

```
1  /*-----
2  * Author:      Dan Cassidy
3  * Date:        2015-06-23
4  * Assignment:  cView-P4
5  * Source File: SimpleConvert.cs
6  * Language:    C#
7  * Course:      CSCI-C 490, C# Programming, MoWe 08:00
8  * Purpose:     Provides simplified variants and extensions of some Convert methods.
9  *-----*/
10
11 namespace System
12 {
13     public static class SimpleConvert
14     {
15         /*-----
16         * Name:      ToByte
17         * Type:       Method
18         * Purpose:   Attempts to convert the given parameter, but returns the default object value if
19         *             it fails for any reason.
20         * Input:     string value, containing the value on which conversion will be attempted.
21         * Output:    byte object containing either the converted value or the default object value.
22         *-----*/
23         public static byte ToByte(string value)
24         {
25             try
26             {
27                 return Convert.ToByte(value);
28             }
29             catch
30             {
31                 return default(byte);
32             }
33         }
34
35         /*-----
36         * Name:      ToByteN
37         * Type:       Method
38         * Purpose:   Attempts to convert the given parameter, but returns a null if it fails for any
39         *             reason.
40         * Input:     string value, containing the value on which conversion will be attempted.
41         * Output:    Nullable<byte> object containing either the converted value or a null.
42         *-----*/
43         public static Nullable<byte> ToByteN(string value)
44         {
45             try
46             {
47                 return Convert.ToByte(value);
48             }
49             catch
50             {
51                 return null;
52             }
53         }
54
55         /*-----
56         * Name:      ToDateTime
57         * Type:       Method
58         * Purpose:   Attempts to convert the given parameter, but returns the default object value if
59         *             it fails for any reason.
60         * Input:     string value, containing the value on which conversion will be attempted.
61         * Output:    DateTime object containing either the converted value or the default object
62         *             value.
63         *-----*/
64         public static DateTime ToDateTime(string value)
65         {
66             try
```

```

67         {
68             return Convert.ToDateTime(value);
69         }
70     catch
71     {
72         return default(DateTime);
73     }
74 }
75
76 /*-----
77  * Name:    ToDateTimeN
78  * Type:    Method
79  * Purpose: Attempts to convert the given parameter, but returns a null if it fails for any
80  *           reason.
81  * Input:   string value, containing the value on which conversion will be attempted.
82  * Output:  Nullable<DateTime> object containing either the converted value or a null.
83  -----*/
84 public static Nullable<DateTime> ToDateTimeN(string value)
85 {
86     try
87     {
88         return Convert.ToDateTime(value);
89     }
90     catch
91     {
92         return null;
93     }
94 }
95
96 /*-----
97  * Name:    ToDecimal
98  * Type:    Method
99  * Purpose: Attempts to convert the given parameter, but returns the default object value if
100  *           it fails for any reason.
101  * Input:   string value, containing the value on which conversion will be attempted.
102  * Output:  Decimal object containing either the converted value or the default object
103  *           value.
104  -----*/
105 public static decimal ToDecimal(string value)
106 {
107     try
108     {
109         return Convert.ToDecimal(value);
110     }
111     catch
112     {
113         return default(decimal);
114     }
115 }
116
117 /*-----
118  * Name:    ToDecimalN
119  * Type:    Method
120  * Purpose: Attempts to convert the given parameter, but returns a null if it fails for any
121  *           reason.
122  * Input:   string value, containing the value on which conversion will be attempted.
123  * Output:  Nullable<Decimal> object containing either the converted value or a null.
124  -----*/
125 public static Nullable<decimal> ToDecimalN(string value)
126 {
127     try
128     {
129         return Convert.ToDecimal(value);
130     }
131     catch
132     {

```

```

133         return null;
134     }
135 }
136
137 /*-----
138  * Name:    ToInt32
139  * Type:    Method
140  * Purpose: Attempts to convert the given parameter, but returns the default object value if
141  *           it fails for any reason.
142  * Input:   string value, containing the value on which conversion will be attempted.
143  * Output:  int object containing either the converted value or the default object value.
144  *-----*/
145 public static int ToInt32(string value)
146 {
147     try
148     {
149         return Convert.ToInt32(value);
150     }
151     catch
152     {
153         return default(int);
154     }
155 }
156
157 /*-----
158  * Name:    ToInt32N
159  * Type:    Method
160  * Purpose: Attempts to convert the given parameter, but returns a null if it fails for any
161  *           reason.
162  * Input:   string value, containing the value on which conversion will be attempted.
163  * Output:  Nullable<int> object containing either the converted value or a null.
164  *-----*/
165 public static Nullable<int> ToInt32N(string value)
166 {
167     try
168     {
169         return Convert.ToInt32(value);
170     }
171     catch
172     {
173         return null;
174     }
175 }
176
177 /*-----
178  * Name:    ToSingle
179  * Type:    Method
180  * Purpose: Attempts to convert the given parameter, but returns the default object value if
181  *           it fails for any reason.
182  * Input:   string value, containing the value on which conversion will be attempted.
183  * Output:  float object containing either the converted value or the default object value.
184  *-----*/
185 public static float ToSingle(string value)
186 {
187     try
188     {
189         return Convert.ToSingle(value);
190     }
191     catch
192     {
193         return default(float);
194     }
195 }
196
197 /*-----
198  * Name:    ToSingleN

```

```
199      * Type:      Method
200      * Purpose: Attempts to convert the given parameter, but returns the default object value if
201      *           it fails for any reason.
202      * Input:   string value, containing the value on which conversion will be attempted.
203      * Output:  Nullable<float> object containing either the converted value or a null.
204      -----*/
205      public static Nullable<float> ToSingleN(string value)
206      {
207          try
208          {
209              return Convert.ToSingle(value);
210          }
211          catch
212          {
213              return null;
214          }
215      }
216  }
217 }
218
```