# **Руководство пользователя и администратора ИС Hasher**

Оглавление

[**Руководство пользователя и администратора ИС Hasher** 1](#_Toc123045786)

[Описание программы 2](#_Toc123045787)

[Общие сведения 2](#_Toc123045788)

[Алгоритмы вычисления контрольной суммы 4](#_Toc123045789)

[Интерфейс программы 4](#_Toc123045790)

[Удаление из базы 7](#_Toc123045791)

[Фиксация изменений в файлах 7](#_Toc123045792)

[Поиск дубликатов 10](#_Toc123045793)

[Архивация системных журналов 11](#_Toc123045794)

[Передача данных 11](#_Toc123045795)

[Поиск новых файлов 13](#_Toc123045796)

[О программе 14](#_Toc123045797)

[**Руководство администратора информационной системы** 15](#_Toc123045798)

[Настройки программы 15](#_Toc123045799)

[Обслуживание базы данных (Microsoft Access). 15](#_Toc123045800)

## Описание программы

Предназначена для контрольного суммирования, проведения инспекционного контроля, контроля целостности и отслеживания изменений версий программных продуктов.

Целями данной программы является:

* обеспечение выполнения пункта 4.3.1 «Требований к обеспечению безопасности автоматизированных систем управления технологическими процессами, сформированные во исполнение норм федерального закона от 26 июля 2017 г. № 187-ФЗ «О безопасности критической информационной инфраструктуры российской федерации»
* повышение надежности объектов критической информационной инфраструктуры.

*Контро́льная су́мма — некоторое значение, рассчитанное по набору данных путём применения определённого алгоритма и используемое для проверки целостности данных при их передаче или хранении. Также контрольные суммы могут использоваться для быстрого сравнения двух наборов данных на неэквивалентность: с большой вероятностью различные наборы данных будут иметь неравные контрольные суммы. Это может быть использовано, например, для обнаружения компьютерных вирусов. Несмотря на своё название, контрольная сумма не обязательно вычисляется путём суммирования.*

## Общие сведения

Информационная система разработана с применением Microsoft .NET Framework 3.5 в среде Visual Studio Express 2012 for Windows Desktop. Данные средства разработки выбраны по причине наилучшей совместимости с системным программным обеспечением используемыми в АСУТП, а также из-за возможности использования без материальных затрат. Главное преимущество компилируемых языков — это скорость исполнения. Компилируемые языки являются самыми эффективными, поскольку они исполняются как машинный код и позволяют использовать аппаратное обеспечение системы. Однако это вводит дополнительные ограничение на написание кода и делает его платформо-зависимым.

В данном случае система подразумевала использование в среде Microsoft Windows, соответственно разумным выбором служит среда разработки, предоставляемая компанией производителем операционной системы. Использование российских сред разработки (CLion) подразумевает материальные затраты на приобретение данного программного продукта, использование других российских сред разработки подразумевает написание кода на не компилируемых языках программирования, установку дополнительного программного обеспечения (виртуальные машины JAVA и т.д.) в системы АСУТП, что не допустимо, также это потребует расхода дополнительных ресурсов системы.

В информационной системе используются алгоритмы:

* **MD5** (англ. Message Digest 5) — 128-битный алгоритм хеширования, разработанный профессором Рональдом Л. Ривестом из Массачусетского технологического института (Massachusetts Institute of Technology, MIT) в 1991 году. Предназначен для создания «отпечатков» или дайджестов сообщения произвольной длины и последующей проверки их подлинности.
* **Циклический избыточный код** (англ. Cyclic redundancy check, CRC) — алгоритм нахождения контрольной суммы, предназначенный для проверки целостности данных. CRC является практическим приложением помехоустойчивого кодирования, основанным на определённых математических свойствах циклического кода.
* **Secure Hash Algorithm 1** — алгоритм криптографического хеширования. Описан в RFC 3174. Для входного сообщения произвольной длины (максимум {\displaystyle 2^{64}-1}2^{64}-1 бит, что примерно равно 2 эксабайта) алгоритм генерирует 160-битное (20 байт) хеш-значение, называемое также дайджестом сообщения, которое обычно отображается как шестнадцатеричное число длиной в 40 цифр. Используется во многих криптографических приложениях и протоколах.
* **SHA-2** (англ. Secure Hash Algorithm Version 2 — безопасный алгоритм хеширования, версия 2) — семейство криптографических алгоритмов — однонаправленных хеш-функций, включающее в себя алгоритмы SHA-224, SHA-256, SHA-384, SHA-512, SHA-512/256 и SHA-512/224. Хеш-функции предназначены для создания «отпечатков» или «дайджестов» для сообщений произвольной длины.
* **Adler-32** - хеш-функция, разработанная Марком Адлером. Является модификацией контрольной суммы Флетчера. Вычисляет значение контрольной суммы в соответствии с RFC 1950 для массива байт или его фрагмента. Данный алгоритм расчёта контрольной суммы отличается от CRC32 производительностью. Adler-32 используется в библиотеке Zlib. Rolling checksum версия функции используется в утилите rsync.
* **RIPEMD-160** (от англ. RACE Integrity Primitives Evaluation Message Digest) — криптографическая хеш-функция, разработанная в Католическом университете Лувена Хансом Доббертином (англ. Hans Dobbertin), Антоном Босселарсом (Antoon Bosselaers) и Бартом Пренелем. Для произвольного входного сообщения функция генерирует 160-разрядное хеш-значение, называемое сводкой сообщения. RIPEMD-160 является улучшенной версией RIPEMD, которая, в свою очередь, использовала принципы MD4 и по производительности сравнима с более распространённой SHA-1. Также существуют 128-, 256- и 320-битные версии этого алгоритма, которые, соответственно, называются RIPEMD-128, RIPEMD-256 и RIPEMD-320. 128-битная версия представляет собой лишь замену оригинальной RIPEMD, которая также была 128-битной и в которой были найдены уязвимости. 256- и 320-битные версии отличаются удвоенной длиной сводки, что уменьшает вероятность коллизий, но при этом функции не являются более криптостойкими. RIPEMD-160 была разработана в открытом академическом сообществе, в отличие от SHA-1 и SHA-2, которые были созданы NSA. С другой стороны, RIPEMD-160 на практике применяется несколько реже, чем SHA-1. Использование RIPEMD-160 не ограничено какими-либо патентами.
* ***HMAC*** *(иногда расшифровывается как англ. hash-based message authentication code, код аутентификации (проверки подлинности) сообщений, использующий хеш-функции, или как англ. keyed-hash message authentication code, код аутентификации сообщений, использующий хеш-функции с ключом) — в информатике (криптографии), один из механизмов проверки целостности информации, позволяющий гарантировать то, что данные, передаваемые или хранящиеся в ненадёжной среде, не были изменены посторонними лицами (см. человек посередине). Механизм HMAC использует имитовставку (MAC), описан в RFC 2104, в стандартах организаций ANSI, IETF, ISO и NIST. MAC — стандарт, описывающий способ обмена данными и способ проверки целостности передаваемых данных с использованием секретного ключа. Два клиента, использующие MAC, как правило, используют общий секретный ключ. HMAC — надстройка над MAC; механизм обмена данными с использованием секретного ключа (как в MAC) и хеш-функций. В названии может уточняться используемая хеш-функция: HMAC-MD5, HMAC-SHA1, HMAC-RIPEMD160 и т. п.*

Программа написана с использованием dotnet 3.5, что позволяет ей работать в системах MS Windows XP и выше.

Программа состоит из 5 файлов:

1. HASHER.exe – файл программы
2. adodb.dll – библиотека для связи с базой данных
3. db.mdb – база данных программы
4. Interop.JRO.dll – библиотека для сжатия базы данных
5. Ionic.Zip.dll – библиотека для работы с архивами

Программа не требует установки, работает из любого каталога. Для запуска необходимо выполнить исполняемый файл - HASHER.exe.

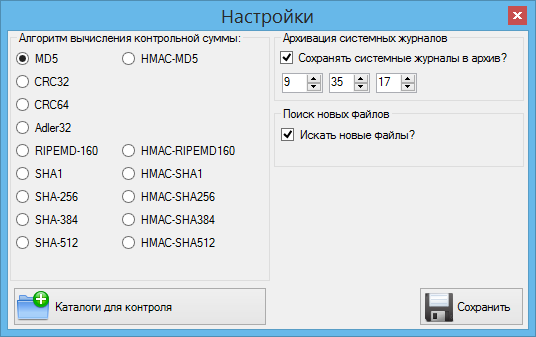
## Алгоритмы вычисления контрольной суммы

В программе используются девять алгоритмов вычисления контрольной суммы фалов:

1. MD5
2. CRC32
3. CRC64
4. SHA256
5. SHA512
6. SHA1
7. SHA384
8. Adler-32
9. RIPEMD-160
10. HMAC-MD5
11. HMAC-SHA1
12. HMAC-SHA256
13. HMAC-SHA384
14. HMAC-SHA512
15. HMAC-RIPEMD-160

