MP3-Фонотека «Sirius Navigator»

Программа управления и навигации МРЗ-архива

О программе

Сириус - ближайший наш сосед и ярчайшая звезда на небе, видимая из любой точки Земли. Сириус - наша путеводная звезда, на которую мы всегда можем положиться и по которой мы можем ориентироваться среди бездонного моря звёзд. Поэтому эта программа названа в честь этой замечательной звезды - чтобы она помогала вам ориентироваться среди бесчисленного количества аудиофайлов. Сириус - звезда двойная. И так же наша программа предназначена как для опытных операторов БД, так и для обычных пользователей. Желаем вам удобного и лёгкого пути!

Программа ориентирована для создания и работы с MP3-фонотекой на основе БД MySQL 5.х. Программа позволяет в автоматическом режиме добавлять\удалять файлы из фонотеки. Проводить сортировку, выборку, поиск по БД. Позволяет выполнять запросы на исполнение и выборку "на лету". Позволяет работать с длинными полями - для добавления изображений и текстов песен в фонотеку. В неё включены функции автоматического создания HTML-страницы для публикации фонотеки в сети в интернет. А так же функции автоматического администрирования БД.

Файлы исходного пакета

Программа:

- SiriusNav.exe исполняемый файл навигатора
- *account.ini* содержит информацию о текущих аккаунтах
- setup.cfg содержит установочные и наладочные инструкции
- library.dll библиотека функций и методов
- Sirius DB.pdf данный мануал

База данных:

• Создаётся\хранится в системных каталогах MySQL

Требования к системе

Hardware:

- i586\i686 процессор (Intel, AMD 1.2GHz+)
- 512 мб RAM
- 64 мб видеокарта с поддержкой DirectX 8+
- Звуковая карта с поддержкой DirectX 8+
- Клавиатура, мышь

•

Software:

- MS Windows (XP\MCE\2003-2005\Vista)
- DirectX 8+
- MySQL 5.x

Установка

- Распакуйте архив <u>SiriusNav</u> в любой доступный вам каталог. Старайтесь выбрать место удобным для использования, именно этот каталог будет использован как хранилище созданных HTML-файлов.
- Для работы <u>SiriusNav</u> необходимо наличие на компьютере БД **MySQL** версии 5.х. Для установки программы проконсультируйтесь с вашим администратором. В простейшем случае выберете тип установки **Typical**, **Standart Configuration**. Введите и заполните по требованию пароль **root**. Данный пользователь\пароль будут использованы при первом запуске **SiriusNav**.

[это важно знать администратору!]

Новую версию MySQL можно скачать здесь:

http://dev.mysql.com/downloads/mysql/

http://www.mysql.ru/download/

Программа разработана под MySQL 5.045, тем не менее рассчитана на работу со всеми БД MySQL версий 5.х (и выше). На момент выхода программы в релиз стабильной версии MySQL 5.1 в поставке Community Server не было. Alpha-Beta версию используйте на собственный страх и риск.

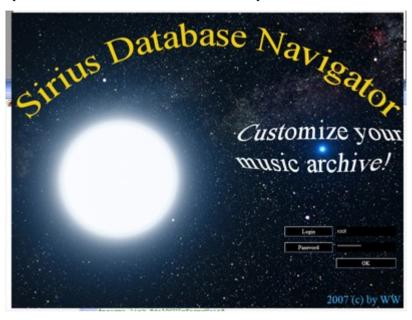
Логин\пароль гоот используется всего один раз при конфигурации <u>SiriusNav</u>. Эта комбинация нигде не сохраняется в программе и не изымается, и не пересылается. Напомним, что хранение учетной записи гоот — прямая и важнейшая обязанность администратора.

Если у вас установлена версия ниже рекомендованной, возможна частичная или полная несовместимость с <u>SiriusNav</u>. При установке двух разных версий MySQL избегайте конфликтов программного обеспечения и выделенных MySQL портов.

- После установки и настройки MvSQL запустите файл SiriusNav.exe
- В поле login\password введите ваши root логин и пароль.
- Если конфликтов не будет обнаружено и комбинация логина и пароля будет верной, то <u>SiriusNav</u> произведёт настройку и выйдет из программы с соответствующим сообщением.
- Запустите программу еще раз и введите логин\пароль администратора, описанные ниже.

