

Клеверенс: RFID для «1С:Предприятия»

Документация по внешней компоненте CleverenceRFID

Апрель 2013 (для версии компоненты 1.2.0.17)

Motorola FX7400



Motorola XR480



Содержание

Введение	6
Возможности компоненты	6
Соответствие стандартам	6
Общие сведения о RFID	7
Обзор частот RFID	7
Считыватели и метки	8
RFID Class 1 Generation 2.....	8
Метки стандарта Class 1 Generation 2.....	10
Что следует знать о UHF RFID Class 1 Generation 2	10
«Кошмар 2050 года»	11
Принципы идентификации объектов с помощью меток Class 1 Generation 2	12
RFID Class 1 Generation 2 для логистики и розницы	13
RFID Class 1 Generation 2 для библиотек	13
Понятие электронного кода объекта	14
Понятие электронного кода продукта (EPC).....	15
Что содержит EPC?.....	16
Способы кодирования EPC.....	16
Примеры маркировки товаров при помощи EPC	17
Понятие уникального кода объекта (UID).....	18
Что содержит UID?	19
Установка и обновление компоненты	20
Установка компоненты автоматически	20
Установка компоненты вручную.....	20
Дополнительная настройка компоненты	21
Обновление компоненты переустановкой.....	21
Обновление компоненты копированием файлов	21
Лицензирование компоненты.....	22
Установка и настройка RFID считывателей	23
Установка и настройка Motorola FX7400	23
Подключение Motorola FX7400 через витую пару (кабель Ethernet, разъем RJ45)	23
Подключение Motorola FX7400 через кабель USB.....	24
Предварительная настройка Motorola FX7400	25
Демонстрационная обработка	27
Тестирование компоненты без RFID-считывателей	27
Тестирование инвентаризации	28
Тестирование записи.....	29
Работа с компонентой	31
Начало работы с компонентой	31
Завершение работы с компонентой	31
Получение лицензии на компоненту	31
Виртуальный режим работы компоненты.....	32
Внешние события компоненты	34
Событие «НайденСчитыватель».....	34
Событие «Чтение»	34

Событие «Запись».....	36
Поиск и подключение RFID-считывателей.....	36
Синхронный поиск считывателей.....	37
Асинхронный поиск считывателей.....	37
Ошибки в работе с компонентой	37
Известные ошибки в работе компоненты.....	38
Чтение меток асинхронно (асинхронная инвентаризация) ничего не читает	38
«Приложение будет закрыто» и обрушение окна «1С:Предприятия»	38
Операции с метками Class 1 Gen 2.....	39
Инвентаризации меток Class 1 Gen 2	39
Синхронная инвентаризация меток.....	39
Асинхронная инвентаризация меток	40
Чтение банков памяти меток Class 1 Gen 2	41
Чтение банка EPC/UII	41
Чтение банка USER	41
Чтение банка TID (запись запрещена).....	42
Чтение и запись банка RESERVED.....	42
Запись банков памяти для меток Class 1 Gen 2	42
Прописывание меток идентификационным кодом EPC/UII	43
Создание новых EPC по данным из базы «1С:Предприятия»	43
Запись сразу в несколько меток	43
Запись только в одну конкретную метку	44
Использование компоненты для задач розницы.....	46
Что можно записать в RFID-метку.....	46
Создание новых EPC по данным о товарах	46
Использование компоненты для библиотечных задач	47
Что можно записать в RFID-метку.....	47
Маркировка библиотечных объектов RFID-метками.....	48
Маркировка библиотечного фонда.....	48
Маркировка читательских билетов (и RFID-карточек)	51
Маркировка библиотечного имущества (столы и стулья)	52
Общий алгоритм маркировки	53
Антикражный механизм для библиотек	54
Выдача и возврат книг	55
Справочник разработчика.....	56
Cleverence.RFID.Api	56
UIIизБиблиотечногоКода (UIIfromISIL).....	56
UIIизБиблиотечногоКода (UIIfromISIL).....	56
СоздатьБиблиотечныйОбъект (CreateLibraryItem).....	56
ОтключитьВсеСчитыватели (DisconnectAllReaders)	56
НайтиСчитыватели (LookupReaders)	57
ПолучитьСчитыватель (GetReader)	57
ПодключитьСчитыватель (ConnectToReader).....	57
ПодключитьСчитывательПодПаролем (ConnectToReaderWithPassword)	57
ВыбратьМетку (FetchTag).....	57
ВыбратьМетки (FetchTags).....	58
EPCизSGTIN (EPCfromSGTIN).....	58
EPCизEAN13 (EPCfromEAN13)	58
EPCизGDTI (EPCfromGDTI)	59
EPCизSSCC (EPCfromSSCC).....	59
НоваяМетка (CreateTag)	59
НоваяМетка (CreateTag)	60
НоваяМетка (CreateTag)	60
СоздатьБиблиотечныйОбъект (CreateLibraryItem).....	60

