

```
using System;
using System.Collections;
using System.ComponentModel;
using System.Drawing;
using System.Data;
using System.Windows.Forms;
using SB54_CSAI.DatabaseAccess;
using SB54_CSAI.Genres;
using System.Data.OleDb;
using System.Net;
using System.IO;
using SB54_CSAI.MP3Lib;
using System.Management;

namespace SB54_CSAI.MediaLibrary
{
    #region uctMP3Editor CLASS
    /// <summary>
    /// uctMediaLibrary is a sub class of a <see cref="System.Windows.Forms.UserControl">
    User Control </see>
    /// that provides host the 2 treeviews <see cref="SB54_CSAI.MediaLibrary.FolderTree">
    FolderTree </see>
    /// and <see cref="SB54_CSAI.MediaLibrary.GenreTree">GenreTree </see> and provides a
    means of viewing
    /// the audio files associated with the 2 treeviews. The user may also add/remove
    folders that are logged
    /// to the <see cref="SB54_CSAI.DatabaseAccess.DBAccess"> database</see> and add folder
    audio files to
    /// the <see cref="SB54_CSAI.MediaPlayer.MediaPlayer">Media Player </see>from here.</
    summary>
    public class uctMediaLibrary : System.Windows.Forms.UserControl
    {
        #region Instance Fields
        //Instance fields
        private System.Windows.Forms.Panel oPnlLeft;
        private System.Windows.Forms.Splitter oSpltV;
        private System.Windows.Forms.Panel oPnlLeftTop;
        private System.Windows.Forms.Button oBtnFolders;
        private System.Windows.Forms.Button oBtnGenres;
        private System.Windows.Forms.ImageList oImgList16;
        private System.Windows.Forms.Panel oPnlFill;
        private System.Windows.Forms.Button oBtnCancel;
        private System.ComponentModel.IContainer components;
        private System.Windows.Forms.ImageList oImgListFldrs;
        private System.Windows.Forms.FolderBrowserDialog oFolderDialog1;
        private System.Windows.Forms.ContextMenu oCtxtMenuFT;
        private System.Windows.Forms.MenuItem oMnuDeleteFolder;
        private System.Windows.Forms.Panel oPnlFillTop;
        private System.Windows.Forms.Splitter oSpltH;
        private System.Windows.Forms.Panel oPnlFillBottomSmall;
        private System.Windows.Forms.Button oBtnAddFolder;
        private System.Windows.Forms.Button oBtnShares;
        private System.Windows.Forms.PictureBox oPicShareSmall;
        private System.Windows.Forms.Panel oPnlRTT;
        private System.Windows.Forms.Label oPnlFillBannerTitle1;
        private System.Windows.Forms.Panel oPnlFillBanner;
        private System.Windows.Forms.PictureBox oPicShareLarge;
        private System.Windows.Forms.Label oPnlFillBannerTitle2;
        private System.Windows.Forms.Label oLblScan;
        private System.Windows.Forms.Panel panell1;
        private System.Windows.Forms.Panel oPnlDetails;
        private System.Windows.Forms.Panel oPnlDetailsbanner;
        private System.Windows.Forms.Label oLblDetailsTitle;
        private System.Windows.Forms.ContextMenu oCtxtDetails;
        private System.Windows.Forms.MenuItem oMnuDetails;
        private System.Windows.Forms.MenuItem oMnuList;
        private System.Windows.Forms.MenuItem oMnuIcon;
        private System.Windows.Forms.MenuItem oMnuThumb;
        private System.Windows.Forms.Button oBtnDetails;
        private System.Windows.Forms.ImageList oImgLV16;
        private System.Windows.Forms.ImageList oImgLV60;
        private System.Windows.Forms.ImageList oImgLV98;
```

```

private System.Windows.Forms.ColumnHeader columnHeader1;
private System.Windows.Forms.ColumnHeader columnHeader2;
private System.Windows.Forms.ColumnHeader columnHeader3;
private System.Windows.Forms.ColumnHeader columnHeader4;
private System.Windows.Forms.Panel oPnlDetails_Top;
private System.Windows.Forms.Label oPnlDetails_TopTXT;
private System.Windows.Forms.Panel oPnlDetailsL;
private System.Windows.Forms.Panel oPnlDetailsFill;
private System.Windows.Forms.Panel oPnlDetailsFillL;
private System.Windows.Forms.PictureBox oPnlDetailsLogo;
private System.Windows.Forms.Label oLblLocation;
private System.Windows.Forms.Label oLblTitle;
private System.Windows.Forms.Label oLblAlbum;
private System.Windows.Forms.Label oLblArtist;
private System.Windows.Forms.Label oLblGenre;
private System.Windows.Forms.Label oLblYear;
private System.Windows.Forms.Label oLblComment;
private System.Windows.Forms.Panel oPnlDetailsFillR;
private System.Windows.Forms.LinkLabel oLnkLocationTXT;
private System.Windows.Forms.Label oLblTitleTXT;
private System.Windows.Forms.LinkLabel oLnkAlbumTXT;
private System.Windows.Forms.LinkLabel oLnkArtistTXT;
private System.Windows.Forms.LinkLabel oLnkGenreTXT;
private System.Windows.Forms.Label oLblYearTXT;
private System.Windows.Forms.Label oLblCommentTXT;
private System.Windows.Forms.ContextMenu oCtxtMenuGT;
private System.Windows.Forms.MenuItem oMnuAddToTrackListFromFT;
private System.Windows.Forms.MenuItem oMnuAddToTrackListFromGT;
private System.Windows.Forms.ToolTip toolTip1;
private GenreTree oTvGenres;
private FolderTree otvFolders;
private TreeNode tvnFolderSelect;
private TreeNode tvnGenreSelect;
private MP3 workingMP3;
private bool ShowDetails =false;
private DBAccess db;
private static ArrayList NewShares = new ArrayList();
private sb54_CSAI.MiscFormComponents.MenuIcons oMnuIcons1;
private static ArrayList TopLevelFolders = new ArrayList();
private System.Windows.Forms.ListView oLvTracks;
private System.Windows.Forms.Button oBtnShowDetails;
private SortedList fileTypeMap = new SortedList();
private bool foldersJustAdded = false;
/// <summary>
/// collection of local hard drives for the PC
/// </summary>
public static ArrayList localDrives = new ArrayList();
/// <summary>
/// AddFilesHandler delegate for the Scan addFiles
/// </summary>
public delegate void AddFilesHandler(object sender, AddFilesEventArgs e);
/// <summary>
/// Raised if the user decides to add a folder to the playe list</see>
/// </summary>
public event AddFilesHandler addFiles;
#endregion
#region Public Constructor
/// <summary>
/// Creates a new uctMediaLibrary, and sets up the 2 TreeViews FolderTree and
GenreTree
/// </summary>
public uctMediaLibrary()
{
    // This call is required by the Windows.Forms Form Designer.
    InitializeComponent();

    setupTrees();
    setupFileTypesMap();
}
#endregion
#region Public Methods/Properties
/// <summary>

```

```

    /// Gets an ArrayList which represents the database stored Root Folders
    /// </summary>
    /// <returns>An ArrayList which represents the database stored Root Folders</
returns>
public static ArrayList TopFolders
{
    get
    {
        return TopLevelFolders;
    }
}

    /// <summary>
    /// Sets the NewShares ArrayList to be the values of the input ArrayList
    /// </summary>
public static ArrayList newShares
{
    set
    {
        NewShares = value;
    }
}

    /// <summary>
    /// Raises the OnAddFiles event for the uctMediaLibrary user control. Any users of
the
    /// uctMediaLibrary may now subscribe to this event using a AddFilesHandler
    /// delegate.
    /// </summary>
    /// <param name="e">The <see cref="SB54_CSAI.MediaLibrary.AddFilesEventArgs"> add
file event arguments</see></param>
public virtual void OnAddFiles(AddFilesEventArgs e)
{
    if (addFiles != null)
    {
        // Invokes the delegates.
        addFiles(this, e);
    }
}

    /// <summary>
    /// Clean up any resources being used.
    /// </summary>
    /// <param name="disposing">true indicates the object should be disposed</param>
protected override void Dispose( bool disposing )
{
    if( disposing )
    {
        if( components != null ) { components.Dispose(); }
    }
    base.Dispose( disposing );
}
#endregion
#region Component Designer generated code
    /// <summary>
    /// Required method for Designer support - do not modify
    /// the contents of this method with the code editor.
    /// </summary>
private void InitializeComponent()
{
    this.components = new System.ComponentModel.Container();
    System.Resources.ResourceManager resources = new System.Resources.
ResourceManager( typeof( uctMediaLibrary ) );
    this.columnHeader1 = new System.Windows.Forms.ColumnHeader();
    this.columnHeader2 = new System.Windows.Forms.ColumnHeader();
    this.columnHeader3 = new System.Windows.Forms.ColumnHeader();
    this.columnHeader4 = new System.Windows.Forms.ColumnHeader();
    this.oPnlLeft = new System.Windows.Forms.Panel();
    this.oTvGenres = new SB54_CSAI.MediaLibrary.GenreTree();
    this.oCtxtMenuGT = new System.Windows.Forms.ContextMenu();
    this.oMnuAddToTrackListFromGT = new System.Windows.Forms.MenuItem();
    this.oImgListFldrs = new System.Windows.Forms.ImageList( this.components );
    this.otvFolders = new SB54_CSAI.MediaLibrary.FolderTree();
}