SiriusNav - Интерфейс

После запуска программы пользователь увидит на экране окно приветствия и программа его попросит ввести имя пользователя и пароль.



В поля **Login и Password** вводятся логин и пароль соответственно. При вводе пароль маскируется символами *. По нажатии кнопки ОК проверяется правильность ввода пароля и логина и привелегии пользователя (**Admin** или **User**). При ошибке ввода или при отсутствии текущей учетной записи выдаётся ошибка и программа автоматически закрывается.

В исходной поставке БД существуют два пользователя:

- Администратор с логином «admin» и паролем «admin»
- Пользователь с логином «guest» и паролем «guest»

При первой же возможности смените пароль администратора!

При первом запуске <u>SiriusNav</u> программа автоматически создаст требуемые аккаунты, таблицы, процедуры и функции. Это в некоторых случаях может занять весьма много времени. Для первого запуска используйте ваш гооt логин и пароль!

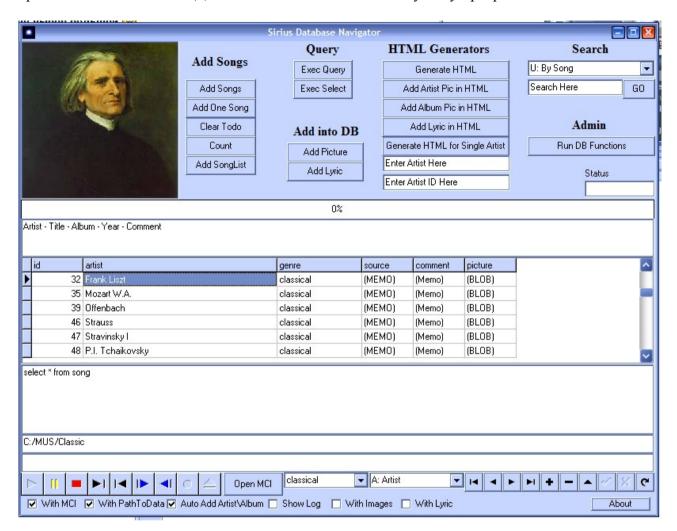
[это важно знать администратору!]

Дополнительные логин и пароль можно добавить в конфигурационный файл *account.ini* соблюдая синтаксис:

[ROLE] LOGIN PASSWORD

Поле ROLE может принимать значения [ADMIN] для привилегий администратора или [USER] для привилегий пользователя. Поля LOGIN и PASSWORD записываются не в открытом виде, а в виде хэш-кода, полученном необратимым шифрованием MySQL5.x функцией PASSWORD('_data_'). Добавляя\изменяя информацию о логине и пароле заранее убедитесь, что создан одноименный пользователь в БД с паролем. Напомним, что любой пользователь с привилегией [ADMIN] может создать как обычных пользователей, так и администраторов с использованием стандартного SQL. И еще раз — убедитесь в корректности пароля и не предоставляйте роли администратора без лишней нужды.

После успешной авторизации пользователь будет проинформирован о своей роли и о времени подключения к БД. Затем он попадет к основному окну программы.



С различными ролями пользователь получит ограниченный доступ к интерфейсу. Так, в роле пользователя недоступны все кнопки, за исключением формы поиска. А так же ограничен доступ к таблицам.

- Большую часть окна занимает таблица (grid), созданная посредством выборки из БД. Это основной инструмент просмотра и администрирования БД Sirius.
- В связке с таблицей работает навигатор по БД (в правом нижнем углу), в режиме администратора позволяющий передвигаться по БД (<, >, |<, >|) а так же добавлять (+), удалять записи (-), принимать изменения (V) и отклонять их (X)
- Непосредственно под таблицей (grid) находится поле ввода, которое предназначено для ввода SQL-запросов. Такие запросы далее будем называть «живые» запросы.
- Под полем запросов находятся служебные поля, в которых отображаются поля из БД <u>SiriusNav</u> типа TEXT. Верхнее поле обычно содержит информацию о пути к файлу\ папке в режиме администратора. Нижнее обычно отображает поля описания артиста\ альбома или текст песни.
- Непосредственно над таблицей (grid) находится поле, в которое обычно записывается лог выполнения операций.

