

```
1  #region LICENSE
2  //
3  // Copyright (c) Microncode.com
4  //
5  // This source is subject to the Microsoft Public License.
6  // See https://opensource.org/licenses/ms-pl.html.
7  // All other rights reserved.
8  //
9  // THIS CODE AND INFORMATION IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, ?
10 // EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE IMPLIED
11 // WARRANTIES OF MERCHANTABILITY AND/OR FITNESS FOR A PARTICULAR PURPOSE.
12 //
13 #endregion
14
15 namespace CSID3TagsExample
16 {
17     using System;
18     using System.Drawing;
19     using System.Windows.Forms;
20
21     /// <summary>
22     /// The main example of using the CSID3Tags library.
23     /// </summary>
24     public partial class frmID3Tags : Form
25     {
26         private CSID3Tags.ID3Tags id3Tags1;
27
28         /// <summary>
29         /// Constructor.
30         /// </summary>
31         public frmID3Tags()
32         {
33             InitializeComponent();
34
35             this.id3Tags1 = new CSID3Tags.ID3Tags();
36         }
37
38         /// <summary>
39         /// Open the multimedia file. The supported format are: avi, mp3, mpc, ?
40         mp4, asf, aiff, aif, aifc, tta, wv, ogg, flac, spx, opus, ape, ?
41         xiph, wma, jpg, jpeg, png, mpg, mpeg and many other multimedia ?
42         files.
43         /// </summary>
44         /// <param name="multimedia_file"></param>
45         public void LoadFile(string multimedia_file)
46         {
47             try
48             {
49                 //Clean the form controls:
50                 ClearAllFields();
51
52                 //Open the media file:
53                 id3Tags1.Open(txtMediaFile.Text);
54
55                 //Get the ID3 tags:
```

```
53         txtTagTitle.Text = iD3Tags1.TagTitle;
54         txtTagAlbum.Text = iD3Tags1.TagAlbum;
55         txtTagComment.Text = iD3Tags1.TagComment;
56         txtTagCopyright.Text = iD3Tags1.TagCopyright;
57         txtTagYear.Text = iD3Tags1.TagYear;
58         txtTagLyrics.Text = iD3Tags1.TagLyrics;
59         txtTagTrackNum.Text = iD3Tags1.TagTrack.ToString();
60
61         cboTagRating.Text = iD3Tags1.RatingStars.ToString();
62
63         DisplayGenres();
64         DisplayArtists();
65         DisplayImages();
66         DisplayComposers();
67
68     }
69     catch (Exception ex)
70     {
71         Console.WriteLine(ex.Message);
72     }
73
74     finally
75     {
76
77     }
78
79 }
80
81 /// <summary>
82 /// Clear all the text boxes fields in the form
83 /// </summary>
84 private void ClearAllFields()
85 {
86     txtTagTitle.Text = "";
87     txtTagAlbum.Text = "";
88     txtTagComment.Text = "";
89     txtTagCopyright.Text = "";
90     txtTagYear.Text = "";
91     txtTagLyrics.Text = "";
92
93     lstTagArtists.Items.Clear();
94     lstTagComposers.Items.Clear();
95     lstTagGenres.Items.Clear();
96     lstTagImages.Items.Clear();
97
98     picImage.Image = null;
99 }
100
101 /// <summary>
102 /// Form load.
103 /// </summary>
104 /// <param name="sender"></param>
105 /// <param name="e"></param>
106 private void frmID3Tags_Load(object sender, EventArgs e)
107 {
108     iD3Tags1.UserName = "Your email";
```

```
109         iD3Tags1.UserKey = "Your registration key";
110     }
111
112     /// <summary>
113     /// Open multimedia file dialog box.
114     /// </summary>
115     /// <param name="sender"></param>
116     /// <param name="e"></param>
117     private void cmdOpenMediaFile_Click(object sender, EventArgs e)
118     {
119         if (openFileDialog1.ShowDialog() == System.Windows.Forms.DialogResult.OK)
120         {
121             txtMediaFile.Text = openFileDialog1.FileName;
122             LoadFile(txtMediaFile.Text);
123         }
124     }
125
126     /// <summary>
127     /// Save the text boxes fields of the form to the selected multimedia
128     /// file.