```

```
this.oPnlLeftTop = new System.Windows.Forms.Panel();
this.oBtnFolders = new System.Windows.Forms.Button();
this.oImgList16 = new System.Windows.Forms.ImageList(this.components);
this.oBtnGenres = new System.Windows.Forms.Button();
this.oBtnCancel = new System.Windows.Forms.Button();
this.oImgLV60 = new System.Windows.Forms.ImageList(this.components);
this.oSpltV = new System.Windows.Forms.Splitter();
this.oPnlFill = new System.Windows.Forms.Panel();
this.oPnlDetails = new System.Windows.Forms.Panel();
this.oPnlDetailsFill = new System.Windows.Forms.Panel();
this.oLblCommentTXT = new System.Windows.Forms.Label();
this.oLblYearTXT = new System.Windows.Forms.Label();
this.oLnkGenreTXT = new System.Windows.Forms.LinkLabel();
this.oLnkArtistTXT = new System.Windows.Forms.LinkLabel();
this.oLnkAlbumTXT = new System.Windows.Forms.LinkLabel();
this.oLblTitleTXT = new System.Windows.Forms.Label();
this.oLnkLocationTXT = new System.Windows.Forms.LinkLabel();
this.oPnlDetailsFillR = new System.Windows.Forms.Panel();
this.oPnlDetailsFillL = new System.Windows.Forms.Panel();
this.oLblComment = new System.Windows.Forms.Label();
this.oLblYear = new System.Windows.Forms.Label();
this.oLblGenre = new System.Windows.Forms.Label();
this.oLblArtist = new System.Windows.Forms.Label();
this.oLblAlbum = new System.Windows.Forms.Label();
this.oLblTitle = new System.Windows.Forms.Label();
this.oLblLocation = new System.Windows.Forms.Label();
this.oPnlDetailsL = new System.Windows.Forms.Panel();
this.oPnlDetailsLogo = new System.Windows.Forms.PictureBox();
this.oPnlDetails_Top = new System.Windows.Forms.Panel();
this.oPnlDetails_TopTXT = new System.Windows.Forms.Label();
this.oPnlDetailsbanner = new System.Windows.Forms.Panel();
this.oLblDetailsTitle = new System.Windows.Forms.Label();
this.oPnlFillBottomSmall = new System.Windows.Forms.Panel();
this.oBtnShares = new System.Windows.Forms.Button();
this.oBtnAddFolder = new System.Windows.Forms.Button();
this.oSpltH = new System.Windows.Forms.Splitter();
this.oPnlFillTop = new System.Windows.Forms.Panel();
this.oLvTracks = new System.Windows.Forms.ListView();
this.oImgLV98 = new System.Windows.Forms.ImageList(this.components);
this.oImgLV16 = new System.Windows.Forms.ImageList(this.components);
this.panell = new System.Windows.Forms.Panel();
this.oBtnShowDetails = new System.Windows.Forms.Button();
this.oBtnDetails = new System.Windows.Forms.Button();
this.oCtxtDetails = new System.Windows.Forms.ContextMenu();
this.oMnuDetails = new System.Windows.Forms.MenuItem();
this.oMnuList = new System.Windows.Forms.MenuItem();
this.oMnuIcon = new System.Windows.Forms.MenuItem();
this.oMnuThumb = new System.Windows.Forms.MenuItem();
this.oPnlFillBanner = new System.Windows.Forms.Panel();
this.oPicShareLarge = new System.Windows.Forms.PictureBox();
this.oPnlFillBannerTitle1 = new System.Windows.Forms.Label();
this.oPnlFillBannerTitle2 = new System.Windows.Forms.Label();
this.oPnlRTT = new System.Windows.Forms.Panel();
this.oLblScan = new System.Windows.Forms.Label();
this.oPicShareSmall = new System.Windows.Forms.PictureBox();
this.oFolderDialog1 = new System.Windows.Forms.FolderBrowserDialog();
this.oMnuIcons1 = new sb54_CSAI.MiscFormComponents.MenuIcons(this.components);
this.oMnuAddToTrackListFromFT = new System.Windows.Forms.MenuItem();
this.oMnuDeleteFolder = new System.Windows.Forms.MenuItem();
this.oCtxtMenuFT = new System.Windows.Forms.ContextMenu();
this.toolTip1 = new System.Windows.Forms.ToolTip(this.components);
this.oPnlLeft.SuspendLayout();
this.oPnlLeftTop.SuspendLayout();
this.oPnlFill.SuspendLayout();
this.oPnlDetails.SuspendLayout();
this.oPnlDetailsFill.SuspendLayout();
this.oPnlDetailsFillL.SuspendLayout();
this.oPnlDetailsL.SuspendLayout();
this.oPnlDetails_Top.SuspendLayout();
this.oPnlDetailsbanner.SuspendLayout();
this.oPnlFillBottomSmall.SuspendLayout();
this.oPnlFillTop.SuspendLayout();
this.panell.SuspendLayout();
```

```

this.oPnlFillBanner.SuspendLayout();
this.oPnlRTT.SuspendLayout();
this.SuspendLayout();
//
// columnHeader1
//
this.columnHeader1.Text = "Directory";
this.columnHeader1.Width = 400;
//
// columnHeader2
//
this.columnHeader2.Text = "Type";
this.columnHeader2.Width = 150;
//
// columnHeader3
//
this.columnHeader3.Text = "Size";
this.columnHeader3.Width = 100;
//
// columnHeader4
//
this.columnHeader4.Text = "Directory";
this.columnHeader4.Width = 400;
//
// oPnlLeft
//
this.oPnlLeft.Controls.Add(this.oTvGenres);
this.oPnlLeft.Controls.Add(this.otvFolders);
this.oPnlLeft.Controls.Add(this.oPnlLeftTop);
this.oPnlLeft.Dock = System.Windows.Forms.DockStyle.Left;
this.oPnlLeft.Location = new System.Drawing.Point(0, 0);
this.oPnlLeft.Name = "oPnlLeft";
this.oPnlLeft.Size = new System.Drawing.Size(318, 584);
this.oPnlLeft.TabIndex = 0;
//
// oTvGenres
//
this.oTvGenres.BackColor = System.Drawing.Color.White;
this.oTvGenres.ContextMenu = this.oCtxtMenuGT;
this.oTvGenres.Dock = System.Windows.Forms.DockStyle.Fill;
this.oTvGenres.ImageList = this.oImgListFldrs;
this.oTvGenres.Location = new System.Drawing.Point(0, 28);
this.oTvGenres.Name = "oTvGenres";
this.oTvGenres.ShowLines = false;
this.oTvGenres.Size = new System.Drawing.Size(318, 556);
this.oTvGenres.TabIndex = 6;
this.toolTip1.SetToolTip(this.oTvGenres, "Using this TreeView you can view your
media library using the classifications sho" +
    "\n\nYou may also add items to the Media Player using a right click");
this.oTvGenres.Visible = false;
this.oTvGenres.MouseDown += new System.Windows.Forms.MouseEventHandler(this.
oTvGenres_MouseDown);
this.oTvGenres.AfterSelect += new System.Windows.Forms.TreeViewEventHandler(this.
.oTvGenres_AfterSelect);
//
// oCtxtMenuGT
//
this.oCtxtMenuGT.MenuItems.AddRange(new System.Windows.Forms.MenuItem[] {
this.oMnuAddToTrackListFromGT});
//
// oMnuAddToTrackListFromGT
//
this.oMnuAddToTrackListFromGT.Index = 0;
this.oMnuIcons1.SetMenuIcons(this.oMnuAddToTrackListFromGT, "5");
this.oMnuAddToTrackListFromGT.OwnerDraw = true;
this.oMnuAddToTrackListFromGT.Text = "Add Selected Folder To Track List";
this.oMnuAddToTrackListFromGT.Click += new System.EventHandler(this.
oMnuAddToTrackListFromGT_Click);
//
// oImgListFldrs
//
this.oImgListFldrs.ColorDepth = System.Windows.Forms.ColorDepth.Depth24Bit;
this.oImgListFldrs.ImageSize = new System.Drawing.Size(20, 20);

