

# LinksPlatform's Platform.Communication.Protocol.Lino Class Library

## 1.1 ./Platform.Communication.Protocol.Lino/IListExtensions.cs

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Runtime.CompilerServices;
5
6 #pragma warning disable CS1591 // Missing XML comment for publicly visible type or member
7
8 namespace Platform.Communication.Protocol.Lino
9 {
10     public static class IListExtensions
11     {
12         [MethodImpl(MethodImplOptions.AggressiveInlining)]
13         public static string Format(this IList<Link> links) => string.Join(Environment.NewLine,
14             ↪ links.Select(l => l.ToString()));
15     }
16 }
```

## 1.2 ./Platform.Communication.Protocol.Lino/Link.cs

```
1 using System;
2 using System.Collections.Generic;
3 using System.Runtime.CompilerServices;
4 using System.Text;
5 using Platform.Collections;
6 using Platform.Collections.Lists;
7
8 #pragma warning disable CS1591 // Missing XML comment for publicly visible type or member
9
10 namespace Platform.Communication.Protocol.Lino
11 {
12     public struct Link : IEquatable<Link>
13     {
14         public string Id
15         {
16             [MethodImpl(MethodImplOptions.AggressiveInlining)]
17             get;
18             [MethodImpl(MethodImplOptions.AggressiveInlining)]
19             set;
20         }
21
22         public IList<Link> Values
23         {
24             [MethodImpl(MethodImplOptions.AggressiveInlining)]
25             get;
26             [MethodImpl(MethodImplOptions.AggressiveInlining)]
27             set;
28         }
29
30         [MethodImpl(MethodImplOptions.AggressiveInlining)]
31         public Link(string id, IList<Link> values) => (Id, Values) = (id, values);
32
33         [MethodImpl(MethodImplOptions.AggressiveInlining)]
34         public Link(IList<Link> values) : this(null, values) { }
35
36         [MethodImpl(MethodImplOptions.AggressiveInlining)]
37         public Link(string id) : this(id, null) { }
38
39         [MethodImpl(MethodImplOptions.AggressiveInlining)]
40         public override string ToString() => Values.IsNullOrEmpty() ? $"{Id}" :
41             ↪ GetLinkValuesString();
42
43         [MethodImpl(MethodImplOptions.AggressiveInlining)]
44         private string GetLinkValuesString() => Id == null ? $"{GetValuesString()}" :
45             ↪ $"{Id}: {GetValuesString()}";
46
47         [MethodImpl(MethodImplOptions.AggressiveInlining)]
48         public string GetValuesString()
49         {
50             var sb = new StringBuilder();
51             for (int i = 0; i < Values.Count; i++)
52             {
53                 if (i > 0)
54                 {
55                     sb.Append(' ');
56                 }
57                 sb.Append(GetValueString(Values[i]));
58             }
59             return sb.ToString();
60         }
61     }
62 }
```

```

59     [MethodImpl(MethodImplOptions.AggressiveInlining)]
60     public static string GetValueString(Link value) => value.ToLinkOrIdString();
61
62     [MethodImpl(MethodImplOptions.AggressiveInlining)]
63     public string ToLinkOrIdString() => Values.IsNullOrEmpty() ? Id : ToString();
64
65     [MethodImpl(MethodImplOptions.AggressiveInlining)]
66     public static implicit operator Link(string value) => new Link(value);
67
68     [MethodImpl(MethodImplOptions.AggressiveInlining)]
69     public static implicit operator Link(ValueTuple<string, IList<Link>> value) => new
70     ↪ Link(value.Item1, value.Item2);
71
72     [MethodImpl(MethodImplOptions.AggressiveInlining)]
73     public override bool Equals(object obj) => obj is Link link ? Equals(link) : false;
74
75     [MethodImpl(MethodImplOptions.AggressiveInlining)]
76     public override int GetHashCode() => (Id, Values.GenerateHashCode()).GetHashCode();
77
78     [MethodImpl(MethodImplOptions.AggressiveInlining)]
79     public bool Equals(Link other) => Id == other.Id && Values.EqualTo(other.Values);
80
81     [MethodImpl(MethodImplOptions.AggressiveInlining)]
82     public static bool operator ==(Link left, Link right) => left.Equals(right);
83
84     [MethodImpl(MethodImplOptions.AggressiveInlining)]
85     public static bool operator !=(Link left, Link right) => !(left == right);
86 }
87 }

```

### 1.3 ./Platform.Communication.Protocol.Lino.Tests/ParserTests.cs

```

1  using Xunit;
2
3  namespace Platform.Communication.Protocol.Lino.Tests
4  {
5      public static class ParserTests
6      {
7          [Fact]
8          public static void ParseAndStringifyTest()
9          {
10             var source = @"(papa (lovesMama: loves mama))
11 (son lovesMama)
12 (daughter lovesMama)
13 (all (love mama))";
14             var parser = new Parser();
15             var links = parser.Parse(source);
16             var target = links.Format();
17             Assert.Equal(source, target);
18         }
19     }
20 }

```

## Index

./Platform.Communication.Protocol.Lino.Tests/ParserTests.cs, 2  
./Platform.Communication.Protocol.Lino/IListExtensions.cs, 1  
./Platform.Communication.Protocol.Lino/Link.cs, 1