>

## ITokenizer.InitializeStopList Method

**C#**

```
void InitializeStopList(  
    string indexFolder  
)
```

### Parameters

*indexFolder*

### See Also

Applies to: [ITokenizer](#)<sub>25</sub>

## RaptorDB.Common Namespace

### Classes

[FastDateTime](#)<sub>27</sub>, [Helper](#)<sub>28</sub>, [SafeDictionary<TKey, TValue>](#)<sub>37</sub>, [SafeSortedList<T, V>](#)<sub>41</sub>

### Interfaces

[IKV<T, V>](#)<sub>41</sub>

## FastDateTime Class

**C#**

```
public static class FastDateTime
```

### Requirements

**Namespace:** [RaptorDB.Common](#)<sub>27</sub>

**Assembly:** Hoot.Standard (in Hoot.Standard.dll)

### Properties

[Now](#)<sub>27</sub>

### Fields

[LocalUtcOffset](#)<sub>27</sub>

## FastDateTime.Now Property

**C#**

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

### See Also

Applies to: [FastDateTime](#)<sub>27</sub>

## LocalUtcOffset Field

**C#**

```
public static TimeSpan LocalUtcOffset
```

### See Also

Applies to: [FastDateTime](#)<sub>27</sub>

## Helper Class

**C#**

```
public static class Helper
```

### Requirements

**Namespace:** [RaptorDB.Common](#)<sub>27</sub>**Assembly:** Hoot.Standard (in Hoot.Standard.dll)

### Methods

[CompareMemCmp](#)<sub>28</sub>, [GetBytes](#)<sub>29</sub>, [GetString](#)<sub>30</sub>, [ToInt16](#)<sub>31</sub>, [ToInt32](#)<sub>32</sub>, [ToInt64](#)<sub>32</sub>

### Fields

[MurMur](#)<sub>33</sub>

## Helper.CompareMemCmp Method

**C#**

```
public static int CompareMemCmp(  
    byte[] left,  
    byte[] right  
)
```

### Parameters

*left**right*

### See Also

Applies to: [Helper](#)<sub>28</sub>

## Helper.GetBytes (String) Method

**C#**

```
public static byte[] GetBytes(  
    string s  
)
```

### Parameters

*s*

### See Also

Applies to: [Helper](#)<sub>28</sub>

## Helper.GetBytes (Int16, Boolean) Method

**C#**

```
public static byte[] GetBytes(  
    short num,  
    bool reverse  
)
```

### Parameters

*num*

*reverse*

### See Also

Applies to: [Helper](#)<sub>28</sub>

## Helper.GetBytes (Int32, Boolean) Method

**C#**

```
public static byte[] GetBytes(  
    int num,  
    bool reverse  
)
```

### Parameters

*num*

*reverse*

### See Also

Applies to: [Helper](#)<sub>28</sub>

## Helper.GetBytes (Int64, Boolean) Method

**C#**

```
public static byte[] GetBytes(  
    long num,  
    bool reverse  
)
```

**Parameters**

*num*

*reverse*

**See Also**

Applies to: [Helper](#)<sub>28</sub>

**Helper.GetString Method****C#**

```
public static string GetString(  
    byte[] buffer,  
    int index,  
    short length  
)
```

**Parameters**

*buffer*

*index*

*length*

**See Also**

Applies to: [Helper](#)<sub>28</sub>

**Helper.ToInt16 (Byte(), Int32) Method****C#**

```
public static short ToInt16(  
    byte[] value,  
    int startIndex  
)
```

