

AnyRAM, «ARD Trade» Production

«ARD Trade», Vitebsk, Belarus

Tel: +375-29-5968565

Website: <http://www.anyram.net>

E-Mail: client@anyram.net



AirWipe – программа для поточной записи бесконтактных карт Mifare и ISO-14443A

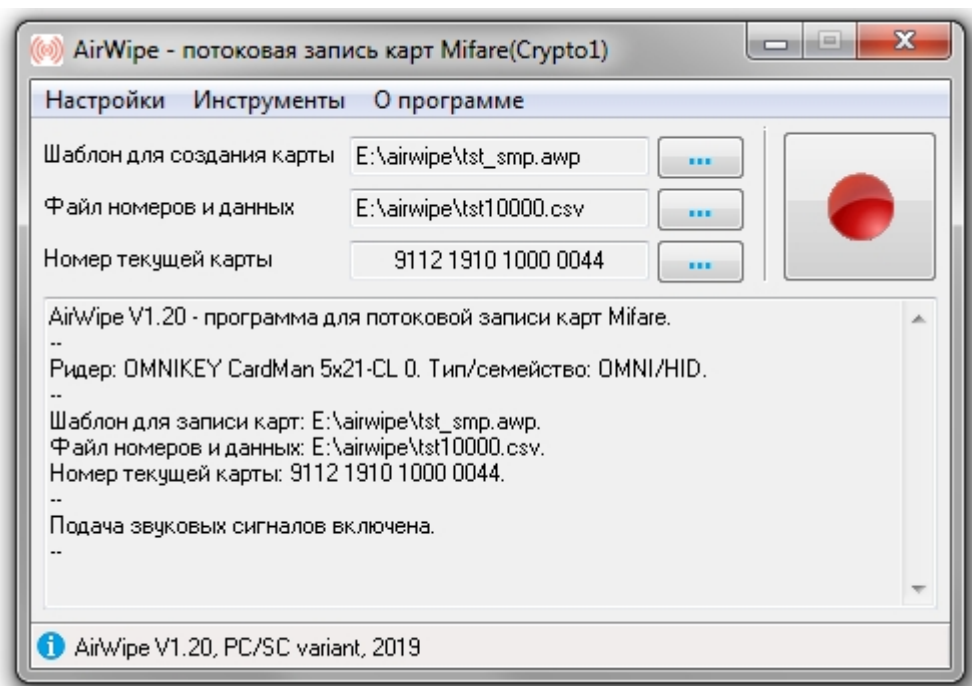
Программа **AirWipe** позволяет массово записывать бесконтактные карты с привязкой содержимого карты к напечатанной на карте информации без дорогостоящего оборудования. ПО позволяет записывать как статические данные, не изменяющиеся от карты к карте, так и динамические, включая уникальные ключи авторизации, изменяющиеся от карты к карте, вести протокол обработанных карт, вызывать внешние приложения для получения и записи секретных данных из внешних источников и многое другое.

Основная идея, заложенная в программу – запись данных в карты, уже прошедших полиграфию, по заранее подготовленному списку. Заранее готовится пачка отпечатанных карт с серийными номерами (или без них), карты в пачке укладываются по порядку, без пересортицы. В Excel подготавливается файл с серийными номерами карт (не обязательно) и данными, которые будут записаны на карту. Также необходимо подготовить скрипт, в котором указывается, куда и как в карте должны быть записаны данные из Excel-файла. Скрипт позволяет очень гибко настроить параметры записи.

Для записи пачки карт, оператор указывает программе Excel-файл с перечнем карт и данных, файл скрипта и номер карты из списка, с которой начнётся запись (не всегда же запись ведётся с начала списка), указать рабочий ридер, если их несколько. Далее, программа указывает номер карты, который надо приложить к ридеру; после записи – просит приложить следующую по списку карту, и т.д.. Если карты идут без пересортицы, то процесс записи выглядит как последовательное прикладывание к ридеру карт из пачки, одна за другой. Контроль – визуальное сравнение номера на экране и номера, напечатанного на карте.

Настройка программы

После запуска программы появляется главное окно диалога, в котором указываются значения основных настроек.



Для работы с программой надо указать **шаблон, файл номеров и данных, номер текущей карты и ридер**: только после этого активируется выбор записи. Выбор текущего номера активируется только после установки файла шаблона и файла номеров и данных.

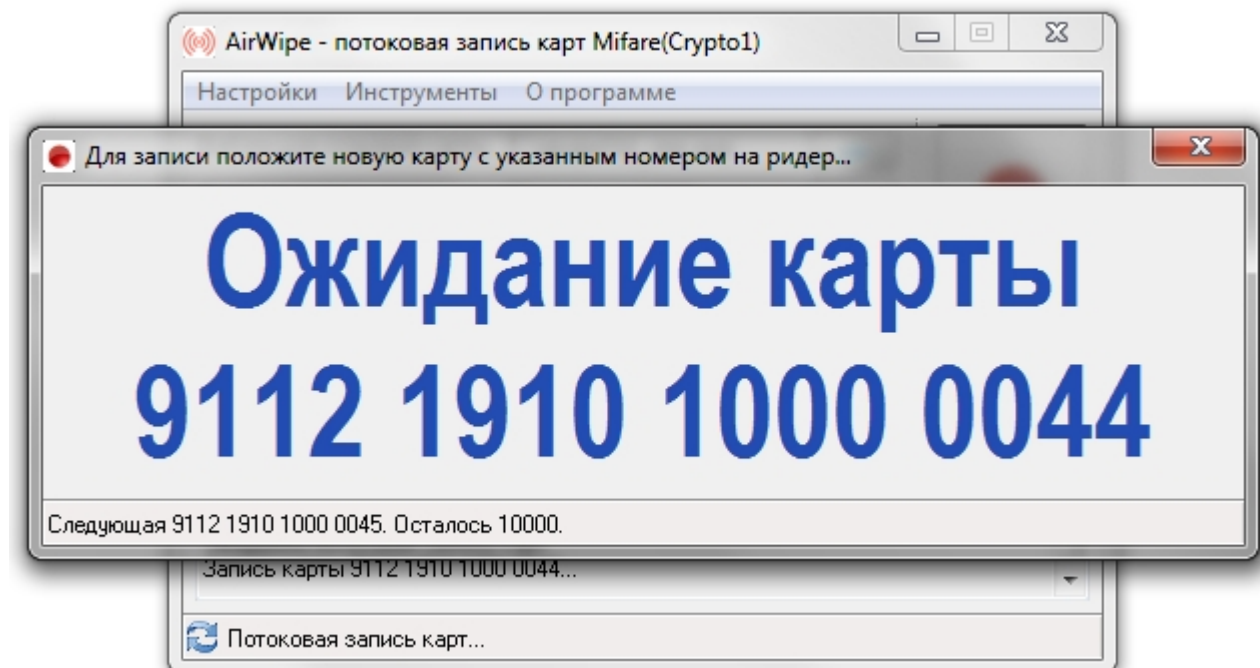
Шаблон, по которому будут записываться данные – это текстовые файлы с расширением **awp**. Устройство и синтаксис файла шаблона приводятся ниже; для начала работы можно воспользоваться демонстрационными файлами, которые входя в инсталляционный пакет.