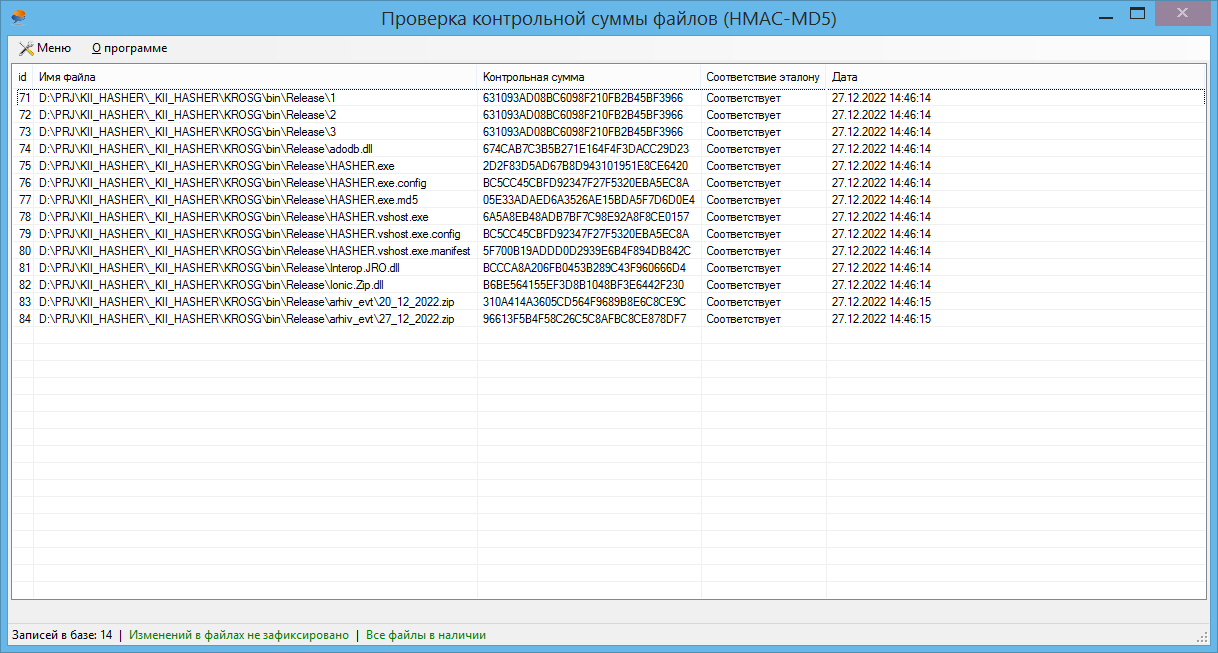
*Для HMAC используется 32 битный постоянный ключ.*

По умолчанию используется алгоритм MD5, изменить текущий алгоритм можно в «Меню\Настройки»

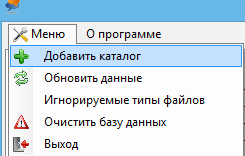


*Не меняйте алгоритм вычисления контрольной суммы при заполненной базе, это приведет к порче сохраненных данных.*

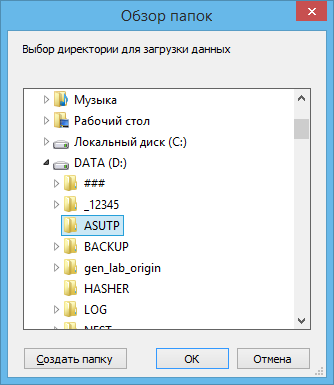
## Интерфейс программы



При первом запуске нужно вызвать пункт меню «Добавить каталог»



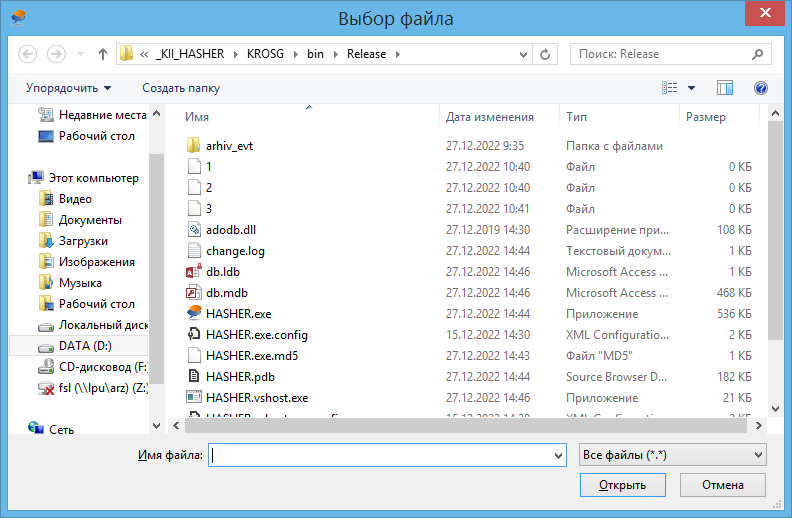
В появившемся диалоговом окне выбрать каталог для добавления



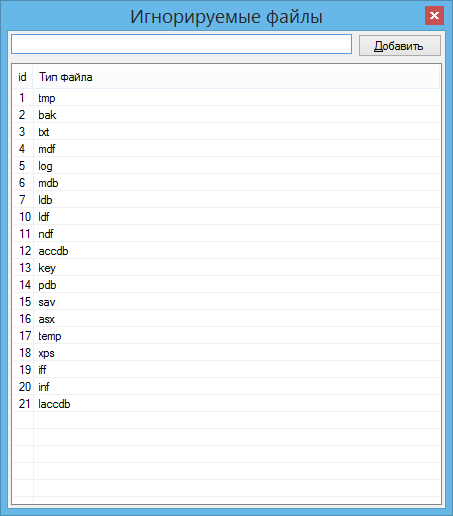
После нажатия кнопки «Ок» произойдет сканирование каталога и просчет контрольных сумм файлов, содержащихся в выбранном каталоге с записью в базу данных.