129     /// </summary>
130     /// <param name="sender"></param>
131     /// <param name="e"></param>
132     private void cmdSave_Click(object sender, EventArgs e)
133     {
134         try
135         {
136             //Set the title:
137             iD3Tags1.TagTitle = txtTagTitle.Text;
138
139             //Set the album name:
140             iD3Tags1.TagAlbum = txtTagAlbum.Text;
141
142             //Set the comment:
143             iD3Tags1.TagComment = txtTagComment.Text;
144
145             //Set the copyright:
146             iD3Tags1.TagCopyright = txtTagCopyright.Text;
147
148             //Set the lyrics:
149             iD3Tags1.TagLyrics = txtTagLyrics.Text;
150
151             //Set the rating tag:
152             float floatrating;
153             if (!float.TryParse(cboTagRating.Text, out floatrating))
154             {
155                 floatrating = 0;
156             }
157             iD3Tags1.RatingStars = floatrating;
158
159             //Set the year:
160             iD3Tags1.TagYear = txtTagYear.Text;
161
162             //Save:
163             iD3Tags1.Set();
```

```
163
164     }
165     catch (Exception ex)
166     {
167         Console.WriteLine(ex.Message);
168     }
169     finally
170     {
171     }
172 }
173
174 }
175
176 /// <summary>
177 /// Select and display the image from the images list when the selected index has been changed.
178 /// </summary>
179 /// <param name="sender"></param>
180 /// <param name="e"></param>
181 private void lstTagImages_SelectedIndexChanged(object sender, EventArgs e)
182 {
183     if (lstTagImages.Items.Count > 0)
184         picImage.Image = id3Tags1.TagImages[lstTagImages.SelectedIndex].Image;
185 }
186
187 /// <summary>
188 /// Select and display the image from the images list when the user clicked on an item in the images list.
189 /// </summary>
190 /// <param name="sender"></param>
191 /// <param name="e"></param>
192 private void lstTagImages_Click(object sender, EventArgs e)
193 {
194     if (lstTagImages.Items.Count > 0)
195         picImage.Image = id3Tags1.TagImages[lstTagImages.SelectedIndex].Image;
196 }
197
198 #region Artists
199
200 private void DisplayArtists()
201 {
202     //Clear the list
203     lstTagArtists.Items.Clear();
204
205     if (id3Tags1.TagArtists == null) return;
206
207     //Add all the items to the list
208     foreach (string artist in id3Tags1.TagArtists)
209         lstTagArtists.Items.Add(artist);
210
211     //Select the first item
212     if (lstTagArtists.Items.Count > 0)
213         lstTagArtists.SelectedIndex = 0;
```

```
214     }
215
216     private void cmdTagArtistsAdd_Click(object sender, EventArgs e)
217     {
218         string value = "";
219         if (Core.InputBox("Add artist", "Artist name:", ref value) == DialogResult.OK)
220         {
221             id3Tags1.TagArtists.Add(value);
222         }
223
224         DisplayArtists();
225     }
226
227     private void cmdTagArtistsRemove_Click(object sender, EventArgs e)
228     {
229         if (lstTagArtists.Items.Count == 0) return;
230
231         id3Tags1.TagArtists.RemoveAt(lstTagArtists.SelectedIndex);
232
233         DisplayArtists();
234     }
235
236     private void cmdTagArtistsUp_Click(object sender, EventArgs e)
237     {
238         int selected_index = 0;
239
240         if (lstTagArtists.Items.Count == 0 || lstTagArtists.SelectedIndex == 0)
241             return;
242
243         selected_index = lstTagArtists.SelectedIndex;
244
245         id3Tags1.TagArtists = Core.Switch(id3Tags1.TagArtists,
246             selected_index, selected_index - 1);
247         DisplayArtists();
248         lstTagArtists.SelectedIndex = selected_index - 1;
249     }
250
251     private void cmdTagArtistsDown_Click(object sender, EventArgs e)
252     {
253         int selected_index = 0;
254
255         if (lstTagArtists.SelectedIndex == (lstTagArtists.Items.Count - 1))
256             return;
257
258         selected_index = lstTagArtists.SelectedIndex;
259
260         id3Tags1.TagArtists = Core.Switch(id3Tags1.TagArtists,
261             selected_index, selected_index + 1);
262         DisplayArtists();
263         lstTagArtists.SelectedIndex = selected_index + 1;
264     }
265
266     #endregion Artists
```

```
265
266     #region Composers
267
268     /// <summary>
269     /// Add all the composers to the Composers list.