```

```
        this.oImgListFldrs.ImageStream = ((System.Windows.Forms.ImageListStreamer)
(resources.GetObject("oImgListFldrs.ImageStream")));
        this.oImgListFldrs.TransparentColor = System.Drawing.Color.Transparent;
        //
        // otvFolders
        //
        this.otvFolders.Dock = System.Windows.Forms.DockStyle.Fill;
        this.otvFolders.ImageList = this.oImgListFldrs;
        this.otvFolders.Location = new System.Drawing.Point(0, 28);
        this.otvFolders.Name = "otvFolders";
        this.otvFolders.Size = new System.Drawing.Size(318, 556);
        this.otvFolders.TabIndex = 5;
        this.toolTip1.SetToolTip(this.otvFolders, "Using this TreeView you can view your
media library using the source folders show" +
        "n.\nYou may also add items to the Media Player using a right click, and
remove fo" +
        "lders should you choose to.");
        this.otvFolders.MouseDown += new System.Windows.Forms.MouseEventHandler(this.
otvFolders_MouseDown);
        this.otvFolders.Scan += new SB54_CSAI.MediaLibrary.FolderTree.
ScanningFolderhandler(this.otvFolders_Scan);
        this.otvFolders.AfterSelect += new System.Windows.Forms.TreeViewEventHandler
(this.otvFolders_AfterSelect);
        //
        // oPnlLeftTop
        //
        this.oPnlLeftTop.Controls.Add(this.oBtnFolders);
        this.oPnlLeftTop.Controls.Add(this.oBtnGenres);
        this.oPnlLeftTop.Controls.Add(this.oBtnCancel);
        this.oPnlLeftTop.Dock = System.Windows.Forms.DockStyle.Top;
        this.oPnlLeftTop.Location = new System.Drawing.Point(0, 0);
        this.oPnlLeftTop.Name = "oPnlLeftTop";
        this.oPnlLeftTop.Size = new System.Drawing.Size(318, 28);
        this.oPnlLeftTop.TabIndex = 3;
        //
        // oBtnFolders
        //
        this.oBtnFolders.FlatStyle = System.Windows.Forms.FlatStyle.Flat;
        this.oBtnFolders.ForeColor = System.Drawing.Color.Black;
        this.oBtnFolders.ImageAlign = System.Drawing.ContentAlignment.MiddleLeft;
        this.oBtnFolders.ImageIndex = 0;
        this.oBtnFolders.ImageList = this.oImgList16;
        this.oBtnFolders.Location = new System.Drawing.Point(9, 3);
        this.oBtnFolders.Name = "oBtnFolders";
        this.oBtnFolders.Size = new System.Drawing.Size(70, 22);
        this.oBtnFolders.TabIndex = 2;
        this.oBtnFolders.Text = "Folders";
        this.oBtnFolders.TextAlign = System.Drawing.ContentAlignment.MiddleRight;
        this.toolTip1.SetToolTip(this.oBtnFolders, "Click here to manage your database
logged directories");
        this.oBtnFolders.Click += new System.EventHandler(this.oBtnFolders_Click);
        //
        // oImgList16
        //
        this.oImgList16.ColorDepth = System.Windows.Forms.ColorDepth.Depth24Bit;
        this.oImgList16.ImageSize = new System.Drawing.Size(16, 16);
        this.oImgList16.ImageStream = ((System.Windows.Forms.ImageListStreamer)
(resources.GetObject("oImgList16.ImageStream")));
        this.oImgList16.TransparentColor = System.Drawing.Color.Transparent;
        //
        // oBtnGenres
        //
        this.oBtnGenres.FlatStyle = System.Windows.Forms.FlatStyle.Flat;
        this.oBtnGenres.ForeColor = System.Drawing.Color.Black;
        this.oBtnGenres.ImageAlign = System.Drawing.ContentAlignment.MiddleLeft;
        this.oBtnGenres.ImageIndex = 1;
        this.oBtnGenres.ImageList = this.oImgList16;
        this.oBtnGenres.Location = new System.Drawing.Point(83, 3);
        this.oBtnGenres.Name = "oBtnGenres";
        this.oBtnGenres.Size = new System.Drawing.Size(84, 22);
        this.oBtnGenres.TabIndex = 3;
        this.oBtnGenres.Text = "Organiser";
        this.oBtnGenres.TextAlign = System.Drawing.ContentAlignment.MiddleRight;
```

```
        this.toolTip1.SetToolTip(this.oBtnGenres, "Click here to view your media using a
classified view");
        this.oBtnGenres.Click += new System.EventHandler(this.oBtnGenres_Click);
        //
        // oBtnCancel
        //
        this.oBtnCancel.FlatStyle = System.Windows.Forms.FlatStyle.Flat;
        this.oBtnCancel.ForeColor = System.Drawing.Color.Red;
        this.oBtnCancel.ImageAlign = System.Drawing.ContentAlignment.MiddleLeft;
        this.oBtnCancel.ImageIndex = 4;
        this.oBtnCancel.ImageList = this.oImgList16;
        this.oBtnCancel.Location = new System.Drawing.Point(218, 3);
        this.oBtnCancel.Name = "oBtnCancel";
        this.oBtnCancel.Size = new System.Drawing.Size(96, 22);
        this.oBtnCancel.TabIndex = 5;
        this.oBtnCancel.Text = "Cancel Scan";
        this.oBtnCancel.TextAlign = System.Drawing.ContentAlignment.MiddleRight;
        this.toolTip1.SetToolTip(this.oBtnCancel, "Cancel the current folder scanning
operation");
        this.oBtnCancel.Click += new System.EventHandler(this.oBtnCancel_Click);
        //
        // oImgLV60
        //
        this.oImgLV60.ColorDepth = System.Windows.Forms.ColorDepth.Depth24Bit;
        this.oImgLV60.ImageSize = new System.Drawing.Size(60, 60);
        this.oImgLV60.ImageStream = ((System.Windows.Forms.ImageListStreamer)(resources.
GetObject("oImgLV60.ImageStream")));
        this.oImgLV60.TransparentColor = System.Drawing.Color.Transparent;
        //
        // oSplTV
        //
        this.oSplTV.BackColor = System.Drawing.SystemColors.Control;
        this.oSplTV.Location = new System.Drawing.Point(318, 0);
        this.oSplTV.Name = "oSplTV";
        this.oSplTV.Size = new System.Drawing.Size(3, 584);
        this.oSplTV.TabIndex = 1;
        this.oSplTV.TabStop = false;
        //
        // oPnlFill
        //
        this.oPnlFill.BackColor = System.Drawing.Color.White;
        this.oPnlFill.Controls.Add(this.oPnlDetails);
        this.oPnlFill.Controls.Add(this.oPnlFillBottomSmall);
        this.oPnlFill.Controls.Add(this.oSpltH);
        this.oPnlFill.Controls.Add(this.oPnlFillTop);
        this.oPnlFill.Dock = System.Windows.Forms.DockStyle.Fill;
        this.oPnlFill.Location = new System.Drawing.Point(321, 0);
        this.oPnlFill.Name = "oPnlFill";
        this.oPnlFill.Size = new System.Drawing.Size(687, 584);
        this.oPnlFill.TabIndex = 2;
        //
        // oPnlDetails
        //
        this.oPnlDetails.Controls.Add(this.oPnlDetailsFill);
        this.oPnlDetails.Controls.Add(this.oPnlDetailsL);
        this.oPnlDetails.Controls.Add(this.oPnlDetails_Top);
        this.oPnlDetails.Controls.Add(this.oPnlDetailsbanner);
        this.oPnlDetails.Dock = System.Windows.Forms.DockStyle.Fill;
        this.oPnlDetails.Location = new System.Drawing.Point(0, 333);
        this.oPnlDetails.Name = "oPnlDetails";
        this.oPnlDetails.Size = new System.Drawing.Size(687, 218);
        this.oPnlDetails.TabIndex = 3;
        //
        // oPnlDetailsFill
        //
        this.oPnlDetailsFill.Controls.Add(this.oLblCommentTXT);
        this.oPnlDetailsFill.Controls.Add(this.oLblYearTXT);
        this.oPnlDetailsFill.Controls.Add(this.oLnkGenreTXT);
        this.oPnlDetailsFill.Controls.Add(this.oLnkArtistTXT);
        this.oPnlDetailsFill.Controls.Add(this.oLnkAlbumTXT);
        this.oPnlDetailsFill.Controls.Add(this.oLblTitleTXT);
        this.oPnlDetailsFill.Controls.Add(this.oLnkLocationTXT);
        this.oPnlDetailsFill.Controls.Add(this.oPnlDetailsFillR);
```

```
this.oPnlDetailsFill.Controls.Add(this.oPnlDetailsFillL);
this.oPnlDetailsFill.Dock = System.Windows.Forms.DockStyle.Fill;
this.oPnlDetailsFill.Location = new System.Drawing.Point(135, 62);
this.oPnlDetailsFill.Name = "oPnlDetailsFill";
this.oPnlDetailsFill.Size = new System.Drawing.Size(552, 156);
this.oPnlDetailsFill.TabIndex = 13;
//
// oLblCommentTXT
//
this.oLblCommentTXT.BackColor = System.Drawing.Color.LightSteelBlue;
this.oLblCommentTXT.Dock = System.Windows.Forms.DockStyle.Top;
this.oLblCommentTXT.Font = new System.Drawing.Font("Microsoft Sans Serif", 8.25F,
, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((System.Byte)
(0)));
this.oLblCommentTXT.ForeColor = System.Drawing.Color.Blue;
this.oLblCommentTXT.Location = new System.Drawing.Point(94, 120);
this.oLblCommentTXT.Name = "oLblCommentTXT";
this.oLblCommentTXT.Size = new System.Drawing.Size(386, 20);
this.oLblCommentTXT.TabIndex = 24;
this.oLblCommentTXT.Text = "Comment";
this.oLblCommentTXT.TextAlign = System.Drawing.ContentAlignment.MiddleLeft;
//
// oLblYearTXT
//
this.oLblYearTXT.BackColor = System.Drawing.Color.WhiteSmoke;
this.oLblYearTXT.Dock = System.Windows.Forms.DockStyle.Top;
this.oLblYearTXT.Font = new System.Drawing.Font("Microsoft Sans Serif", 8.25F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((System.Byte)(0)))
;
this.oLblYearTXT.ForeColor = System.Drawing.Color.Blue;
this.oLblYearTXT.Location = new System.Drawing.Point(94, 100);
this.oLblYearTXT.Name = "oLblYearTXT";
this.oLblYearTXT.Size = new System.Drawing.Size(386, 20);
this.oLblYearTXT.TabIndex = 23;
this.oLblYearTXT.Text = "Year";
this.oLblYearTXT.TextAlign = System.Drawing.ContentAlignment.MiddleLeft;
//
// oLnkGenreTXT
//
this.oLnkGenreTXT.BackColor = System.Drawing.Color.LightSteelBlue;
this.oLnkGenreTXT.Dock = System.Windows.Forms.DockStyle.Top;
this.oLnkGenreTXT.Location = new System.Drawing.Point(94, 80);
this.oLnkGenreTXT.Name = "oLnkGenreTXT";
this.oLnkGenreTXT.Size = new System.Drawing.Size(386, 20);
this.oLnkGenreTXT.TabIndex = 22;
this.oLnkGenreTXT.TabStop = true;
this.oLnkGenreTXT.Text = "linkLabel1";
this.oLnkGenreTXT.TextAlign = System.Drawing.ContentAlignment.MiddleLeft;
this.toolTip1.SetToolTip(this.oLnkGenreTXT, "Click here to navigate to the Genre
folder for this MP3");
this.oLnkGenreTXT.LinkClicked += new System.Windows.Forms.
LinkLabelLinkClickedEventHandler(this.oLnkGenreTXT_LinkClicked);
//
// oLnkArtistTXT
//
this.oLnkArtistTXT.BackColor = System.Drawing.Color.WhiteSmoke;
this.oLnkArtistTXT.Dock = System.Windows.Forms.DockStyle.Top;
this.oLnkArtistTXT.Location = new System.Drawing.Point(94, 60);
this.oLnkArtistTXT.Name = "oLnkArtistTXT";
this.oLnkArtistTXT.Size = new System.Drawing.Size(386, 20);
this.oLnkArtistTXT.TabIndex = 18;
this.oLnkArtistTXT.TabStop = true;
this.oLnkArtistTXT.Text = "linkLabel1";
this.oLnkArtistTXT.TextAlign = System.Drawing.ContentAlignment.MiddleLeft;
this.toolTip1.SetToolTip(this.oLnkArtistTXT, "Click here to navigate to the
Artist folder for this MP3");
this.oLnkArtistTXT.LinkClicked += new System.Windows.Forms.
LinkLabelLinkClickedEventHandler(this.oLnkArtistTXT_LinkClicked);
//
// oLnkAlbumTXT
//
this.oLnkAlbumTXT.BackColor = System.Drawing.Color.LightSteelBlue;
this.oLnkAlbumTXT.Dock = System.Windows.Forms.DockStyle.Top;
```

```
this.oLnkAlbumTXT.Location = new System.Drawing.Point(94, 40);
this.oLnkAlbumTXT.Name = "oLnkAlbumTXT";
this.oLnkAlbumTXT.Size = new System.Drawing.Size(386, 20);
this.oLnkAlbumTXT.TabIndex = 17;
this.oLnkAlbumTXT.TabStop = true;
this.oLnkAlbumTXT.Text = "linkLabel1";
this.oLnkAlbumTXT.TextAlign = System.Drawing.ContentAlignment.MiddleLeft;
this.toolTip1.SetToolTip(this.oLnkAlbumTXT, "Click here to navigate to the Album
folder for this MP3");
this.oLnkAlbumTXT.LinkClicked += new System.Windows.Forms.
LinkLabelLinkClickedEventHandler(this.oLnkAlbumTXT_LinkClicked);
//
// oLblTitleTXT
//
this.oLblTitleTXT.BackColor = System.Drawing.Color.WhiteSmoke;
this.oLblTitleTXT.Dock = System.Windows.Forms.DockStyle.Top;
this.oLblTitleTXT.Font = new System.Drawing.Font("Microsoft Sans Serif", 8.25F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((System.Byte)(0)))
;
this.oLblTitleTXT.ForeColor = System.Drawing.Color.Blue;
this.oLblTitleTXT.Location = new System.Drawing.Point(94, 20);
this.oLblTitleTXT.Name = "oLblTitleTXT";
this.oLblTitleTXT.Size = new System.Drawing.Size(386, 20);
this.oLblTitleTXT.TabIndex = 16;
this.oLblTitleTXT.Text = "Title";
this.oLblTitleTXT.TextAlign = System.Drawing.ContentAlignment.MiddleLeft;
//
// oLnkLocationTXT
//
this.oLnkLocationTXT.BackColor = System.Drawing.Color.LightSteelBlue;
this.oLnkLocationTXT.Dock = System.Windows.Forms.DockStyle.Top;
this.oLnkLocationTXT.Font = new System.Drawing.Font("Microsoft Sans Serif", 8.
25F, System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, ((System.Byte)
(0)));
this.oLnkLocationTXT.Location = new System.Drawing.Point(94, 0);
this.oLnkLocationTXT.Name = "oLnkLocationTXT";
this.oLnkLocationTXT.Size = new System.Drawing.Size(386, 20);
this.oLnkLocationTXT.TabIndex = 15;
this.oLnkLocationTXT.TabStop = true;
this.oLnkLocationTXT.Text = "linkLabel1";
this.oLnkLocationTXT.TextAlign = System.Drawing.ContentAlignment.MiddleLeft;
this.toolTip1.SetToolTip(this.oLnkLocationTXT, "Click here to navigate to the
Location folder for this MP3");
this.oLnkLocationTXT.LinkClicked += new System.Windows.Forms.
LinkLabelLinkClickedEventHandler(this.oLnkLocationTXT_LinkClicked);
//
// oPnlDetailsFillR
//
this.oPnlDetailsFillR.Dock = System.Windows.Forms.DockStyle.Right;
this.oPnlDetailsFillR.Location = new System.Drawing.Point(480, 0);
this.oPnlDetailsFillR.Name = "oPnlDetailsFillR";
this.oPnlDetailsFillR.Size = new System.Drawing.Size(72, 156);
this.oPnlDetailsFillR.TabIndex = 14;
//
// oPnlDetailsFillL
//
this.oPnlDetailsFillL.Controls.Add(this.oLblComment);
this.oPnlDetailsFillL.Controls.Add(this.oLblYear);
this.oPnlDetailsFillL.Controls.Add(this.oLblGenre);
this.oPnlDetailsFillL.Controls.Add(this.oLblArtist);
this.oPnlDetailsFillL.Controls.Add(this.oLblAlbum);
this.oPnlDetailsFillL.Controls.Add(this.oLblTitle);
this.oPnlDetailsFillL.Controls.Add(this.oLblLocation);
this.oPnlDetailsFillL.Dock = System.Windows.Forms.DockStyle.Left;
this.oPnlDetailsFillL.Location = new System.Drawing.Point(0, 0);
this.oPnlDetailsFillL.Name = "oPnlDetailsFillL";
this.oPnlDetailsFillL.Size = new System.Drawing.Size(94, 156);
this.oPnlDetailsFillL.TabIndex = 13;
//
// oLblComment
//
this.oLblComment.BackColor = System.Drawing.Color.LightSteelBlue;
this.oLblComment.Dock = System.Windows.Forms.DockStyle.Top;
```

```
        this.oLblComment.Font = new System.Drawing.Font("Microsoft Sans Serif", 8.25F,
System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, ((System.Byte)(0)));
        this.oLblComment.Location = new System.Drawing.Point(0, 120);
        this.oLblComment.Name = "oLblComment";
        this.oLblComment.Size = new System.Drawing.Size(94, 20);
        this.oLblComment.TabIndex = 6;
        this.oLblComment.Text = "Comment";
        this.oLblComment.TextAlign = System.Drawing.ContentAlignment.MiddleLeft;
        //
        // oLblYear
        //
        this.oLblYear.BackColor = System.Drawing.Color.WhiteSmoke;
        this.oLblYear.Dock = System.Windows.Forms.DockStyle.Top;
        this.oLblYear.Font = new System.Drawing.Font("Microsoft Sans Serif", 8.25F,
System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, ((System.Byte)(0)));
        this.oLblYear.Location = new System.Drawing.Point(0, 100);
        this.oLblYear.Name = "oLblYear";
        this.oLblYear.Size = new System.Drawing.Size(94, 20);
        this.oLblYear.TabIndex = 5;
        this.oLblYear.Text = "Year";
        this.oLblYear.TextAlign = System.Drawing.ContentAlignment.MiddleLeft;
        //
        // oLblGenre
        //
        this.oLblGenre.BackColor = System.Drawing.Color.LightSteelBlue;
        this.oLblGenre.Dock = System.Windows.Forms.DockStyle.Top;
        this.oLblGenre.Font = new System.Drawing.Font("Microsoft Sans Serif", 8.25F,
System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, ((System.Byte)(0)));
        this.oLblGenre.Location = new System.Drawing.Point(0, 80);
        this.oLblGenre.Name = "oLblGenre";
        this.oLblGenre.Size = new System.Drawing.Size(94, 20);
        this.oLblGenre.TabIndex = 4;
        this.oLblGenre.Text = "Genre";
        this.oLblGenre.TextAlign = System.Drawing.ContentAlignment.MiddleLeft;
        //
        // oLblArtist
        //
        this.oLblArtist.BackColor = System.Drawing.Color.WhiteSmoke;
        this.oLblArtist.Dock = System.Windows.Forms.DockStyle.Top;
        this.oLblArtist.Font = new System.Drawing.Font("Microsoft Sans Serif", 8.25F,
System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, ((System.Byte)(0)));
        this.oLblArtist.Location = new System.Drawing.Point(0, 60);
        this.oLblArtist.Name = "oLblArtist";
        this.oLblArtist.Size = new System.Drawing.Size(94, 20);
        this.oLblArtist.TabIndex = 3;
        this.oLblArtist.Text = "Artist";
        this.oLblArtist.TextAlign = System.Drawing.ContentAlignment.MiddleLeft;
        //
        // oLblAlbum
        //
        this.oLblAlbum.BackColor = System.Drawing.Color.LightSteelBlue;
        this.oLblAlbum.Dock = System.Windows.Forms.DockStyle.Top;
        this.oLblAlbum.Font = new System.Drawing.Font("Microsoft Sans Serif", 8.25F,
System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, ((System.Byte)(0)));
        this.oLblAlbum.Location = new System.Drawing.Point(0, 40);
        this.oLblAlbum.Name = "oLblAlbum";
        this.oLblAlbum.Size = new System.Drawing.Size(94, 20);
        this.oLblAlbum.TabIndex = 2;
        this.oLblAlbum.Text = "Album";
        this.oLblAlbum.TextAlign = System.Drawing.ContentAlignment.MiddleLeft;
        //
        // oLblTitle
        //
        this.oLblTitle.BackColor = System.Drawing.Color.WhiteSmoke;
        this.oLblTitle.Dock = System.Windows.Forms.DockStyle.Top;
        this.oLblTitle.Font = new System.Drawing.Font("Microsoft Sans Serif", 8.25F,
System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, ((System.Byte)(0)));
        this.oLblTitle.Location = new System.Drawing.Point(0, 20);
        this.oLblTitle.Name = "oLblTitle";
        this.oLblTitle.Size = new System.Drawing.Size(94, 20);
        this.oLblTitle.TabIndex = 1;
        this.oLblTitle.Text = "Title";
        this.oLblTitle.TextAlign = System.Drawing.ContentAlignment.MiddleLeft;
```

```
//
// oLblLocation
//
this.oLblLocation.BackColor = System.Drawing.Color.LightSteelBlue;
this.oLblLocation.Dock = System.Windows.Forms.DockStyle.Top;
this.oLblLocation.Font = new System.Drawing.Font("Microsoft Sans Serif", 8.25F,
System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, ((System.Byte)(0)));
this.oLblLocation.Location = new System.Drawing.Point(0, 0);
this.oLblLocation.Name = "oLblLocation";
this.oLblLocation.Size = new System.Drawing.Size(94, 20);
this.oLblLocation.TabIndex = 0;
this.oLblLocation.Text = "Location";
this.oLblLocation.TextAlign = System.Drawing.ContentAlignment.MiddleLeft;
//
// oPnlDetailsL
//
this.oPnlDetailsL.Controls.Add(this.oPnlDetailsLogo);
this.oPnlDetailsL.Dock = System.Windows.Forms.DockStyle.Left;
this.oPnlDetailsL.Location = new System.Drawing.Point(0, 62);
this.oPnlDetailsL.Name = "oPnlDetailsL";
this.oPnlDetailsL.Size = new System.Drawing.Size(135, 156);
this.oPnlDetailsL.TabIndex = 12;
//
// oPnlDetailsLogo
//
this.oPnlDetailsLogo.Image = ((System.Drawing.Image)(resources.GetObject(
"oPnlDetailsLogo.Image")));
this.oPnlDetailsLogo.Location = new System.Drawing.Point(16, 21);
this.oPnlDetailsLogo.Name = "oPnlDetailsLogo";
this.oPnlDetailsLogo.Size = new System.Drawing.Size(100, 100);
this.oPnlDetailsLogo.TabIndex = 0;
this.oPnlDetailsLogo.TabStop = false;
//
// oPnlDetails_Top
//
this.oPnlDetails_Top.Controls.Add(this.oPnlDetails_TopTXT);
this.oPnlDetails_Top.Dock = System.Windows.Forms.DockStyle.Top;
this.oPnlDetails_Top.Location = new System.Drawing.Point(0, 18);
this.oPnlDetails_Top.Name = "oPnlDetails_Top";
this.oPnlDetails_Top.Size = new System.Drawing.Size(687, 44);
this.oPnlDetails_Top.TabIndex = 11;
//
// oPnlDetails_TopTXT
//
this.oPnlDetails_TopTXT.BackColor = System.Drawing.Color.White;
this.oPnlDetails_TopTXT.Dock = System.Windows.Forms.DockStyle.Fill;
this.oPnlDetails_TopTXT.Font = new System.Drawing.Font("Microsoft Sans Serif",
11.25F, System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, ((System.Byte)(0)));
this.oPnlDetails_TopTXT.ForeColor = System.Drawing.Color.Black;
this.oPnlDetails_TopTXT.Location = new System.Drawing.Point(0, 0);
this.oPnlDetails_TopTXT.Name = "oPnlDetails_TopTXT";
this.oPnlDetails_TopTXT.Size = new System.Drawing.Size(687, 44);
this.oPnlDetails_TopTXT.TabIndex = 0;
this.oPnlDetails_TopTXT.Text = "label1";
this.oPnlDetails_TopTXT.TextAlign = System.Drawing.ContentAlignment.MiddleLeft;
//
// oPnlDetailsbanner
//
this.oPnlDetailsbanner.BackColor = System.Drawing.Color.CornflowerBlue;
this.oPnlDetailsbanner.Controls.Add(this.oLblDetailsTitle);
this.oPnlDetailsbanner.Dock = System.Windows.Forms.DockStyle.Top;
this.oPnlDetailsbanner.Location = new System.Drawing.Point(0, 0);
this.oPnlDetailsbanner.Name = "oPnlDetailsbanner";
this.oPnlDetailsbanner.Size = new System.Drawing.Size(687, 18);
this.oPnlDetailsbanner.TabIndex = 10;
//
// oLblDetailsTitle
//
this.oLblDetailsTitle.BackColor = System.Drawing.Color.Transparent;
this.oLblDetailsTitle.Font = new System.Drawing.Font("Microsoft Sans Serif", 12F,
System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, ((System.Byte)(0)));
this.oLblDetailsTitle.ForeColor = System.Drawing.Color.White;
```

```
this.oLblDetailsTitle.Location = new System.Drawing.Point(4, -1);
this.oLblDetailsTitle.Name = "oLblDetailsTitle";
this.oLblDetailsTitle.Size = new System.Drawing.Size(319, 21);
this.oLblDetailsTitle.TabIndex = 6;
this.oLblDetailsTitle.Text = "Details View";
//
// oPnlFillBottomSmall
//
this.oPnlFillBottomSmall.BackColor = System.Drawing.SystemColors.Control;
this.oPnlFillBottomSmall.Controls.Add(this.oBtnShares);
this.oPnlFillBottomSmall.Controls.Add(this.oBtnAddFolder);
this.oPnlFillBottomSmall.Dock = System.Windows.Forms.DockStyle.Bottom;
this.oPnlFillBottomSmall.Location = new System.Drawing.Point(0, 551);
this.oPnlFillBottomSmall.Name = "oPnlFillBottomSmall";
this.oPnlFillBottomSmall.Size = new System.Drawing.Size(687, 33);
this.oPnlFillBottomSmall.TabIndex = 2;
//
// oBtnShares
//
this.oBtnShares.FlatStyle = System.Windows.Forms.FlatStyle.Flat;
this.oBtnShares.ForeColor = System.Drawing.Color.Black;
this.oBtnShares.ImageAlign = System.Drawing.ContentAlignment.MiddleLeft;
this.oBtnShares.ImageIndex = 2;
this.oBtnShares.ImageList = this.oImgList16;
this.oBtnShares.Location = new System.Drawing.Point(7, 6);
this.oBtnShares.Name = "oBtnShares";
this.oBtnShares.Size = new System.Drawing.Size(100, 22);
this.oBtnShares.TabIndex = 6;
this.oBtnShares.Text = "Share Files...";
this.oBtnShares.TextAlign = System.Drawing.ContentAlignment.MiddleRight;
this.toolTip1.SetToolTip(this.oBtnShares, "Click here to add folders to your
database logged files");
this.oBtnShares.Click += new System.EventHandler(this.oBtnShares_Click);
//
// oBtnAddFolder
//
this.oBtnAddFolder.FlatStyle = System.Windows.Forms.FlatStyle.Flat;
this.oBtnAddFolder.ForeColor = System.Drawing.Color.Black;
this.oBtnAddFolder.ImageAlign = System.Drawing.ContentAlignment.MiddleLeft;
this.oBtnAddFolder.ImageIndex = 3;
this.oBtnAddFolder.ImageList = this.oImgList16;
this.oBtnAddFolder.Location = new System.Drawing.Point(111, 6);
this.oBtnAddFolder.Name = "oBtnAddFolder";
this.oBtnAddFolder.Size = new System.Drawing.Size(88, 22);
this.oBtnAddFolder.TabIndex = 5;
this.oBtnAddFolder.Text = "Add Folder";
this.oBtnAddFolder.TextAlign = System.Drawing.ContentAlignment.MiddleRight;
this.toolTip1.SetToolTip(this.oBtnAddFolder, "Click here to add a single folder
to your database logged files");
this.oBtnAddFolder.Click += new System.EventHandler(this.oBtnAddFolder_Click);
//
// oSpltH
//
this.oSpltH.BackColor = System.Drawing.SystemColors.Control;
this.oSpltH.Dock = System.Windows.Forms.DockStyle.Top;
this.oSpltH.Location = new System.Drawing.Point(0, 330);
this.oSpltH.Name = "oSpltH";
this.oSpltH.Size = new System.Drawing.Size(687, 3);
this.oSpltH.TabIndex = 1;
this.oSpltH.TabStop = false;
//
// oPnlFillTop
//
this.oPnlFillTop.BackColor = System.Drawing.SystemColors.Control;
this.oPnlFillTop.Controls.Add(this.oLvTracks);
this.oPnlFillTop.Controls.Add(this.panell);
this.oPnlFillTop.Controls.Add(this.oPnlFillBanner);
this.oPnlFillTop.Controls.Add(this.oPnlRTT);
this.oPnlFillTop.Dock = System.Windows.Forms.DockStyle.Top;
this.oPnlFillTop.Location = new System.Drawing.Point(0, 0);
this.oPnlFillTop.Name = "oPnlFillTop";
this.oPnlFillTop.Size = new System.Drawing.Size(687, 330);
this.oPnlFillTop.TabIndex = 0;
```

```
//
// oLvTracks
//
this.oLvTracks.AllowColumnReorder = true;
this.oLvTracks.Columns.AddRange(new System.Windows.Forms.ColumnHeader[] {
this.columnHeader1,
this.columnHeader2,
this.columnHeader3,
this.columnHeader4});
this.oLvTracks.Dock = System.Windows.Forms.DockStyle.Fill;
this.oLvTracks.FullRowSelect = true;
this.oLvTracks.LargeImageList = this.oImgLV98;
this.oLvTracks.Location = new System.Drawing.Point(0, 130);
this.oLvTracks.MultiSelect = false;
this.oLvTracks.Name = "oLvTracks";
this.oLvTracks.Size = new System.Drawing.Size(687, 200);
this.oLvTracks.SmallImageList = this.oImgLV16;
this.oLvTracks.Sorting = System.Windows.Forms.SortOrder.Ascending;
this.oLvTracks.TabIndex = 9;
this.toolTip1.SetToolTip(this.oLvTracks, "Currently selected track view");
this.oLvTracks.View = System.Windows.Forms.View.Details;
this.oLvTracks.ColumnClick += new System.Windows.Forms.ColumnClickEventHandler
(this.oLvTracks_ColumnClick);
this.oLvTracks.SelectedIndexChanged += new System.EventHandler(this.
oLvTracks_SelectedIndexChanged);
//
// oImgLV98
//
this.oImgLV98.ColorDepth = System.Windows.Forms.ColorDepth.Depth24Bit;
this.oImgLV98.ImageSize = new System.Drawing.Size(98, 98);
this.oImgLV98.ImageStream = ((System.Windows.Forms.ImageListStreamer)(resources.
GetObject("oImgLV98.ImageStream")));
this.oImgLV98.TransparentColor = System.Drawing.Color.Transparent;
//
// oImgLV16
//
this.oImgLV16.ColorDepth = System.Windows.Forms.ColorDepth.Depth24Bit;
this.oImgLV16.ImageSize = new System.Drawing.Size(17, 18);
this.oImgLV16.ImageStream = ((System.Windows.Forms.ImageListStreamer)(resources.
GetObject("oImgLV16.ImageStream")));
this.oImgLV16.TransparentColor = System.Drawing.Color.Transparent;
//
// panell
//
this.panell.Controls.Add(this.oBtnShowDetails);
this.panell.Controls.Add(this.oBtnDetails);
this.panell.Dock = System.Windows.Forms.DockStyle.Top;
this.panell.Location = new System.Drawing.Point(0, 103);
this.panell.Name = "panell";
this.panell.Size = new System.Drawing.Size(687, 27);
this.panell.TabIndex = 10;
//
// oBtnShowDetails
//
this.oBtnShowDetails.BackColor = System.Drawing.SystemColors.Control;
this.oBtnShowDetails.FlatStyle = System.Windows.Forms.FlatStyle.Flat;
this.oBtnShowDetails.ForeColor = System.Drawing.Color.Black;
this.oBtnShowDetails.ImageAlign = System.Drawing.ContentAlignment.MiddleLeft;
this.oBtnShowDetails.ImageIndex = 11;
this.oBtnShowDetails.ImageList = this.oImgList16;
this.oBtnShowDetails.Location = new System.Drawing.Point(116, 2);
this.oBtnShowDetails.Name = "oBtnShowDetails";
this.oBtnShowDetails.Size = new System.Drawing.Size(112, 24);
this.oBtnShowDetails.TabIndex = 6;
this.oBtnShowDetails.Text = "Show Details";
this.toolTip1.SetToolTip(this.oBtnShowDetails, "Show the details for a selected
track");
this.oBtnShowDetails.Click += new System.EventHandler(this.
oBtnShowDetails_Click);
//
// oBtnDetails
//
this.oBtnDetails.BackColor = System.Drawing.SystemColors.Control;
```

```
this.oBtnDetails.ContextMenu = this.oCtxtDetails;
this.oBtnDetails.FlatStyle = System.Windows.Forms.FlatStyle.Flat;
this.oBtnDetails.ForeColor = System.Drawing.Color.Black;
this.oBtnDetails.ImageAlign = System.Drawing.ContentAlignment.MiddleLeft;
this.oBtnDetails.ImageIndex = 7;
this.oBtnDetails.ImageList = this.oImgList16;
this.oBtnDetails.Location = new System.Drawing.Point(0, 2);
this.oBtnDetails.Name = "oBtnDetails";
this.oBtnDetails.Size = new System.Drawing.Size(112, 24);
this.oBtnDetails.TabIndex = 12;
this.oBtnDetails.Text = "Detail View";
this.oBtnDetails.TextAlign = System.Drawing.ContentAlignment.MiddleRight;
this.toolTip1.SetToolTip(this.oBtnDetails, "Change the view of the currently
shown tracks");
this.oBtnDetails.Click += new System.EventHandler(this.oBtnDetails_Click);
//
// oCtxtDetails
//
this.oCtxtDetails.MenuItems.AddRange(new System.Windows.Forms.MenuItem[] {
this.oMnuDetails,
this.oMnuList,
this.oMnuIcon,
this.oMnuThumb});
//
// oMnuDetails
//
this.oMnuDetails.Index = 0;
this.oMnuIcons1.SetMenuIcons(this.oMnuDetails, "7");
this.oMnuDetails.OwnerDraw = true;
this.oMnuDetails.Text = "Detail View";
this.oMnuDetails.Click += new System.EventHandler(this.oMnuDetails_Click);
//
// oMnuList
//
this.oMnuList.Index = 1;
this.oMnuIcons1.SetMenuIcons(this.oMnuList, "8");
this.oMnuList.OwnerDraw = true;
this.oMnuList.Text = "List View";
this.oMnuList.Click += new System.EventHandler(this.oMnuList_Click);
//
// oMnuIcon
//
this.oMnuIcon.Index = 2;
this.oMnuIcons1.SetMenuIcons(this.oMnuIcon, "9");
this.oMnuIcon.OwnerDraw = true;
this.oMnuIcon.Text = "Icon View";
this.oMnuIcon.Click += new System.EventHandler(this.oMnuIcon_Click);
//
// oMnuThumb
//
this.oMnuThumb.Index = 3;
this.oMnuIcons1.SetMenuIcons(this.oMnuThumb, "10");
this.oMnuThumb.OwnerDraw = true;
this.oMnuThumb.Text = "Thumbnail View";
this.oMnuThumb.Click += new System.EventHandler(this.oMnuThumb_Click);
//
// oPnlFillBanner
//
this.oPnlFillBanner.BackColor = System.Drawing.Color.CornflowerBlue;
this.oPnlFillBanner.Controls.Add(this.oPicShareLarge);
this.oPnlFillBanner.Controls.Add(this.oPnlFillBannerTitle1);
this.oPnlFillBanner.Controls.Add(this.oPnlFillBannerTitle2);
this.oPnlFillBanner.Dock = System.Windows.Forms.DockStyle.Top;
this.oPnlFillBanner.Location = new System.Drawing.Point(0, 50);
this.oPnlFillBanner.Name = "oPnlFillBanner";
this.oPnlFillBanner.Size = new System.Drawing.Size(687, 53);
this.oPnlFillBanner.TabIndex = 9;
//
// oPicShareLarge
//
this.oPicShareLarge.Image = ((System.Drawing.Image)(resources.GetObject(
"oPicShareLarge.Image")));
this.oPicShareLarge.Location = new System.Drawing.Point(6, 1);
```

```
this.oPicShareLarge.Name = "oPicShareLarge";
this.oPicShareLarge.Size = new System.Drawing.Size(48, 48);
this.oPicShareLarge.TabIndex = 9;
this.oPicShareLarge.TabStop = false;
//
// oPnlFillBannerTitle1
//
this.oPnlFillBannerTitle1.BackColor = System.Drawing.Color.Transparent;
this.oPnlFillBannerTitle1.Font = new System.Drawing.Font("Microsoft Sans Serif", 12F, System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, ((System.Byte)
(0)));
this.oPnlFillBannerTitle1.ForeColor = System.Drawing.Color.White;
this.oPnlFillBannerTitle1.Location = new System.Drawing.Point(61, 4);
this.oPnlFillBannerTitle1.Name = "oPnlFillBannerTitle1";
this.oPnlFillBannerTitle1.Size = new System.Drawing.Size(319, 21);
this.oPnlFillBannerTitle1.TabIndex = 6;
this.oPnlFillBannerTitle1.Text = "Welcome To Your Media Library !";
//
// oPnlFillBannerTitle2
//
this.oPnlFillBannerTitle2.BackColor = System.Drawing.Color.Transparent;
this.oPnlFillBannerTitle2.Font = new System.Drawing.Font("Microsoft Sans Serif", 8.25F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((System.
Byte)(0)));
this.oPnlFillBannerTitle2.ForeColor = System.Drawing.Color.White;
this.oPnlFillBannerTitle2.Location = new System.Drawing.Point(64, 27);
this.oPnlFillBannerTitle2.Name = "oPnlFillBannerTitle2";
this.oPnlFillBannerTitle2.Size = new System.Drawing.Size(319, 14);
this.oPnlFillBannerTitle2.TabIndex = 10;
this.oPnlFillBannerTitle2.Text = "Feel free to explore the Media Library";
//
// oPnlRTT
//
this.oPnlRTT.Controls.Add(this.oLblScan);
this.oPnlRTT.Controls.Add(this.oPicShareSmall);
this.oPnlRTT.Dock = System.Windows.Forms.DockStyle.Top;
this.oPnlRTT.Location = new System.Drawing.Point(0, 0);
this.oPnlRTT.Name = "oPnlRTT";
this.oPnlRTT.Size = new System.Drawing.Size(687, 50);
this.oPnlRTT.TabIndex = 6;
//
// oLblScan
//
this.oLblScan.Font = new System.Drawing.Font("Microsoft Sans Serif", 8.25F, System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, ((System.Byte)(0)));
this.oLblScan.Location = new System.Drawing.Point(26, 8);
this.oLblScan.Name = "oLblScan";
this.oLblScan.Size = new System.Drawing.Size(653, 41);
this.oLblScan.TabIndex = 5;
this.oLblScan.Text = "label1";
//
// oPicShareSmall
//
this.oPicShareSmall.Image = ((System.Drawing.Image)(resources.GetObject("oPicShareSmall.Image")));
this.oPicShareSmall.Location = new System.Drawing.Point(5, 7);
this.oPicShareSmall.Name = "oPicShareSmall";
this.oPicShareSmall.Size = new System.Drawing.Size(16, 16);
this.oPicShareSmall.TabIndex = 7;
this.oPicShareSmall.TabStop = false;
//
// oMnuIcons1
//
this.oMnuIcons1.ImageList = this.oImgList16;
//
// oMnuAddToTrackListFromFT
//
this.oMnuAddToTrackListFromFT.Index = 0;
this.oMnuIcons1.SetMenuIcons(this.oMnuAddToTrackListFromFT, "5");
this.oMnuAddToTrackListFromFT.OwnerDraw = true;
this.oMnuAddToTrackListFromFT.Text = "Add Selected Folder To Track List";
this.oMnuAddToTrackListFromFT.Click += new System.EventHandler(this.oMnuAddToTrackList_Click);
```

```

        //
        // oMnuDeleteFolder
        //
        this.oMnuDeleteFolder.Index = 1;
        this.oMnuIcons1.SetMenuIcons(this.oMnuDeleteFolder, "6");
        this.oMnuDeleteFolder.OwnerDraw = true;
        this.oMnuDeleteFolder.Text = "Delete Selected Folder";
        this.oMnuDeleteFolder.Click += new System.EventHandler(this.
oMnuDeleteFolder_Click);
        //
        // oCtxtMenuFT
        //
        this.oCtxtMenuFT.MenuItems.AddRange(new System.Windows.Forms.MenuItem[] {
this.oMnuAddToTrackListFromFT,
this.oMnuDeleteFolder});
        //
        // uctMediaLibrary
        //
        this.BackColor = System.Drawing.SystemColors.Control;
        this.Controls.Add(this.oPnlFill);
        this.Controls.Add(this.oSplTV);
        this.Controls.Add(this.oPnlLeft);
        this.Name = "uctMediaLibrary";
        this.Size = new System.Drawing.Size(1008, 584);
        this.Resize += new System.EventHandler(this.uctMediaLibrary_Resize);
        this.Load += new System.EventHandler(this.uctMediaLibrary_Load);
        this.oPnlLeft.ResumeLayout(false);
        this.oPnlLeftTop.ResumeLayout(false);
        this.oPnlFill.ResumeLayout(false);
        this.oPnlDetails.ResumeLayout(false);
        this.oPnlDetailsFill.ResumeLayout(false);
        this.oPnlDetailsFillL.ResumeLayout(false);
        this.oPnlDetailsL.ResumeLayout(false);
        this.oPnlDetails_Top.ResumeLayout(false);
        this.oPnlDetailsbanner.ResumeLayout(false);
        this.oPnlFillBottomSmall.ResumeLayout(false);
        this.oPnlFillTop.ResumeLayout(false);
        this.panell.ResumeLayout(false);
        this.oPnlFillBanner.ResumeLayout(false);
        this.oPnlRTT.ResumeLayout(false);
        this.ResumeLayout(false);
    }
    #endregion
    #region Private Methods
    #region GUI User interactions

    /// <summary>
    /// Call the CancelFolderScanning method of the <see cref="SB54_CSAI.MediaLibrary.
FolderTree">FolderTree </see>
    /// to try and cancel the current folder scanning
    /// </summary>
    /// <param name="sender">The source object that raised the event</param>
    /// <param name="e">The event arguments</param>
    private void oBtnCancel_Click(object sender, System.EventArgs e)
    {
        //cancel scanning and setup GUI components to new states
        otvFolders.CancelFolderScanning(false);
        oBtnCancel.Visible = false;
        oBtnShares.Enabled = true;
        oBtnAddFolder.Enabled = true;
        oBtnFolders.Enabled = true;
        oBtnGenres.Enabled = true;
        otvFolders.Enabled = true;
        oLblScan.Text = "You have cancelled the folder scanning operation, \n" +
            "So only those folders that were scanned up until you cancelled are now
available";
    }

    /// <summary>
    /// Displays a new folder browser to the user and if they select a new folder,
    /// add the new folder to the list of root folders and start the
    /// <see cref="SB54_CSAI.MediaLibrary.FolderTree">FolderTree </see>