**Parameters**

*value*

*startIndex*

**See Also**

Applies to: [Helper](#)<sub>28</sub>

## Helper.ToInt16 (Byte(), Int32, Boolean) Method

**C#**

```
public static short ToInt16(  
    byte[] value,  
    int startIndex,  
    bool reverse  
)
```

### Parameters

*value*

*startIndex*

*reverse*

### See Also

Applies to: [Helper](#)<sub>28</sub>

## Helper.ToInt32 (Byte(), Int32, Boolean) Method

**C#**

```
public static int ToInt32(  
    byte[] value,  
    int startIndex,  
    bool reverse  
)
```

### Parameters

*value*

*startIndex*

*reverse*

### See Also

Applies to: [Helper](#)<sub>28</sub>

## Helper.ToInt32 (Byte(), Int32) Method

**C#**

```
public static int ToInt32(  
    byte[] value,  
    int startIndex  
)
```

### Parameters

*value*

*startIndex*

### See Also

Applies to: [Helper](#)<sub>28</sub>

## Helper.ToInt64 (Byte(), Int32, Boolean) Method

**C#**

```
public static long ToInt64(  
    byte[] value,  
    int startIndex,  
    bool reverse  
)
```

### Parameters

*value*

*startIndex*

*reverse*

### See Also

Applies to: [Helper](#)<sub>28</sub>

## Helper.ToInt64 (Byte(), Int32) Method

**C#**

```
public static long ToInt64(  
    byte[] value,  
    int startIndex  
)
```



## Parameters

*value*

*startIndex*

## See Also

Applies to: [Helper](#)<sub>28</sub>

## MurMur Field

**C#**

```
new public static MurmurHash2Unsafe MurMur
```

## See Also

Applies to: [Helper](#)<sub>28</sub>

## SafeDictionary<TKey, TValue> Class

**C#**

```
public class SafeDictionary<TKey, TValue> : IKV<TKey, TValue>
```

## Type Parameters

*TKey*

*TValue*

## Requirements

**Namespace:** [RaptorDB.Common](#)<sub>27</sub>

**Assembly:** Hoot.Standard (in Hoot.Standard.dll)

## Constructors

[SafeDictionary<TKey, TValue>](#)<sub>37</sub>

## Properties

[Item](#)<sub>37</sub>

## Methods

[Add](#)<sub>37</sub>, [Clear](#)<sub>37</sub>, [Count](#)<sub>37</sub>, [GetEnumerator](#)<sub>37</sub>, [GetValue](#)<sub>37</sub>, [Keys](#)<sub>37</sub>, [Remove](#)<sub>37</sub>, [TryGetValue](#)<sub>37</sub>

## SafeDictionary<TKey, TValue> Constructor

**C#**

```
public SafeDictionary()
```

### See Also

Applies to: [SafeDictionary<TKey, TValue>](#)<sup>37</sup>

## SafeDictionary<TKey, TValue>(Int32) Constructor

**C#**

```
public SafeDictionary(  
    int capacity  
)
```

### Parameters

*capacity*

### See Also

Applies to: [SafeDictionary<TKey, TValue>](#)<sup>37</sup>

## SafeDictionary<TKey, TValue>.Item Property

**C#**

```
public TValue this[  
    TKey key  
] {get; set;}
```

### Parameters

*key*

### See Also

Applies to: [SafeDictionary<TKey, TValue>](#)<sup>37</sup>

## SafeDictionary<TKey, TValue>.Add Method

**C#**

```
public void Add(  
    TKey key,  
    TValue value  
)
```

**Parameters**

*key*

*value*

**Implements**

[IKV.Add](#)<sub>42</sub>

**See Also**

Applies to: [SafeDictionary<TKey, TValue>](#)<sub>37</sub>

**SafeDictionary<TKey, TValue>.Clear Method****C#**

```
public void Clear()
```

**Implements**

[IKV.Clear](#)<sub>42</sub>

**See Also**

Applies to: [SafeDictionary<TKey, TValue>](#)<sub>37</sub>

**SafeDictionary<TKey, TValue>.Count Method****C#**

```
public int Count()
```

**Implements**

[IKV.Count](#)<sub>42</sub>

**See Also**

Applies to: [SafeDictionary<TKey, TValue>](#)<sub>37</sub>

**SafeDictionary<TKey, TValue>.GetEnumerator Method****C#**

```
public IEnumerator<KeyValuePair<TKey, TValue>> GetEnumerator()
```