Библиотеки (Libraries)	60
Версия (Version)	60
Язык (Culture).....	60
ВиртуальныйРежим (VirtualMode)	60
LookupTagParams (LookupTagParams)	60
ФильтрыEPC (EpcFilterValues)	60
AFI (AFI)	61
Cleverence.GS1.AttributeBits	61
Cleverence.GS1.Epc	61
Строка (ToString).....	61
Схема (EpcScheme)	61
AttributeBits (AttributeBits).....	61
КодКомпании (Company)	61
Ссылка (Reference).....	61
СерийныйНомер (Serial).....	61
Фильтр (FilterValue).....	61
URI (URI).....	61
SGTIN (SGTIN).....	61
GDTI (GDTI)	61
SSCC (SSCC).....	62
Cleverence.GS1.EpcFilterValue	62
Схема (EpcScheme)	63
Значение (Value).....	63
Тип (Type).....	63
Представление (DisplayName).....	63
Cleverence.GS1.EpcScheme.....	63
Cleverence.ISO.Afi	64
Cleverence.ISO.AfiCollection.....	64
Количество (Count).....	64
Элемент (get_Item)	64
Добавить (Add).....	64
Удалить (Remove).....	64
УдалитьПоИндексу (RemoveAt)	64
Добавить (Add).....	64
Cleverence.ISO.LibraryItem.....	65
СформироватьUSERБанк (ToUSER_BANK).....	65
УникальныйКод (PrimaryItemIdentifier).....	65
ISIL (ISIL).....	65
Наименование (Title)	65
МестоНаПолке (ShelfLocation).....	65
РазмерНабора (SetSize)	65
ИндексВНаборе (SetIndex)	65
Cleverence.ISO.LibraryUii	65
Закодировать (Encode)	65
УникальныйКод (PrimaryItemIdentifier).....	65
ISIL (ISIL).....	66
ТипИспользования (TypeOfUsage).....	66
IsValid (IsValid).....	66
ErrorString (ErrorString)	66
DataElements (DataElements).....	66
Cleverence.ISO.TypeOfUsage	66
КодКласса (ClassCode).....	68
ИмяКласса (ClassName).....	68

КодИспользования (UsageCode).....	68
Описание (UsageDescription).....	69
Cleverence.ISO.Uii.....	69
AFI (AFI)	69
Value (Value).....	69
Cleverence.RFID.LookupTagParams.....	69
IsValid (IsValid).....	69
ErrorString (ErrorString)	69
БинарноеПредставление (BinaryString)	69
Cleverence.RFID.RfidAntennaInfo	69
Ид (Id)	69
Разрешена (IsEnabled).....	69
Подсоединена (IsConnected)	69
КоэффициентНаправленности (Gain)	69
ВходнаяМощность (TransmitPower).....	69
Cleverence.RFID.RfidAntennaInfoCollection	69
Количество (Count).....	69
Элемент (get_Item)	69
Добавить (Add).....	70
Удалить (Remove).....	70
УдалитьПоИндексу (RemoveAt)	70
Cleverence.RFID.RfidChip	70
Cleverence.RFID.RfidReader	70
Подключить (Connect).....	70
ПодключитьПодПаролем (ConnectWithPassword)	70
ПолучитьВозможности (GetCapabilities)	70
ИнвентаризоватьМетки (LookupTags)	71
НачатьИнвентаризацию (LookupTagsBegin).....	71
ОкончитьИнвентаризацию (LookupTagsEnd)	71
ВыбратьМетку (FetchTag)	71
ВыбратьМетки (FetchTags)	72
ПрочестьБанкEPCU11 (ReadEPCU11).....	72
ЗаписатьEPCU11 (WriteEPCU11).....	72
ЗаписатьEPCU11noTID (WriteEPCU11forTID)	73
ПрочестьБанкRESERVED (ReadRESERVED).....	73
ПрочестьБанкTID (ReadTID).....	73
ReadTIDs (ReadTIDs).....	74
ПрочестьБанкUSER (ReadUSER).....	74
ReadUSERS (ReadUSERS)	74
ВключитьАнтенну (EnableAntenna).....	75
ВыключитьАнтенну (DisableAntenna)	75
УстановитьВходнуюМощностьДляАнтенны (SetTransmitPower)	75
Отключить (Disconnect).....	75
LookupTagParams (LookupTagParams).....	76
Имя (DisplayName).....	76
Ид (Id).....	76
Url (Url)	76
Модель (Model).....	76
Подключен (IsConnected).....	76
ИдетИнвентаризация (LookupTagsInProgress).....	76
РазрешатьПовторныеЧтения (AllowRepetitiveReads).....	76
Cleverence.RFID.RfidReaderCapabilities	76
Антенны (Antennas)	76
Cleverence.RFID.RfidReaderCollection	76