Можно добавить несколько каталогов.

*Так же есть возможность добавления отдельного файла: меню\добавить файл.*

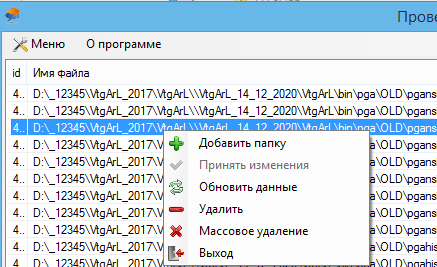


В программе имеется возможность фильтрации по типу файлов, «Меню\Игнорируемые типы файлов»



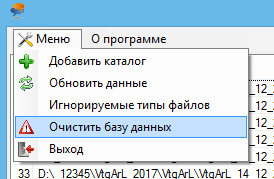
## Удаление из базы

1. Удалить отдельный файл



Выделяем в списке файл, нажимаем правую клавишу мыши «Удалить»

1. Очистка базы данных



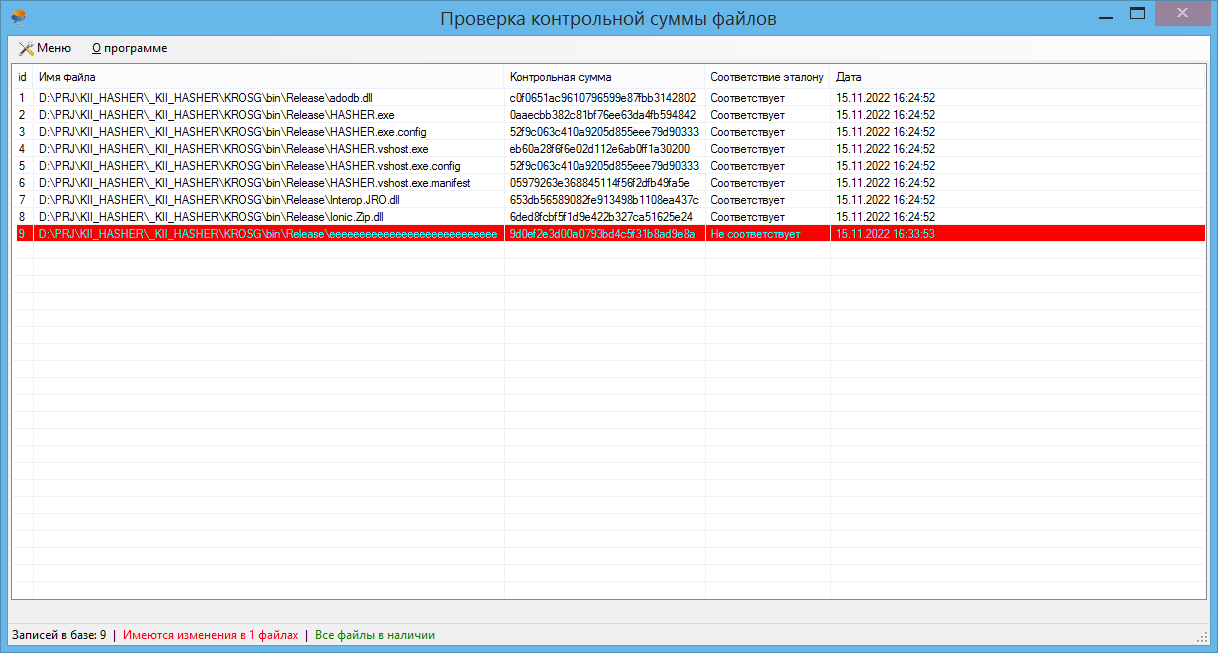
После нажатия произойдет полная очистка базы данных (т.е. все данные будут удалены)

## Фиксация изменений в файлах

Повторное сканирование файлов происходит:

1. При повторном запуске программы
2. Вызывается вручную в пункте меню «Обновить данные»

При выявленных изменениях в листе измененные данные помечаются красным цветом:

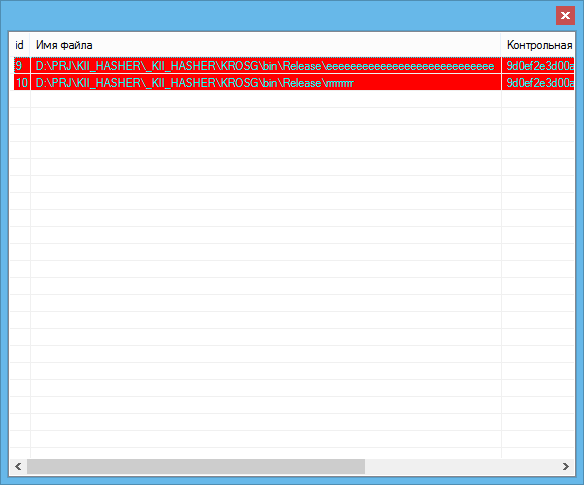


Отсутствующие (удаленные) файлы помечаются серым цветом.

