

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace TestReadAccessDB
{
    public partial class frmCheckSong : Form
    {
        public frmCheckSong()
        {
            InitializeComponent();
        }

        private void CboSongs_SelectedIndexChanged(object sender, EventArgs e)
        {
            // OK button enabled IF something is selected
            cmdOK.Enabled = cboSongs.SelectedIndex != -1;
        } // CboSongs_SelectedIndexChanged

        private void CmdOK_Click(object sender, EventArgs e)
        {
            // pass information back to main form before closing
            if (radSelectSong.Checked)
            {
                FrmMain.Gl.cSongType = 'S';
                FrmMain.Gl.sSongTitle = cboSongs.SelectedItem.ToString();
            }
            else if (radNewSong.Checked)
            {
                FrmMain.Gl.cSongType = 'N';
            }
            else if (radEnterSong.Checked)
            {
                FrmMain.Gl.cSongType = 'E';
                FrmMain.Gl.sSongTitle = txtNewSongTitle.Text;
            }
            else
            {
                MessageBox.Show("CmdOK_Click, this should never happen.");
                this.Close();
            }
        } // CmdOK_Click

        private void FrmCheckSong_Load(object sender, EventArgs e)
        {
            // get all the song titles from DB
            ADODB.Recordset rs = new ADODB.Recordset();
            lblSongTitle.Text = FrmMain.Gl.sSongTitle;
            rs.Open("SELECT SongTitle FROM Songs ORDER BY SongTitle",
                FrmMain.Gl.cn, ADODB.CursorTypeEnum.adOpenDynamic,
```

```

        ADODB.LockTypeEnum.adLockOptimistic);
    // load the song titles into the combo box
    rs.MoveFirst();
    while (!rs.EOF)
    {
        cboSongs.Items.Add(rs.Fields["SongTitle"].Value);
        rs.MoveNext();
    }
    // if song title ends with 3 dots,
    // calling next line will try to find it
    RadioButtonsCheckChanged();
} // FrmCheckSong_Load

private void RadEnterSong_CheckedChanged(object sender, EventArgs e)
{
    if (radEnterSong.Checked)
        RadioButtonsCheckChanged();
} // radEnterSong_CheckedChanged

private void RadioButtonsCheckChanged()
{
    if (radSelectSong.Checked)
    {
        cboSongs.Enabled = true;
        txtNewSongTitle.Enabled = false;
        cmdOK.Enabled = false;
        // last 4 chars of string: mystring.Substring(mystring.Length - 4);
        // if song searched for ends with 3 dots, try to find what it's a
        // shortened version of
        if (lblSongTitle.Text.Substring(lblSongTitle.Text.Length - 1) == "..." ||
            lblSongTitle.Text.Substring(lblSongTitle.Text.Length - 3) == "...")
        {
            cboSongs.SelectedIndex = 0; // 1st song; next line will blow up if
            blank
            while (string.Compare(cboSongs.SelectedItem.ToString(),
                lblSongTitle.Text) == -1)
            {
                cboSongs.SelectedIndex += 1;
                cboSongs.SelectedIndex -= 1;
            } // if song title ends with 3 dots, 1 or 3 characters
        }
    }
    else if (radNewSong.Checked)
    {
        cboSongs.Enabled = false;
        txtNewSongTitle.Enabled = false;
        cmdOK.Enabled = true;
    }
    else // radEnterSong.Checked
    {
        cboSongs.Enabled = false;
        cboSongs.SelectedIndex = -1; // nothing selected
        txtNewSongTitle.Enabled = true;
        cmdOK.Enabled = true;
    }
}

```

```
    }  
} // RadioButtonsCheckChanged  
  
private void RadNewSong_CheckedChanged(object sender, EventArgs e)  
{  
    if (radNewSong.Checked)  
        RadioButtonsCheckChanged();  
} // radNewSong_CheckedChanged  
  
private void RadSelectSong_CheckedChanged(object sender, EventArgs e)  
{  
    if (radSelectSong.Checked)  
        RadioButtonsCheckChanged();  
} // radSelectSong_CheckedChanged  
} // class  
} // namespace
```