

# Namespace LlamaLogic.ValveDataFormat

## Classes

### [VdfKeyValuePair](#)

Represents a key-value pair in a VDF document

### [VdfNode](#)

Represents a node in a VDF document

### [VdfSection](#)

Represents a section in a VDF document

# Class VdfKeyValuePair

Namespace: [LlamaLogic.ValveDataFormat](#)

Assembly: LlamaLogic.ValveDataFormat.dll

Represents a key-value pair in a VDF document

```
public sealed class VdfKeyValuePair : VdfNode, IEquatable<VdfNode>,
    IEquatable<VdfKeyValuePair>
```




## Inheritance

[object](#)  ← [VdfNode](#)  ← VdfKeyValuePair

## Implements

[IEquatable](#)  <[VdfNode](#)>, [IEquatable](#)  <[VdfKeyValuePair](#)>

## Inherited Members

[VdfNode.Deserialize\(StreamReader\)](#), [VdfNode.DeserializeAsync\(StreamReader\)](#), [VdfNode.Parse\(string\)](#), [VdfNode.TryParse\(string, out IReadOnlyList<VdfNode>\)](#), [VdfNode.TrailingComment](#), [VdfNode.Serialize\(StreamWriter\)](#), [VdfNode.SerializeAsync\(StreamWriter\)](#), [VdfNode.ToString\(\)](#), [object.Equals\(object, object\)](#) , [object.GetType\(\)](#) , [object.ReferenceEquals\(object, object\)](#) 

## Constructors

### VdfKeyValuePair()

Represents a key-value pair in a VDF document

```
public VdfKeyValuePair()
```

## Properties

### Key

Gets/sets the key of the key-value pair

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

Property Value

[string](#)↗

## KeyTrailingComment

Gets/sets the comment following the key

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

Property Value

[string](#)↗

## Value

Gets/sets the value of the key-value pair

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

Property Value

[object](#)↗

## Methods

### Equals(VdfKeyValuePair?)

Indicates whether the current object is equal to another object of the same type.

```
public bool Equals(VdfKeyValuePair? other)
```

Parameters

other [VdfKeyValuePair](#)

An object to compare with this object.

Returns

[bool](#)

[true](#) if the current object is equal to the other parameter; otherwise, [false](#).

## Equals(VdfNode?)

Indicates whether the current object is equal to another object of the same type.

```
public override bool Equals(VdfNode? other)
```

Parameters

other [VdfNode](#)

An object to compare with this object.

Returns

[bool](#)

[true](#) if the current object is equal to the other parameter; otherwise, [false](#).

## Equals(object?)

Determines whether the specified object is equal to the current object.

```
public override bool Equals(object? obj)
```

Parameters

obj [object](#)

The object to compare with the current object.

Returns

[bool](#)

[true](#) if the specified object is equal to the current object; otherwise, [false](#).

## GetHashCode()

Serves as the default hash function.

```
public override int GetHashCode()
```

Returns

[int](#)

A hash code for the current object.

## Operators

`operator ==(VdfKeyValuePair?, VdfKeyValuePair?)`

```
public static bool operator ==(VdfKeyValuePair? left, VdfKeyValuePair? right)
```

Parameters

`left` [VdfKeyValuePair](#)

`right` [VdfKeyValuePair](#)

Returns

[bool](#)

`operator !=(VdfKeyValuePair?, VdfKeyValuePair?)`

```
public static bool operator !=(VdfKeyValuePair? left, VdfKeyValuePair? right)
```

## Parameters

**left** [VdfKeyValuePair](#)

**right** [VdfKeyValuePair](#)

## Returns

[bool](#)

# Class VdfNode

Namespace: [LlamaLogic.ValveDataFormat](#)

Assembly: LlamaLogic.ValveDataFormat.dll

Represents a node in a VDF document

```
public class VdfNode : IEquatable<VdfNode>
```

## Inheritance

[object](#)  ← VdfNode

## Implements

[IEquatable](#)  <[VdfNode](#)>

## Derived

[VdfKeyValuePair](#), [VdfSection](#)

## Inherited Members

[object.Equals\(object, object\)](#)  , [object.GetType\(\)](#)  , [object.MemberwiseClone\(\)](#)  ,  
[object.ReferenceEquals\(object, object\)](#) 

# Properties

## TrailingComment

Gets/sets the comment which will follow the [VdfNode](#)

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

## Property Value

[string](#) 

# Methods

## Deserialize(StreamReader)

Deserializes a list of [VdfNode](#) from a specified `reader`

```
public static IReadOnlyList<VdfNode> Deserialize(StreamReader reader)
```

## Parameters

`reader` [StreamReader](#)

## Returns

[IReadOnlyList](#) <[VdfNode](#)>

# DeserializeAsync(StreamReader)

Deserializes a list of [VdfNode](#) from a specified `reader` asynchronously

```
public static Task<IReadOnlyList<VdfNode>> DeserializeAsync(StreamReader reader)
```

## Parameters

`reader` [StreamReader](#)

## Returns

[Task](#) <[IReadOnlyList](#) <[VdfNode](#)> >

# Equals(VdfNode?)

Indicates whether the current object is equal to another object of the same type.

```
public virtual bool Equals(VdfNode? other)
```

## Parameters

`other` [VdfNode](#)

An object to compare with this object.



## Returns

[bool](#)

[true](#) if the current object is equal to the **other** parameter; otherwise, [false](#).

## Equals(object?)

Determines whether the specified object is equal to the current object.

```
public override bool Equals(object? obj)
```

## Parameters

**obj** [object](#)

The object to compare with the current object.

