Оглавление:

1. Цели и задачи системы
2. Интерфейс пользователя
3. Интерфейс администратора
4. Логика работы
5. Технические особенности
6. Задел на будущее или планирование развития

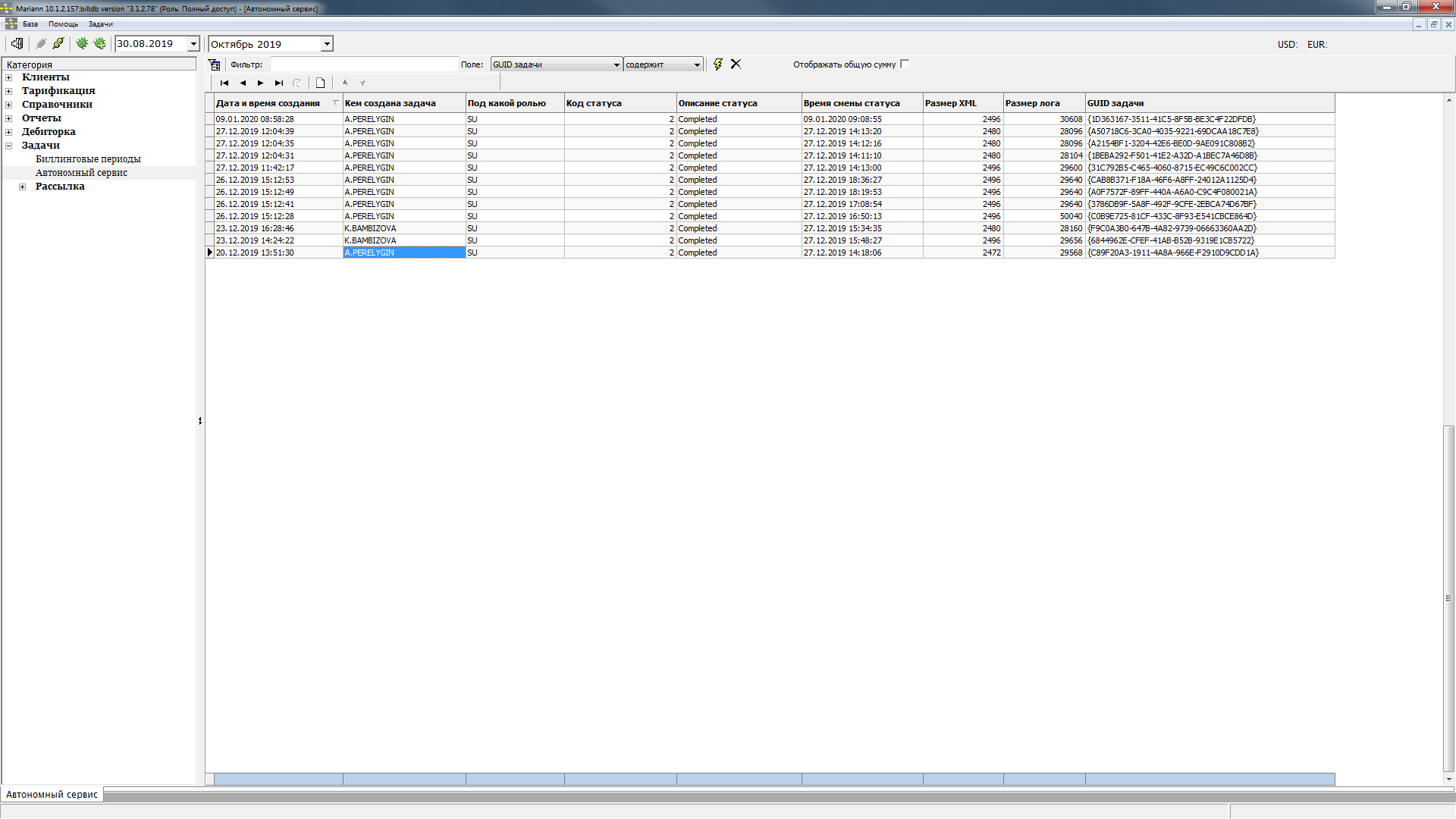
Термины:

* **Сервис**: Это автономная программа, запускающая на выполняющая библиотеки “dll”.
* **библиотека dll** : Созданная библиотека Delphi2007 выполняющая определённую операцию.
* **Задача**: Группа операций (**шагов**), собранная в единый блок для выполнения в одном процессе.
* **Процесс:** Промежуток времени и ресурс, выделенный для выполнения **задачи**.
* Этап: Законченная функция или процедура внутри каждой **библиотека dll**
* **Шаг:** Исполнение одной **библиотеки dll** в рамках **задачи** с одним **набором параметров**
* **Набор параметров:** Набор параметров (переменных) определенный для каждого **шага.**
* **Интерфейс**: Графическая оболочка взаимодействия с пользователем.
* **Статус**: Итоговая оценка текущего статуса выполнения **задачи** и имеющая код и описание.
* **GUID**: Уникальный идентификатор **задачи** связывающий **шаги** задачи для отображения в собственных **интерфейсах**, и определяющий уникальный ресурс для **задачи**.

1. Цели и задачи системы:

**Сервис** необходим для выполнения длительных и объемных задач на стороне автономного сервера в защищённом режиме с эксклюзивными правами доступа к данным и ресурсам и с полным отображением логов работы каждого этапа процесса. Важно помнить, что каждая задача может содержать более одного этапа выполнения и переход к следующему этапу возможен лишь если выполнен предыдущий со статусом “**удача**”. Для каждого этапа выполнения существует свой набор передаваемых параметров как входных, так и выходных, что обеспечивает универсальность выполнения. Статистика выполнения вложенных этапов добавляется к общему логу задачи, что позволяет провести анализ не только последовательность исполнения, но и понять сколько раз, когда и сколько времени заняло выполнение каждого этапа и даже каждой вложенной функции внутри библиотеки (подключение к бд, копирование, выполнение SQL). Сам **сервис** выполняет каждую **задачу** в отдельном потоке, что позволяет одновременную обработку нескольких **задач**. Последовательность выполнения **задач** происходит по принципу первая пришла первая взята в работу, а время завершения зависит только от количества и уровня сложности вложенных **шагов**. Каждая задача может содержать 1 и более **шагов** выполнения. Каждый шаг содержит свой **набор параметров**, и ссылку на свою **библиотеку dll** исполнителя и только одну. Разработка своей **библиотеки dll** позволяет обеспечить высокую степень надёжности выполнения, а общий принцип построения объектов позволяет минимизировать объём исполняемого кода и соответственно сделать сам процесс доработки более защищенным от простых ошибок. Статистику и результат выполнения **задач** пользователь видит в общем интерфейсе, что позволяет легче обнаружить ошибки, допущенные им и быстрее реагировать при их появлении. Сам **сервис** имеет алгоритм расчета времени допустимой активности работы, что позволяет ему останавливать и запускать процессы в соответствии с интервалами активности и остановки необходимой для обеспечения выполнения несовместимых процессов всей системы. Каждая задача имеет свой уникальный GUID, что позволяет идентифицировать результаты выполнения задачи в разных таблицах и разных СУБД а так же иметь собственное пространство для промежуточных данных. Важно помнить, что запуск и результат не требует ожидания пользователя, создавшего задачу, а, следовательно, это экономит ему время для работы с другими бизнес-процессами. Итого, цель данной системы, это выполнения большого количества сложных задач на стороне автономного сервера, с исключением внешних мешающих факторов, доводя риск ошибок к нулю.