Файл номеров и данных – это текстовый файл **csv**-формата, с точкой с запятой в качестве разделителя полей, экспортируемый из Microsoft Excel. Обязательно наличие первого поля, которое будет выведено на экран в качестве опознавателя текущей карты, в которую производится запись. Обычно это номер карты, но в этом поле можно написать что угодно, например, “**следующая карта**”, если все карты одинаковые.

Пример содержимого csv-файла номеров и данных:

карта;	номер карты;	ключ
9112 1910 1000 0044;	5B0C130A0A00002C;	6CE184BE2329
9112 1910 1000 0045;	5B0C130A0A00002D;	F1499052AED6
9112 1910 1000 0046;	5B0C130A0A00002E;	A6B3E9999999

Первое поле “9112 1910 1000 0044” будет выведено на экран при записи карты.



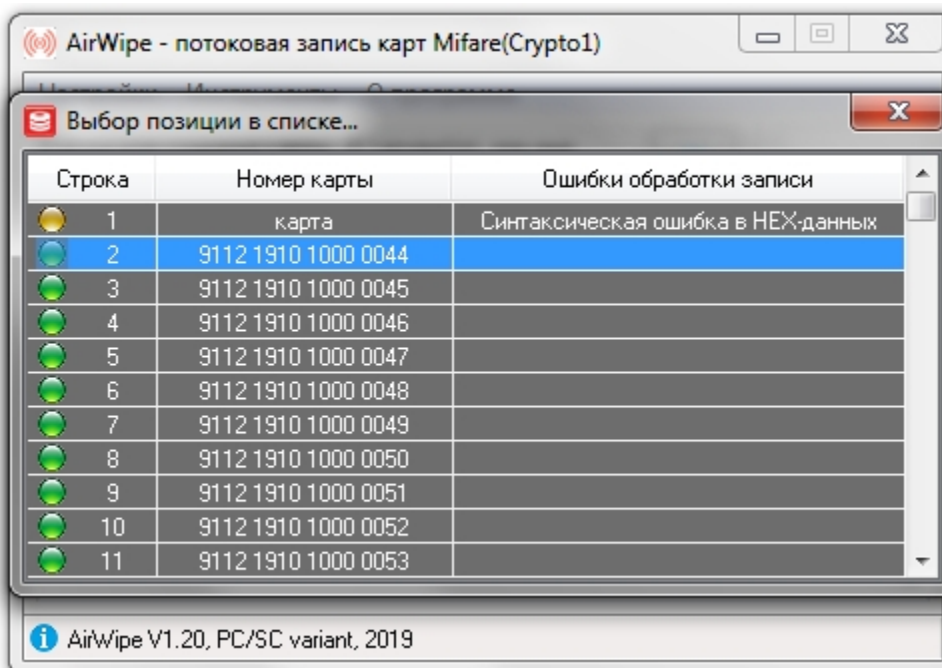
Номер карты “5B0C130A0A00002C” и ключ авторизации “6CE184BE2329” – это поля реальных данных, которые записываются в карту в соответствии с полем **FORMAT** в awp-файле шаблона.

FORMAT=N1:0:0 N1:3:10

Значение **N1:0:0** (номер карты) устанавливает, что первое поле данных – это данные в HEX-формате, которые пишутся в сектор 1 блок 0 со смещения 0. Значение **N1:3:10** (ключ авторизации) указывает, что HEX-данные ключа записываются в сектор 1 блок 3 со смещения 10 - здесь хранится В-ключ авторизации сектора 1. Подробное описание команды приводится ниже, с устройством карт Mifare можно ознакомиться в соответствующих технических описаниях.

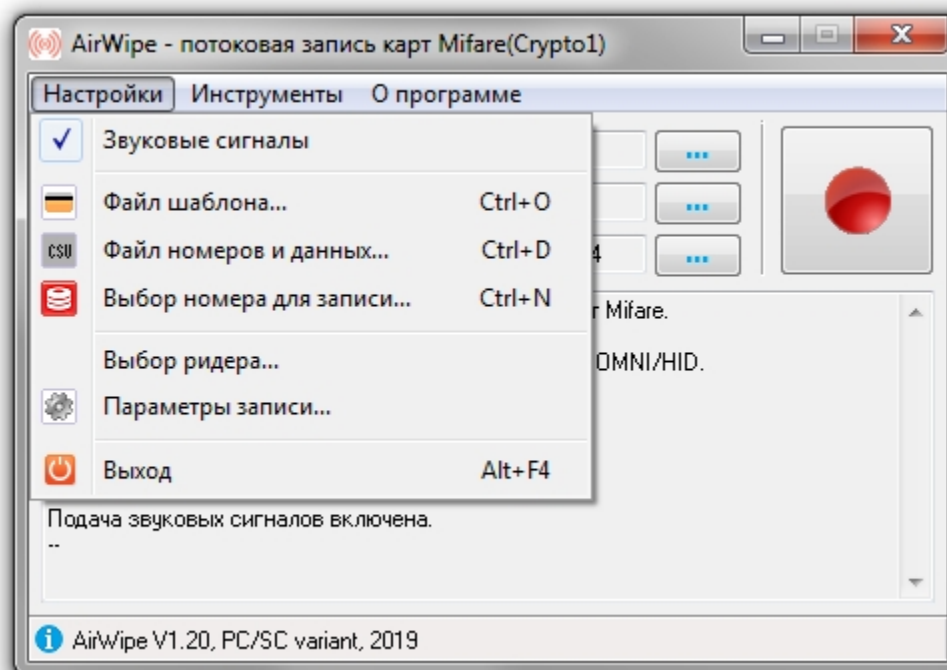
После выбора рабочих файлов, следует задать номер, с которого будут записываться карты. Для этого в память будет загружен файл номеров и данных, а затем каждая его запись будет проанализирована на соответствие шаблону (скрипту). Если в процессе анализа будет обнаружена ошибка соответствия шаблону, то в списке выбора записи “светофор” будет красного или жёлтого цвета (в зависимости от

“тяжести” ошибки), а в поле “Ошибки обработки записи” будет указано краткое описание проблемы.

