```csharp
#region LICENSE
//
// Copyright (c) Microncode.com
//
// This source is subject to the Microsoft Public License.
// See https://opensource.org/licenses/ms-pl.html.
// All other rights reserved.
//
// THIS CODE AND INFORMATION IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, 
// EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE IMPLIED
// WARRANTIES OF MERCHANTABILITY AND/OR FITNESS FOR A PARTICULAR PURPOSE.
//
#endregion

namespace CSID3TagsExample
{
    using System;
    using System.Drawing;
    using System.Windows.Forms;

    /// <summary>
    /// The main example of using the CSID3Tags library.
    /// </summary>
    public partial class frmID3Tags : Form
    {
        private CSID3Tags.ID3Tags iD3Tags1;

        /// <summary>
        /// Constructor.
        /// </summary>
        public frmID3Tags()
        {
            InitializeComponent();

            this.iD3Tags1 = new CSID3Tags.ID3Tags();
        }

        /// <summary>
        /// Open the multimedia file. The supported format are: avi, mp3, mpc, mp4, asf, aiff, aif, aifc, tta, wv, ogg, flac, spx, opus, ape, xiph, wma, jpg, jpeg, png, mpg, mpeg and many other multimedia files.
        /// </summary>
        /// <param name="multimedia_file"></param>
        public void LoadFile(string multimedia_file)
        {
            try
            {
                //Clean the form controls:
                ClearAllFields();

                //Open the media file:
                iD3Tags1.Open(txtMediaFile.Text);

                //Get the ID3 tags:
                txtTagTitle.Text = iD3Tags1.TagTitle;
                txtTagAlbum.Text = iD3Tags1.TagAlbum;
                txtTagComment.Text = iD3Tags1.TagComment;
                txtTagCopyright.Text = iD3Tags1.TagCopyright;
                txtTagYear.Text = iD3Tags1.TagYear;
                txtTagLyrics.Text = iD3Tags1.TagLyrics;
                txtTagTrackNum.Text = iD3Tags1.TagTrack.ToString();

                cboTagRating.Text = iD3Tags1.RatingStars.ToString();

                DisplayGenres();
                DisplayArtists();
                DisplayImages();
                DisplayComposers();

            }
            catch (Exception ex)
            {
                Console.WriteLine(ex.Message);
            }

            finally
            {

            }

        }

        /// <summary>
        /// Clear all the text boxes fields in the form
        /// </summary>
        private void ClearAllFields()
        {
            txtTagTitle.Text = "";
            txtTagAlbum.Text = "";
            txtTagComment.Text = "";
            txtTagCopyright.Text = "";
            txtTagYear.Text = "";
            txtTagLyrics.Text = "";

            lstTagArtists.Items.Clear();
            lstTagComposers.Items.Clear();
            lstTagGenres.Items.Clear();
            lstTagImages.Items.Clear();

            picImage.Image = null;
        }

        /// <summary>
        /// Form load.
        /// </summary>
        /// <param name="sender"></param>
        /// <param name="e"></param>
        private void frmID3Tags_Load(object sender, EventArgs e)
        {
            iD3Tags1.UserName = "Your email";
            iD3Tags1.UserKey = "Your registration key";
        }

        /// <summary>
        /// Open multimedia file dialog box.
        /// </summary>
        /// <param name="sender"></param>
        /// <param name="e"></param>
        private void cmdOpenMediaFile_Click(object sender, EventArgs e)
        {
            if (openFileDialog1.ShowDialog() == System.Windows.Forms.DialogResult.OK)
            {
                txtMediaFile.Text = openFileDialog1.FileName;
                LoadFile(txtMediaFile.Text);
            }
        }

        /// <summary>
        /// Save the text boxes fields of the form to the selected multimedia file.
