My Rss Player项目文档

# 程序功能描述及使用场景

1. 欢迎界面



当应用程序第一次打开时，会出现如上的欢迎界面。

1. RssReader部分



功能说明：如上图所示，Rss阅读器部分实现的功能有：1.根据输入的Rss链接抓取Rss信息；2.根据关键词搜索对应的Rss新闻；3.跳转到特定的Rss网页查看详细的Rss信息。

使用场景：如上图所示为程序Rss阅读器的界面，点击按钮我们可以添加我们感兴趣的Rss站点， 在图中的文本框中输入Rss站点的链接即可。



点击我们可以对RssItem的信息进行查找，点击我们可以在网页上浏览我们感性兴趣的新闻详细内容。

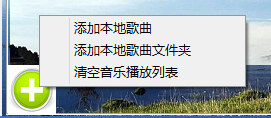
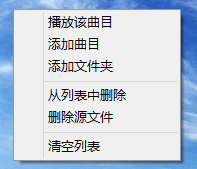
1. MusicPlayer部分



功能说明：1.播放、暂停、快进、快退、上一曲、下一曲音乐播放控制；2.添加本地歌曲、以及本地文件夹歌曲、清空音乐播放列表；3.根据关键字搜索播放列表中的歌曲；4.可以选择特定的音乐播放模式；5.音乐播放音量的控制；6.桌面歌词的显示；7.应用程序托盘实现。

使用场景：

**播放控制：**如上图所示，即为程序的音乐播放器界面，其中这一组按钮是控制音乐播放的按钮。从左至右依次是上一曲、快退、播放/暂停、停止、快进、下一曲。

**添加音乐、搜索音乐：**点击按钮之后，我们可以向音乐播放列表中添加曲目、或者清空音乐列表。点击按钮，我们可以在出现的音乐搜索框中查找已添加的本地曲目。我们可以在图上的搜索框中输入我们要查询的信息，音乐播放列表中要是有符合项，那么这一项将会被置到音乐列表头。同时我们还可以在播放列表区域右击鼠标，在弹出的菜单项中我们可以进行对应的操作。

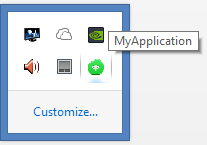
**播放模式：**点击音乐播放模式复选框，我们可以选择音乐的播放模式。在这里有四种模式供我们选择。

**音量控制：**点击按钮，我们可以左右滑动音量条来调节音乐播放音量的大小。

**桌面歌词：**点击按钮，可以打开或者关闭桌面歌词的显示。

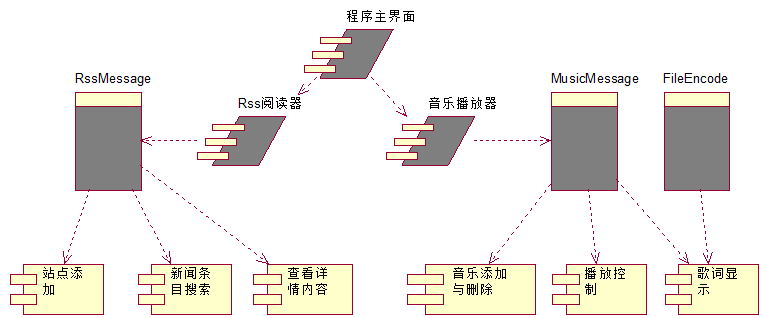


**控制按钮：**如图所示，程序右上角的三个按钮分别表示退出程序、最小化到任务栏、最小化到托盘。

**托盘功能：**在托盘中我们也可以找到该程序的图标，右击改程序图标我们可以退出该程序。

# 程序架构设计

改程序的架构设计如下图所示：



程序主界面包含两个子界面，这两个子界面分别是主界面的tab选项卡。两个最主要的实现类为RssMessage类和MusicMessage类；其中RssMessage类中封装了Rss站点的新闻链接，新闻详情等具体信息，程序中还有一个RssFunction.cpp文件，这个文件中包含了Rss解析、及将解析出的Rss信息显示到界面，以及存储Rss信息到本地磁盘的函数；MusicMessage类中包含了音乐的本地全路径、本地短路径等基本信息，除此之外还包含了音乐播放控制的函数，音量控制函数，以及歌词显示函数。在歌词显示时，还用到了字符转码子程序，因为不同播放器歌词文件的编码是不一样的，比如：QQ音乐的歌词编码为UTF-8编码，而酷狗音乐的是ACALL编码，FileEncode子程序做的就是将读入的歌词字符统一转化为UTF-8编码，使程序的歌词显示不会出现乱码。

# 模块的功能描述

程序主要分为两大功能模块：Rss阅读器模块和Music音乐播放器模块。下面简述一下各个子模块间的功能：

Rss阅读器模块：

1. 添加站点模块：添加对应的Rss站点链接，之后我们便可以查看所添加Rss站点的最新新闻程序。
2. 新闻条目搜索模块：根据输入的关键字搜索对应的新闻条目（是根据新闻条目的标题来检索的），被搜索到的条目对应字体会变成红色。
3. 查看新闻详情模块：让用户直接链接到网上对应的新闻详情页面，直接查看新闻的详细内容。