**Implements**

[IKV.GetEnumerator](#)<sub>43</sub>

## See Also

Applies to: [SafeDictionary<TKey, TValue>](#)<sub>37</sub>

## SafeDictionary<TKey, TValue>.GetValue Method

**C#**

```
public TValue GetValue(  
    TKey key  
)
```

## Parameters

*key*

## Implements

[IKV.GetValue](#)<sub>43</sub>

## See Also

Applies to: [SafeDictionary<TKey, TValue>](#)<sub>37</sub>

## SafeDictionary<TKey, TValue>.Keys Method

**C#**

```
public TKey[] Keys()
```

## Implements

[IKV.Keys](#)<sub>43</sub>

## See Also

Applies to: [SafeDictionary<TKey, TValue>](#)<sub>37</sub>

## SafeDictionary<TKey, TValue>.Remove Method

**C#**

```
public bool Remove(  
    TKey key  
)
```

## Parameters

*key*

## Implements

[IKV.Remove](#)<sub>43</sub>

## See Also

Applies to: [SafeDictionary<TKey, TValue>](#)<sub>37</sub>

## SafeDictionary<TKey, TValue>.TryGetValue Method

**C#**

```
public bool TryGetValue(  
    TKey key,  
    out TValue value  
)
```

## Parameters

*key*

*value*

## Implements

[IKV.TryGetValue](#)<sub>44</sub>

## See Also

Applies to: [SafeDictionary<TKey, TValue>](#)<sub>37</sub>

## SafeSortedList<T, V> Class

**C#**

```
public class SafeSortedList<T, V> : IKV<T, V>
```

## Type Parameters

*T*

*V*

## Requirements

**Namespace:** [RaptorDB.Common](#)<sub>27</sub>

**Assembly:** Hoot.Standard (in Hoot.Standard.dll)

## Properties

[Item](#)<sub>41</sub>

## Methods

[Add](#)<sub>41</sub>, [Clear](#)<sub>41</sub>, [Count](#)<sub>41</sub>, [GetEnumerator](#)<sub>41</sub>, [GetKey](#)<sub>41</sub>, [GetValue](#)<sub>41</sub>, [Keys](#)<sub>41</sub>, [Remove](#)<sub>41</sub>, [TryGetValue](#)<sub>41</sub>

## SafeSortedList<T, V>.Item Property

**C#**

```
public V this[
    T key
] {get; set;}
```

## Parameters

*key*

## See Also

Applies to: [SafeSortedList<T, V>](#)<sub>41</sub>

## SafeSortedList<T, V>.Add Method

**C#**

```
public void Add(
    T key,
    V val
)
```

## Parameters

*key*

*val*

## Implements

[IKV.Add](#)<sub>42</sub>

## See Also

Applies to: [SafeSortedList<T, V>](#)<sub>41</sub>

## SafeSortedList<T, V>.Clear Method

**C#**

```
public void Clear()
```

## Implements

[IKV.Clear](#)<sub>42</sub>

## See Also

Applies to: [SafeSortedList<T, V>](#)<sub>41</sub>

## SafeSortedList<T, V>.Count Method

**C#**

```
public int Count()
```

## Implements

[IKV.Count](#)<sub>42</sub>

## See Also

Applies to: [SafeSortedList<T, V>](#)<sub>41</sub>

## SafeSortedList<T, V>.GetEnumerator Method

**C#**

```
public IEnumerator<KeyValuePair<T, V>> GetEnumerator()
```

## Implements

[IKV.GetEnumerator](#)<sub>43</sub>

## See Also

Applies to: [SafeSortedList<T, V>](#)<sub>41</sub>

## SafeSortedList<T, V>.GetKey Method

**C#**

```
public T GetKey(  
    int index  
)
```

## Parameters

*index*

## See Also

Applies to: [SafeSortedList<T, V>](#)<sub>41</sub>

## SafeSortedList<T, V>.GetValue (Int32) Method

**C#**

```
public V GetValue(  
    int index  
)
```

### Parameters

*index*

### Implements

[Error! Hyperlink reference not valid.](#)

### See Also

Applies to: [SafeSortedList<T, V>](#)<sub>41</sub>

## SafeSortedList<T, V>.GetValue (T) Method

**C#**

```
public V GetValue(  
    T key  
)
```

### Parameters

*key*

### Implements

[IKV.GetValue](#)<sub>43</sub>

### See Also

Applies to: [SafeSortedList<T, V>](#)<sub>41</sub>

## SafeSortedList<T, V>.Keys Method

**C#**

```
public T[] Keys()
```

### Implements

[IKV.Keys](#)<sub>43</sub>

### See Also

Applies to: [SafeSortedList<T, V>](#)<sub>41</sub>



## SafeSortedList<T, V>.Remove Method

**C#**

```
public bool Remove(  
    T key  
)
```

### Parameters

*key*

### Implements

[IKV.Remove](#)<sub>43</sub>

### See Also

Applies to: [SafeSortedList<T, V>](#)<sub>41</sub>

## SafeSortedList<T, V>.TryGetValue Method

**C#**

```
public bool TryGetValue(  
    T key,  
    out V value  
)
```

### Parameters

*key*

*value*

### Implements

[IKV.TryGetValue](#)<sub>44</sub>

### See Also

Applies to: [SafeSortedList<T, V>](#)<sub>41</sub>

## IKV<T, V> Interface

**C#**

```
public interface IKV<T, V>
```

### Type Parameters

*T*

V

## Requirements

Namespace: [RaptorDB.Common](#)<sub>27</sub>

Assembly: Hoot.Standard (in Hoot.Standard.dll)