        /// </summary>
        /// <param name="sender"></param>
        /// <param name="e"></param>
        private void cmdSave_Click(object sender, EventArgs e)
        {
            try
            {
                //Set the title:
                iD3Tags1.TagTitle = txtTagTitle.Text;

                //Set the album name:
                iD3Tags1.TagAlbum = txtTagAlbum.Text;

                //Set the comment:
                iD3Tags1.TagComment = txtTagComment.Text;

                //Set the copyright:
                iD3Tags1.TagCopyright = txtTagCopyright.Text;

                //Set the lyrics:
                iD3Tags1.TagLyrics = txtTagLyrics.Text;

                //Set the rating tag:
                float floatrating;
                if (!float.TryParse(cboTagRating.Text, out floatrating))
                {
                    floatrating = 0;
                }
                iD3Tags1.RatingStars = floatrating;

                //Set the year:
                iD3Tags1.TagYear = txtTagYear.Text;

                //Save:
                iD3Tags1.Set();

            }
            catch (Exception ex)
            {
                Console.WriteLine(ex.Message);
            }
            finally
            {

            }

        }

        /// <summary>
        /// Select and display the image from the images list when the selected index has been changed.
        /// </summary>
        /// <param name="sender"></param>
        /// <param name="e"></param>
        private void lstTagImages_SelectedIndexChanged(object sender, EventArgs e)
        {
            if (lstTagImages.Items.Count > 0)
                picImage.Image = iD3Tags1.TagImages[lstTagImages.SelectedIndex].Image;
        }

        /// <summary>
        /// Select and display the image from the images list when the user clicked on an item in the images list.
        /// </summary>
        /// <param name="sender"></param>
        /// <param name="e"></param>
        private void lstTagImages_Click(object sender, EventArgs e)
        {
            if (lstTagImages.Items.Count > 0)
                picImage.Image = iD3Tags1.TagImages[lstTagImages.SelectedIndex].Image;
        }

        #region Artists

        private void DisplayArtists()
        {
            //Clear the list
            lstTagArtists.Items.Clear();

            if (iD3Tags1.TagArtists == null) return;

            //Add all the items to the list
            foreach (string artist in iD3Tags1.TagArtists)
                lstTagArtists.Items.Add(artist);

            //Select the first item
            if (lstTagArtists.Items.Count > 0)
                lstTagArtists.SelectedIndex = 0;
        }

        private void cmdTagArtistsAdd_Click(object sender, EventArgs e)
        {
            string value = "";
            if (Core.InputBox("Add artist", "Artist name:", ref value) == DialogResult.OK)
            {
                iD3Tags1.TagArtists.Add(value);
            }

            DisplayArtists();
        }

        private void cmdTagArtistsRemove_Click(object sender, EventArgs e)
        {
            if (lstTagArtists.Items.Count == 0) return;

            iD3Tags1.TagArtists.RemoveAt(lstTagArtists.SelectedIndex);

            DisplayArtists();
        }

        private void cmdTagArtistsUp_Click(object sender, EventArgs e)
        {
            int selected_index = 0;

            if (lstTagArtists.Items.Count == 0 || lstTagArtists.SelectedIndex == 0)
                return;

            selected_index = lstTagArtists.SelectedIndex;

            iD3Tags1.TagArtists = Core.Switch(iD3Tags1.TagArtists, selected_index, selected_index - 1);
            DisplayArtists();
            lstTagArtists.SelectedIndex = selected_index - 1;
        }

        private void cmdTagArtistsDown_Click(object sender, EventArgs e)
        {
            int selected_index = 0;

            if (lstTagArtists.SelectedIndex == (lstTagArtists.Items.Count - 1))
                return;

            selected_index = lstTagArtists.SelectedIndex;

            iD3Tags1.TagArtists = Core.Switch(iD3Tags1.TagArtists, selected_index, selected_index + 1);
            DisplayArtists();
            lstTagArtists.SelectedIndex = selected_index + 1;
        }

        #endregion Artists

        #region Composers

        /// <summary>
        /// Add all the composers to the Composers list.
