```
LinksPlatform's Platform.Communication.Protocol.Lino Class Library
     ./csharp/Platform.Communication.Protocol.Lino/ILinksGroupListExtensions.cs
   using System.Collections.Generic;
   using System.Runtime.CompilerServices;
   #pragma warning disable CS1591 // Missing XML comment for publicly visible type or member
4
   namespace Platform.Communication.Protocol.Lino
6
        public static class ILinksGroupListExtensions
9
            [MethodImpl(MethodImplOptions.AggressiveInlining)]
10
            public static List<Link> ToLinksList(this IList<LinksGroup> groups)
11
12
                var list = new List<Link>();
13
                for (var i = 0; i < groups.Count; i++)</pre>
1.5
                    groups[i].AppendToLinksList(list);
16
                return list;
18
            }
19
       }
   }
21
1.2
     ./csharp/Platform.Communication.Protocol.Lino/IListExtensions.cs
   using Platform.Collections;
   using System;
2
   using System.Collections.Generic;
using System.Linq;
4
   using System.Runtime.CompilerServices;
   #pragma warning disable CS1591 // Missing XML comment for publicly visible type or member
   namespace Platform.Communication.Protocol.Lino
9
10
        public static class IListExtensions
12
            [MethodImpl(MethodImplOptions.AggressiveInlining)]
13
            public static string Format(this IList<Link> links) => string.Join(Environment.NewLine,
14
               links.Select(1 => 1.ToString()));
1.5
            [MethodImpl(MethodImplOptions.AggressiveInlining)]
16
            public static string Format(this IList<Link> links, bool lessParentheses)
18
                if (lessParentheses == false)
19
20
21
                    return links.Format();
                }
22
                else
23
24
                    return string.Join(Environment.NewLine, links.Select(1 =>
25
                     → 1.ToString().TrimSingle('(').TrimSingle(')')));
                }
            }
27
        }
28
   }
29
     ./csharp/Platform.Communication.Protocol.Lino/Link.cs
   using System;
         System Collections Generic;
   using
   using System.Runtime.CompilerServices;
3
   using System. Text;
         Platform.Collections;
5
   using
   using Platform.Collections.Lists;
   #pragma warning disable CS1591 // Missing XML comment for publicly visible type or member
   namespace Platform.Communication.Protocol.Lino
10
   {
11
12
        public struct Link : IEquatable<Link>
13
            public string Id
15
                [MethodImpl(MethodImplOptions.AggressiveInlining)]
16
17
                [MethodImpl(MethodImplOptions.AggressiveInlining)]
18
19
20
            public IList<Link> Values
```

```
{
    [MethodImpl(MethodImplOptions.AggressiveInlining)]
    [MethodImpl(MethodImplOptions.AggressiveInlining)]
    set;
}
[MethodImpl(MethodImplOptions.AggressiveInlining)]
public Link(string id, IList<Link> values) => (Id, Values) = (id, values);
[MethodImpl(MethodImplOptions.AggressiveInlining)]
public Link(IList<Link> values) : this(null, values) { }
[MethodImpl(MethodImplOptions.AggressiveInlining)]
public Link(params Link[] values) : this(null, values) { }
[MethodImpl(MethodImplOptions.AggressiveInlining)]
public Link(string id) : this(id, null) { }
[MethodImpl(MethodImplOptions.AggressiveInlining)]
public override string ToString() => Values.IsNullOrEmpty() ? $\bar{\sqrt{Id}}\)" :

→ GetLinkValuesString();
[MethodImpl(MethodImplOptions.AggressiveInlining)]
private string GetLinkValuesString() => Id == null ? $\frac{\$}{\}"({GetValuesString()})" :
   $\"({Id}: {GetValuesString()})\";
[MethodImpl(MethodImplOptions.AggressiveInlining)]
public string GetValuesString()
    var sb = new StringBuilder();
    for (int i = 0; i < Values.Count; i++)</pre>
        if (i > 0)
        {
            sb.Append(' ');
        sb.Append(GetValueString(Values[i]));
    }
    return sb.ToString();
}
[MethodImpl(MethodImplOptions.AggressiveInlining)]
public Link Simplify()
    if (Values.IsNullOrEmpty())
    {
        return this;
    }
    else if (Values.Count == 1)
        return Values[0];
    }
    else
        var newValues = new Link[Values.Count];
        for (int i = 0; i < Values.Count; i++)</pre>
        {
            newValues[i] = Values[i].Simplify();
        return new Link(Id, newValues);
    }
}
[MethodImpl(MethodImplOptions.AggressiveInlining)]
public Link Combine(Link other) => new Link(this, other);
[MethodImpl(MethodImplOptions.AggressiveInlining)]
public static string GetValueString(Link value) => value.ToLinkOrIdString();
[MethodImpl(MethodImplOptions.AggressiveInlining)]
public string ToLinkOrIdString() => Values.IsNullOrEmpty() ? Id : ToString();
[MethodImpl(MethodImplOptions.AggressiveInlining)]
public static implicit operator Link(string value) => new Link(value);
[MethodImpl(MethodImplOptions.AggressiveInlining)]
public static implicit operator Link(ValueTuple<string, IList<Link>> value) => new

→ Link(value.Item1, value.Item2);
```