音乐播放器模块：

1. 音乐播放控制：对音乐基本播放逻辑的实现，如快进快退、上一曲、下一曲、播放\暂停、停止等逻辑。
2. 添加曲目：程序支持从本地文件或者本地文件夹中导入音乐曲目。
3. 歌词文件编码转化：程序支持读入不同格式的歌词文件（程序内部都将他们转化成了UTF-8编码），使歌词显示不会出现乱码。
4. 音量控制：程序支持音乐播放音量大小的调节。
5. 删除曲目：程序支持从列表删除或者从本地文件删除音乐曲目。

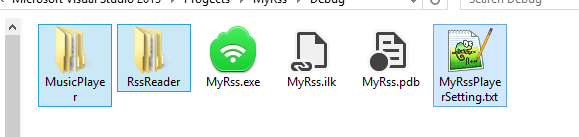
主程序模块：

1. Tab选项卡：程序使用Tab选项卡实现Rss阅读界面与音乐播放界面的切换。
2. 托盘管理：程序支持最小化到托盘的功能。
3. 状态栏显示：程序的状态栏中将会显示一些与程序相关的信息，如程序开启状态等。

# 程序设计原则

该程序使用了面向对象的程序设计原则，实现了程序中各个模块之间的低耦合性，在程序的开发过程中，我尽量复用了代码，减少了重复代码的行数，例如程序中图片按钮的显示（以及点击效果）这一部分，我就统一的将这些代码写在了void DrawBkgndToButton(HWND hwnd, WPARAM odBtnID, LPDRAWITEMSTRUCT lpDrawItemStruct) 函数中，这样使程序中按钮图像载入都集中到这一个函数中来，这样既增强了代码的可读性，又减少了代码的行数。另外我将对音乐播放操作控制的函数也加入了MusicMessage类中，（原来写MusicMessage类的时候只是想在其中存放音乐文件的相关信息）这些播放控制函数都是MusicMessage类的静态函数，可以直接通过MusicMessage类名来调用，这样将类的实体信息以及操作集合到了MusicMessage类中，方便了对音乐文件实体的信息操作。

# 技术细节描述

1. 程序的音乐播放逻辑实现主要是靠MCIERROR mciSendString (   
   LPCTSTR lpszCommand, // 指令字符串  
   LPTSTR lpszReturnString, // 返回信息字符串  
   UINT cchReturn, // 返回信息字符串长度  
   HANDLE hwndCallback // 回叫窗口的句柄  
   ); 这个函数，在第一个参数（指令字符串）中指定的音乐文件路径名必须是短路径格式（8.3格式文件名），在Windows8.1操作系统中8.3格式文件名会被禁用，这会导致GetShortPathName函数调用失败，得不到文件的短路径，从而程序的音乐播放功能彻底失效，（参考链接：<http://msdn.microsoft.com/en-us/library/aa364989(VS.85).aspx>） ，可以将8.3格式文件名功能打开，再重新向磁盘中考入音乐文件，程序功能即恢复正常。（8.3格式文件功能打开方式可参考：<http://jingyan.baidu.com/article/295430f10e6bc80c7e00503c.html>）。
2. 程序在许多地方使用了自定义的Windows消息函数（#define WM\_XXX WM\_USER+N）,这些自定义的消息函数主要是用来保证子窗口与主窗口之间的位置一致性，例如在点击音量按钮的时候会弹出音量调节框，这时候移动主窗口，那么音量调节子窗口也必须随主窗口移动，不能出现错位，程序中的很多WM自定义消息干的就是这件事情。
3. 程序在运行目录会自动生成

MusicPlayer和RssReader文件夹，以及MyRssPlayerSetting.txt文件，在 MyRssPlayerSetting.txt实际上设置的是控制程序欢迎界面弹出的标志（实现只有在第一次使用该程序时，欢迎界面才会弹出）；MusicPlayer文件夹下的src子文件夹下的MusicListMessage.mls文件中记录了程序音乐播放列表的信息，该文件的格式如下：

音乐文件名

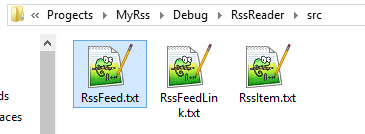
音乐短路径名称

音乐全路径名称

音乐文件名

音乐短路径名称

音乐全路径名称

音乐文件的写入信息的实现在InitMusicPlayer.cpp文件的bool SaveMusicPlayerMessage(HWND hwnd) 函数中；RssReader文件夹下src子文件夹中有这三个文件，其中RssFeed.txt保存的Rss站点的名称信息，RssFeedLink.txt保存的是Rss站点的链接信息，而RssItem中保存的则是每个Rss站点对应的新闻列表的详情信息，RssReader信息的写入逻辑在InitRssReader.cpp中的bool SaveRssMsg(HWND hwnd) 函数中实现。