270     /// </summary>
271     private void DisplayComposers()
272     {
273         //Clear the list
274         lstTagComposers.Items.Clear();
275
276         if (id3Tags1.TagComposers == null) return;
277
278         //Add all the items to the list
279         foreach (string composer in id3Tags1.TagComposers)
280             lstTagComposers.Items.Add(composer);
281
282         //Select the first item
283         if (lstTagComposers.Items.Count > 0)
284             lstTagComposers.SelectedIndex = 0;
285     }
286
287     /// <summary>
288     /// Add new Composer
289     /// </summary>
290     /// <param name="sender"></param>
291     /// <param name="e"></param>
292     private void cmdComposersAdd_Click(object sender, EventArgs e)
293     {
294         string value = "";
295         if (Core.InputBox("Add composer", "Composer name:", ref value) == DialogResult.OK)
296         {
297             id3Tags1.TagComposers.Add(value);
298         }
299
300         DisplayComposers();
301     }
302
303     /// <summary>
304     /// Remove the selected Composer
305     /// </summary>
306     /// <param name="sender"></param>
307     /// <param name="e"></param>
308     private void cmdComposersRemove_Click(object sender, EventArgs e)
309     {
310         if (lstTagComposers.Items.Count == 0) return;
311
312         id3Tags1.TagComposers.RemoveAt(lstTagComposers.SelectedIndex);
313
314         DisplayComposers();
315     }
316
317     /// <summary>
318     /// Move the selected composer index in the list one stage up.
319     /// </summary>
```

```
320     /// <param name="sender"></param>
321     /// <param name="e"></param>
322     private void cmdTagComposersUp_Click(object sender, EventArgs e)
323     {
324         int selected_index = 0;
325
326         if (lstTagComposers.Items.Count == 0 ||
327             lstTagComposers.SelectedIndex == 0)
328             return;
329
330         selected_index = lstTagComposers.SelectedIndex;
331
332         id3Tags1.TagComposers = Core.Switch(id3Tags1.TagComposers,
333             selected_index, selected_index - 1);
334         DisplayComposers();
335         lstTagComposers.SelectedIndex = selected_index - 1;
336     }
337
338     /// <summary>
339     /// Move the selected composer index in the list one stage down.
340     /// </summary>
341     /// <param name="sender"></param>
342     /// <param name="e"></param>
343     private void cmdTagComposersDown_Click(object sender, EventArgs e)
344     {
345         int selected_index = 0;
346
347         if (lstTagComposers.SelectedIndex == (lstTagComposers.Items.Count
348             - 1))
349             return;
350
351         selected_index = lstTagComposers.SelectedIndex;
352
353         id3Tags1.TagComposers = Core.Switch(id3Tags1.TagComposers,
354             selected_index, selected_index + 1);
355         DisplayComposers();
356         lstTagComposers.SelectedIndex = selected_index + 1;
357     }
358
359     #endregion Composers
360
361     #region Genres
362
363     private void DisplayGenres()
364     {
365         //Clear the list
366         lstTagGenres.Items.Clear();
367
368         if (id3Tags1.TagGenres == null) return;
369
370         //Add all the items to the list
371         foreach (string genre in id3Tags1.TagGenres)
372             lstTagGenres.Items.Add(genre);
373
374         //Select the first item
375         if (lstTagGenres.Items.Count > 0)
```

```
372         lstTagGenres.SelectedIndex = 0;
373
374     }
375
376     private void cmdGenresAdd_Click(object sender, EventArgs e)
377     {
378         if (cboTagGenres.Text != "")
379             id3Tags1.TagGenres.Add(cboTagGenres.Text);
380
381         DisplayGenres();
382     }
383
384     private void cmdGenresRemove_Click(object sender, EventArgs e)
385     {
386         if (lstTagGenres.Items.Count == 0) return;
387
388         id3Tags1.TagGenres.RemoveAt(lstTagGenres.SelectedIndex);
389
390         DisplayGenres();
391     }
392
393
394
395     private void cmdTagGenresUp_Click(object sender, EventArgs e)
396     {
397         int selected_index = 0;
398
399         if (lstTagGenres.Items.Count == 0 || lstTagGenres.SelectedIndex == 0)
400             return;
401
402         selected_index = lstTagGenres.SelectedIndex;
403
404         id3Tags1.TagGenres = Core.Switch(id3Tags1.TagGenres, selected_index, selected_index - 1);
405         DisplayGenres();
406         lstTagGenres.SelectedIndex = selected_index - 1;
407     }
408
409     private void cmdTagGenresDown_Click(object sender, EventArgs e)
410     {
411         int selected_index = 0;
412
413         if (lstTagGenres.SelectedIndex == (lstTagGenres.Items.Count - 1))
414             return;
415
416         selected_index = lstTagGenres.SelectedIndex;
417
418         id3Tags1.TagGenres = Core.Switch(id3Tags1.TagGenres, selected_index, selected_index + 1);
419         DisplayGenres();
420         lstTagGenres.SelectedIndex = selected_index + 1;
421     }
422
423     #endregion Genres
424
```



```

425     #region Images
426
427     private void DisplayImages()
428     {
429         //Clear the list
430         lstTagImages.Items.Clear();
431
432         if (id3Tags1.TagImages == null) return;
433
434         //Add all the items to the list
435         foreach (var img in id3Tags1.TagImages)
436             lstTagImages.Items.Add(img.Mime_Type + "\t" + img.Type + "\t" +
437                                     + img.Description);
438
439         //Select the first item
440         if (lstTagImages.Items.Count > 0)
441         {
442             lstTagImages.SelectedIndex = 0;
443             picImage.Image = id3Tags1.TagImages
444                             [lstTagImages.SelectedIndex].Image;
445         }
446
447         /// <summary>
448         /// Add an image to the images tag of the multimedia file. You can add
449         /// an image from a file (AddImageFromFile) or from an Image object.