Количество (Count).....	76
Элемент (get_Item).....	76
Добавить (Add).....	76
Удалить (Remove).....	76
УдалитьПоИндексу (RemoveAt)	76
ДобавитьПоUrl (AddByUrl).....	77
Cleverage.RFID.RfidTag	77
TagId (TagId).....	77
Считыватель (Reader)	77
Объект (Identity)	77
НомерАнтенны (AntennaId).....	77
Время (FirstTimeSeen)	77
Счетчик (SeenCount)	77
RSSI (PeakRSSI).....	77
Cleverage.RFID.RfidTagCollection	77
Количество (Count).....	77
Элемент (get_Item)	77
Добавить (Add).....	78
Удалить (Remove).....	78
УдалитьПоИндексу (RemoveAt)	78
Добавить (Add).....	78
Cleverage.RFID.VirtualRfidTag	78

Введение

Настоящая документация описывает работу с внешней компонентой «AddIn.CleverenceRFID» из кода «1С:Предприятия» в клиентском режиме для реализации проектов по бесконтактному учету в логистике, в рознице и для библиотек.

Возможности компоненты

На текущий момент в версии 1.1.0.2 компонента может быть использована для работы с RFID-считывателями следующих моделей:

- **Motorola FX7400;**
- **Motorola XR480.**

Оба считывателя работают с метками Class 1 Generation 2 на частотах UHF (860-960 МГц), см. более подробно в разделах ниже.

Компонента позволяет выполнять следующие действия:

1. Работа в виртуальном режиме (демонстрация возможностей без наличия под рукой реальных RFID-считывателей и меток);
2. Поиск RFID-считывателей в локальной сети;
3. Подключение к RFID-считывателям по локальной сети (или через VPN);
4. Инвентаризация RFID-меток синхронная и асинхронная;
5. Чтение банков памяти RFID-меток;
6. Запись банков памяти RFID-меток.

ПРИМЕЧАНИЕ: Текущая версия компонента позволяет работать только в «клиентском режиме», т.е. компонента создается на клиенте «1С:Предприятия» и работает на той машине, на которой запущен клиент «1С:Предприятия». Серверный вариант работы компонента, в котором возможна фоновая работа RFID-считывателей в отсутствие запущенных клиентов «1С:Предприятия», пока только планируется к выходу.

Соответствие стандартам

При кодировании и чтении меток компонентой используются положения следующих стандартов:

ISO/IEC 18000-6:2010 - Информационные технологии. Радиочастотная идентификация для управления элементом данных. Часть 6. Параметры для связи через радиointерфейс на частотах от 860 МГц до 960 МГц.

ISO/IEC 15961 - RFID для управления объектами. Протокол передачи данных - прикладной интерфейс.

ISO/IEC 15962 - RFID для управления объектами. Протокол правил кодировки данных и логических функций памяти.

EPCglobal TDS 1.5 - GS1 Tag Data Standard. Определение EPC. Определение данных для хранения на метках Class 1 Gen 2.

ISO 28560-1:2011 - Информация и документация. Радиочастотная идентификация в библиотеках. Часть 1. Элементы данных и общие руководящие указания по применению.

ISO 15511:2011 - Информация и документация. Международный стандартный идентификатор для библиотек и связанных с ними организаций (ISIL).

Общие сведения о RFID

RFID («Эр Эф Ай Ди», Radio Frequency Identification – «радио-частотная идентификация») – это набор разнородных технологий и стандартов, обеспечивающих бесконтактную идентификацию чего-либо по радио-каналу на расстояниях от нескольких сантиметров до сотен метров путем обмена данными между специальным устройством чтения, называемым RFID-считыватель, и специальными метками с антенной и микрочипом, которые наносятся на интересующие объекты.

При этом понятие «RFID-считыватель» (RFID-ридер), микрочипы и радио-диапазон используемых частот – пожалуй, единственное, что объединяет между собой все те технологии, стандарты и оборудование, которые могут скрываться за аббревиатурой RFID.