23

2.5

27

28 29

30

32

33

34

37 38

39

40

42

43

45

46

47

48

49 50

51

52 53

54

56 57

58

60

61 62

63

64 65

66

67

68

69 70

71

72

7.3

74 75

76

77

78

79 80

81

82

84

86

87

88

89

91

92 93

94

95 96

97

```
[MethodImpl(MethodImplOptions.AggressiveInlining)]
            public override bool Equals(object obj) => obj is Link link ? Equals(link) : false;
101
             [MethodImpl(MethodImplOptions.AggressiveInlining)]
103
            public override int GetHashCode() => (Id, Values.GenerateHashCode()).GetHashCode();
104
105
             [MethodImpl(MethodImplOptions.AggressiveInlining)]
106
            public bool Equals(Link other) => Id == other.Id && Values.EqualTo(other.Values);
107
108
             [MethodImpl(MethodImplOptions.AggressiveInlining)]
109
            public static bool operator ==(Link left, Link right) => left.Equals(right);
110
111
             [MethodImpl(MethodImplOptions.AggressiveInlining)]
112
            public static bool operator !=(Link left, Link right) => !(left == right);
114
115
     ./csharp/Platform.Communication.Protocol.Lino/LinksGroup.cs
    using System;
    using System.Collections.Generic;
    using System.Runtime.CompilerServices;
 3
    using Platform.Collections;
 4
    using Platform.Collections.Lists;
    #pragma warning disable CS1591 // Missing XML comment for publicly visible type or member
    namespace Platform.Communication.Protocol.Lino
 9
10
        public struct LinksGroup : IEquatable<LinksGroup>
11
12
            public Link Link
13
14
                 [MethodImpl(MethodImplOptions.AggressiveInlining)]
15
                 [MethodImpl(MethodImplOptions.AggressiveInlining)]
17
                 set;
18
            }
19
20
            public IList<LinksGroup> Groups
21
22
23
                 [MethodImpl(MethodImplOptions.AggressiveInlining)]
24
                 [MethodImpl(MethodImplOptions.AggressiveInlining)]
25
26
                 set;
            }
27
28
             [MethodImpl(MethodImplOptions.AggressiveInlining)]
29
            public LinksGroup(Link link, IList<LinksGroup> groups)
30
31
                 Link = link;
32
                 Groups = groups;
33
             }
34
35
             [MethodImpl(MethodImplOptions.AggressiveInlining)]
36
            public LinksGroup(Link link) : this(link, null) { }
38
             [MethodImpl(MethodImplOptions.AggressiveInlining)]
39
            public static implicit operator List<Link>(LinksGroup value) => value.ToLinksList();
41
             [MethodImpl(MethodImplOptions.AggressiveInlining)]
42
            public List<Link> ToLinksList()
43
44
                 var list = new List<Link>();
45
                 AppendToLinksList(list);
                 return list;
47
             }
49
             [MethodImpl(MethodImplOptions.AggressiveInlining)]
            public void AppendToLinksList(List<Link> list) => AppendToLinksList(list, Link, this);
51
52
             [MethodImpl(MethodImplOptions.AggressiveInlining)]
53
            public static void AppendToLinksList(List<Link> list, Link dependency, LinksGroup group)
54
55
                 list.Add(dependency);
                 var groups = group.Groups;
57
                 if (!groups.IsNullOrEmpty())
58
59
                     for (int i = 0; i < groups.Count; i++)</pre>
60
```

```
var innerGroup = groups[i];
62
                         AppendToLinksList(list, dependency.Combine(innerGroup.Link), innerGroup);
                     }
64
                }
65
            }
67
            [MethodImpl(MethodImplOptions.AggressiveInlining)]
68
            public override bool Equals(object obj) => obj is LinksGroup linksGroup ?

→ Equals(linksGroup) : false;

70
            [MethodImpl(MethodImplOptions.AggressiveInlining)]
71
            public override int GetHashCode() => (Link, Groups.GenerateHashCode()).GetHashCode();