1. Интерфейс пользователя:

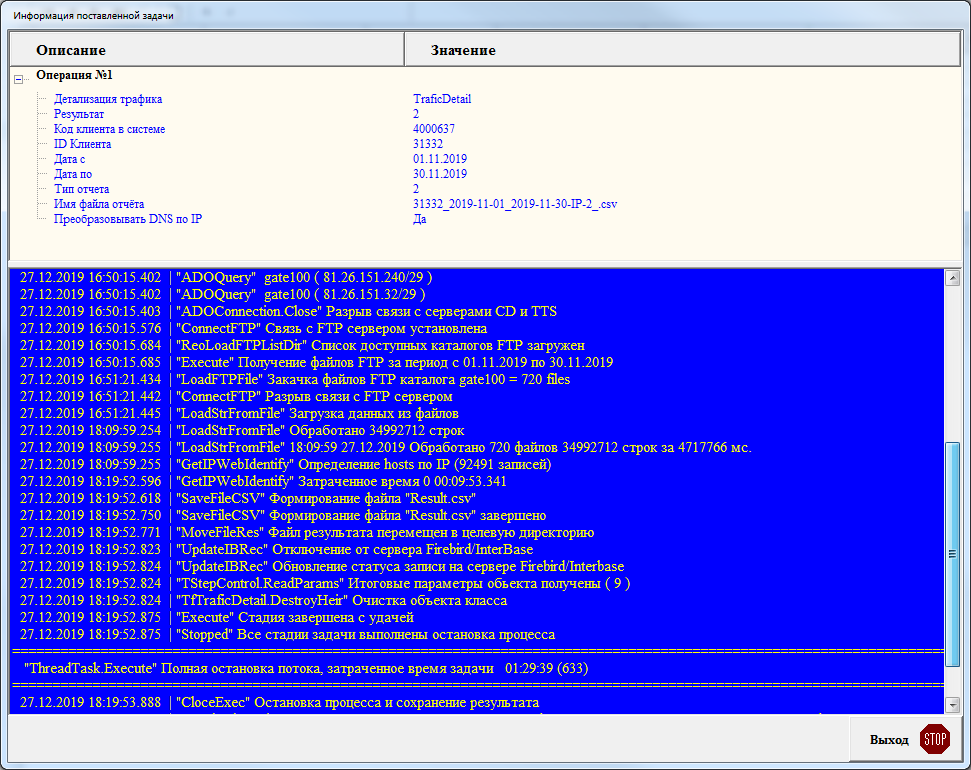


**Интерфейс** пользователя обеспечен в программе «BillClient.exe». В основном меню «Задачи» в подменю «Автономный сервис». Каждый пользователь видит список задач, созданный только им самим. Пользователи с ролью «Полный доступ» видят все **задачи** независимо от имени входа. В данном меню в столбцах «Код статуса» и «Описание статуса» можно понять в каком статусе находиться сейчас задача.

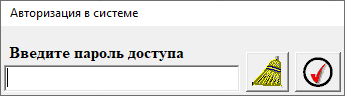
1. **Код статуса** = -1 **Описание статуса** =”Error” Означает что выполнение одного из этапов завершено с ошибкой и требует отреагировать на ситуацию как можно скорее.
2. **Код статуса** = 0 **Описание статуса** = “Service” Означает что задача находиться в режиме ожидания взятия в работу.
3. **Код статуса** = 1 **Описание статуса** = “Execute” Говорит о том что **задача** находиться в обработке и стоит подождать результата выполнения
4. **Код статуса** = 2 **Описание статуса** = “Completed” Данный статус повествует о корректном завершении выполнения **задачи** с результатом «Успех».
5. **Код статуса** = 0 **Описание статуса** =”Stopped Thread” Означает что **задача** находиться в режиме ожидания взятия в работу но при этом уже ранее была взята в работу но была остановлена в результате остановки работы автономного сервиса.