Обзор частот RFID

На практике все используемые в настоящее время технологии RFID условно подразделяются на следующие группы:

LF	Low-Frequency (низкие частоты)	Технологии и оборудование для работы на частотах условно 125-134 КГц. Грубое проникновение электромагнитных волн на таких частотах в воду (8 метров), живую ткань (2 м) и металлы (~2 мм) обеспечивает их применение в такой области как маркировка скота, домашних и прочих животных. А возможность ограничения зоны считывания несколькими сантиметрами позволяет применять их в системах контроля доступа (но не всегда подходит для смарткарт, т.к. передача данных на такой частоте слишком медленная).
HF	High-Frequency (высокие частоты)	Технологии и оборудование для работы на частотах условно 5-7 МГц, 13,56 МГц. Возможность ограничения зоны считывания несколькими сантиметрами позволяет применять их в системах контроля доступа и оплаты – пропуска, карточки, ключи домофона, сотовые телефоны и т.п.
UHF	Ultra High-Frequency (сверхвысокие частоты)	Технологии и оборудование для работы на частотах 433 МГц, 860-960 МГц, 2,4-2,45 ГГц и 5,2-5,8 ГГц.

С одной стороны, высокая частота волны (большое число колебаний в секунду) означает, что за секунду времени передается больше информации.

С другой стороны, длина волны (расстояние от одного «гребня» до другого) на высокой частоте заметно короче – такие волны «разбиваются» о гораздо более мелкие препятствия, и гораздо больше материалов оказываются для них непроницаемыми.

На практике **скорость важнее, т.к. позволяет за секунду прочесть больше меток**. В тысячу раз выше частота – в тысячу раз больше меток.

Кроме того, частоты UHF позволяют использовать более короткие антенны (чем короче длина волны, тем короче должна быть длина антенны, чтобы она «почувствовала волнение»).

Поэтому, несмотря на то, что человеческое тело, фольга, бочки с жидкостью и т.д. представляют для волн UHF непреодолимое препятствие, **именно UHF и комбинированные UHF/HF RFID технологии развиваются наиболее активно**.

От используемой частоты напрямую зависят скорость и расстояние передачи данных, а также габариты антенн (как у считывателей, так и у меток). Чем ниже частота, тем длиннее должны быть антенны. А чем выше скорость передачи данных, тем быстрее должны работать микрочипы считывателей и меток. Именно скорость чипов долгое время сдерживала развитие технологий на частотах UHF.

Для работы чипа необходимо электропитание. Электропитание может поступить либо из батарейки (тогда это называется «активная метка») либо из энергии волн, которыми RFID-считыватель облучает метки (и тогда это «пассивная метка»). Пассивные метки наиболее распространены, т.к. дешевле в производстве и использовании. Опять же, чем выше частота, тем быстрее чип пассивной метки заряжается от энергии волн.

Считыватели и метки

Любая RFID-система состоит из RFID-считывателя (стационарного или мобильного) и RFID-меток (чаще всего в виде тонких этикеток или толстых корпусных меток). RFID-принтеры, используемые для печати на рулоны RFID-этикеток, можно отнести к RFID-считывателям.

Все без исключения RFID-считыватели одновременно являются и «RFID-писателями», т.к. и чтение, и запись, происходят путем отправки RFID-считывателем определенных команд по радиointерфейсу. Эти команды принимаются и исполняются микрочипами меток.

Важной частью RFID-считывателя являются антенны (или одна антенна), которые во многом определяют дальность и качество считывания. От формы и материала антенны зависит распределение энергии излучаемой волны в пространстве (условно говоря, где вокруг антенны будет пусто, а где густо). Основной характеристикой антенны является её направленность (по-английски «gain») – число, условно показывающее насколько сильно энергия, излучаемая антенной, сконцентрирована в определенном направлении. Без потери физического смысла можно сказать, что если gain = 1, то излучение равномерно распределено во все стороны, а если gain > 19, то большая часть энергии направляется куда-то сконцентрированным лучом.

RFID Class 1 Generation 2

Оборудование и метки RFID могут работать на тех или иных частотах (см. «Обзор частот RFID»). На текущий момент компонента ориентирована только на частоты UHF (Ultra High-Frequency, сверхвысокие частоты) в применении к меткам стандарта Class 1 Gen 2 (стандарт ISO 18000-6C).

UHF RFID – это технологии и оборудование для работы на частотах 433 МГц, 860-960 МГц, 2,4-2,45 ГГц и 5,2-5,8 ГГц (из них Class 1 Gen 2 только на частотах 860-960 МГц). В настоящее время широко распространено мнение, что UHF – наиболее перспективные частоты и технологии для маркировки практически чего угодно.

Стандарт Class 1 Generation 2 (Class 1 Gen 2 или просто Gen2, второе поколение первого класса) – это набор документов, разработанных коммерческой организацией «EPCglobal, Inc.»¹, в которых подробно описано своего рода «техническое задание» на чипы RFID-меток и RFID-считыватели (стандарт ISO18000-6).