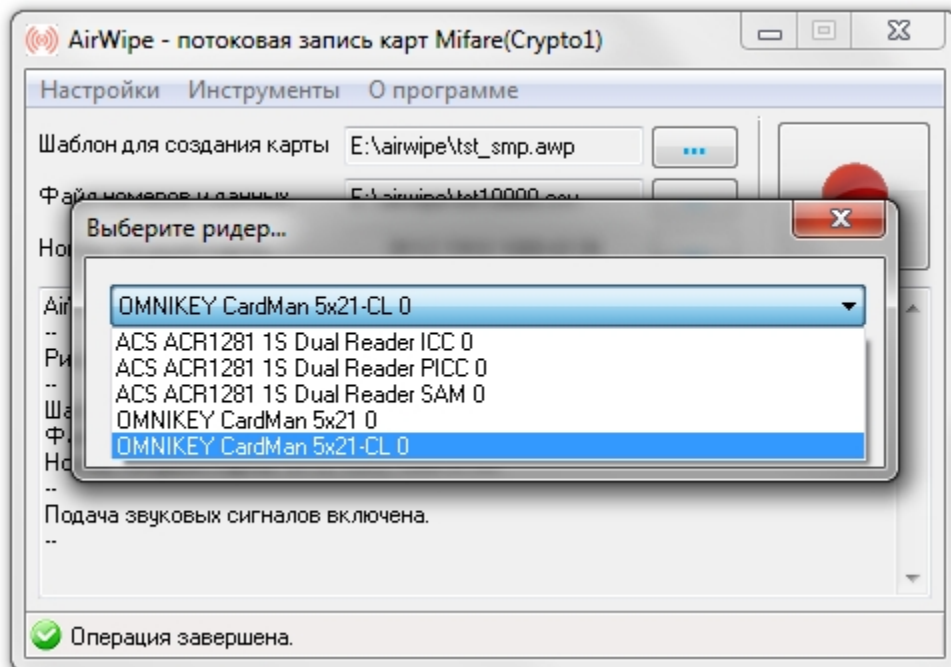


Обычно в первой строке csv-файла указываются названия полей, поэтому первичный анализ показывает, что номер карты – “карта”, и, поскольку поля описания не интерпретируются как HEX-данные, анализатор сигнализирует об ошибке в первой строке. Для устранения ошибки при записи карт можно указать, что запись начинается со второй строки (как это сделано на скриншоте), а можно в текстовом редакторе в файле номеров и данных удалить первую строку с описанием полей.

Выбор рабочего ридера производится через меню “Настройка – Выбор ридера”.



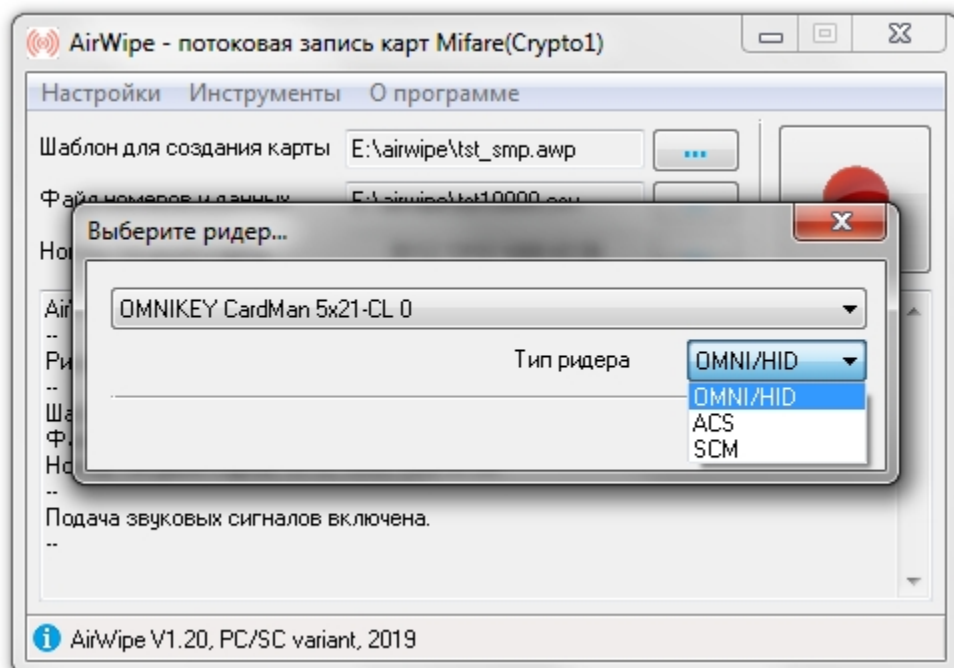
Некоторые версии программы не содержат этого пункта меню, т.к. предназначены для работы с узкоспециализированными поточными ридерами, например, ТК-D8, ICT3K и т.п.. В версии **AirWipe** для PC/SC ридеров, ридер выбирается из выпадающего списка всех ридеров, присутствующих в системе.



Обычно PC/SC ридеры универсальны и могут считывать как контактные, так и бесконтактные карты, поэтому, при подключении такого ридера, в системе будут зарегистрированы два или даже три устройства в перечне которых обязательно будет ридер бесконтактных карт. Выбирать надо именно его.