**Все отличные показатели от перечисленных требуют уведомления разработчика с передачей «GUID задачи».**

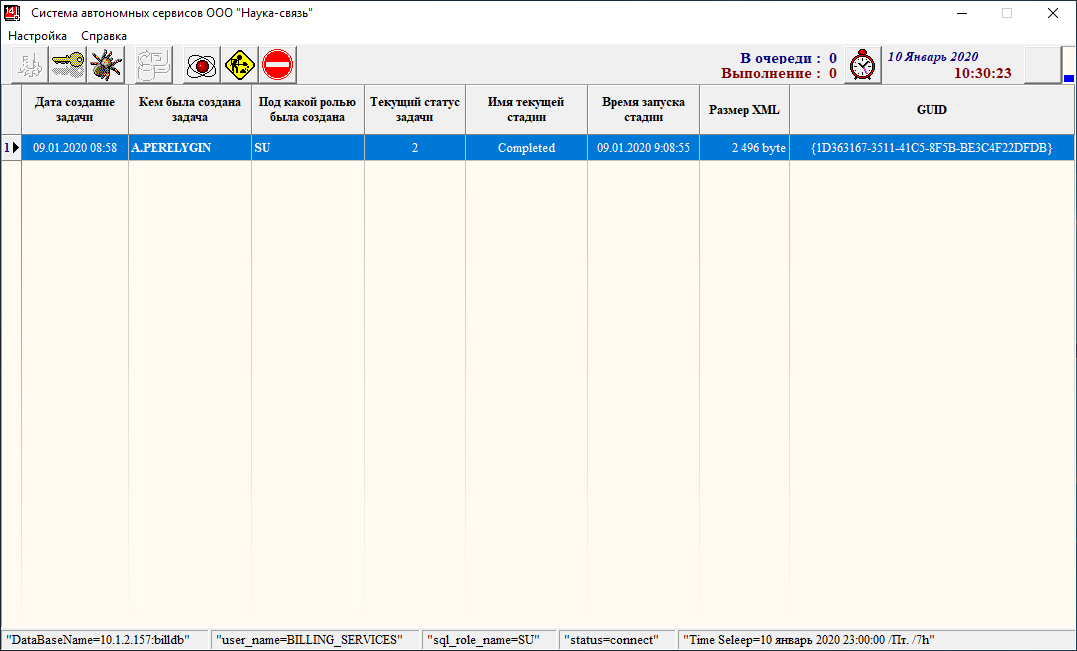
При двойном нажатии на строку задачи выходит окно.



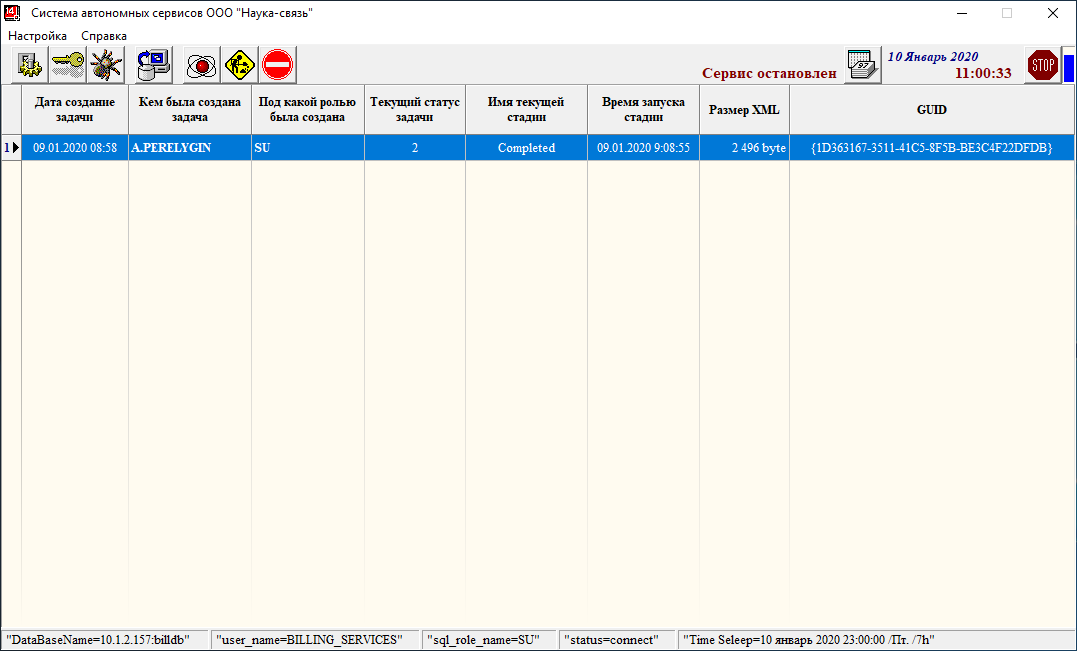
Данное окно содержит список **шагов** «Операция №1», **список параметров** со значением этих параметров, и отображения статистики выполнения **задачи** с временными рамками и промежуточными данными. Создание **задачи** производиться в соответствии с дополнениями в модуле и явного интерфейса не имеют.