## Methods

[Add](#)<sub>42</sub>, [Clear](#)<sub>42</sub>, [Count](#)<sub>42</sub>, [GetEnumerator](#)<sub>43</sub>, [GetValue](#)<sub>43</sub>, [Keys](#)<sub>43</sub>, [Remove](#)<sub>43</sub>, [TryGetValue](#)<sub>44</sub>

## IKV<T, V>.Add Method

C#

```
void Add(  
    T key,  
    V value  
)
```

## Parameters

*key*

*value*

## See Also

Applies to: [IKV<T, V>](#)<sub>41</sub>

## IKV<T, V>.Clear Method

C#

```
void Clear()
```

## See Also

Applies to: [IKV<T, V>](#)<sub>41</sub>

## IKV<T, V>.Count Method

C#

```
int Count()
```

## See Also

Applies to: [IKV<T, V>](#)<sub>41</sub>

## IKV<T, V>.GetEnumerator Method

**C#**

```
IEnumerator<KeyValuePair<T, V>> GetEnumerator()
```

### See Also

Applies to: [IKV<T, V>](#)<sub>41</sub>

## IKV<T, V>.GetValue Method

**C#**

```
V GetValue(  
    T key  
)
```

### Parameters

*key*

### See Also

Applies to: [IKV<T, V>](#)<sub>41</sub>

## IKV<T, V>.Keys Method

**C#**

```
T[] Keys()
```

### See Also

Applies to: [IKV<T, V>](#)<sub>41</sub>

## IKV<T, V>.Remove Method

**C#**

```
bool Remove(  
    T key  
)
```

### Parameters

*key*

## See Also

Applies to: [IKV<T, V>](#)<sub>41</sub>

## IKV<T, V>.TryGetValue Method

**C#**

```
bool TryGetValue(  
    T key,  
    out V val  
)
```

## Parameters

*key*

*val*

## See Also

Applies to: [IKV<T, V>](#)<sub>41</sub>

## RaptorDB.Filters Namespace

### Classes

[HtmlFilter](#)<sub>45</sub>, [NoFilter](#)<sub>46</sub>

### HtmlFilter Class

**C#**

```
public class HtmlFilter : IHootFilter
```

### Requirements

**Namespace:** [RaptorDB.Filters](#)<sub>45</sub>

**Assembly:** Hoot.Standard (in Hoot.Standard.dll)

### Methods

[FilterText](#)<sub>45</sub>, [InitializeFilter](#)<sub>45</sub>

### HtmlFilter.FilterText Method

Filter Html Text

**C#**

```
public string FilterText(  
    string input  
)
```

### Parameters

*input*

### Implements

[IHootFilter.FilterText](#)<sub>24</sub>

### See Also

Applies to: [HtmlFilter](#)<sub>45</sub>

### HtmlFilter.InitializeFilter Method

Initialize the filter, Not used

**C#**

```
public void InitializeFilter(  
    string filterPath = null
```

```
)
```

### Parameters

*filterPath*

### Implements

[IHootFilter.InitializeFilter](#)<sub>24</sub>

### See Also

Applies to: [HtmlFilter](#)<sub>45</sub>

## NoFilter Class

**C#**

```
public class NoFilter : IHootFilter
```

### Requirements

**Namespace:** [RaptorDB.Filters](#)<sub>45</sub>

**Assembly:** Hoot.Standard (in Hoot.Standard.dll)

### Methods

[FilterText](#)<sub>46</sub>, [InitializeFilter](#)<sub>47</sub>

## NoFilter.FilterText Method

Just return the Text in default Filter

**C#**

```
public string FilterText(  
    string input  
)
```

### Parameters

*input*

### Implements

[IHootFilter.FilterText](#)<sub>24</sub>

### See Also

Applies to: [NoFilter](#)<sub>46</sub>

## NoFilter.InitializeFilter Method

Do nothing in the default filter

**C#**

```
public void InitializeFilter(  
    string filterPath  
)
```

### Parameters

*filterPath*

Path to full text folder

### Implements

[IHootFilter.InitializeFilter](#)<sub>24</sub>

### See Also

Applies to: [NoFilter](#)<sub>46</sub>

## h00t Namespace

### Classes

[Document](#)<sub>48</sub>

### Document Class

C#

```
public class Document
```

### Requirements

Namespace: [h00t](#)<sub>48</sub>

Assembly: Hoot.Standard (in Hoot.Standard.dll)