```

```

/// scanning with all the root folders as root nodes
/// </summary>
/// <param name="sender">The source object that raised the event</param>
/// <param name="e">The event arguments</param>
private void oBtnAddFolder_Click(object sender, System.EventArgs e)
{
    try
    {
        DialogResult r = (oFolderDialog1.ShowDialog(this));
        //if user selected a new folder, store it and scan it
        if (r == DialogResult.OK)
        {
            if (! oFolderDialog1.SelectedPath.Equals(string.Empty))
            {
                //get the selected directory
                DirectoryInfo d = new DirectoryInfo (oFolderDialog1.SelectedPath);

                //If the folder is a different PC, normally denoted \\
                //then inform user this is not allowed
                if (d.FullName.StartsWith("\\\"))
                {
                    MessageBox.Show("The folder " + d.FullName + "\n" +
                        "can not be used by the Media Library\r\n\r\n" +
                        "Only local directories may be used",
                        "Invalid Folder", MessageBoxButtons.OK,
                        MessageBoxIcon.Error);
                    return;
                }

                //If the folder is not local, then inform user this is not allowed
                if (! localDrives.Contains(d.FullName.Substring(0,2)))
                {
                    MessageBox.Show("The folder " + d.FullName + "\n" +
                        "can not be used by the Media Library\r\n\r\n" +
                        "Only local directories may be used",
                        "Invalid Folder", MessageBoxButtons.OK,
                        MessageBoxIcon.Error);
                    return;
                }

                //get the size of the selected directory, and alert user how much
                //they may be asking to scan, remember there is nothing to stop them
                selecting
                //C:\ which could take a very very long time
                double dSize = (DirectorySize(d) / Math.Pow(1024,2));
                DialogResult mr = MessageBox.Show("You are about to scan " + dSize.
                ToString("###,###,###,### MB ") +
                "of disk space, this may take some time, are you sure you wish
                to proceed",
                "CONFIRMATION Required", MessageBoxButtons.YesNo,
                MessageBoxIcon.Question);

                //if user still wish to scan this much disk space, proceed to scan
                it
                if (mr == DialogResult.Yes)
                {
                    TopLevelFolders.Add(oFolderDialog1.SelectedPath.ToString());
                    setTeeTopLevelFolders();
                    //set flag so GenreTree knows that it needs to rebuild all its
                    nodes for new folders
                    foldersJustAdded = true;
                }
            }
        }
    }
}

catch(Exception ex)
{
    MessageBox.Show("File handle error",
        "oBtnAddFolder_Click", MessageBoxButtons.OK,
        MessageBoxIcon.Error);
    return;
}