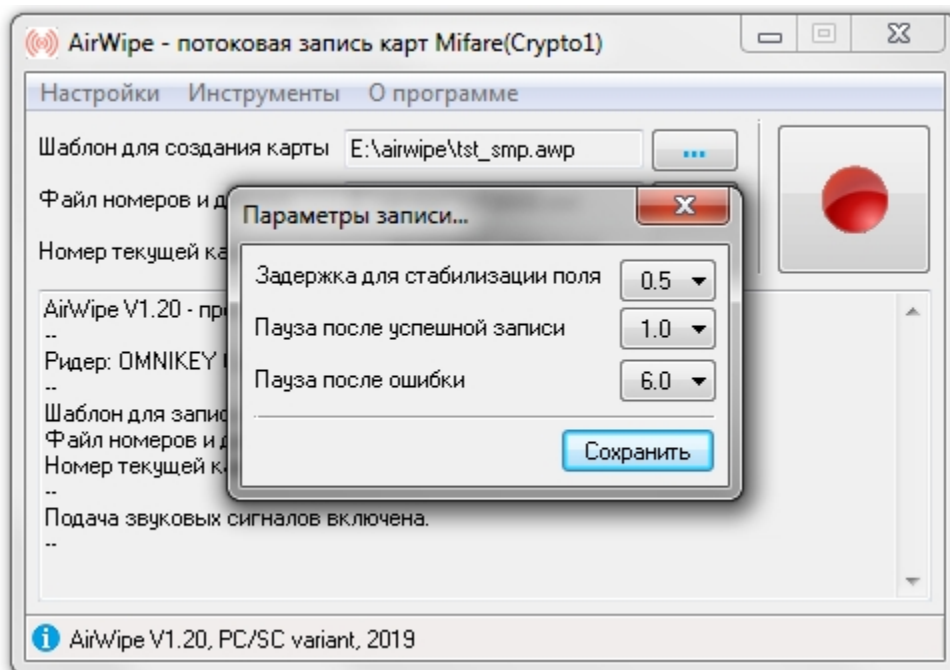
Как определить, какое устройство следует выбрать? Универсального ответа нет. Для ридеров **Omnikey Cardman** или **HID** ридер бесконтактных карт – это устройство с аббревиатурой **CL** в конце (contactless). Для ридеров **ACS**, **SCM** – это устройства с аббревиатурами **PICC** и **NFC** в строке названия устройства. Правильность выбора можно проверить войдя в консоль через меню "Инструменты - Консоль" и проверив связь с картой вручную (обычно нажав на "?", чтобы узнать тип карты).

После выбора рабочего ридера обязательно следует указать протокол обмена, иначе возможны ошибки при чтении или записи карт.



Обычно, программа сама правильно определяет тип ридера и устанавливает соответствующий протокол обмена по названию ридера. Однако, некоторые производители OEM-версий ридеров выдумывают названия, по которым невозможно определить семейство к которому принадлежит ридер – в этом случае следует указать протокол вручную.

При постоянной работе с ридером для ускорения процесса записи может потребоваться более тонкая настройка параметров записи. Это можно сделать через меню “Настройка – Параметры записи”.



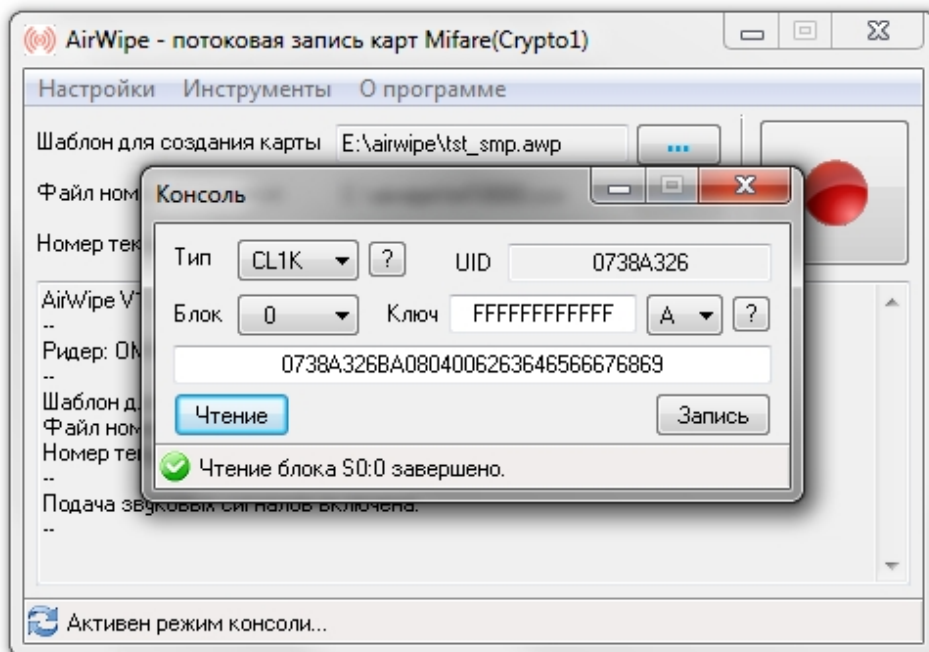
Задержка для стабилизации поля устанавливает время, через которое будет произведена запись данных после обнаружения карты. Современные бесконтактные карты – сложные, часто микропроцессорные, устройства, которые стараются сохранять низкое энергопотребление. Для пробуждения и ответа о присутствии, карте достаточно очень небольшой напряжённости электромагнитного поля, однако, для полноценной работы карте требуется мощное поле и хорошее потокоцепление с ридером. Поэтому при поднесении карты к ридеру, ридер обнаруживает карту на некотором расстоянии от рабочей зоны: расстояние достаточно для обнаружения карты, но недостаточно для записи данных. Задержка для стабилизации поля позволяет установить время после обнаружения, через которое карта точно будет в рабочей зоне и готова для записи.

Пауза после успешной записи – это время для вывода сообщения “Карта записана”. Для ускорения процесса записи это время можно сократить до нуля; в обучающих целях можно увеличить до 10с.

Пауза после ошибки – время для вывода надписи “Ошибка”. Его можно сократить до 1с, а можно увеличить до 10с. В любом случае, ошибка – это нестандартная ситуация, карту на которой произошла ошибка записи надо отложить, сделать отбраковку, а на это надо время.

После установки настроек можно приступать к записи карт, но если уверенности в правильности шаблона и данных нет, можно запустить тест записи (“Инструменты – Тест”), который проверит запись данных на карты без фактической записи – это очень удобно делать в отладочных целях, когда окончательный шаблон находится в стадии разработки и отладки.

Для ручной работы с данными на картах можно использовать встроенную консоль (“Инструменты - Консоль”). Консоль позволяет определять тип карты, подбирать ключи авторизации, менять данные как карты, так и трейлера блоками по 16 байт. В специальном поле в консоли выводится UID карты. Консоль поддерживает карты Mifare с шифрованием Crypto1 и карты Ultralight.