### Constructors

[Document](#)<sub>49</sub>

### Properties

[DocNumber](#)<sub>49</sub>, [FileName](#)<sub>49</sub>, [ModifiedDate](#)<sub>49</sub>, [Text](#)<sub>49</sub>

### Methods

[ToString](#)<sub>50</sub>

### Fields

[FileSize](#)<sub>50</sub>

### Document(FileInfo, String) Constructor

C#

```
public Document(  
    FileInfo fileinfo,  
    string text  
)
```

### Parameters

*fileinfo*

*text*

### See Also

Applies to: [Document](#)<sub>48</sub>



## Document Constructor

**C#**

```
public Document()
```

### See Also

Applies to: [Document](#)<sub>48</sub>

## Document.DocNumber Property

**C#**

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

### See Also

Applies to: [Document](#)<sub>48</sub>

## Document.FileName Property

**C#**

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

### See Also

Applies to: [Document](#)<sub>48</sub>

## Document.ModifiedDate Property

**C#**

```
public DateTime ModifiedDate {get; set;}
```

### See Also

Applies to: [Document](#)<sub>48</sub>

## Document.Text Property

**C#**

```
[XmlIgnore()]  
public string Text {get; set;}
```

## See Also

Applies to: [Document](#)<sub>48</sub>

## Document.ToString Method

**C#**

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

## See Also

Applies to: [Document](#)<sub>48</sub>

## FileSize Field

**C#**

```
public long FileSize
```

## See Also

Applies to: [Document](#)<sub>48</sub>

## Index

- Add Method {RaptorDB.Common.IKV<T, V>} 42
- Add Method {RaptorDB.Common.SafeDictionary<TKey, TValue>} 37
- Add Method {RaptorDB.Common.SafeSortedList<T, V>} 41
- Clear Method {RaptorDB.Common.IKV<T, V>} 42
- Clear Method {RaptorDB.Common.SafeDictionary<TKey, TValue>} 37
- Clear Method {RaptorDB.Common.SafeSortedList<T, V>} 41
- CompareMemCmp Method 28
- CompressBitmapBytes Field 7
- CompressDocumentOverKiloBytes Field 7
- Count Method {RaptorDB.Common.IKV<T, V>} 42
- Count Method {RaptorDB.Common.SafeDictionary<TKey, TValue>} 37
- Count Method {RaptorDB.Common.SafeSortedList<T, V>} 41
- DefaultStringKeySize Field 8
- DocMode Property {RaptorDB.HootConfig} 19
- DocMode Property {RaptorDB.IHootConfig} 22
- DocNumber Property 49
- Document Constructor 49
- Document (FileInfo, String) Constructor 48
- Document Class 48
- DocumentCount Property 12
- FastDateTime Class 27
- Fetch<T> Method 13
- FileName Property {RaptorDB.HootConfig} 20
- FileName Property {RaptorDB.IHootConfig} 23
- FileName Property {hOot.Document} 49
- FileSize Field 50
- FilterText Method {RaptorDB.Filters.HtmlFilter} 45
- FilterText Method {RaptorDB.Filters.NoFilter} 46
- FilterText Method {RaptorDB.IHootFilter} 24
- FindDocumentFileNames Method 14
- FindDocuments<T> Method 14
- FindRows Method 14
- FlushStorageFileImmediately Field 8
- FreeBitmapMemoryOnSave Field 8
- FreeMemory Method 15
- GenerateWordFreq Method {RaptorDB.ITokenizer} 25
- GenerateWordFreq Method {RaptorDB.tokenizer} 21
- GetBytes (Int16, Boolean) Method 29
- GetBytes (Int32, Boolean) Method 29
- GetBytes (Int64, Boolean) Method 29
- GetBytes (String) Method 29
- GetEnumerator Method {RaptorDB.Common.IKV<T, V>} 43
- GetEnumerator Method {RaptorDB.Common.SafeDictionary<TKey, TValue>} 37
- GetEnumerator Method {RaptorDB.Common.SafeSortedList<T, V>} 41
- GetKey Method 41
- GetString Method 30
- GetValue (Int32) Method {RaptorDB.Common.SafeSortedList<T, V>} 41
- GetValue (T) Method {RaptorDB.Common.SafeSortedList<T, V>} 41
- GetValue Method {RaptorDB.Common.IKV<T, V>} 43
- GetValue Method {RaptorDB.Common.SafeDictionary<TKey, TValue>} 37
- Global Class 7
- Helper Class 28
- HighFrequencyKVDiskBlockSize Field 8
- Hoot (IHootConfig) Constructor 11
- Hoot (IHootConfig, ITokenizer) Constructor 12
- Hoot (String, String, Boolean) Constructor 10
- Hoot (String, String, Boolean, ITokenizer) Constructor 11
- Hoot Class 10
- Hoot.Standard Reference 6
- HootConfOptions Property 12
- HootConfig Class 19
- HootConfig Constructor 19
- HtmlFilter Class 45
- IHootConfig Interface 22
- IHootFilter Interface 24
- IKV<T, V> Interface 41
- ITokenizer Interface 25
- IgnoreNumerics Property {RaptorDB.HootConfig} 20
- IgnoreNumerics Property {RaptorDB.IHootConfig} 23
- Index (Document, Boolean) Method 16
- Index (Document, Boolean, IHootFilter) Method 15
- Index (Int32, String) Method 16
- Index (Int32, String, IHootFilter) Method 15
- IndexPath Property {RaptorDB.HootConfig} 20
- IndexPath Property {RaptorDB.IHootConfig} 23
- InitializeFilter Method {RaptorDB.Filters.HtmlFilter} 45
- InitializeFilter Method {RaptorDB.Filters.NoFilter} 47
- InitializeFilter Method {RaptorDB.IHootFilter} 24
- InitializeStopList Method {RaptorDB.ITokenizer} 25
- InitializeStopList Method {RaptorDB.tokenizer} 22
- IsIndexed Method 17
- Item Property {RaptorDB.Common.SafeDictionary<TKey, TValue>} 37
- Item Property {RaptorDB.Common.SafeSortedList<T, V>} 41
- Keys Method {RaptorDB.Common.IKV<T, V>} 43
- Keys Method {RaptorDB.Common.SafeDictionary<TKey, TValue>} 37