Просмотреть измененные файлы можно нажав на надпись в строке состояния «»



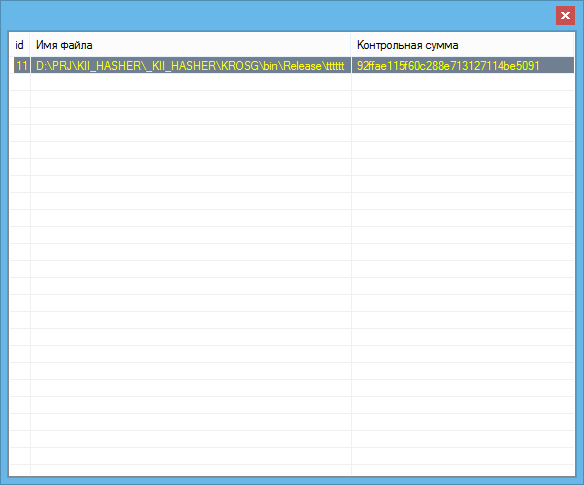
После нажатия появиться форма:



Просмотреть удаленные файлы можно нажав на надпись в строке состояния «Удаленные файлы»



После нажатия появиться форма:

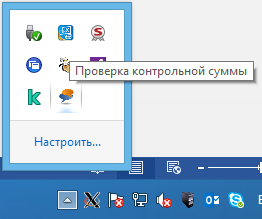


В строке состояния программы указано количество измененных и удаленных фалов.

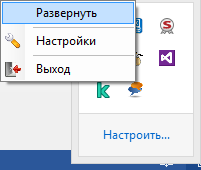


1. Так же в программе предусмотрена проверка данных по таймеру, проверка осуществляется каждые три часа (00, 03, 06, 09, 12, 15, 18, 21) с записью обнаруженных изменений в файл change.log, находящийся в каталоге программы.

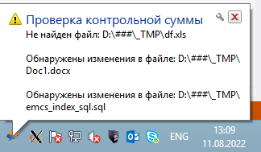
Для этого необходимо не закрывать программу, а свернуть, она будет находится в системном лотке:



Развернуть программу можно щелкнув по иконке, либо вызвав контекстное меню:

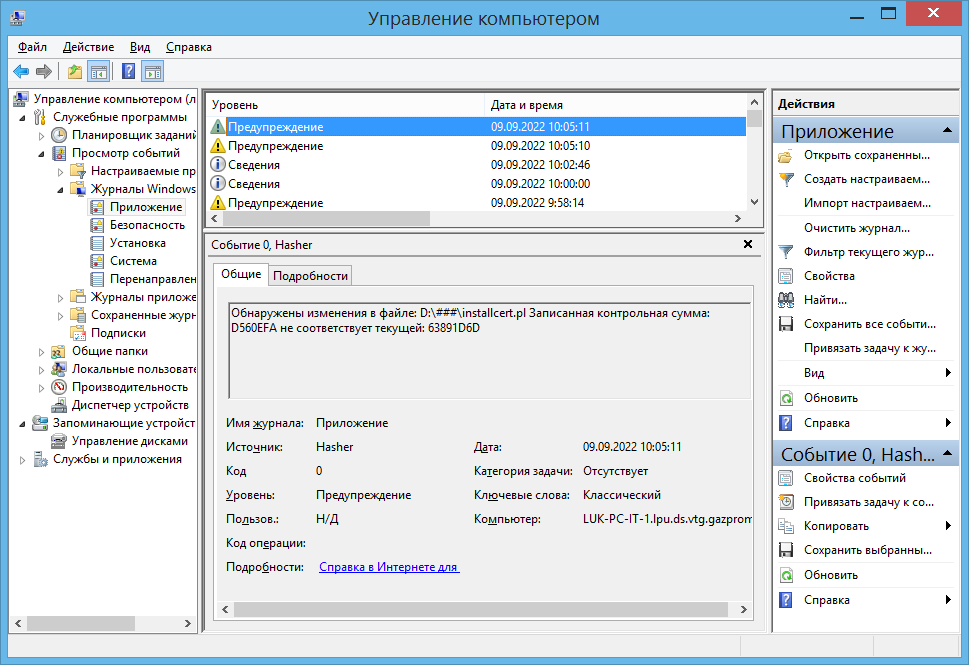


При обнаружении изменений в системном лотке появиться сообщение:

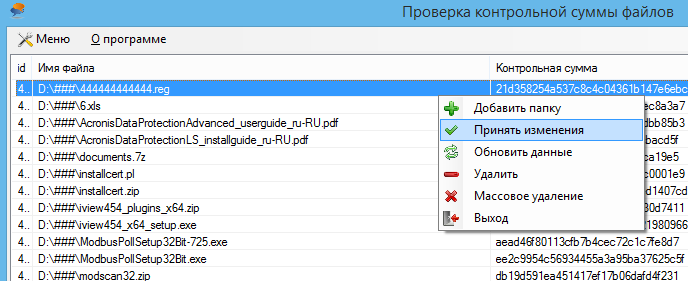


и будет произведена запись в журнал, если запись в журнале уже существует, то повторно запись делаться не будет.