450         /// </summary>
451         /// <param name="sender"></param>
452         /// <param name="e"></param>
453         private void cmdImagesAdd_Click(object sender, EventArgs e)
454         {
455             openFileDialog1.Filter = "(Image Files)|*.jpg;*.jpeg;*.png|jpg, |
456                                     *.jpg|jpeg, | *.jpeg|png, | *.png";
457             if (openFileDialog1.ShowDialog() ==
458                 System.Windows.Forms.DialogResult.OK)
459             {
460                 //From a file:
461                 Image img = id3Tags1.AddImageFromFile
462                             (openFileDialog1.FileName);
463
464                 this.picImage.Image = img;
465
466                 //From an Image object:
467                 /*
468                 Image image = Image.FromFile(openFileDialog1.FileName);
469                 this.picImage.Image = image;
470                 string minetype = Core.GetMimeType(image);
471                 ImageTag imgobj = new ImageTag(image, minetype,
472                                                 "id3Tags1.Tagcom", CSID3TagsLib.ImageType.FrontCover);
473                 id3Tags1.TagImages.Add(imgobj);
474                 */
475             }
476
477             DisplayImages();
478         }
479     }

```

```
474
475
476     private void cmdImagesRemove_Click(object sender, EventArgs e)
477     {
478         if (lstTagImages.Items.Count == 0) return;
479
480         id3Tags1.TagImages.RemoveAt(lstTagImages.SelectedIndex);
481
482         DisplayImages();
483     }
484
485     private void cmdTagImagesUp_Click(object sender, EventArgs e)
486     {
487         int selected_index = 0;
488
489         if (lstTagImages.Items.Count == 0 || lstTagImages.SelectedIndex == 0)
490             return;
491
492         selected_index = lstTagImages.SelectedIndex;
493
494         id3Tags1.TagImages = Core.Switch(id3Tags1.TagImages,
495                                         selected_index, selected_index - 1);
496         DisplayImages();
497         lstTagImages.SelectedIndex = selected_index - 1;
498     }
499
500     private void cmdTagImagesDown_Click(object sender, EventArgs e)
501     {
502         int selected_index = 0;
503
504         if (lstTagImages.SelectedIndex == (lstTagImages.Items.Count - 1))
505             return;
506
507         selected_index = lstTagImages.SelectedIndex;
508
509         id3Tags1.TagImages = Core.Switch(id3Tags1.TagImages,
510                                         selected_index, selected_index + 1);
511
512         DisplayImages();
513         lstTagImages.SelectedIndex = selected_index + 1;
514     }
515
516     #endregion Images
517
518     /// <summary>
519     /// Clear all the ID3 tags from the multimedia file.
520     /// </summary>
521     /// <param name="sender"></param>
522     /// <param name="e"></param>
523     private void cmdClear_Click(object sender, EventArgs e)
524     {
525         if (txtMediaFile.Text == "") return;
526
527         DialogResult dialogResult = MessageBox.Show("'Clear All Tags' will
528             delete all the ID3 tags in the multimedia file, continue?",
```

```
        "Warning", MessageBoxButtons.OKCancel);
526     if (dialogResult == DialogResult.OK)
527     {
528         id3Tags1.ClearAllTags();
529         LoadFile(txtMediaFile.Text);
530
531         picImage.Image = null;
532     }
533
534 }
535
536 /// <summary>
537 /// Order a license.
538 /// </summary>
539 /// <param name="sender"></param>
540 /// <param name="e"></param>
541 private void cmdOrderLicense_Click(object sender, EventArgs e)
542 {
543     /// In order to get your own UserName and RegKey and to distribute ↗
544     /// this library
545     /// with your own projects for commercial or any other purpose, ↗
546     /// please order a license at:
547     /// https://www.microncode.com/developers/cs-id3-tags/?cmd=order
548     System.Diagnostics.Process.Start("https://www.microncode.com/
549     developers/cs-id3-tags/?cmd=order");
550 }
551 }
```