Keys Method {RaptorDB.Common.SafeSortedList<T, V>}

41

LocalUtcOffset Field 27

ModifiedDate Property 49

MurMur Field 33

NoFilter Class 46

Now Property 27

OptimizeIndex Method 17

PageItemCount Field 9

Query Method 17

RaptorDB Namespace 7

RaptorDB.Common Namespace 27

RaptorDB.Filters Namespace 45

Remove Method {RaptorDB.Common.IKV<T, V>} 43

Remove Method

{RaptorDB.Common.SafeDictionary<TKey, TValue>}

37

Remove Method {RaptorDB.Common.SafeSortedList<T,

V>} 41

RemoveDocument (Int32) Method 18

RemoveDocument (String) Method 18

SafeDictionary<TKey, TValue> Constructor 37

SafeDictionary<TKey, TValue> (Int32) Constructor 37

SafeDictionary<TKey, TValue> Class 37

SafeSortedList<T, V> Class 41

Save Method 18

SaveAsBinaryJSON Field 9

SaveIndexToDiskTimerSeconds Field 9

Shutdown Method 19

SplitStorageFilesMegaBytes Field 9

Text Property 49

ToInt16 (Byte(), Int32) Method 31

ToInt16 (Byte(), Int32, Boolean) Method 31

ToInt32 (Byte(), Int32) Method 32

ToInt32 (Byte(), Int32, Boolean) Method 32

ToInt64 (Byte(), Int32) Method 32

ToInt64 (Byte(), Int32, Boolean) Method 32

ToString Method 50

TryGetValue Method {RaptorDB.Common.IKV<T, V>} 44

TryGetValue Method

{RaptorDB.Common.SafeDictionary<TKey, TValue>}

37

TryGetValue Method

{RaptorDB.Common.SafeSortedList<T, V>} 41

UseLessMemoryStructures Field 10

UseStopList Property {RaptorDB.HootConfig} 21

UseStopList Property {RaptorDB.IHootConfig} 23

WordCount Property 13

Words Property 13

hOOt Namespace 48

tokenizer Class 21