Так же осуществляется запись в системный журнал:

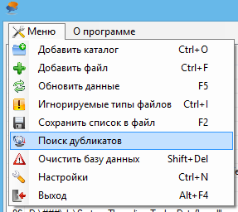


Если данные изменения произведены корректно (т.е. Вами), к примеру, это конфигурационный файл, либо файл мнемосхемы, измененные согласно регламента, то можно принять данные изменения вызвав контекстное меню:

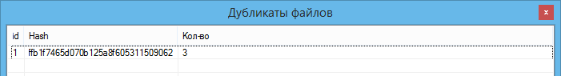


После нажатия кнопки изменения будут записаны в базу, с изменением даты и времени.

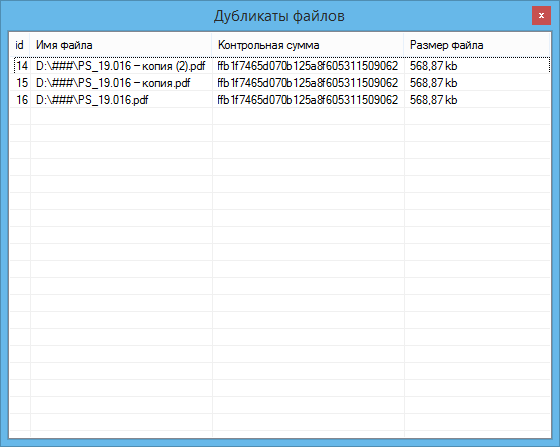
## Поиск дубликатов



Производится поиск дубликатов файлов в базе, сравнивается контрольная сумма файлов. Если обнаруживаются дубликаты появляется окно со списком дубликатов:

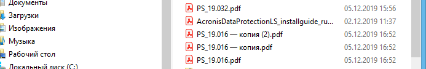


Двойной клик на строке откроет окно со списком файлов:



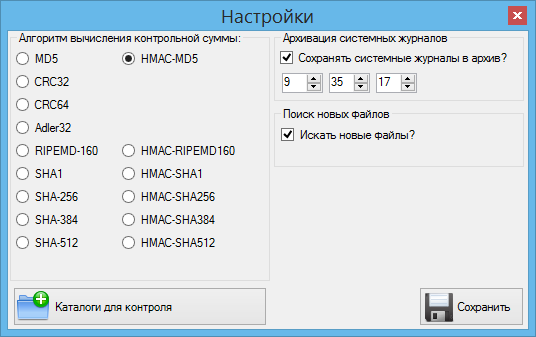
В списке виден путь до файла, контрольная сумма и размер файла.

Двойной клик на строке откроет каталог, в котором находится файл.



## Архивация системных журналов

В меню\Настройки установить галочку «Сохранять системные журналы в архив» и установить время создания архива журналов:

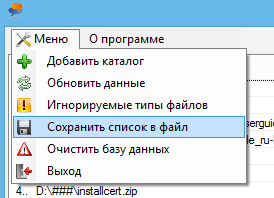


В каталоге программы будет создан каталог «arhiv\_evt» в который будут добавляться ежесуточные архивы системных журналов. Формат имени файла – дата\_месяц\_год.zip

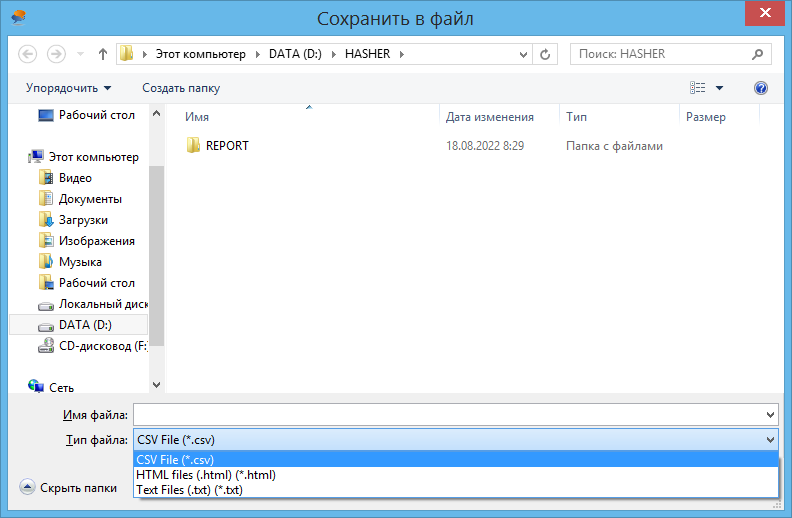
## Передача данных

В программе имеется возможность сохранения данных в формат CSV, HTML и TXT:

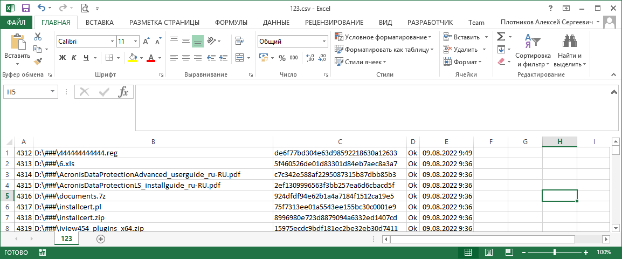
«Меню\Сохранить список в файл»

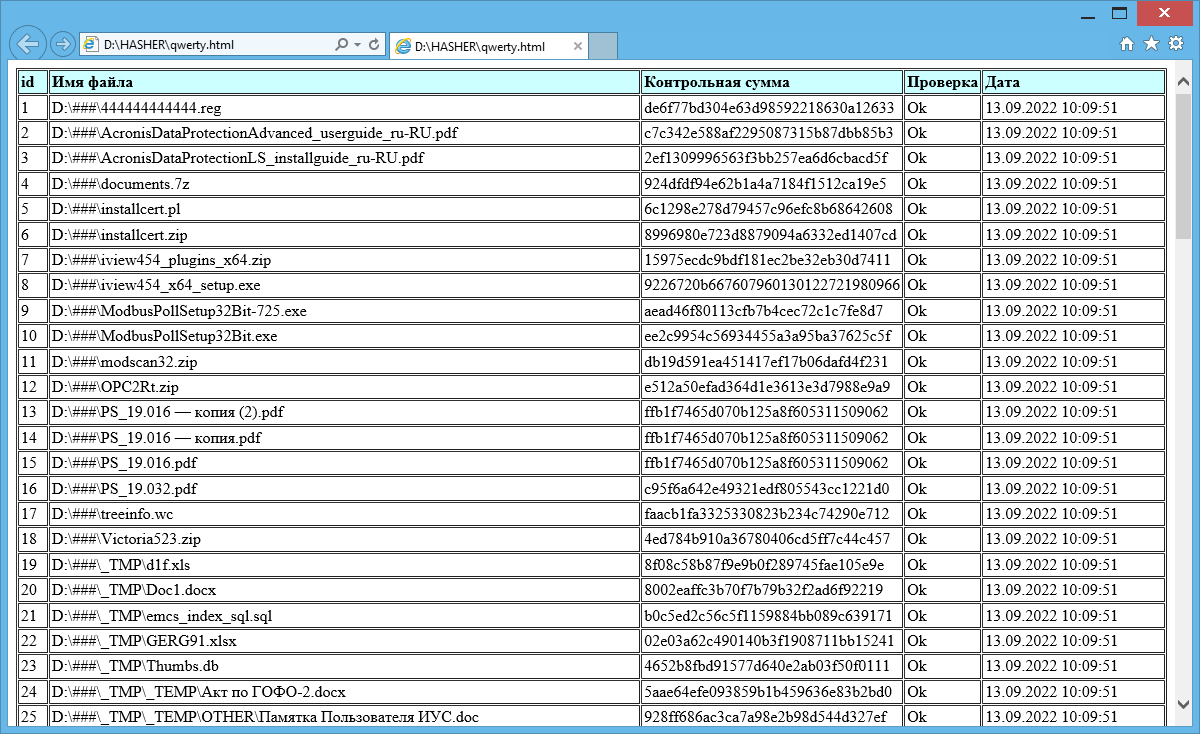


При нажатии на данный пункт меню появится диалоговое окно с предложением сохранить файл, выбираем путь сохранения и имя фала, нажимаем «Сохранить»