        /// </summary>
        private void DisplayComposers()
        {
            //Clear the list
            lstTagComposers.Items.Clear();

            if (iD3Tags1.TagComposers == null) return;

            //Add all the items to the list
            foreach (string composer in iD3Tags1.TagComposers)
                lstTagComposers.Items.Add(composer);

            //Select the first item
            if (lstTagComposers.Items.Count > 0)
                lstTagComposers.SelectedIndex = 0;
        }

        /// <summary>
        /// Add new Composer
        /// </summary>
        /// <param name="sender"></param>
        /// <param name="e"></param>
        private void cmdComposersAdd_Click(object sender, EventArgs e)
        {
            string value = "";
            if (Core.InputBox("Add composer", "Composer name:", ref value) == DialogResult.OK)
            {
                iD3Tags1.TagComposers.Add(value);
            }

            DisplayComposers();
        }

        /// <summary>
        /// Remove the selected Composer
        /// </summary>
        /// <param name="sender"></param>
        /// <param name="e"></param>
        private void cmdComposersRemove_Click(object sender, EventArgs e)
        {
            if (lstTagComposers.Items.Count == 0) return;

            iD3Tags1.TagComposers.RemoveAt(lstTagComposers.SelectedIndex);

            DisplayComposers();
        }

        /// <summary>
        /// Move the selected composer index in the list one stage up.
        /// </summary>
        /// <param name="sender"></param>
        /// <param name="e"></param>
        private void cmdTagComposersUp_Click(object sender, EventArgs e)
        {
            int selected_index = 0;

            if (lstTagComposers.Items.Count == 0 || lstTagComposers.SelectedIndex == 0)
                return;

            selected_index = lstTagComposers.SelectedIndex;

            iD3Tags1.TagComposers = Core.Switch(iD3Tags1.TagComposers, selected_index, selected_index - 1);
            DisplayComposers();
            lstTagComposers.SelectedIndex = selected_index - 1;
        }

        /// <summary>
        /// Move the selected composer index in the list one stage down.
        /// </summary>
        /// <param name="sender"></param>
        /// <param name="e"></param>
        private void cmdTagComposersDown_Click(object sender, EventArgs e)
        {
            int selected_index = 0;

            if (lstTagComposers.SelectedIndex == (lstTagComposers.Items.Count - 1))
                return;

            selected_index = lstTagComposers.SelectedIndex;

            iD3Tags1.TagComposers = Core.Switch(iD3Tags1.TagComposers, selected_index, selected_index + 1);
            DisplayComposers();
            lstTagComposers.SelectedIndex = selected_index + 1;
        }

        #endregion Composers

        #region Genres

        private void DisplayGenres()
        {
            //Clear the list
            lstTagGenres.Items.Clear();

            if (iD3Tags1.TagGenres == null) return;

            //Add all the items to the list
            foreach (string genre in iD3Tags1.TagGenres)
                lstTagGenres.Items.Add(genre);

            //Select the first item
            if (lstTagGenres.Items.Count > 0)
```

```
                lstTagGenres.SelectedIndex = 0;

        }

        private void cmdGenresAdd_Click(object sender, EventArgs e)
        {
            if (cboTagGenres.Text != "")
                iD3Tags1.TagGenres.Add(cboTagGenres.Text);

            DisplayGenres();
        }

        private void cmdGenresRemove_Click(object sender, EventArgs e)
        {
            if (lstTagGenres.Items.Count == 0) return;

            iD3Tags1.TagGenres.RemoveAt(lstTagGenres.SelectedIndex);

            DisplayGenres();
        }



        private void cmdTagGenresUp_Click(object sender, EventArgs e)
        {
            int selected_index = 0;

            if (lstTagGenres.Items.Count == 0 || lstTagGenres.SelectedIndex == 0)
                return;

            selected_index = lstTagGenres.SelectedIndex;

            iD3Tags1.TagGenres = Core.Switch(iD3Tags1.TagGenres, selected_index, selected_index - 1);
            DisplayGenres();
            lstTagGenres.SelectedIndex = selected_index - 1;
        }

        private void cmdTagGenresDown_Click(object sender, EventArgs e)
        {
            int selected_index = 0;

            if (lstTagGenres.SelectedIndex == (lstTagGenres.Items.Count - 1))
                return;

            selected_index = lstTagGenres.SelectedIndex;

            iD3Tags1.TagGenres = Core.Switch(iD3Tags1.TagGenres, selected_index, selected_index + 1);
            DisplayGenres();
            lstTagGenres.SelectedIndex = selected_index + 1;
        }

        #endregion Genres

        #region Images

        private void DisplayImages()
        {
            //Clear the list
            lstTagImages.Items.Clear();

            if (iD3Tags1.TagImages == null) return;

            //Add all the items to the list
            foreach (var img in iD3Tags1.TagImages)
                lstTagImages.Items.Add(img.Mine_Type + "\t" + img.Type + "\t" + img.Description);

            //Select the first item
            if (lstTagImages.Items.Count > 0)
            {
                lstTagImages.SelectedIndex = 0;
                picImage.Image = iD3Tags1.TagImages[lstTagImages.SelectedIndex].Image;
            }

        }

        /// <summary>
        /// Add an image to the images tag of the multimedia file. You can add an image from a file (AddImageFromFile) or from an Image object.