¹ «EPCglobal, Inc.» – американская компания, владельцами которой на паритетных началах являются международная организация «GS1» (занимающаяся вопросами стандартизации в области штрихкодирования) и американская организация «GS1 US» (представительство «GS1» в США).

Основное в стандарте Class 1 Generation 2:

1. Чипы меток и считыватели должны работать на частотах UHF 860-960 МГц (при этом и считывателям, и чипам меток, теоретически не запрещается в дополнение к UHF поддерживать и любые другие частоты);
2. Чип каждой метки должен иметь свой уникальный идентификационный номер, прошитый производителем еще на стадии производства (при этом в стандарте прописана структура номера: он должен начинаться на «E2», содержать номер производителя и номер модели. EPCglobal занимается регистрацией производителей чипов и выдает им те самые номера производителя);
3. Чипы меток должны поддерживать не только чтение, но и запись данных;
4. В чипе должен присутствовать специальный банк памяти для хранения уникального идентификатора маркируемого объекта (так называемого EPC/UII, см. ниже), прошиваемый на этапе начала пользования меткой (этот банк записывается **пользователем метки**. т.е. это не номер чипа, записываемый производителем, и вообще не имеет к номеру чипа никакого отношения, см. пояснения ниже);
5. Чипы в метках должны позволять задавать пароль доступа на чтение или запись данных;
6. Чипы в метках должны позволять «прожигать» данные намертво, так чтобы их уже нельзя было переписать;
7. Чипы в метках должны позволять безвозвратно стирать с них информацию, производить так называемое «убийство» метки (в данном случае самоубийство);
8. Чипы в метках должны позволять задавать пароль на эту функцию «убийства», в дополнение к паролю на доступ к чтению/записи.

На уровне радио-протокола обмена между чипами меток Class 1 Gen 2 и считывателем UHF можно производить следующие операции:

1. Операция инвентаризации меток (самая быстрая и надежная);
2. Операция чтения содержимого конкретных банков чипа RFID-метки;
3. Операция записи какого-то заранее известного значения (константы) в определенные места конкретных банков чипа RFID-метки (пишет все чипы, которые подошли под заранее задаваемый критерий);
4. Операция «прожига намертво» содержимого конкретного банка чипа RFID-метки;
5. Операция блокирования банков или убийства чипа RFID-метки.

Метки стандарта Class 1 Generation 2

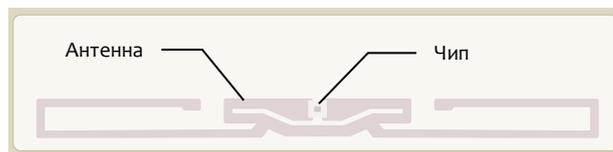
Стандарт на Class 1 Generation 2 описывает только частоты, протоколы обмена и некоторые алгоритмы работы (или советы по алгоритмам) для чипов, используемых в RFID-метках. Помимо чипа, метка состоит из антенны и субстрата для крепления метки на объект. В стандарте ничего не сказано о креплении на металл, надежности приклеивания или размерах метки. Всё, что нужно для надежного крепления и хорошего считывания, – отдается на откуп производителя конкретных меток.

Метки могут быть любыми: тонкие самоклеющиеся бумажные и синтетические в виде этикетки, толстые пластмассовые корпусные, стеклянные вживляемые, съедобные и т.д.

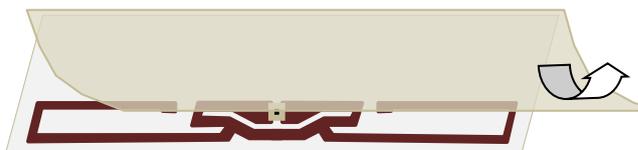
RFID-метка в виде этикетки, вид спереди



RFID-метка в виде этикетки, вид на просвет



RFID-метка в виде этикетки, вид сзади (подложка отклеена)



RFID-метка в виде этикетки, наклеена на объект



Чип в метке стандарта Class 1 Gen 2 позволяет читать из 4х банков памяти разного размера и писать в 3 из них. Тот единственный банк, из которого можно только читать (но нельзя писать), – это банк №2, в котором должен храниться уникальный номер чипа, присваиваемый еще на стадии производства чипа.

Структура данных на метке стандарта Gen2 (4 банка памяти):

Банк №0	(RESERVED)	Пароль уничтожения (4 байта)	Пароль для чтения (4 байта)	+что-нибудь еще...
Банк №1	(EPC)	Заголовок (4 байта)	EPC/UII* (от 12 байт)	XPC + еще...
Банк №2	(TID)	уникальный номер чипа (от 4х до 12ти байт)		
Банк №3	(USER)	зависит от метки: может отсутствовать, а может быть в несколько килобайт		

* EPC - Electronic Product Code (Электронный код продукта), UII – Unique Item Identifier (Уникальный идентификатор объекта), см. специальный раздел ниже.