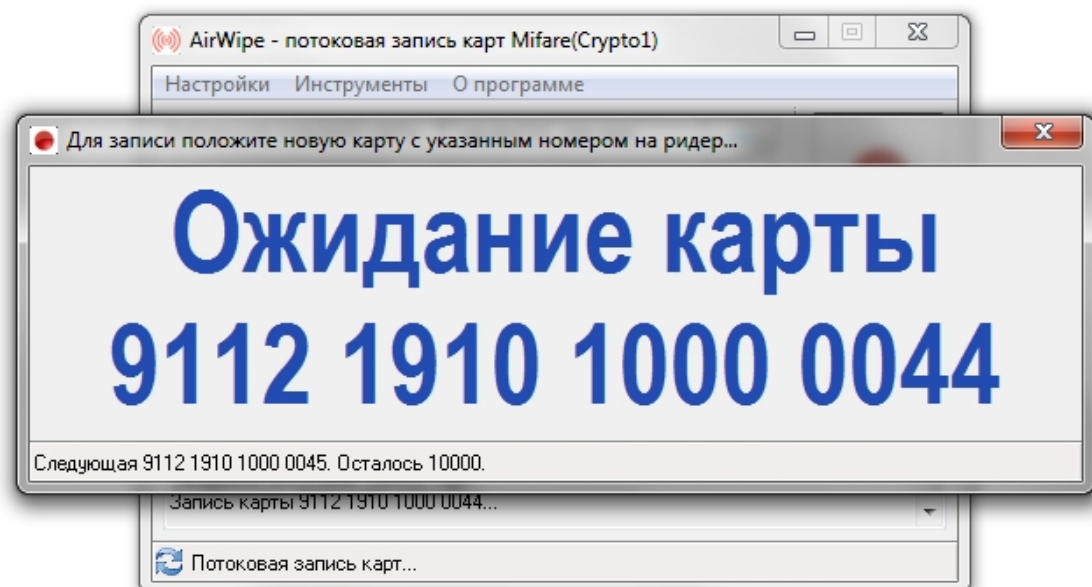


Тип карты можно установить вручную или попробовать определить автоматически, нажав “?” напротив селектора типов. Поддерживаются следующие типы карт: UL – Mifare Ultralight, CL1K – Mifare Classic 1Kb, CL4K – Mifare Classic 4Kb, MINI – Mifare Mini, PL2S1 – Mifare Plus 2Kb Security Level 1, PL4S1 – Mifare Plus 4Kb Security Level 1, PL2S2 – Mifare Plus 2Kb Security Level 2, PL4S2 – Mifare Plus 4Kb Security Level 2, Smart 1K EV1/2, Smart 2K, Smart 4K, Smart MX.

Селектор “Блок” устанавливает **абсолютный** номер блока с которым будут производиться операции в мэппинге карты. Для карт типа Mifare Ultralight этот селектор выбирает начальную страницу, начиная с которой данные будут загружены в поле редактирования; в поле редактирования загружаются 4 страницы. Адресацию блоков следует смотреть в техническом описании соответствующего типа карт. В строке статуса консоли отображается результат операции, причём абсолютная нумерация блока переводится в адрес вида сектор-блок с начальным символом “S”, если адресуется обычный блок данных, и символом “T”, если адресуется трейлер.

В поле “Ключ” устанавливается 12 цифр ключа авторизации, типа ключа (A или B) селектором рядом. Если значение ключа неизвестно, можно попробовать подобрать его из списка предустановленных ключей, нажав “?” рядом с селектором.

Когда шаблон отлажен, и тестовая запись успешно завершена, для поточной записи карт следует выбрать “Инструменты - Запись” или нажать на кнопку “Запись” в главном окне диалога.



Устройство файла шаблона (расширение awp)

В файле шаблона предусмотрено два типа секций: конфигурационная секция **[CONFIG]** и множество секций **[SECTOR]** для описания статических данных. Секция конфигурации – одна, а секций секторов может быть до 40 (максимальное количество секторов на карте), но в одной секции можно описать сразу несколько секторов. Текущая секция заканчивается при объявлении новой секции или при завершении файла.

Пример полного ini-файла.

```
[CONFIG]                                     # секция настроек
LOGLEVEL=8                                   # подробность лога; лог отсутствует при 0
LOGFILE="C:\AirWipe\test.log"               # путь к лог-файлу
DATADIR="C:\AirWipe\"                       # каталог с входными данными при подстановке по '@'
WORKDIR="C:\AirWipe\work"                  # каталог передачи временных файлов
PREEXE="C:\AirWipe\crypto.exe -o EMI"      # программа обработки данных карт перед записью
POSTEXE="C:\AirWipe\emission.exe -f "      # программа обработки данных карт после записи

ACCESS="RO"                                  # режим доступа к карте: OFF, RO или RW
CARDTYPE="CL4K"                             # тип обрабатываемых карт: CL1K, PL2S1 и т.д.
DEFKEYCM=61                                 # HEX-код команды авторизации к "пустой" карте
DEFKEY=FFFFFFFF                             # значение ключа авторизации "пустой" карты

DELIMITER=;                                # символ-разделитель данных
FORMAT=I14:0:2 S14:1:0 H14:2:4             # описание динамических данных из файла данных

[SECTOR 14,16-31]                           # секция данных для указанных секторов
D0,D1,D2=00000000000000000000000000000000 # данные для блоков 0,1,2 в секторе
T=A0A1A2A3A4A5FF078069B0B1B2B3B4B5        # значение трейлера сектора

[SECTOR 15]                                 # секция данных для сектора 15
D0,D1,D2=FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF # данные для блоков 0,1,2 в секторе 15
T=@def_trailer.dat                         # данные берутся из файла def_trailer.dat

[SECTOR 32]                                 # секция данных для сектора 32
D0-D14=000000000000000000000000000000000 # все блоки данных прописываются нулями
T=FFFFFFFFFFFFFFFF078069FFFFFFFF          # данные для трейлера

@def_sector.dat                             # данные берутся из файла def_sector.dat
```

Комментарии следуют после символа “#” или символов “//” до конца строки. Возможно использование составной строки, указывая переход символом “\”. При задании параметров возможен отступ от начала строки до начала поля-описателя, также есть возможность разделять пробелами логические конструкции, использовать строчные или прописные буквы. Ниже приводится описание секции конфигурации эквивалентное приведённому ранее.

```
[ config ]                                  // секция настроек
loglevel = 8                                // подробность лога; лог отсутствует при 0
logfile = c:\airwipe\proclass.log           // путь к лог-файлу
datadir = c:\airwipe\                      // каталог с входными данными при подстановке по '@'
workdir = c:\airwipe\work                  // каталог передачи временных файлов
preexe = c:\airwipe\crypto.exe -o emi     // программа обработки данных карт перед записью
postexe = c:\airwipe\emission.exe -f      // программа обработки данных карт после записи

access = ro                                 // режим доступа к карте: off, ro или rw

cardtype = CL4K                             // тип обрабатываемых карт
defkeycm = 61                               // hex-код команды авторизации к "пустой" карте
defkey = ffffffff                           // значение ключа авторизации "пустой" карты

delimiter = ";"                             // символ-разделитель данных
format = I14:0:2 S14:1:0 H14:2:4           // описание динамических данных из файла данных
```

Строковые константы могут задаваться двумя способами: с использованием двойных кавычек и без. В случае использования кавычек, открывающие кавычки ставятся сразу перед значением строковой константы, а закрывающие - сразу после значения строковой константы. Если кавычки не используются, то значение строковой константы принимается с первого отличного от пробела символа и до конца строки, исключая завершающие пробелы. Символы двойных кавычек должны отсутствовать в содержимом строковых констант.