```

```

    }

    /// <summary>
    /// Takes a DirectoryInfo parameter and returns the size in bytes
    /// of this Directory, all sub folders are used in the returned size
    /// </summary>
    /// <param name="dirInfo">The DirectoryInfo to find the size of</param>
    /// <returns>The size in Bytes of the dirInfo parameter</returns>
    private static long DirectorySize(System.IO.DirectoryInfo dirInfo)
    {
        try
        {
            long total = 0;

            foreach(System.IO.FileInfo file in
                dirInfo.GetFiles())
                total += file.Length;

            foreach(System.IO.DirectoryInfo dir in
                dirInfo.GetDirectories())
                total += DirectorySize(dir);

            return total;
        }
        catch(Exception ex)
        {
            MessageBox.Show("File handle error",
                "DirectorySize", MessageBoxButtons.OK,
                MessageBoxIcon.Error);
            return 0;
        }
    }

    /// <summary>
    /// Clear any previously selected TreeNode and get the data for the new
    /// selected TreeNode
    /// </summary>
    /// <param name="sender">The source object that raised the event</param>
    /// <param name="e">The event arguments</param>
    private void otvFolders_AfterSelect(object sender, System.Windows.Forms.
TreeViewEventArgs e)
    {
        //if there is a node selected from last select, clear it highlighted color
        if (tvnFolderSelect != null)
        {
            tvnFolderSelect.BackColor = Color.White;
        }
        //process the newly selected node
        newFolderNodeSelected();
    }

    /// <summary>
    /// Displays the FolderTree context menu if the user right clicks over the
    /// FolderTree area
    /// </summary>
    /// <param name="sender">The source object that raised the event</param>
    /// <param name="e">The event arguments</param>
    private void otvFolders_MouseDown(object sender, System.Windows.Forms.MouseEventArgs
e)
    {
        if (e.Button == MouseButton.Right)
        {
            if (otvFolders.SelectedNode is MP3TreeNode)
            {
                oCtxtMenuFT.Show(otvFolders,new Point(e.X,e.Y));
            }
        }
    }

    /// <summary>
    /// Occurs when the user selects the delete menu for the FolderTree.