Пяти и больше банков в метке Gen2 быть не может, т.к. в протоколе общения считывателя с меткой номер банка кодируется всего двумя битами (итого получаются сочетания «00», «01», «10» и «11»).

Что следует знать о UHF RFID Class 1 Generation 2

О технологии UHF RFID следует знать следующее:

1. UHF позволяет передавать в секунду больше данных – это позволят читать больше меток за меньшее время;
2. UHF читает метки с большего расстояния, чем HF или LF;
3. Для меток UHF нет необходимости придумывать свои схемы кодирования, изобретать таблицы поиска соответствия и т.п. – существуют стандарты GS1 Tag Data Standard и ISO/IEC, в которых всё уже придумано;

4. Не существует никакого «кода RFID метки» и тем более «уникального кода RFID метки». Существуют только две вещи: 1) электронный код маркированного объекта, записанный на RFID-метке, и 2) уникальный номер чипа, используемого в метке.
5. Метки UHF, которые вы купите у поставщика, все будут иметь одинаковый «код метки». Любое ТЗ, в котором предполагается привязывать какие-то «уникальные коды меток» к товарам или имуществу - ересь (см. пояснения ниже).

Последние три пункта требуют пояснения. Дело в том, что «код метки» у меток Class 1 Generation 2 (меток Gen2) под 860-960 МГц, который называется TAG ID и отображается при чтении как длинная строка цифр и букв (например, «3024000003320C4063A23312»), – это не просто цифры, а **шестнадцатеричная запись бинарного представления универсального электронного кода (EPC или UID)**, на который должна быть нанесена купленная вами метка. Метка идентифицирует не себя, а объект, на который её клеят. И перед началом использования метки от вас ожидают, что вы самостоятельно **пропишете в метку** свои EPC или UID (т.е. код товара/объекта/контейнера/документа, а также (опционально) его серийный номер), после чего метка при чтении будет возвращать именно их (см. ниже «Понятие электронного кода объекта»).

Кратко, EPC хранится в метке и содержит информацию о товаре, компании-производителе (или компании-владельце) и серийном номере конкретной единицы каждого товара или упаковки.

Более подробно о EPC с примерами маркировки см. раздел «Понятие электронного кода продукта (EPC)».

Более подробно использование компоненты RFID от Клеверенс для задач розницы рассмотрено в разделе «Использование компоненты для задач розницы».

«Кошмар 2050 года»

Представим, что на дворе 2050 год и всё-всё вокруг промаркировано RFID-метками UHF. Эта ситуация сильно отличается от той, когда всё-всё вокруг промаркировано штрихкодами.

Используя штрихкоды почти всегда можно было закрыть глаза на все стандарты. Исключениями служат только производство для продаж в супермаркетах, маркировка грузов в международной логистике и прочие крупные затеи. Для внутреннего учета всегда можно было использовать самодельные штрихкоды. Это разрешено, т.к. сканер штрихкодов читает только то, на что мы его направили, и внезапное чтение штрихкода с пачки сигарет в кармане кладовщика никогда не было проблемой.

С RFID всё не так! Каждая пачка сигарет, каждый документ – любая мелочь будет прочитана вашим считывателем как своя. И если вам в голову пришла светлая идея прошивать свои метки кодами «1, 2, 3, 4, 5...» – будьте уверены, вы такие не одни. Ваш склад и офис будет наполнен сотнями меток, все с номерами «1, 2, 3...», и все непонятно где наклеены. Вот это подарок!

Внедряя «взрослую» систему придется и к делу подойти «по-взрослому»: использовать международные стандарты.

Принципы идентификации объектов с помощью меток Class 1 Generation 2

Архитекторы UHF RFID разрабатывали всю систему исходя из следующего принципа идентификации:

1. **Перед маркировкой** объекта в метку записывается **уникальный номер идентифицируемого объекта** (в виде EPC или UII, см. специальный раздел ниже). Метка крепится к объекту. Таким образом, EPC/UII в метке идентифицирует объект, к которому прикреплена метка. Метки инвентаризуются по EPC/UII, со скоростью тысяча меток в секунду;
2. В метке дополнительно хранится **уникальный номер чипа**, который **идентифицирует саму метку**, а не тот объект, к которому она прикреплена. Номер чипа никак не связан с EPC/UII и не имеет никакого отношения к идентифицируемому объекту.