• Поля лога находится индикатор прогресса выполнения операции. При длительных процессах в нём отображается ход выполнения операции.

• Группа кнопок Add songs

- Add Songs предназначено для добавления множества песен в список обработки. После нажатия пользователю будет выведен диалог с предложением выбрать каталог, откуда будут добавлены песни. Песни добавляются при надобности либо без вложенных каталогов, либо с вложенными каталогами неограниченной глубины. Добавляются песни имеющие только расширение *.mp3. После добавления нескольких песен загорится окошко Status и отобразиться количество песен в очереди.
- Add One Song предназначено для добавление песен по одной. После нажатия пользователю будет выведен диалог с предложением выбрать песню. Добавляются песни имеющие только расширение *.mp3. После добавления песни загориться окошко Status и отобразиться количество песен в очереди.
- Clear Todo убирает очередь в списке обработки. После очистки очереди погаснет окошко Status — и отобразится «None».
- Count показывает сколько песен находится в очереди \ сколько песен было добавлено за прошлый процесс добавления.
- Add SongList предназначено для добавления песен в БД. Добавление происходит автоматически и может занять большое количество времени. При добавлении песен из очереди обработки в БД автоматически считываются значения из ID3v1 mp3-тэга. По их результатам составляется информация о песне название, исполнитель и пр.. В случае отсутствия-повреждения тэга запись всё равно заносится в БД, но с меткой **F**, означающее возможно неправильный ввод в БД. Для возможного исключения данной ситуации используются различные варианты обработки данных, которые будут описаны ниже. Во время добавления состояние показывает индикатор прогресса.

[это важно знать администратору!]

Метки в таблице **songs** нужны для возможной пост-обработки записи. Зная некорректную запись, её можно проще и быстрее найти и исправить. Для этого достаточно в режиме администратора использовать запросы вида:

select * from song where full_data='F'

delete from song where full_data='F'

• Группа кнопок Query

- о **Exec Query** предназначена для исполнения «живых» запросов, не возвращающих набор данных (все запросы без *select*). Текст запроса программа читает из поля ввода запросов.
- Exec Select предназначена для исполнения «живых» запросов, возвращающих набор данных (все запросы с *select*). Текст запроса программа читает из поля ввода запросов.

• Группа кнопок **Add into DB**

Аdd Picture — предназначена для добавления изображения в БД. При нажатии открывается диалог, в котором пользователю предлагается выбрать рисунок формата *.jpg или *.bmp. Не рекомендуется использовать большие изображения (более 300x300px). При превышении предела в (500x500px или 30.000sqpx)

- программа просто не добавит картинку в БД. В зависимости от открытой таблицы album или artist и текущей выбранной записи в неё и будет добавлено изображение.
- Add Lyric предназначена для добавления массивов текста в БД. При нажатии открывается диалог, в котором пользователю предлагается выбрать текст формата *.txt или *.html. В зависимости от открытой таблицы song, album или artist и текущей выбранной записи в неё и будет добавлен текст.

[это важно знать администратору!]

Ограничение на размер изображения и длину текста не случайно. Оно вызвано ограничением на бинарные поля в MySQL. Несмотря на ограничение, использование больших объемов информации в БД не рекомендуется. Это увеличит как загрузку сервера, так и время поиска.