```

```

/// Deletes a specific folder from the FolderTree and database
/// </summary>
/// <param name="sender">The source object that raised the event</param>
/// <param name="e">The event arguments</param>
private void oMnuDeleteFolder_Click(object sender, System.EventArgs e)
{
    //ensure the details panel is hidden, as it may be left shown for a folder that
    //no longer exists
    displayDetailPanel(false);
    //Delete a specific folder from the FolderTree and database
    TreeNode n = otvFolders.SelectedNode;
    otvFolders.SelectedNode.Remove();
    db.RemoveFolder(((MP3TreeNode)n).FullPath.Trim());
    removeLoggedFolder(n);
    TopLevelFolders.Remove(n.Text);
}

/// <summary>
/// Keeps the shown panel(s) in graphical proportion
/// </summary>
/// <param name="sender">The source object that raised the event</param>
/// <param name="e">The event arguments</param>
private void uctMediaLibrary_Resize(object sender, System.EventArgs e)
{
    //keep the shown panel in graphical proportion
    if (!ShowDetails)
        oPnlFillTop.Height = this.Height - oPnlFillBottomSmall.Height;
    else
        oPnlFillTop.Height = this.Height/2;

    Application.DoEvents();
}

/// <summary>
/// Shows the details sub panel using the currently selected
/// track, if there is one. If an illegal state is found the
/// details sub panel will NOT be shown
/// </summary>
/// <param name="sender">The source object that raised the event</param>
/// <param name="e">The event arguments</param>
private void oBtnShowDetails_Click(object sender, System.EventArgs e)
{
    try
    {
        //and there is a track
        if (oLvTracks.SelectedItems.Count ==0)
        {
            MessageBox.Show("You must have a track selected to view details",
                "Selection ERROR", MessageBoxButtons.OK,
                MessageBoxIcon.Error);
            return;
        }

        //and the track selected is MP3
        FileInfo fi = new FileInfo(oLvTracks.SelectedItems[0].Text);
        if (!fi.Extension.ToLower().Equals(".mp3"))
        {
            MessageBox.Show("The details view is for MP3 files only",
                "Incorrect Filetype", MessageBoxButtons.OK,
                MessageBoxIcon.Information);
            displayDetailPanel(false);
            return;
        }
        //All conditions above are satisfied, so user simply wants to toggle state
        //of the current visibility of the Details Panel, so use the flag and toggle
it
        ShowDetails = !ShowDetails;
        displayDetailPanel(ShowDetails);
    }
    catch(Exception ex)
    {
        MessageBox.Show("File handle error",
            "oBtnShowDetails_Click", MessageBoxButtons.OK,

```

```
        MessageBoxIcon.Error);
        return;
    }
}

/// <summary>
/// Shows the details panel if the input parameter is true, otherwise hides the
/// details panel
/// </summary>
/// <param name="display">True is the details panel is to be displayed</param>
private void displayDetailPanel(bool display)
{
    //If the user has ask to see details
    if (display)
    {
        //show the details, this method returns false if the track is not MP3 or
        //can not be shown, such as empty item name
        if (setUpDetails(oLvTracks.SelectedItems[0].Text))
        {
            setMP3Details();
            oPnlFillTop.Height = this.Height/2;
            oPnlDetails.Visible = true;
            oSpltH.Visible = true;
        }
    }
    //The users has turned off the details view, so hide it
    else
    {
        oPnlFillTop.Height = this.Height - oPnlFillBottomSmall.Height;
        oPnlDetails.Visible = false;
        oSpltH.Visible = false;
    }
}

/// <summary>
/// Shows the oBtnDetails ContextMenu
/// </summary>
/// <param name="sender">The source object that raised the event</param>
/// <param name="e">The event arguments</param>
private void oBtnDetails_Click(object sender, System.EventArgs e)
{
    oBtnDetails.ContextMenu.Show(oBtnDetails,new Point(0,oBtnDetails.Height - 2));
}

/// <summary>
/// Change the track ListView view to details
/// </summary>
/// <param name="sender">The source object that raised the event</param>
/// <param name="e">The event arguments</param>
private void oMnuDetails_Click(object sender, System.EventArgs e)
{
    oLvTracks.SmallImageList = oImgLV16;
    setButton(7,(sender as MenuItem).Text.ToString());
    oLvTracks.View = View.Details;
}

/// <summary>
/// Change the track ListView view to List
/// </summary>
/// <param name="sender">The source object that raised the event</param>
/// <param name="e">The event arguments</param>
private void oMnuList_Click(object sender, System.EventArgs e)
{
    columnHeader1.Width = 415;

    oLvTracks.SmallImageList = oImgLV16;
    setButton(8,(sender as MenuItem).Text.ToString());
    oLvTracks.View = View.List;
    oLvTracks.Invalidate();
    oLvTracks.Refresh();
}
```

```

    }

    /// <summary>
    /// Change the track ListView view to LargeIcon
    /// </summary>
    /// <param name="sender">The source object that raised the event</param>
    /// <param name="e">The event arguments</param>
    private void oMnuIcon_Click(object sender, System.EventArgs e)
    {
        oLvTracks.LargeImageList = oImgLV60;
        setButton(9,(sender as MenuItem).Text.ToString());
        oLvTracks.View = View.LargeIcon;
    }

    /// <summary>
    /// Change the track ListView view to Thumbnail
    /// </summary>
    /// <param name="sender">The source object that raised the event</param>
    /// <param name="e">The event arguments</param>
    private void oMnuThumb_Click(object sender, System.EventArgs e)
    {
        oLvTracks.LargeImageList = oImgLV98;
        setButton(10,(sender as MenuItem).Text.ToString());
        oLvTracks.View = View.LargeIcon;
    }

    /// <summary>
    /// If the user clicks on a track ListView column, sort the track ListView using
    /// the column index
    /// </summary>
    /// <param name="sender">The source object that raised the event</param>
    /// <param name="e">The event arguments</param>
    private void oLvTracks_ColumnClick(object sender, System.Windows.Forms.
ColumnClickEventArgs e)
    {
        // Set the ListViewItemSorter property to a new ListViewItemComparer object.
        this.oLvTracks.ListViewItemSorter = new ListViewItemComparer(e.Column);
        // Call the sort method to manually sort the column based on the
        ListViewItemComparer implementation.
        oLvTracks.Sort();
    }

    /// <summary>
    /// Raise an event that event subscribers can use. The event is intended to
    /// be used to add files to the <see cref="SB54_CSAI.MediaPlayer.uctMediaPlayer">
media player</see>
    /// </summary>
    /// <param name="sender">The source object that raised the event</param>
    /// <param name="e">The event arguments</param>
    private void oMnuAddToTrackList_Click(object sender, System.EventArgs e)
    {
        //get the files
        string[] files = ((MP3TreeNode)otvFolders.SelectedNode).PlayableFiles;
        addNodeFiles(files);
    }

    /// <summary>
    /// Get the details for the currently selected track list listview item
    /// </summary>
    /// <param name="sender">The source object that raised the event</param>
    /// <param name="e">The event arguments</param>
    private void oLvTracks_SelectedIndexChanged(object sender, System.EventArgs e)
    {
        ListView.SelectedListViewItemCollection co = oLvTracks.SelectedItems;

        //As the listview oLvTracks MultiSelect property has been set to false
        permanently
        //there will only ever be 1 item selected, so quite safe to iterate ALL selected
        //items
        foreach ( ListViewItem item in co )