Значения констант можно неоднократно менять (для секции **[CONFIG]** возможность неоднократного изменения только в пределах секции).

Максимальная длина строки (одиночной или сумма длин составных строк) не должна превышать 16384 символов. Максимальный размер **awp**-файла и файлов-ссылок не должен превышать 1Мб. Допустимо использование любого количества вложенности файлов-ссылок, однако, следует избегать перекрёстных ссылок в файлах для предотвращения рекурсии: рекурсия недопустима.

Секция **[CONFIG]**

Параметр **LOGLEVEL** по умолчанию принимает значение 0 при котором лог-файл не создаётся. **LOGLEVEL** желательно указывать самым первым, т.к. именно от его значения зависит подробность создаваемого лога. Если указать значение **LOGLEVEL** после других команд, то трассировка предыдущих команд будет утеряна.

Следующей по важности является команда **LOGFILE**. Именно после указания имени лог-файла создаётся файл трассировки выполнения. Если указать имя файла после каких-то других команд, то в случае ошибки в предыдущих командах лог файл не будет сохранён, т.е. о причинах возникновения ошибки прочесть будет негде. Промежуточные каталоги, указанные в пути к лог-файлу, должны существовать: при создании лога создаётся только файл лога без промежуточных компонентов пути к нему.

При старте интерпретатора шаблона выделяется блок памяти (1Мб), в который сбрасываются данные лога. Изменение значения **LOGLEVEL** приводит к изменению подробности лога сразу после указания каждого последующего значения. Многократное указание имени лог-файла **LOGFILE** не ведёт к созданию нового файла: файл будет создан только один раз. По умолчанию значение пути к лог-файлу пусто.

DATADIR по умолчанию принимает значение текущего каталога программы **AirWipe**, обычно это каталог в котором находится программа, но необязательно. Каталог **DATADIR** указывает на место хранения файлов-ссылок. Каталог может быть указан как с завершающим символом “\”, так и без него.

WORKDIR по умолчанию принимает значение временного каталога системы, через который передаются файлы обмена с данными карты и дополнительными данными для пре- и пост- процессинга.

Значение **PREEXE** по умолчанию пустое: препроцессинг (обработка данных карты перед записью) не используется. Для указания необходимости препроцессинга, необходимо задать полный путь к исполняемому файлу с указанием опций командной строки.

Непосредственно перед вызовом препроцессинговой программы создаётся текстовый файл обмена, в который помещаются данные карты. Формирование строки вызова внешней программы происходит конкатенацией строки **PREEXE** с полным путём(**WORKDIR**) и именем файла обмена. Префикс файла обмена состоит из 3х букв и принимает значение “PRE”, остальная часть имени выбирается системой случайным образом. Для приведённого выше файла конфигурации, строка запускаемого процесса будет выглядеть так:

```
c:\airwipe\crypto.exe -o emi c:\airwipe\work\pref4c2.tmp
```

После завершения процесса, указанного в **PREEXE**, интерпретатор считывает файл обмена и удаляет его (если **LOGLEVEL** > 6, то файл обмена не удаляется).

Значение **POSTEXE** по умолчанию пустое: постпроцессинг (обработка данных карты после записи) не используется. Для указания необходимости постпроцессинга, необходимо задать полный путь к исполняемому файлу с указанием опций командной строки. Функционал постпроцессинга аналогичен препроцессингу, но префикс файла обмена принимает значение "PST".

В **ACCESS** указывается режим доступа к ридеру: **OFF** – доступ к ридеру не производится, однако, данные могут быть переданы в пре-/пост- процессинговый софт, который может произвести запись самостоятельно. В режиме **RO (Read Only)** перед вызовом препроцессинга считывает UID карты. В режиме **RW (Read- Write)** ридер после вызова препроцессинга записывает данные в карту. По умолчанию активен режим **OFF**.

В **CARDTYPE** указывается тип карты. Тип следует указывать точно, т.к. тип карты в принтере отслеживается для того, чтобы не было пересортицы. По умолчанию установлен тип **UNKNOWN** (неизвестный тип карты). Интерпретатор поддерживает карты следующих типов: **CL1K**(Mifare Classic 1Kb), **CL4K**(Mifare Classic 4Kb), **MINI**(Mini), **PL2S1**(Mifare Plus 2Kb Security Level 1), **PL4S1**(Mifare Plus 4Kb Security Level 1), **PL2S2** (Mifare Plus 2Kb Security Level 2), **PL4S2**(Mifare Plus 4Kb Security Level 2), **SM1K**(Smart 1K), **UL**(Mifare Ultralight).

В **DEFKEYCM** устанавливается код команды авторизации для записи карты. Для стандартных карт это **60** для авторизации по ключу А и **61** для авторизации по ключу В. Карты с повышенной секретностью часто используют нестандартные коды команды авторизации: 71 для карт FUDAN, 6F – Holtek и т.п. По умолчанию принимает значение **60**(авторизация по ключу А).

По умолчанию **DEFKEY** принимает значение транспортного ключа **FFFFFFFFFFFF**, однако, при записи подготовленных карт это значение может быть другим.

DELIMITER – символ разделитель полей в строке динамических данных. Интерпретируется как однобайтовый ASCII-символ. По умолчанию установлен код пробела (0x20).

В **FORMAT** через пробел задаются формат и количество интерпретируемых динамических данных, принимаемых из Прокарта. По умолчанию параметр пуст.

Динамические данные из Прокарта передаются в виде текстовой строки в которой данные разделяются символом, заданным в **DELIMITER**. Тип, адрес и количество передаваемых полей данных указывается в поле **FORMAT** с помощью дескрипторов данных. Возможны 3 типа дескрипторов: **I** (integer), **S** (string), **H** (hex). После дескриптора указывается адрес, по которому эти данные должны быть размещены: **s:d:o**, **s** – сектор (sector), **d** – номер блока в секторе (datablock), **o** - смещение в блоке (offset). Указание дескриптора без адреса размещения данных не имеет смысла, поэтому под дескриптором понимается совместное описание типа данных и адрес их размещения.

Дескриптор **I** позволяет интерпретировать переданные текстовые данные как беззнаковое десятичное число от 0 до 4294967295 (всегда 4 байта). Данные в карту записываются в порядке младший-старший. Например, **I14:0:2** интерпретирует входную строку как число, которое будет записано в сектор 14, со 2го байта блока 0.