## Returns

[bool](#)

[true](#) if the specified object is equal to the current object; otherwise, [false](#).

## GetHashCode()

Serves as the default hash function.

```
public override int GetHashCode()
```

## Returns

[int](#)

A hash code for the current object.

## Parse(string)

Parses a list of [VdfNode](#) from a specified [string](#)

```
public static IReadOnlyList<VdfNode> Parse(string s)
```

## Parameters

**s** [string](#)

The [string](#) to parse

## Returns

[IReadOnlyList](#) <[VdfNode](#)>

## Serialize(StreamWriter)

Serializes this [VdfNode](#) to the specified **writer**

```
public void Serialize(StreamWriter writer)
```

## Parameters

**writer** [StreamWriter](#)

## SerializeAsync(StreamWriter)

Serializes this [VdfNode](#) to the specified **writer** asynchronously

```
public ValueTask SerializeAsync(StreamWriter writer)
```

## Parameters

**writer** [StreamWriter](#)

## Returns

[ValueTask](#)

## ToString()

Returns a string that represents the current object.

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

Returns

[string](#) 

A string that represents the current object.

## TryParse(string?, out IReadOnlyList<VdfNode>)

Tries to parse a list of [VdfNode](#) from a specified [string](#) 

```
public static bool TryParse(string? s, out IReadOnlyList<VdfNode> result)
```

Parameters

**s** [string](#) 



The [string](#)  to parse

**result** [IReadOnlyList](#)  [<VdfNode>](#)

The list of [VdfNode](#) if **s** was successfully parsed

Returns

[bool](#) 

[true](#)  if **s** was successfully parsed; otherwise, [false](#) 

## Operators

operator ==(VdfNode?, VdfNode?)

```
public static bool operator ==(VdfNode? left, VdfNode? right)
```

Parameters

left [VdfNode](#)

right [VdfNode](#)

Returns

[bool](#)

## operator !=(VdfNode?, VdfNode?)

```
public static bool operator !=(VdfNode? left, VdfNode? right)
```

Parameters

left [VdfNode](#)

right [VdfNode](#)

Returns

[bool](#)

# Class VdfSection

Namespace: [LlamaLogic.ValveDataFormat](#)

Assembly: LlamaLogic.ValveDataFormat.dll


Represents a section in a VDF document

```
public sealed class VdfSection : VdfNode, IEquatable<VdfNode>, IEquatable<VdfSection>
```




## Inheritance

[object](#)  ← [VdfNode](#)  ← VdfSection

## Implements

[IEquatable](#)  <[VdfNode](#)>, [IEquatable](#)  <[VdfSection](#)>

## Inherited Members

[VdfNode.Deserialize\(StreamReader\)](#), [VdfNode.DeserializeAsync\(StreamReader\)](#), [VdfNode.Parse\(string\)](#), [VdfNode.TryParse\(string, out IReadOnlyList<VdfNode>\)](#), [VdfNode.TrailingComment](#), [VdfNode.Serialize\(StreamWriter\)](#), [VdfNode.SerializeAsync\(StreamWriter\)](#), [VdfNode.ToString\(\)](#), [object.Equals\(object, object\)](#) , [object.GetType\(\)](#) , [object.ReferenceEquals\(object, object\)](#) 

# Properties

## this[int]

Gets/sets [Value](#) of the [VdfKeyValuePair](#) at the specified [index](#) of the section

```
public object this[int index] { get; set; }
```

## Parameters

[index](#) [int](#) 

## Property Value

[object](#) 

## this[string]

Gets/sets [Value](#) of the [VdfKeyValuePair](#) with the specified **key**

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

### Parameters

**key** [string](#)<sup>↗</sup>

### Property Value

[object](#)<sup>↗</sup>

## Nodes

Gets the list of [VdfNode](#) in the [VdfSection](#)

```
public IList<VdfNode> Nodes { get; }
```

### Property Value

[IList](#)<sup>↗</sup> <[VdfNode](#)>

## OpeningTrailingComment

Gets/sets the comment following the opening character ('{') of the section

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

### Property Value

[string](#)<sup>↗</sup>

## Methods

## Equals(VdfNode?)

Indicates whether the current object is equal to another object of the same type.

```
public override bool Equals(VdfNode? other)
```

### Parameters

**other** [VdfNode](#)

An object to compare with this object.

### Returns

[bool](#)

[true](#) if the current object is equal to the **other** parameter; otherwise, [false](#).

## Equals(VdfSection?)

Indicates whether the current object is equal to another object of the same type.

```
public bool Equals(VdfSection? other)
```

### Parameters

**other** [VdfSection](#)

An object to compare with this object.

### Returns

[bool](#)

[true](#) if the current object is equal to the **other** parameter; otherwise, [false](#).

## Equals(object?)

Determines whether the specified object is equal to the current object.

```
public override bool Equals(object? obj)
```

## Parameters

**obj** [object](#)

The object to compare with the current object.

## Returns

[bool](#)

[true](#) if the specified object is equal to the current object; otherwise, [false](#).

## GetHashCode()

Serves as the default hash function.

```
public override int GetHashCode()
```

## Returns

[int](#)

A hash code for the current object.

## Operators

### operator ==(VdfSection?, VdfSection?)

```
public static bool operator ==(VdfSection? left, VdfSection? right)
```

## Parameters

**left** [VdfSection](#)

**right** [VdfSection](#)



Returns

[bool](#)

operator !=(VdfSection?, VdfSection?)

```
public static bool operator !=(VdfSection? left, VdfSection? right)
```

Parameters

**left** [VdfSection](#)

**right** [VdfSection](#)

Returns

[bool](#)