72
73
            [MethodImpl(MethodImplOptions.AggressiveInlining)]
            public bool Equals(LinksGroup other) => Link == other.Link &&
7.5
                Groups.EqualTo(other.Groups);
            [MethodImpl(MethodImplOptions.AggressiveInlining)]
77
            public static bool operator ==(LinksGroup left, LinksGroup right) => left.Equals(right);
78
79
            [MethodImpl(MethodImplOptions.AggressiveInlining)]
80
            public static bool operator !=(LinksGroup left, LinksGroup right) => !(left == right);
81
        }
   }
83
1.5
     ./csharp/Platform.Communication.Protocol.Lino.Tests/ParserTests.cs
   using Xunit;
2
   namespace Platform.Communication.Protocol.Lino.Tests
3
4
        public static class ParserTests
6
            [Fact]
            public static void ParseAndStringifyTest()
9
                var source = 0"(papa (lovesMama: loves mama))
10
    (son lovesMama)
11
    (daughter lovesMama)
12
    (all (love mama))";
13
                var parser = new Parser();
14
15
                var links = parser.Parse(source);
                var target = links.Format();
16
                Assert. Equal(source, target);
17
            }
18
19
            [Fact]
20
            public static void ParseAndStringifyWithLessParenthesesTest()
21
22
                var source = 0"lovesMama: loves mama
   papa lovesMama
24
25
   son lovesMama
   daughter lovesMama
26
27
   all (love mama)";
                var parser = new Parser();
2.8
                var links = parser.Parse(source);
29
                var target = links.Format(lessParentheses: true);
30
                Assert.Equal(source, target);
31
            }
32
            [Fact]
34
            public static void SignificantWhitespaceTest()
35
                var source = @"
37
38
   users
        user1
39
            id
40
                43
42
            name
                first
43
                     John
44
                last
45
                     Williams
46
            location
47
                New York
48
49
            age
50
        user2
51
52
            id
                56
53
            name
```

```
first
55
56
                  middle
                      Petrovich
58
                  last
                      Ivanov
             location
61
                  Moscow
63
                  20";
64
                  var target = 0"(users)
65
    (users user1)
66
    ((users user1) id)
67
    (((users user1) id) 43)
68
    ((users user1) name)
69
    (((users user1) name) first)
    ((((users user1) name) first) John)
71
    (((users user1) name) last)
((((users user1) name) last) Williams)
72
    ((users user1) location)
74
    (((users user1) location) (New York))
75
    ((users user1) age)
76
    (((users user1) age) 23)
    (users user2)
78
    ((users user2) id)
79
    (((users user2) id) 56)
    ((users user2) name)
81
    (((users user2) name) first)
82
    ((((users user2) name) first) Igor)
83
    (((users user2) name) middle)
    ((((users user2) name) middle) Petrovich)
(((users user2) name) last)
((((users user2) name) last) Ivanov)
85
86
    ((users user2) location)
    (((users user2) location) Moscow)
89
    ((users user2) age)
    (((users user2) age) 20)";
                  var parser = new Parser();
92
                  var links = parser.Parse(source);
93
                  var formattedLinks = links.Format();
94
                  Assert.Equal(target, formattedLinks);
95
             }
96
        }
97
    }
```

Index

- ./csharp/Platform.Communication.Protocol.Lino.Tests/ParserTests.cs, 4
 ./csharp/Platform.Communication.Protocol.Lino/ILinksGroupListExtensions.cs, 1
 ./csharp/Platform.Communication.Protocol.Lino/IListExtensions.cs, 1
 ./csharp/Platform.Communication.Protocol.Lino/Link.cs, 1
 ./csharp/Platform.Communication.Protocol.Lino/LinksGroup.cs, 3