```

```
        {
            setupDetails(item.Text);
            setMP3Details();
        }
    }

    /// <summary>
    /// Show the FoldersTree treeview and hide GenresTree treeview
    /// </summary>
    /// <param name="sender">The source object that raised the event</param>
    /// <param name="e">The event arguments</param>
    private void oBtnFolders_Click(object sender, System.EventArgs e)
    {
        oTvFolders.Visible = true;
        oTvGenres.Visible = false;
    }

    /// <summary>
    /// Show the GenresTree treeview and hide FoldersTree treeview
    /// </summary>
    /// <param name="sender">The source object that raised the event</param>
    /// <param name="e">The event arguments</param>
    private void oBtnGenres_Click(object sender, System.EventArgs e)
    {
        oTvFolders.Visible = false;
        oTvGenres.Visible = true;
        oTvGenres.Nodes.Clear();
        oTvGenres.InitTree();
        oTvGenres.ExpandAll();
    }

    /// <summary>
    /// Find the Album node in the GenresTree treeview
    /// </summary>
    /// <param name="sender">The source object that raised the event</param>
    /// <param name="e">The event arguments</param>
    private void oLnkAlbumTXT_LinkClicked(object sender, System.Windows.Forms.
LinkLabelLinkClickedEventArgs e)
    {
        //have to rebuild genres, to ensure it is up to date with just added folders
        if (foldersJustAdded)
        {
            oTvGenres.Nodes.Clear();
            oTvGenres.InitTree();
            oTvGenres.ExpandAll();
            foldersJustAdded=false;
        }
        //go find the correct node
        selectLinkGenreTreeNode(oTvGenres.Nodes[0],oLnkAlbumTXT.Text);
        oTvFolders.Visible = false;
        oTvGenres.Visible = true;
    }

    /// <summary>
    /// Find the Artist node in the GenresTree treeview
    /// </summary>
    /// <param name="sender">The source object that raised the event</param>
    /// <param name="e">The event arguments</param>
    private void oLnkArtistTXT_LinkClicked(object sender, System.Windows.Forms.
LinkLabelLinkClickedEventArgs e)
    {
        //have to rebuild genres, to ensure it is up to date with just added folders
        if (foldersJustAdded)
        {
            oTvGenres.Nodes.Clear();
            oTvGenres.InitTree();
            oTvGenres.ExpandAll();
            foldersJustAdded=false;
        }
        //go find the correct node
        selectLinkGenreTreeNode(oTvGenres.Nodes[0],oLnkArtistTXT.Text);
        oTvFolders.Visible = false;
        oTvGenres.Visible = true;
    }
}
```

```

    }

    /// <summary>
    /// Find the Genre node in the GenresTree treeview
    /// </summary>
    /// <param name="sender">The source object that raised the event</param>
    /// <param name="e">The event arguments</param>
    private void oLnkGenreTXT_LinkClicked(object sender, System.Windows.Forms.
LinkLabelLinkClickedEventArgs e)
    {
        //have to rebuild genres, to ensure it is up to date with just added folders
        if (foldersJustAdded)
        {
            oTvGenres.Nodes.Clear();
            oTvGenres.InitTree();
            oTvGenres.ExpandAll();
            foldersJustAdded=false;
        }
        //go find the correct node
        selectLinkGenreTreeNode(oTvGenres.Nodes[0],oLnkGenreTXT.Text);
        otvFolders.Visible = false;
        oTvGenres.Visible = true;
    }

    /// <summary>
    /// Clear any previously selected TreeNode and get the data for the new
    /// selected TreeNode
    /// </summary>
    /// <param name="sender">The source object that raised the event</param>
    /// <param name="e">The event arguments</param>
    private void oTvGenres_AfterSelect(object sender, System.Windows.Forms.
TreeViewEventArgs e)
    {
        //if there is a node selected from last select, clear it highlighted color
        if (tvnGenreSelect != null)
        {
            tvnGenreSelect.BackColor = Color.White;
        }
        //if the currently selected node is correct type and is holding track data
        bool validNode = (oTvGenres.SelectedNode is ClassifiedTreeNode) &&
            ((ClassifiedTreeNode)oTvGenres.SelectedNode).hasPlayableFiles();
        //then we can use this treenode to provide tracks to media player, so allow
menus items
        oMnuAddToTrackListFromGT.Enabled = validNode;

        //Exploring genre treeview
        if (validNode)
        {
            //show the correct status text,and the nodes playable files for the current
node
            ClassifiedTreeNode tn = (ClassifiedTreeNode)oTvGenres.SelectedNode;
            this.oLvTracks.Items.Clear();
            oLblScan.Text = "Exploring " + tn.Text;
            showFilesForNode( tn.PlayableFiles);
        }
        else
        {
            this.oLvTracks.Items.Clear();
            oLblScan.Text = "Exploring " + oTvGenres.SelectedNode.Text;
        }
    }

    /// <summary>
    /// Displays the FolderTree context menu if the user right clicks over the
    /// FolderTree area
    /// </summary>
    /// <param name="sender">The source object that raised the event</param>
    /// <param name="e">The event arguments</param>
    private void oTvGenres_MouseDown(object sender, System.Windows.Forms.MouseEventHandler
e)
    {
        if (e.Button == MouseButtons.Right)
    
```

```

        {
            if (oTvGenres.SelectedNode is ClassifiedTreeNode)
            {
                oCtxtMenuGT.Show(oTvGenres,new Point(e.X,e.Y));
            }
        }
    }

    /// <summary>
    /// Raise an event that event subscribers can use. The event is intended to
    /// be used to add files to the <see cref="SB54_CSAI.MediaPlayer.uctMediaPlayer">
media player</see>
    /// </summary>
    /// <param name="sender">The source object that raised the event</param>
    /// <param name="e">The event arguments</param>
    private void oMnuAddToTrackListFromGT_Click(object sender, System.EventArgs e)
    {
        string[] files = ((ClassifiedTreeNode)oTvGenres.SelectedNode).PlayableFiles;
        addNodeFiles(files);
    }

    /// <summary>
    /// Find the directory node in the FoldersTree treeview
    /// </summary>
    /// <param name="sender">The source object that raised the event</param>
    /// <param name="e">The event arguments</param>
    private void oLnkLocationTXT_LinkClicked(object sender, System.Windows.Forms.
LinkLabelLinkClickedEventArgs e)
    {
        selectLinkFolderTreeNode(otvFolders.Nodes[0],oLnkLocationTXT.Text);
        otvFolders.Visible = true;
        oTvGenres.Visible = false;
        newFolderNodeSelected();
    }

    /// <summary>
    /// Shows the user a <see cref="SB54_CSAI.MediaLibrary.FrmShares">window</see>
    /// that allows them to create new root folders that will be used by the
    /// <see cref="SB54_CSAI.MediaLibrary.FolderTree"> FolderTree</see>
    /// </summary>
    /// <param name="sender"></param>
    /// <param name="e"></param>
    private void oBtnShares_Click(object sender, System.EventArgs e)
    {
        try
        {
            //clear all shares
            NewShares.Clear();
            FrmShares f = new FrmShares();
            DialogResult r = (f.ShowDialog(this));
            f.Dispose();
            //if user selected ok
            if (r == DialogResult.OK)
            {
                //if there was at least one root node, get the FoldersTree to scan it
                if (NewShares.Count > 0)
                {
                    long dAllSize=0;
                    //build the list of root nodes
                    foreach(string s in NewShares)
                    {
                        TopLevelFolders.Add(s);
                        DirectoryInfo d = new DirectoryInfo (s);
                        dAllSize += DirectorySize(d);
                    }
                    double dSize = (dAllSize / Math.Pow(1024,2));
                    //get the size of the selected directory, and alert user how much
                    //they may be asking to scan, remember there is nothing to stop them
selecting
                    //C:\ which could take a very very long time
                    DialogResult mr = MessageBox.Show("You are about to scan " + dSize.
ToString("###,###,###,### MB ") +

```



```

        oTvGenres.Nodes.Clear();
        oTvGenres.InitTree();
    }

    /// <summary>
    /// Retrieves all the previously stored root folders to allow the
    /// <see cref="SB54_CSAI.MediaLibrary.FolderTree">FolderTree </see> to
    /// create the required FolderTree nodes
    /// </summary>
    /// <param name="sender">The source object that raised the event</param>
    /// <param name="e">The event arguments</param>
    private void uctMediaLibrary_Load(object sender, System.EventArgs e)
    {
        //get all local drives
        getLocalDrives();
        //set up main panels
        oPnlFillTop.Height = this.Height - oPnlFillBottomSmall.Height;
        //create DataBase access object
        db = new DBAccess();

        try {
            //Get all the previously stored root folders, there may actually be none.
            //For example if this is the 1st time the ReMP3 application has been run,
            //then there would not be any stored folders
            DataSet ds = db.GetRootFolders(Dns.GetHostName());
            foreach(DataRow dr in ds.Tables["Folders"].Rows)
            {
                TopLevelFolders.Add(dr["folderName"].ToString());
            }
        }
        catch(Exception ex)
        {
            MessageBox.Show("Problem with DB access-\n\n connection: "
                + "\r\n\r\n query: "
                + "\r\n\r\n\r\n" + ex.ToString(),
                "Database Access ERROR", MessageBoxButtons.OK,
                MessageBoxIcon.Error);
        }
        //rescan the stored root nodes, this keeps the users root files up to date
        //It like a FileSystemWatcher but one that works without the need for the ReMP3
        //application to be running all the time. Simply gets the old stored roots on
        //startup, then rescan adding anything that is new since last time.
        setTeeTopLevelFolders();
    }

    /// <summary>
    /// Gets a list of all local drives using a ManagementObjectSearcher object
    /// combined with a SELECT * From Win32_LogicalDisk query
    /// </summary>
    private void getLocalDrives()
    {
        //Local disk type drive = 3
        const int LocalDisk = 3;
        //get a list of all drives on users system, then filter out Local Drives and add
        //them to the localDrives collection
        ManagementObjectSearcher query = new ManagementObjectSearcher("SELECT * From
Win32_LogicalDisk ");
        ManagementObjectCollection queryCollection = query.Get();
        foreach ( ManagementObject mo in queryCollection)
        {
            int diskType = int.Parse(mo["DriveType"].ToString());
            if (diskType == LocalDisk)
            {
                localDrives.Add(mo["Name"].ToString());
            }
        }
    }

    /// <summary>
    /// Starts the <see cref="SB54_CSAI.MediaLibrary.FolderTree">FolderTree </see>
    /// scanning with all the root folders as root nodes
    /// </summary>

```

```

private void setTeeTopLevelFolders()
{
    oTvFolders.ScanFolders(TopLevelFolders);
    oBtnFolders.Enabled = false;
    oBtnGenres.Enabled = false;
}

/// <summary>
/// Subscribed<see cref="SB54_CSAI.MediaLibrary.FolderTree">FolderTree </see>
/// </summary>Scan event. Simply disables/enables certain GUI control based
/// on whether the FolderTree is currently scanning
/// <param name="sender">The source object that raised the event</param>
/// <param name="e">The event arguments</param>
private void oTvFolders_Scan(object sender, ScanningEventArgs e)
{
    //Simply disables/enables certain GUI control based
    //on whether the FolderTree is currently scanning
    oBtnCancel.Visible = e.Scanning;
    oBtnShares.Enabled = !e.Scanning;
    oBtnAddFolder.Enabled = !e.Scanning;
    oBtnFolders.Enabled = !e.Scanning;
    oBtnGenres.Enabled = !e.Scanning;
    oTvFolders.Enabled = !e.Scanning;
    oLblScan.Text = e.FolderName;
    if (!e.Scanning)
    {
        oTvFolders.ExpandAll();
    }
}

/// <summary>
/// Displays the text for a newly selected treeview node and also
/// all the nodes playable files
/// </summary>
private void newFolderNodeSelected()
{
    //if the currently selected node is correct type and is holding track data
    bool validNode = (oTvFolders.SelectedNode is MP3TreeNode) &&
        ((MP3TreeNode)oTvFolders.SelectedNode).hasPlayableFiles();
    //then we can use this treenode to provide tracks to media player, so allow
    menus items
    oMnuAddToTrackListFromFT.Enabled = validNode;
    //if the node is a valid one
    if (validNode)
    {
        //show the correct status text, and the nodes playable files
        MP3TreeNode tn = (MP3TreeNode)oTvFolders.SelectedNode;
        this.oLvTracks.Items.Clear();
        oLblScan.Text = "Exploring " + tn.FullPath;
        //show the nodes playable files
        showFilesForNode( tn.PlayableFiles);
    }
    else
    {
        this.oLvTracks.Items.Clear();
        oLblScan.Text = "Exploring " + oTvFolders.SelectedNode.Text;
    }
}

/// <summary>
/// Displays the list of trings that represent the current
/// nodes files within the main GUI
/// </summary>
/// <param name="files">an array of strings that represent the current
/// nodes files</param>
private void showFilesForNode(string[] files)
{
    try
    {
        System.Windows.Forms.ListViewItem listViewItem;
        //loop through all the files displaying the correct details
    }
}