1. Интерфейс администратора: 

Авторизация в **сервисе**, позволяет отрыть общий интерфейс для управления выгрузкой или остановкой системы либо запуском отдельных **задач**. При запусках **сервиса** программа прячется в трей, откуда вызвать ее можно двойным нажатием на пиктограмму в трее и вводом пароля системного администратора **сервиса**. Важно помнить, что независимо находиться ли программа в скрытом режиме или в режиме администрирования все процессы, связанные с выполнением фоновых задач, запускаются автоматически. Отображение того что программа не зависла занимается прогресс бар в правом верхнем углу, и изменении времени на часах в программе. Отображение что система находиться в режиме поиска и постановки **задач** в очередь символизирует кнопка с пиктограммой будильника и отображения значения «В очереди».

**Основной интерфейс программы:** 

Основное отображение списка доступно в режиме администратора и несет скорее информативный характер. Список отображает **задачи**, у которых дата создания больше или ровно текущей за минусом количества дней, указанных в настройках. В верхнем правом углу находиться кнопка выгрузки программы и доступна лишь после остановки обработки очереди **задач** (Производиться это путем нажатия кнопки с пиктограммой будильник при этом пиктограмма на кнопке будет изменена, повторный запуск очереди путем повторного нажатия этой же кнопки). Остановка обработки очереди задач символизируется надписью «Сервис остановлен» и кнопка с пиктограммой будильника меняет на пиктограмму календаря.

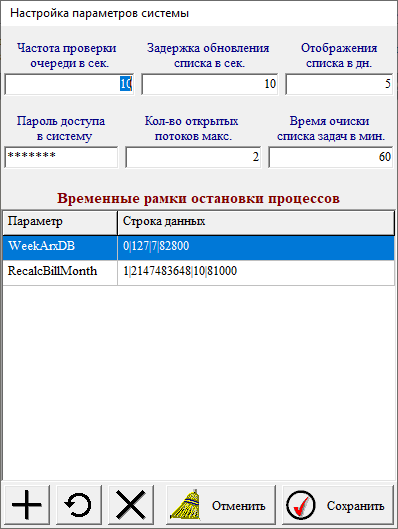


В режиме обработки очереди **задач** будут недоступны кнопки настройки системы и настройки связи с БД. Связано это с тем что большая часть настроек напрямую влияет на выполнения **задач** и обеспечения результата очереди, так что доступность будет только после остановки соответствующего **сервиса**.

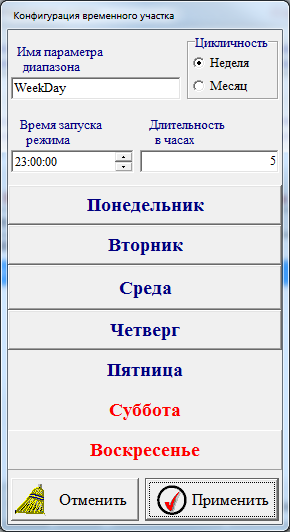
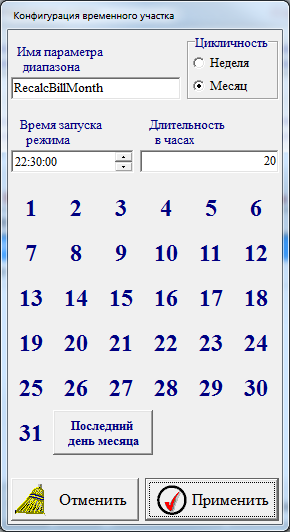
Блокировка/Разблокировка интерфейса администратора производиться путем нажатия кнопки с пиктограммой ключа, при этом сама система остается в том же состоянии и производиться только блокировка отображения списка и активация кнопок управления **интерфейса**.