• Группа кнопок HTML Generators

- **Generate HTML** предназначено для создания HTML-файлов на основе БД <u>SiriusNav</u> для последующей публикации в сети интернет. При нажатии на кнопку происходит генерация HTML-файлов в каталоге *HTML*. В папку *ARTISTS* добавляются изображения исполнителей, в *ALBUM* обложки альбомов, а в *LYRICS* тексты песен. Всё это связывает финальный файл *index.html*. Процесс при разных параметрах занимает различное время. В зависимости от количества песен в БД процесс может занять очень большое количество времени. В течение которого прогресс отображается на индикаторе.
- Add Artist Pic in HTML предназначено для добавления изображений из БД в каталог *ARTIST*. Аналогично со встроенным процессом в **Generate HTML**. В зависимости от количества исполнителей в БД процесс может занять очень большое количество времени. В течение которого прогресс отображается на индикаторе.
- Add Album Pic in HTML предназначено для добавления изображений из БД в каталог *ALBUM*. Аналогично со встроенным процессом в **Generate HTML**. В зависимости от количества альбомов в БД процесс может занять очень большое количество времени. В течение которого прогресс отображается на индикаторе.
- Add Lyric in HTML предназначено для добавления текста песен из БД в каталог *LYRIC*. Аналогично со встроенным процессом в **Generate HTML**. В зависимости от количества песен в БД процесс может занять очень большое количество времени. В течение которого прогресс отображается на индикаторе.
- о Generate HTML for Single Artist предназначено для создания HTML-файла одного артиста. В качестве параметров используются значения нижележащих полей. В поле сразу под кнопкой вводится артист, страницу которого необходимо создать. В следующем поле вводится уникальный ID артиста (INT), который при процедуре Generate HTML задаётся автоматически. Поэтому используйте уникальный цифровой ID артиста для того, чтобы данный пункт не затёр существующие файлы. В папке HTML будет создан файл с именем введённого вами ID.

• Группа Search

• Выпадающий список — критерий поиска. В зависимости от выбора поиск производится по нужным таблицам, параметрам или функциям. В роле пользователя доступны только поля, начинающиеся с индекса U.

- о **Поле ввода** сюда вводится ключево слово для поиска.
- о **Кнопка GO** старт поиска. В зависимости от количества записей в БД может занять продолжительное время.

• Группа кнопок **Admin**

- о **Run DB Functions** запускает механизм администрирования БД (удаление «пустых» артистов и альбомов, дубликатов, добавление несуществующих в таблицах asrtist или album исполнителей\альбомов, присутствующих в таблице song). Нажатие кнопки вызовет к показу предупреждению, что данный процесс может занять продолжительное время и может привести к зависанию компьютера. Использование данного инструмента не даёт 100% гарантии в успешном завершении операции. При произведении требуемых операций прогресс отображается на индикаторе.
- AES Memo [optional] данная кнопка доступна только при наличии исходных кодов. Предназначена для кодирования информации, помещенной в поле запроса. Данные из поля кодируются алгоритмом AES (128 бит) и сохраняются под именем, введённым в поле «Enter Artist Here».

[это важно знать администратору!]

AES Memo не поставляется в комплекте с БД <u>SiriusNav</u>. Данный инструмент служит для кодирования важной информации. Тоже самое можно воспроизвести функцией AES_ENCRYPT('_data_', '_key_')

• Группа медиа-проигрывателя

- Медиа-проигрыватель доступен только администратору и доступен при работе с выборкой из таблицы song. Стандартные кнопки медиапроигрывателя -«перемотка», «играть», «пауза» и «стоп».
- Данные для медиа-проигрывателя берутся из текущей записи таблицы song, тогда для открытия файла используется кнопка Open MCI, и затем нажатие кнопки «играть» (>).

[это важно знать администратору!]

Медиа-проигрыватель - удобный инструмент для администратора, позволяющий без лишнего поиска по диску ознакомится с той или иной песней.

• Группа переключателей (расположена внизу)

- With MCI при добавлении песен в БД вычисляется их длина в формате (чч:мм:сс). Является обязательным для осмысленной БД <u>SiriusNav</u>.
- With PathToData в случае отсутствия\повреждения ID3v1 тэга данные об исполнителе\альбоме\годе берутся из значений каталогов. Так, например путь к некоторой песни <u>C:\MUS\Shocking Blue\(1970\) At Home\Scorpio Dance.mp3</u> будет преобразован в параметры Artist = Shocking Blue, Album = At Home, Year = 1970. При этом нужно отдавать отчет, что при отсутствии тэга запись всё равно получится «ложной» и ей будет поставлен флажок «F». Песня запишется как Unknown Song. Альбомы и года обрабатываются в виде: \((YEAR) Album\) или \(YEAR Album\) или \(Album\)