```

```

        foreach (string s in files)
        {
            FileInfo fi = new FileInfo(s);
            //create a new listitem with Image index 0
            float f = (float)fi.Length / 1024;
            listViewItem = new System.Windows.Forms.ListViewItem(
                new string[] {
                    fi.Name,
                    setupFileTypesMap(fi.Extension.ToString()),
                    f.ToString("###,###,###") + " KB",
                    fi.DirectoryName}, 0);
            //add the items to the listview control
            this.oLvTracks.Items.Add(listViewItem);
        }
    }
    catch(Exception ex)
    {
        MessageBox.Show("The files could not be found",
            "showFilesForNode", MessageBoxButtons.OK,
            MessageBoxIcon.Error);
        return;
    }
}

/// <summary>
/// Sets up internal collection with Audio Files type information
/// </summary>
private void setupFileTypesMap()
{
    //add Window Audio Files
    fileTypes.Add(".wav", "Windows Audio File");
    fileTypes.Add(".snd", "Windows Audio File");
    fileTypes.Add(".au", "Windows Audio File");
    fileTypes.Add(".aif", "Windows Audio File");
    fileTypes.Add(".aifc", "Windows Audio File");
    fileTypes.Add(".aiff", "Windows Audio File");
    fileTypes.Add(".wma", "Windows Audio File");
    fileTypes.Add(".mp3", "Windows Audio File");

    //add Window Media Files
    fileTypes.Add(".asf", "Windows Media File");
    fileTypes.Add(".wm", "Windows Media File");
}

/// <summary>
/// Takes an file extension and returns the file type full name
/// </summary>
/// <param name="key">a string representing the file extension, ".wav" for example</param>
/// <returns>a string representing the file type, "Windows Audio File" for example</returns>
private string setupFileTypesMap(string key)
{
    //add Window Audio Files
    int idx = fileTypes.IndexOfKey(key.ToLower());
    return (string)fileTypes.GetByIndex(idx);
}

/// <summary>
/// Takes a TreeNode and removes all the <see cref="SB54_CSAI.DatabaseAccess.DBAccess">database stored information</see>
/// </summary>for the given node
/// <param name="n">The TreeNode to remove data for</param>
private void removeLoggedFolder(TreeNode n)
{
    //loop through all the subnodes
    foreach (TreeNode nd in n.Nodes)
    {
        if (nd is MP3TreeNode)
        {
            //and remove the logged data
            db.RemoveFolder(((MP3TreeNode)nd).FullPath.Trim());
        }
    }
}

```

```

    }
    //do it RECURSIVELY
    removeLoggedFolder(nd);
}
}

/// <summary>
/// Displays the correct meta data for the working MP3
/// </summary>
private void setMP3Details()
{
    try
    {
        oPnlDetails_TopTXT.Text = (workingMP3.fileFileName == null ? string.Empty :
workingMP3.fileFileName.Trim() );
        oLnkLocationTXT.Text = (workingMP3.filePath == null ? string.Empty :
workingMP3.filePath.Trim() );
        oLblTitleTXT.Text = (workingMP3.id3Title == null ? string.Empty : workingMP3
.id3Title.Trim() );
        oLnkAlbumTXT.Text = (workingMP3.id3Album == null ? string.Empty : workingMP3
.id3Album.Trim() );
        oLnkArtistTXT.Text = (workingMP3.id3Artist == null ? string.Empty :
workingMP3.id3Artist.Trim() );
        int id3Genre = (workingMP3.id3Genre == 0 ? 0 : (int)workingMP3.id3Genre);
        oLnkGenreTXT.Text = (id3Genre == 0 ? string.Empty : oTvFolders.getGenreName
(id3Genre));
        oLblYearTXT.Text = (workingMP3.id3Year == null ? string.Empty : workingMP3.
id3Year.ToString().Trim() );
        oLblCommentTXT.Text = (workingMP3.id3Comment == null ? string.Empty :
workingMP3.id3Comment.Trim() );
        //disable links that have no id3 tag data
        oLnkAlbumTXT.Enabled = workingMP3.id3Album != null;
        oLnkArtistTXT.Enabled = workingMP3.id3Artist != null;
        oLnkGenreTXT.Enabled = id3Genre != 0;
    }
    catch(Exception ex)
    {
        MessageBox.Show("File handle error",
            "setMP3Details", MessageBoxButtons.OK,
            MessageBoxIcon.Error);
        return;
    }
}

/// <summary>
/// Takes a currFile filename and sees if MP3 details can be retrieved for this file
/// </summary>
/// <param name="currFile">the current file</param>
/// <returns>true if details were available for the current file</returns>
private bool setupDetails(string currFile)
{
    try
    {
        string fullPath="";
        //IF the currently shown treeview is oTvFolders, can get the full file path
        //by simply asking the MP3TreeNode the fullpath name
        if (oTvFolders.Visible == true)
        {
            fullPath = ((MP3TreeNode)oTvFolders.SelectedNode).FullPath + @"\" +
currFile;
        }
        //IF the currently shown treeview is oTvGenres, it is a little bit more
painful, as this node
        //could hold files from all over the place, at various directories, so a
loop thorough all
        //files in the currently selected node has been done to find the file
that matches the
        //selected listview item, and then need to use the file found as the
fullpath string used to
        //create details
        else if (oTvGenres.Visible == true)
        {

```

```

        if (oTvGenres.SelectedNode != null)
        {
            string[] files = ((ClassifiedTreeNode)oTvGenres.SelectedNode).
PlayableFiles;
            foreach (string s in files)
            {
                if (s.IndexOf(currFile,0) != -1)
                {
                    //found the file, so copy the details
                    fullFilePath=s;
                    break;
                }
            }
        }
        //Got a fullFilePath now, so attempt to get the MP3 details for it
        if (!fullFilePath.Equals(string.Empty))
        {
            //check the file is an MP3 file
            FileInfo fi = new FileInfo(fullFilePath);
            if (fi.Extension.ToLower().Equals(".mp3"))
            {
                //Is MP3 so read its TAG
                workingMP3 = new MP3(fi.DirectoryName, fi.Name);
                MP3Tag.readMP3Tag (ref workingMP3);
                //all ok
                return true;
            }
            else
            {
                //not an MP3
                return false;
            }
        }
        //NO DETAILS FOUND
        return false;
    }
    catch(Exception ex)
    {
        MessageBox.Show("File handle error",
            "setupDetails", MessageBoxButtons.OK,
            MessageBoxIcon.Error);
        return false;
    }
}

/// <summary>
/// Sets the oBtnDetails with the parameters supplied
/// </summary>
/// <param name="imgIdx">The image list index</param>
/// <param name="NewCaption">The new caption</param>
private void setButton(int imgIdx, string NewCaption)
{
    oBtnDetails.ImageIndex = imgIdx;
    oBtnDetails.Text = NewCaption;
    oBtnDetails.Invalidate();
    oBtnDetails.Refresh();
}

/// <summary>
/// Searches the TreeNode specified by tree for the text specified by
/// nodetext and if found will select the node where the text was found
/// </summary>
/// <param name="tree">The TreeNode to find the new selected node in</param>
/// <param name="nodetext">The node name to try and select</param>
private void selectLinkGenreTreeNode(TreeNode tree, string nodetext)
{
    //loop through all child nodes for the specified TreeNode
    foreach (TreeNode n in tree.Nodes)
    {
        //if a match is found, make the active node the selected node
        if ( n.Text.ToLower().Trim().Equals(nodetext.ToLower().Trim()) )
        {

```

```

        oTvGenres.SelectedNode = n;
        oTvGenres.SelectedNode.BackColor = SystemColors.Highlight;
        tvnGenreSelect =n;
        return;
    }
    else
    {
        //node was not found, so need to look RECURSIVELY
        selectLinkGenreTreeNode(n,nodetext);
    }
}

/// <summary>
/// Searches the TreeNode specified by tree for the text specified by
/// nodetext and if found will select the node where the text was found
/// </summary>
/// <param name="tree">The TreeNode to find the new selected node in</param>
/// <param name="nodetext">The node name to try and select</param>
private void selectLinkFolderTreeNode(TreeNode tree, string nodetext)
{
    //loop through all child nodes for the specified TreeNode
    foreach (TreeNode n in tree.Nodes)
    {
        if (n is MP3TreeNode)
        {
            //if a match is found, make the active node the selected node
            if ( ((MP3TreeNode)n).FullPath.Equals(nodetext))
            {
                otvFolders.SelectedNode = n;
                otvFolders.SelectedNode.BackColor = SystemColors.Highlight;
                tvnFolderSelect =n;
                return;
            }
            else
            {
                //node was not found, so need to look RECURSIVELY
                selectLinkFolderTreeNode(n,nodetext);
            }
        }
    }
}
#endregion
#endregion
#region ListViewItemComparer INNER CLASS
/// <summary>
/// Implements the manual sorting of items by columns.
/// </summary>
class ListViewItemComparer : IComparer
{
    #region Instance Fields
    //Instance Fields
    private int col;
    #endregion
    #region Public Constructors
    /// <summary>
    /// Default Constructor, assumed ListView column 0 is to be sorted
    /// </summary>
    public ListViewItemComparer()
    {
        col=0;
    }
    /// <summary>
    /// Column assigned constructor
    /// </summary>
    /// <param name="column">The ListView column to sort</param>
    public ListViewItemComparer(int column)
    {
        col=column;
    }
    #endregion
    #region Public Methods

```

```

        /// <summary>
        /// Compares two objects and returns a value indicating whether one is less than
        '
        /// equal to or greater than the other.
        /// </summary>
        /// <returns>Less than zero x is less than y, Zero z = y,
        /// Greater than zero x is greater than y</returns>
        public int Compare(object x, object y)
        {
            //must not be a number so, use string comaprison
            return String.Compare(((ListViewItem)x).SubItems[col].Text, ((ListViewItem)
y).SubItems[col].Text);
        }
        #endregion
    }
    #endregion
}
#region AddFilesEventArgs CLASS
/// <summary>
/// Provides the AddFilesEventArgs used with the <see cref="SB54_CSAI.MediaLibrary.
uctMediaLibrary">
/// Media Library Control</see> addFiles event. The addFiles event is used to add files
to the
/// <see cref="SB54_CSAI.MediaPlayer.uctMediaPlayer">media player control</see>
/// </summary>
public class AddFilesEventArgs : EventArgs
{
    #region Instance Fields
    ///instance fields
    private string[] files;
    #endregion
    #region Public Constructor
    /// <summary>
    /// Constructs a new AddFilesEventArgs object using the parameters provided
    /// </summary>
    /// <param name="files"></param>
    /// </summary>
    public AddFilesEventArgs(string[] files)
    {
        this.files = files;
    }
    #endregion
    #region Public Methods/Properties
    /// <summary>
    /// Returns an array of string representing the playable files that should be
    /// added to the <see cref="SB54_CSAI.MediaPlayer.uctMediaPlayer">
    /// media player</see>
    /// </summary>
    /// <returns>An array of string representing the playable files</returns>
    public string[] PlayableFiles
    {
        get { return files;}
    }
    #endregion
}
#endregion
#endregion
}

```