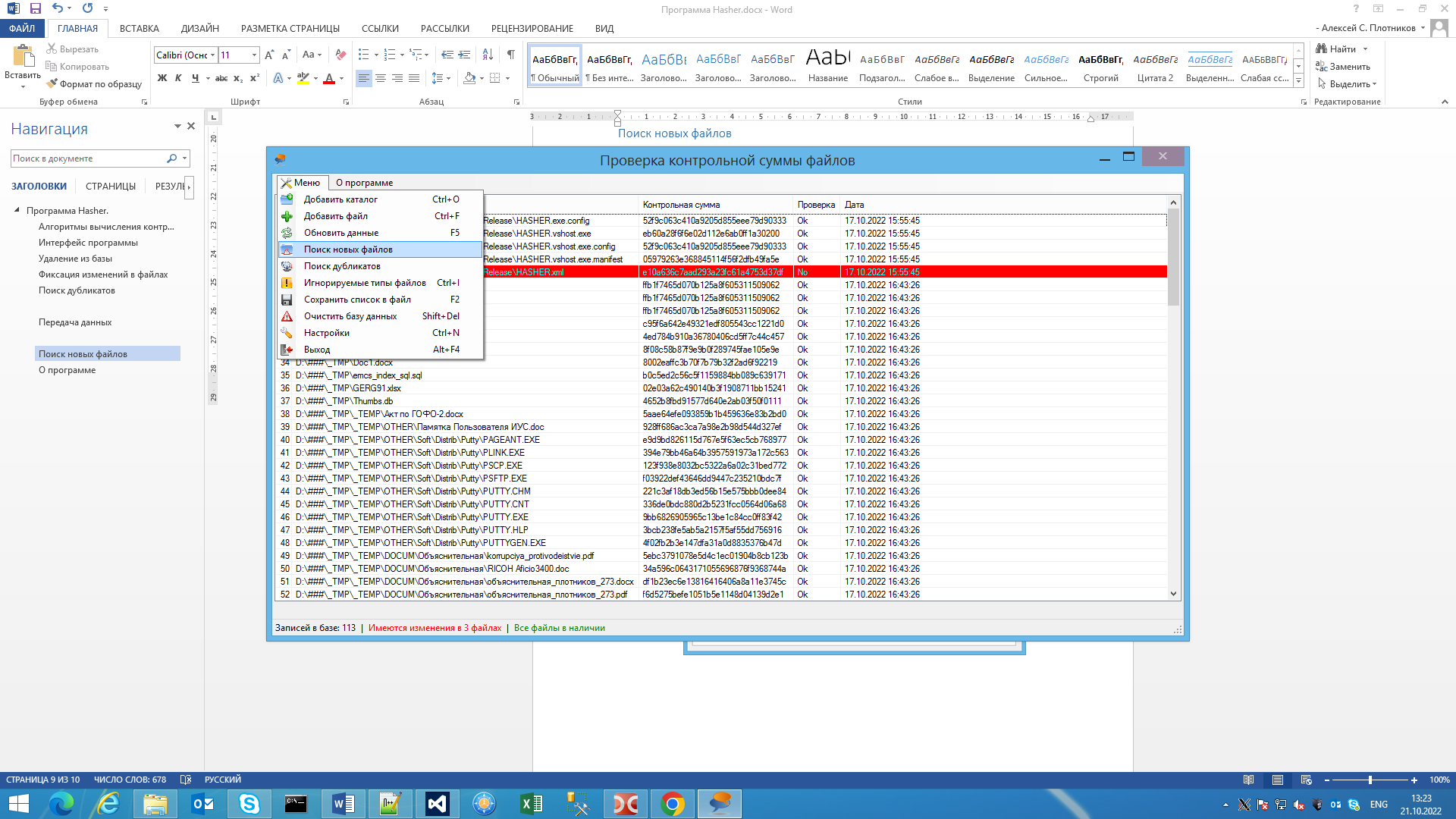
После выполнения функции сохранения в указанном месте появиться файл.





Данный файл можно использовать для хранения и последующего контроля.

## Поиск новых файлов



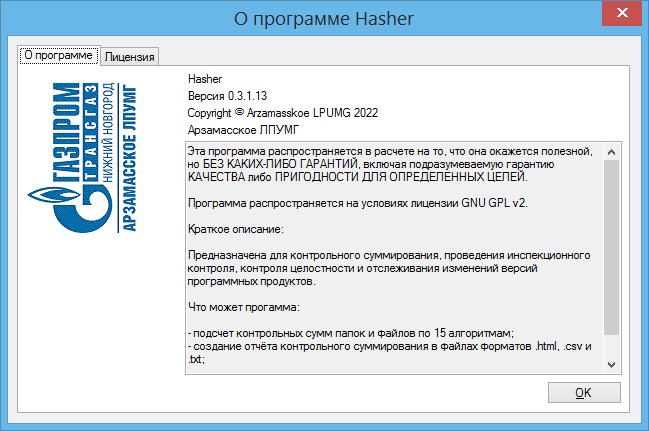
При выборе данного пункта, произойдет повторное сканирование каталогов, файлы, отсутствующие в базе, будут добавлены.

Так же поиск новых файлов осуществляется по таймеру (в случае указания в настройках программы, иначе простая проверка) каждые три часа (00, 03, 06, 09, 12, 15, 18, 21) с записью обнаруженных новых файлов change.log и системный журнал.

## О программе

Эта программа распространяется в расчете на то, что она окажется полезной, но БЕЗ КАКИХ-ЛИБО ГАРАНТИЙ, включая подразумеваемую гарантию КАЧЕСТВА либо ПРИГОДНОСТИ ДЛЯ ОПРЕДЕЛЕННЫХ ЦЕЛЕЙ.

Программа распространяется на условиях лицензии GNU GPL v2.



# **Руководство администратора информационной системы**

Установка программного обеспечения производится путем копирования файлов необходимых для работы в каталог.

## Настройки программы

- выбрать алгоритм расчета контрольной суммы;

- указать каталоги для контроля;

- добавить необходимые для контроля файлы.

Если требуется автоматический запуск программы, необходимо создать ярлык приложения и поместить его в автозапуск операционной системы Microsoft Windows.

## Обслуживание базы данных (Microsoft Access).

Базы данных могут быстро увеличиваться в размере по мере использования, что иногда снижает их быстродействие. Кроме того, они могут повреждаться. Для предотвращения и исправления таких проблем при выгрузке приложения производится автоматическая проверка состояния базы данных, в случае обнаружения проблем проблемы в автоматическом режиме устраняются.