- Auto Add Artist\Album при добавлении песни исполнитель или альбом, к которому принадлежит песня добавляются в соответствующие таблицы автоматически
- Show Log при добавлении в БД показывает лог добавления песен, функции обработки в поле лога.
- With Images при создании HTML-файлов в процессе будут добавлены так же избражения исполнителей и обложки альбомов (при отмеченной галочке в режиме поиска для пользователей будут доступны изображения)
- With Lyric при создании HTML-файлов в процессе будут добавлены так же и тексты песен

• Выпадающие списки

- Жанр в <u>SiriusNav</u> при изменении жанра исполнителя или альбома изменяется жанр так же и у принадлежащих ему песен или альбомов
- Таблицы отображает в таблице (grid) нужные таблицы, которые отмечены (А
 для администратора) и (U пользователя)
- **Кнопка About** показывает справку о программе.

Исходные таблицы БД

Реальные (table)

SONG - Песни		
ID	ID песни (PRIMARY KEY)	INT NOT NULL
ARTIST	Исполнитель	VARCHAR (40) NOT NULL
SONG	Песня	VARCHAR (40) NOT NULL
ALBUM	Альбом	VARCHAR (40)
YEAR	Год	INT (4)
GENRE	Жанр	ENUM
SOURCE	Путь к файлу	TEXT
TIME	Длина песни	TIME
LYRIC	Текст Песни	TEXT
COMMENT	Коммент (тэг-коммент)	VARCHAR (40)
FULL_DATA	Статус Корректности	VARCHAR (1)

ALBUM – Альбомы		
ID	ID альбома (PRIMARY KEY)	INT NOT NULL
ARTIST	Исполнитель	VARCHAR (40) NOT NULL
ALBUM	Альбом	VARCHAR (40) NOT NULL
YEAR	Год	INT (4)
GENRE	Жанр	ENUM
SOURCE	Путь к альбому	TEXT
COMMENT	Описание альбома	TEXT
COVER	Обложка (Картинка)	LONGBLOB

ARTIST - Исполнители		
ID	ID исполнителя (PRIMARY KEY)	INT NOT NULL
ARTIST	Исполнитель	VARCHAR (40) NOT NULL
GENRE	Жанр	ENUM
SOURCE	Путь к папке исполнителя	TEXT
COMMENT	Описание исполнителя	TEXT

PICTURE	Изображение (картинка)	LONGBLOB

Искусственные (view)

user_cut_songs - Песни		
ARTIST	Исполнитель	VARCHAR (40) NOT NULL
SONG	Песня	VARCHAR (40) NOT NULL
ALBUM	Альбом	VARCHAR (40)
YEAR	Год	INT (4)
TIME	Длина песни	TIME
	·	

user_cut_albums(_full) - Альбомы		
ARTIST	Исполнитель	VARCHAR (40) NOT NULL
ALBUM	Альбом	VARCHAR (40) NOT NULL
YEAR	Год	INT (4)
TOTAL	Песен в альбоме	CALCULATED
COVER	Обложка (Картинка)	LONGBLOB

user_cut_artist(_full) - Исполнители		
ARTIST	Исполнитель	VARCHAR (40) NOT NULL
SONGS	Песен у исполнителя	CALCULATED
ALBUMS	Альбомов у исполнителя	CALCULATED
PICTURE	Изображение (картинка)	LONGBLOB

Функции и процедуры

В SiriusNav входят следующие процедуры и функции:

- **insert_song_album ()** добавляет те альбомы, которые не содержатся в таблице album, но имеются в таблице song [npouedypa]
- insert_song_artist () добавляет тех исполнителей, которые не содержатся в таблице artist, но имеются в таблице song [процедура]
- **insert_album_artist** () добавляет тех исполнителей, которые не содержатся в таблице artist, но имеются в таблице album [npouedypa]
- delete_empty_album () удаляет альбомы, в которых нет песен [процедура]
- delete empty artist () удаляет исполнителей, у которых нет песен [процедура]
- **delete artist dupe ()** удаляет дубликаты исполнителей в таблице artist [процедура]
- delete_album_dupe (flag INT) удаляет дубликаты альбомов в таблице album [процедура]. Параметр flag:
 - 0 с учётом года
 - o 1 без учета года
- **delete_song_dupe** (**flag INT**) удаляет дубликаты песен в таблице song [*npoцедура*]. Параметр **flag**:
 - о **0** с учетом исполнителя, альбома, года
 - о **1** с учетом исполнителя, альбома, года, длительности
 - о **2** с учетом исполнителя, альбома, года, длительности, пути
 - o **3** с учетом исполнителя, альбома, года, пути
 - 4 с учетом исполнителя, альбома
 - 5 с учетом исполнителя
- return_time (param VARCHAR (40), flag INT, flag2 INT) возвращает длительность звучания для артиста, альбома или жанра [функция].
 - о Параметр **param -** отвечает за шаблон, по которому будет произведена выборка жанр, исполнитель или альбом
 - о Параметр **flag** критерий выборки
 - **0** длительность всех песен в БД <u>SiriusNav</u>
 - 1 длительность по альбому
 - 2 длительность по исполнителю
 - 3 длительность по жанру
 - о Параметр **flag2** параметры возврата
 - **0** возвращает в формате xx days xx hours xx minutes xx seconds
 - **■** 1 возвращает в формате *dd:hh:mm:ss*
 - 1 возвращает в формате xx hours xx minutes xx seconds
- html_print (param VARCHAR (40), number INT) возвращает текст HTML страницы в формате LONGTEXT [процедура]
 - о Параметр **param** отвечает за шаблон, по которому будет произведена выборка исполнителя
 - о Параметр **number** присваивает уникальный ID исполнителю, для которого производится выборка и создание HTML

Программное обеспечение, использовавшееся в разработке

Программа разработана в среде Borland Builder 6.01 Enterprise на языке ANSI C++\Borland C\Objective C, в разрабоке использвались также:

- Borland Image Editor <u>http://www.borland.com/</u>
- MySQL 5.0.45 http://www.mysql.ru/
- Adobe Photoshop 10 CS3
- Adobe ImageReady 10 CS3 http://www.adobe.com
- OpenOffice.org 2.3 Writer http://www.openoffice.org
- Opera 9.23 http://www.opera.com
- Microsoft Notepad
- Microsoft Calculator <u>http://www.microsoft.com</u>
- Miranda 0.7.0.PR#10 http://www.miranda-im.org
- Mozilla ThunderBird 2.0.0.6 http://www.mozilla.org
- NullSoft WinAmp 5.35 http://www.winamp.com/
- Jim-Player 2.2, 3.0 Beta http://illusionist7.narod.ru/jim.html
- AnyDAC Library http://www.da-soft.com/content/view/45/145/
- MyBuilder http://www.crlab.com/mybuilder

Прочее

Данная программа разрабатывалась с целью каталогизировать более чем 20.000 MP3 файлов. В ручном режиме на это могли уйти

недели, месяцы, а то и годы. В задачу было поставлено: автоматическая обработка информации, максимально снижающая уровень ошибок, а так же последующее создание на основе полученных данных кода HTML для публикации в сети. Нечто подобное в простейшем варианте было реализовано в Jim 2.х, но и в случае плеера создавать реальный каталог было весьма неудобно из-за перещелкивания альбомов, а так же из-за невозможности создания логичных и однозначных связей в БД. Для этих целей и была разработана SiriusNav.

В данной версии БД нет жесткой привязки к таблицам songartist-album. Эту задачу выполняют встроенные функции процедуры, а не внешние и уникальные ключи. Это связано с тем, что из-за некорректности тэгов или их различий (ведь порой один и тот же файл, имеющий разный тэг будет восприниматься как различные песни, альбомы, исполнители) нельзя привязывать таблицы из-за их каскадных свойств. Так, на "двойника" исполнителя могут быть записаны песни, в то время как исполнитель уже числится в бд, но под другим именем. Это не рационально и неудобно – вначале переименовывать песню, потом отслеживать связи. Именно поэтому автоматизация с помощью функций более достойный выход из задачи.

Программа основана на новейших разработках в мире IT, отвечает самым современным требованиям, но и запрашивает достаточно высокие аппаратные ресурсы.

В заключение хочу выразить благодарность кафедре Прикладной Математики МИРЭА за информационную поддержку и консультации, в особенности Скляру А.. Отдельное спасибо Гуренко В. aka Hobboth'y за идеи и поддержку:)