Правильное понимание того, как устроен «уникальный код» RFID метки Class 1 Gen 2 (UHF RFID):

- 1) «Уникальный код» метки (TAG ID) сам по себе не уникален. Большинство производителей поставят вам метки (рулон или коробку) с совершенно идентичными кодами. Ожидается, что вы сами пропишете код товара/контейнера/документа и его уникальный серийный номер в код RFID-метки, и только после этого метки при чтении станут разными;
- 2) «Уникальный код» метки (TAG ID) уже содержит в себе поля, которые необходимо заполнить вашими кодом товара/объекта/контейнера/документа согласно вашей базе, а также серийным номером помечаемого объекта (опционально). На все случаи жизни уже придуманы и приняты стандарты кодирования. Придумывать какие-то свои схемы кодирования и записывать их в код метки не запрещается, но и не приветствуется;
- 3) Ожидается, что уникальность обеспечивает система идентификации в вашей организации, а не производитель меток. Ожидается, что вы сами пропишете код товара/объекта или контейнера/документа и его уникальный серийный номер в код RFID-метки перед наклейкой её на интересующий объект. Прошитая метка будет нести в себе ваш код и серийный номер, будет возвращать их при инвентаризации;
- 4) Прочитанный код любой метки, например «3024000003320C4063A23312», следует декодировать и вытащить из него информацию о маркированном объекте;

В компоненте CleverenceRFID уже реализовано большинство стандартных методов кодирования/декодирования, и вы уже имеете возможность работать с метками в терминах кодов товаров, номеров книг, серийных номеров, штрихкодов EAN13 и т.п.

- 5) Если «уникальный код» метки (TAG ID) записали за вас – значит это не уникальный код, и ценность такого кода минимальна.

Но продавец говорит мне, что каждая его метка имеет уникальный код!

Продавец в данном случае говорит об уникальном номере у каждого RFID-чипа, используемого им при производстве меток. У любого чипа RFID-метки Class 1 Gen 2 (UHF RFID) согласно стандарту ISO18000-6 должен быть уникальный код. У любой такой метки в любой момент можно узнать уникальный номер используемого в ней чипа, но это значительно более медленная и ненадежная операция, чем инвентаризация RFID, и полагаться на неё в инвентаризации нельзя. Инвентаризация по кодам товаров позволяет читать до 1000 меток/сек. Инвентаризация по номерам чипов едва ли прочтет 5 меток/сек и почти никогда не прочтет все метки, если их лежит больше 1 шт. Уникальные номера чипов можно хранить в базе и использовать для определения того, что метка не была подменена на другую, если вдруг возникли такие подозрения. Для целей инвентаризации исключительно всегда следует использовать банк EPC/UII, в котором метка хранит код объекта, на который она нанесена, и который возвращается считывателю при инвентаризации.

Наше ТЗ содержит простую таблицу соответствия кодов меток объектам нашей базы данных!

Ваше ТЗ ересь (в прямом смысле), т.к. идет против международных стандартов. Код метки не является случайным уникальным числом, а представляет собой «карточку», которую следует заполнить данными из вашей базы. В зависимости от характера маркируемых вами объектов (товары это, документы, книги или другое имущество) заранее предусмотрены стандартные схемы заполнения этой «карточки». Другие банки меток (в частности, пользовательский банк) не предусмотрены для быстрой инвентаризации, их используют для других задач (хранение расширенной информации для операций с отдельными метками, выявление клонов меток и подделок). Для целей инвентаризации исключительно всегда следует использовать банк EPC/UII, в котором метка хранит данные объекта, на который она нанесена, и который возвращается считывателю при инвентаризации. Изобретение своих схем кодирования не запрещается, но и не приветствуется.

RFID Class 1 Generation 2 для логистики и розницы

RFID для задач логистики и розницы регулируется стандартом EPCglobal Tag Data Standard. В основе системы лежит понятие EPC, который прописывается в RFID-метки для маркировки логистических единиц (коробок, паллет, транспорта, транспортных документов, местоположений и т.п.) и розничных товаров (сами товары, комплекты, мелкие аксессуары внутри коробки с товаром и т.п.).

Кратко, EPC хранится в метке и содержит информацию о товаре, компании-производителе (или компании-владельце) и серийном номере конкретной единицы каждого товара или упаковки.

Более подробно о EPC с примерами маркировки см. раздел «Понятие электронного кода продукта (EPC)».

Более подробно использование компоненты RFID от Клеверенс для задач розницы рассмотрено в разделе «Использование компоненты для задач розницы».