        /// </summary>
        /// <param name="sender"></param>
        /// <param name="e"></param>
        private void cmdImagesAdd_Click(object sender, EventArgs e)
        {
            openFileDialog1.Filter = "(Image Files)|*.jpg;*.jpeg;*.png|jpg, | *.jpg|jpeg, | *.jpeg|png, | *.png";
            if (openFileDialog1.ShowDialog() == System.Windows.Forms.DialogResult.OK)
            {
                //From a file:
                Image img = iD3Tags1.AddImageFromFile(openFileDialog1.FileName);

                this.picImage.Image = img;

                //From an Image object:
                /*
                Image image = Image.FromFile(openFileDialog1.FileName);
                this.picImage.Image = image;
                string minetype = Core.GetMineType(image);
                ImageTag imgobj = new ImageTag(image, minetype, "iD3Tags1.Tagcom", CSID3TagsLib.ImageType.FrontCover);
                iD3Tags1.TagImages.Add(imgobj);
                */
            }

            DisplayImages();
        }
```

```csharp


        private void cmdImagesRemove_Click(object sender, EventArgs e)
        {
            if (lstTagImages.Items.Count == 0) return;

            iD3Tags1.TagImages.RemoveAt(lstTagImages.SelectedIndex);

            DisplayImages();
        }

        private void cmdTagImagesUp_Click(object sender, EventArgs e)
        {
            int selected_index = 0;

            if (lstTagImages.Items.Count == 0 || lstTagImages.SelectedIndex == 0)
                return;

            selected_index = lstTagImages.SelectedIndex;

            iD3Tags1.TagImages = Core.Switch(iD3Tags1.TagImages, selected_index, selected_index - 1);
            DisplayImages();
            lstTagImages.SelectedIndex = selected_index - 1;
        }

        private void cmdTagImagesDown_Click(object sender, EventArgs e)
        {
            int selected_index = 0;

            if (lstTagImages.SelectedIndex == (lstTagImages.Items.Count - 1))
                return;

            selected_index = lstTagImages.SelectedIndex;

            iD3Tags1.TagImages = Core.Switch(iD3Tags1.TagImages, selected_index, selected_index + 1);

            DisplayImages();
            lstTagImages.SelectedIndex = selected_index + 1;
        }

        #endregion Images

        /// <summary>
        /// Clear all the ID3 tags from the multimedia file.
        /// </summary>
        /// <param name="sender"></param>
        /// <param name="e"></param>
        private void cmdClear_Click(object sender, EventArgs e)
        {
            if (txtMediaFile.Text == "") return;

            DialogResult dialogResult = MessageBox.Show("'Clear All Tags' will delete all the ID3 tags in the multimedia file, continue?",
              "Warning", MessageBoxButtons.OKCancel);
            if (dialogResult == DialogResult.OK)
            {
                iD3Tags1.ClearAllTags();
                LoadFile(txtMediaFile.Text);

                picImage.Image = null;
            }

        }

        /// <summary>
        /// Order a license.
        /// </summary>
        /// <param name="sender"></param>
        /// <param name="e"></param>
        private void cmdOrderLicense_Click(object sender, EventArgs e)
        {
            /// In order to get your own UserName and RegKey and to distribute this library
            /// with your own projects for commercial or any other purpose, please order a license at:
            /// https://www.microncode.com/developers/cs-id3-tags/?cmd=order

            System.Diagnostics.Process.Start("https://www.microncode.com/developers/cs-id3-tags/?cmd=order");
        }
    }
}
```