Кнопка с пиктограммой паука прячет программу в трей, при этом программа будет переведена в режим блокировки **интерфейса** администратора в любом случае.

Как уже говорилось ранее, кнопка настройки доступна после остановки режима обработки очереди задач и имеет пиктограмму шестерёнки. Данная настройка нужна для ввода основных параметров **системы**.

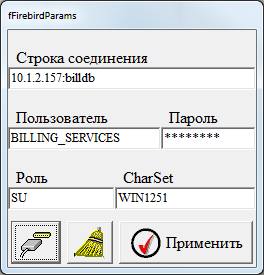


1. Параметр «Частота проверки очереди в сек.» позволяет скоординировать с какой частотой режим обработки очереди задач будет проверять добавление новых задач в случае если есть свободные потоки для их выполнения.
2. Параметр «Задержка обновления сек.» позволяет указать с какой частотой будет обновляться отображение списка задач в режиме администратора.
3. Параметр «Отображения списка в дн.» Указывает сколько последних дней стоит отобразить в списке **задач** в режиме администратора.
4. Параметр «Кол-во открытых потоков макс.» Соответствует максимальному количеству **задач**, обрабатываемых одновременно в режиме обработки очереди.
5. Параметр «Время очистки списка **задач** в мин.» Данный параметр указывает за какое время до автоматической/плановой приостановки сервиса **задачи** больше не будут браться в работу, и будет выполняться только ранее запущенные.
6. Параметр «Временные рамки остановки процессов» Это список временных рамок, при которых планируется приостановить сервис и закрыть все выполняемые **задачи** независимо от стадии завершения (При этом **задача** переводиться в **статус** 0) а связь с основной БД прекращается. Данный список имеет свой собственный механизм управления параметрами и вызывается одной из кнопок с низу с пиктограммами плюс и круговая стрелка.



Каждый параметр имеет условие цикличности либо проверку дня недели, либо проверку дня месяца, соответствующий день будет выбран путём клика на соответствующей дню пиктограмме. Важно помнить, что должен быть выбран хотя бы один день. Последний день месяца нужен для выбора соответствующего дня так как в каждом месяце разное кол-во дней. Так же следует помнить, что время запуска режима не может иметь все нули, как и длительность в часах, а также параметр должен иметь уникальное имя. Ожидаемое время и длительность остановки система выбирает сама по методу наиболее близкое и отображает в статус баре (Правая нижняя запись) в котором указана дата и время срабатывания, день недели и на какой промежуток времени сервис будет остановлен. Если в параметрах нет ни одного диапазона, система напишет соответствующую ошибку и будет остановлена совсем (Time Sleep=Error).

1. Настройка доступа к СУБД производиться путем нажатия кнопки с пиктограммой компьютер с цилиндром и стрелкой и вызывает соответствующее окно настройки



После ввода соответствующих параметров необходимо нажать кнопку с пиктограммой соединить для тестового соединения, только после этого кнопка сохранения с пиктограммой красная галочка в кругу станет доступна и введённые параметры смогут быть сохранены.

1. Следующие 3 кнопки призваны управлять задачами в списке, который отображается только в режиме администратора. Кнопка со значком атома отображает меню описанное ранее «Информация поставленной задачи» и несет только информационный характер. Кнопка человека с лопатой в желтом ромбе становиться активной только для записей со статусом отличного от 1, и позволяет перевести статус задачи из любого в 0, для последующего выполнения, при этом логирование будет дополнено. Кнопка с минусом в красном кругу позволяет удалить задачу совсем если её статус отлично от 1.
2. При нажатии крестика в правом верхнем углу (Стандартная кнопка выгрузки приложения), программа сворачивается в трей.
3. Логика работы:

Стороннее приложение (в данном случае программа «BillClient.exe».) формирует файл xml с параметрами для **сервиса** и создает запись в таблице «RDB$SERVICE» в поле «DATA\_XML» кладет данный xml, а в поле «GUID» генерирует новый GUID, это и будет **задача**. В свою очередь если нет препятствующих факторов серверная часть системы проверяет данную таблицу на записи со статусом 0. Далее если количество открытых потоков меньше указанного максимального параметра **сервис** меняет статус на 1, при этом он загружает данные xml и log в создаваемый объект «TTask». Объект «TTask» при создании считывает xml и забирает параметры, внутри себя создает объекты **шагов** и заполняет их **наборами параметров** в соответствии с данными полученными из xml. Следующий этап, это создание и запуск выделенного потока для **задачи**. Внутри потока последовательно запускаются **шаги,** пока результат не вернёт ошибку или пока не останется ни одного **шага**. После чего в базу возвращается результат в качестве обновленного xml и файла лога. Итогом становиться закрытие потока и уничтожения объекта «TTask» с очисткой выделенных ресурсов.

1. Технические особенности:
2. **Библиотека dll** соответствует имени задачи без расширение, так что имя dll не должно быть на русском и быть максимально коротким и максимально понятным с расширением «.dll».
3. Перед подключением **библиотека dll** к **сервису** следует провести ее тестирование модулем «TestService.exe».
4. **Библиотека dll** должна содержать external function для основного объекта класса «TStepControl» в модуле uStepControl.pas (пример реализации библиотеки dllZero.dproj).
5. Необходимо перекрыть метод очистки объекта, при этом поля этого объекта должны быть зачищены включая массивы, record, string и. т. д. в противном случае это приведёт к утечкам памяти **сервиса**.
6. В параметрах должны быть указаны только уникальные данные для **задачи**, общие параметры (параметры доступа к db, путь к каталогу выгрузки и.т.д) необходимо хранить в файлах «.ini» во избежание ненужной пересылки данных через xml.
7. Базовый класс «TStepControl» обеспечивает наполнение файла лога, обращение к function «SetLog» должно быть во всех ключевых моментах обработки.
8. Передаваемые и получаемые параметры содержаться в базовом классе, не надо создавать что-то дополнительное во избежание путаницы.
9. Обновление библиотек производиться путем полной выгрузки программы из памяти, а не остановки сервиса (Важно помнить что все **Библиотека dll** Windows блокирует сама).
10. Дополнительные формы библиотеки надо стараться делать как можно более универсальными, что бы их можно было использовать как «ComUtils» (Путь к каталогу, настройка FTP, настройка ADO и. т. д).
11. Сторонние библиотеки (файлы с расширением “dll”), не относящиеся к задачам должны быть сведены к минимуму и лежать в директории с исполняемым файлом сервиса.
12. Все изменения в сервисе и исполняемых модулях должны фиксироваться в svn и в данном документе.
13. Запрет на обращение Ole ()
14. Задел на будущее или планирование развития:
15. Необходимо сделать логирование самим **сервисом** (запуск, остановка, выгрузка взятие **задач** и.т.д).
16. Разделить **задачи** на приоритеты отделить пользовательские и системные, создать зависимости исполняемых механизмов между **задачами**.
17. Создать механизм регистрации новых модулей **библиотек dll** и добавления и редактирования **списков параметров**.
18. Необходимо продумать именование **задач** для понимания интерфейса его создавшего.
19. Желательно добавить механизм для циклически выполняемых **задач**, с определением глубины и частоты циклов.
20. Отладить механизм корректной выгрузки сервиса при перегрузке машины и доработать службу ре-стартующую сервис при перегрузке или зависания **сервиса**.
21. Крайне необходимо доработать механизм сжатия xml и log данных при работе с db.
22. Доработать выпадающую справочную систему.
23. Доработать библиотеки для формирования отчетов «FastReport», отсылки email, выполнения скриптов SQL хотя бы Firebird в режиме ожидания результата с динамическими параметрами.
24. Передача объекта «TStepControl» между Step в рамках задачи если обращение имеет один и тот же исполняемый модуль.
25. По принципу локальных параметров этапов нужно создать глобальные параметры только массивы и глобальные для задачи.