Дескриптор **S** позволяет напрямую записать текстовые данные в карту: один текстовый ASCII-символ интерпретируется в один байт данных; длина строки определяется символом-разделителем или окончанием длины передаваемого буфера.

Дескриптор **H** позволяет записать в карту любые данные: пара hex-цифр интерпретируется как байт; следует следить, чтобы тетрады передавались полностью для предотвращения неоднозначности интерпретации последнего байта; длина hex-строки определяется символом-разделителем или окончанием длины передаваемого

буфера. Например, строка **DEADBEEFA** будет интерпретирована в числа **DE_AD_BE_EF**, т.к. **A** не имеет тетрадной пары.

Например, из csv-файла номеров и данных была передана строка данных:

9112 1910 1000 0044 ;5634 ;WATCHTOWER ;DEADBEEF.

Параметр **DELIMITER** – точка с запятой, параметр **FORMAT = I14:0:2 S14:1:0 H14:2:4** позволяет интерпретировать принятые данные

- 1) – сообщение оператору о том, что записывается карта с номером “**9112 1910 1000 0044**” (можно было написать “Hello!”, на запись карты это не влияет);
- 2) **5634** – целое число 5634, которое в hex-формате займёт 4 байта и будет равно **0x00001602**, а в карте будет записано в абсолютный блок 56 (сектор 14, блок 0 от начала сектора), начиная со 2го байта (отсчёт с 0);
- 3) **WATCHTOWER** – интерпретируется как ASCII строка длиной 10 символов **\x57, \x41, \x54, \x43, \x48, \x54, \x4F, \x57, \x45, \x52**, записываемая в абсолютный блок 57 (сектор 14, блок 1 от начала сектора) с самого начала блока, т.к. установлено смещение 0;
- 4) **DEADBEAF** – интерпретируется как hex-строка **DE_AD_BE_AF** длиной 4 байта, записываемая в абсолютный блок 58 (сектор 14, блок 2 от начала сектора) со смещения 4 (пропускаем 4 байта от начала).

```
S14:00 (B056) - 000002160000000000000000000000000000
S14:01 (B057) - 5741544348544F5745520000000000000000
S14:02 (B058) - 00000000DEADBEEF00000000000000000000
```

Остальные данные будут проигнорированы, т.к. строка формата их не описывает. В случае нехватки данных, описываемых строкой формата, программа вернёт ошибку.

Секции [SECTOR nn]

В секциях **[SECTOR nn]** описываются статические данные, которые будут записаны на карту. Максимальный диапазон изменения номера сектора после дескриптора **SECTOR**, от 0 до 39, однако, у реальных карт значение последнего сектора зависит от типа: **CL1K** – 15, **CL4K** – 39, **PL2Sn** – 31, **PL4Sn** – 39, **MINI** – 4. Следует следить, чтобы не было «лишних» данных, несоответствующих типу карты, в противном случае интерпретатор шаблона сообщит об ошибке.

При описании данных сектора допустимо перечисление номеров секторов через запятую или указание диапазона номеров, например,

```
[SECTOR 0, 1, 2, 3] или
[SECTOR 0 - 4, 5, 7, 9, 12-15 ]
```

После открытия секции, следует описание статических данных после дескриптора **D** с указанием номера блока в секторе (0...14); перечисление блоков можно производить как через запятую, так и списками. Статические данные описываются hex-строками по 16 байт в каждой. Длина hex-строки обязательно должна быть 16 байт, иначе интерпретатор шаблона вернёт ошибку. Трейлер секции описывается после дескриптора **T** так же, как и данные, hex-строкой длиной 16 байт.

```
[SECTOR 0, 1, 2-39]
```

```
D0 = 0000000000000000000000000000AA55 # данные для блока 0
D1,D2=FFFFFFFFFFFFFFFFFFFFFFFFFFFFFF # данные для блоков 1,2 D3,D4-
D14=00000000000000000000000000000000 # данные для блоков 3-14
T = A0A1A2A3A4A5FF078069B0B1B2B3B4B5 # значение трейлера сектора
```

Приведённая выше запись инициализирует все сектора карты **CL4K**, несмотря на то, что размер секторов в карте различен. Это происходит потому, что номер блока данных, лежащий за пределами сектора, игнорируется. Также игнорируются данные, указанные для блока 0 сектора 0 (на реальных картах этот блок доступен только для чтения). Подобный подход позволяет упростить написание инициализации карт.

Ограничений на повтор объявления секций и данных в них нет (в отличие от секции **[CONFIG]**), однако, в итоговом дампе будут данные, которые объявлялись в шаблоне последними (порядок обработки awp-файла слева-направо, сверху вниз).

Использование ссылок

В шаблоне вместо прямого указания данных возможно использование ссылок на файлы с требуемыми данными.

```
[SECTOR 15] # секция данных для сектора 15
D0,D1,D2=FFFFFFFFFFFFFFFFFFFFFFFFFFFFFF # данные для блоков 0,1,2 в секторе 15
T=@def_trailer.dat # данные берутся из файла def_trailer.dat
```

В приведённом выше примере показано, что данные для трейлера берутся из файла **def_trailer.dat**, в котором находится строка **A0A1A2A3A4A5FF078069B0B1B2B3B4B5**.

Файлы-ссылки ищутся в каталоге заданном параметром **DATADIR** в секции **[CONFIG]**. Для исключения некорректной интерпретации имени файла, его можно заключить в двойные кавычки. В этом случае допустимо указание дополнительного пути к файлу.

```
T=@"card_trailers\def_trailer.dat"
```

Ссылки бывают однострочные и многострочные. Выше приведён пример однострочной ссылки: даже если в файле будет больше одной строки, использована будет только первая. Максимальная длина данных, которые будут загружены по такой ссылке – не более 16384 байт, включая данные до символа '@'. Все ссылки после знака '=' интерпретируются как однострочные.

Файл может быть интерпретирован как многострочная ссылка, если ссылка будет единственной записью в строке.

```
[SECTOR 0] D0,D1,D2=00000000000000000000000000000000
T=A0A1A2A3A4A5FF078069B0B1B2B3B4B5
```

```
[SECTOR 14]
@defsector.dat
```

```
[SECTOR 15] D0,D1,D2=FFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
T=FFFFFFFFFFFFFFFF078069FFFFFFFFFFFFFF
```

Ссылка **@defsector.dat** будет интерпретирована как многострочная ссылка: содержимое файла **defsector.dat** будет загружено в память и вставлено строкой ниже после строки со ссылкой.

Следует избегать использования ссылок до явной инициализации каталога **DATADIR**. Также следует избегать использования в файлах рекурсивных ссылок.

Файл обмена для внешней обработки данных

Для обмена данными при пре- и постпроцессинге используется специальный формат обмена данными. Файл, передаваемый на обработку, имеет следующий вид:

```
STATUS: WAIT
TYPE: CL4K
UID: 04AE3267CB0282
READER: "SCM Microsystems Inc. SCR331-DI USB ContactlessReader 0"
```

```
B001: 00000000000000000000000000000000
B003: A0A1A2A3A4A5FF078069B0B1B2B3B4B5
B056: 00000216000000000000000000000000
B057: 5741544348544F574552000000000000
B058: 00000000DEADBEEF0000000000000000
```

Поле **STATUS** отражает текущее состояние данных в файле и принимает значения **WAIT**, **OK**, **FAIL** (в момент передачи процессингу - только **WAIT**).

Поля **TYPE**, **UID**, **READER** являются информационными для программ - обработчиков, чтобы они могли правильно произвести свои вычисления или записать данные в обход программы. Значения этих и других неизвестных полей будут игнорироваться при считывании содержимого файла после обработки данных. В поле **TYPE** указывается тип обрабатываемой карты, в поле **UID** – UID карты. В поле **READER** указывается **полное** имя ридера, с которым ведётся работа.

Поля **B001**, **B002** и т.д. описывают записываемые блоки карты; к неописанным блокам доступ не будет производиться. Данные описываются hex-строкой длиной 16 байт.

После формирования файла с данными, программа освобождает хэндл ридера и запускает процесс, указанный в поле **PREEXE**, ожидая его завершения. После завершения, считывается переданный ранее файл: он должен быть перезаписан программой пре- или постпроцессинга. В считанном файле проверяется поле **STATUS**: для продолжения работы его значение должно быть равно **OK**. Если значение статуса отличается от **OK**, дальнейшая обработка будет прекращена, и интерпретатор шаблона вернёт ошибку.

Если статус **OK**, то будут считаны данные **Bnn**(необязательные поля), наличие других полей не проверяется, значения в них игнорируются. Если значение поля **PREEXE** не задано, то файл с данными для обработки не формируется. После считывания, файл обмена удаляется, если **LOGLEVEL** < 7.

Данные, полученные после препроцессинга, будут записаны на карту только в случае установки режима **RW** для доступа к карте в поле **ACCESS** секция **[CONFIG]**. В противном случае процесс записи данных на карту будет пропущен.

Далее вызывается постпроцессинг: его работа не отличается от препроцессинга, за исключением того, что даже если в файле обмена будут указаны данные, они будут проигнорированы – смысл есть только в считывании поля **STATUS** после постпроцессинга.

Порядок выполнения шаблона

Чтение новой строки данных запускает процесс обработки шаблона. Процесс начинается с инициализации внутренних данных (установки значений по умолчанию):
тип карты устанавливается в **CARDTYPE - UNKNOWN**;
команда авторизации **DEFKEYCM - 60**;
разделитель данных **DELIMITER – “;”**;
все блоки данных в карте неактивны и обнулены;
все трейлеры карты неактивны и инициализированы строкой
FFFFFFFFFFFFFFFF07869FFFFFFFFFFFFF (ключ А и В - **FFFFFFFFFFFF**);
в **DATADIR** копируется путь по которому лежит вызывающая программа;
в **WORKDIR** копируется путь к временному каталогу системы; остальные
данные обнулены и пусты.

Выделяется буфер объемом 256Кбайт для сохранения сообщений в памяти, пока не открыт лог-файл.

В случае невозможности открыть файл шаблона, выдается сообщение об ошибке. Ошибочным также считается файл шаблона размером менее 8 байт и более 1Мбайта.

После загрузки шаблона в память в лог добавляется сообщение с датой и временем старта сессии обработки шаблона.

Далее происходит разбор и интерпретация данных шаблона с выдачей ошибок в лог-файл. **LOGLEVEL** управляет подробностью отладочных сообщений: при **LOGLEVEL > 4** в лог-файл выводятся установленные значения конфигурации, при **LOGLEVEL > 5** выводится листинг строк файла шаблона и ссылок в процессе их обработки, при **LOGLEVEL > 7** выводится инициализированный статическими данными дамп карты.

Последующая проверка типа карты производится на соответствие объема карты с максимальным адресом при задании статических данных.

Далее разбираются входные (динамические) данные в соответствии со строкой формата **FORMAT**. Если **LOGLEVEL > 3**, в лог-файл будут записаны входные данные и строка формата. Если **LOGLEVEL > 5**, то в логе также будут указаны параметры буфера динамических данных, а также расшифровка, как входные данные были интерпретированы, чтобы убедиться что строка – это строка, число – это число. После разбора переданных динамических данных, если **LOGLEVEL > 7** выводится дамп карты, инициализированный динамическими данными.

Если в поле **ACCESS** указаны режимы **RO** или **RW**, далее происходит чтение типа карты и UID с выдачей соответствующих ошибок, если они обнаружены (отсутствие подключения к ридеру, отсутствие карты, неверный тип карты и т.п.). Если ошибок не обнаружено, то в выходной буфер данных для Прокарта записывается текстовая строка с UID-ом карты (hex-формат).

Проверяется поле **PREEXE**: если поле содержит командную строку для вызова программы препроцессинга, то в каталог **WORKDIR** записывается файл обмена и вызывается внешний обработчик данных. Интерпретатор шаблона ожидает завершения дочернего процесса, и, после завершения, считывает, а затем удаляет файл обмена. При **LOGLEVEL > 3** указывается время вызова процесса, если **LOGLEVEL > 6**, то удаление не производится. Если **LOGLEVEL > 7** выводится дамп карты, считанный после препроцессинга.

Если в поле **ACCESS** указан режим **RW**, производится запись дампа данных в карту (если есть что записывать). При **LOGLEVEL** > 2 в логе указывается время начала и окончания записи данных. Если запись данных завершается ошибкой, то производится попытка доступа к сектору по ключам **A** и **B** из трейлера этого сектора, даже при неактивности трейлера: такой подход позволяет реализовать доступ к данным на картах, не прошедших подготовку к печати - достаточно указать содержимое трейлера при описании статических или динамических данных.

Проверяется поле **POSTEXE**: если поле содержит командную строку для вызова программы постпроцессинга: алгоритм работы постпроцессинга такой же, как и препроцессинга, с той разницей, что постпроцессор может повлиять только на статус завершения обработки.

Работа интерпретатора шаблона завершается выдачей сообщения в лог об окончании сессии с указанием времени завершения и удержанием на экране сообщения об окончании записи.