RFID Class 1 Generation 2 для библиотек

RFID для библиотечных задач регулируется стандартом ISO 28560. В основе системы лежит понятие UII, который формируется на основе следующих полей (все поля строковые):

1. Номер библиотечного объекта (книги, журнала, диска, читательского билета и т.п.), уникальный в рамках библиотеки – **обязательно**;
2. ISIL библиотеки (International Standard Identifier for Libraries and Related Organisations – Международный стандартный идентификатор для библиотек и других связанных организаций, в России ISIL'ы выдаются ГПНТБ) – **необязательно**;
3. ISBN, ISSN или другой штрихкод – **необязательно**;
4. Тип использования объекта (что это: книга, читательский билет и т.п.) – **необязательно**.

Т.е. минимально можно использовать любой строковой код, уже используемый в учете в библиотеке, и не использовать ISIL и пр. При этом понятно (и это указано в стандарте), что отказ от внедрения ISIL будет означать, что прошитые метки окажутся уникальны только в рамках конкретной библиотеки. Номера книги из других библиотек, если они тоже не будут использовать ISIL, могут совпасть с «нашими» номерами, что вызовет проблемы (см. «Кошмар 2050 года»).

В настоящий момент не существует стандарта на использование UHF RFID в библиотеках. Стандарт ISO 28560 не запрещает использовать UHF, но конкретные схемы кодирования и прочее даны только для HF RFID (в разделах ISO 28560-2 и ISO 28560-3). Реализация Клеверенс полностью

соответствует основной части ISO 28560, а для кодирования использует элементы стандартов ISO 28560-2 и ISO 15962.

Внедрения RFID в библиотеке будет состоять из следующих шагов:

1. Все книги, журналы, диски и т.п. должны быть промаркированы RFID-метками, а затем прошиты правильно сформированным UHF. Все новые поступления также должны маркироваться и прошиваться (см. «Принципы идентификации объектов с помощью меток Class 1 Generation 2»);
2. Читательские билеты могут быть заменены на RFID-карточки;
3. Если использовать «антикражный бит», то при выдаче и возврате книг банк EPC/UHF книг следует перепрошивать, чтобы изменить в них этот «антикражный бит».

Более подробно использование компоненты RFID от Клеверенс в библиотеках рассмотрено в разделе «Использование компоненты для библиотечных задач».

Понятие электронного кода объекта

Архитекторы UHF RFID на метках Class 1 Gen 2 выстроили довольно сложную систему идентификации объектов, которая базируется на существующих стандартах международных организаций GS1 и ISO/IEC по идентификации товаров, грузовых контейнеров, автомобилей, книг, авиабагажа и т.п. Результирующие решения, реализованные в «железе» Class 1 Gen 2, кардинально отличаются от того, что ожидает от RFID-учета любой неподготовленный заранее «технарь».

В первую очередь это касается вопроса об «уникальных номерах меток».

На заре UHF RFID предполагалось, что RFID-метки будут служить простой альтернативой GS1 штрихкодам, и что прошиваться и наноситься на товары они будут еще на этапе производства. Просто вместо того, чтобы заказывать обычные бирки или пачки с заранее напечатанным EAN13, производитель будет заказывать «умные» бирки со штрихкодом + встроенными чипами с заранее прописанным аналогом EAN13, совершенно одинаковым для всех чипов. Никакой уникальности меток не предполагалось.

В системе с неуникальными метками человек должен был подходить с товаром на кассу и система пробивала бы его – какой-то просто «по штрихкоду», какой-то «по штрихкоду из RFID-метки». При этом в Class 1 Gen 2 была реализована возможность подсчета точного количества меток на кассе, даже если все метки идентичны.

Недостатком такой системы служат непреодолимые сложности при инвентаризации. Хотя на кассе система точно подсчитывает количество товара, этого невозможно сделать в мобильном режиме в торговом зале, т.к. человек со считывателем может несколько раз пройти мимо одной и той же полки с разрывом в несколько минут, и система несколько раз «пробьет» товар на ней в результате инвентаризации.

Решением проблемы является назначение каждой единицы товара уникального кода, в дополнение к коду товара. Таким кодом был выбран *серийный номер* – числовой для меток в 128 бит и строковой для меток большего объема.

Ответственность за назначение серийных номеров была возложена на пользователя меток. Т.е. для внедрения RFID-учета от организации требуется организовать у себя систему учета по серийным номерам. Серийные номера могут быть «фиктивными», т.е. никак не отражаться в учете и назначаться просто по счетчику, а могут быть реальными.

Теперь, заказывая бобину с бирками для джинсов или паллету высечки для пакетов с вином, производитель должен получить упаковку, в RFID-чипах которой помимо условного EAN13 будет нарастающим итогом храниться серийный номер каждого экземпляра продукции.

Для учета библиотечных фондов, основных средств, сотрудников и т.п. были придуманы еще более сложные академические схемы, основанные на системе Relative OIDs и классификации наименования ASN.1

Но **во всех случаях предполагается, что код в метку прошьет пользователь метки**, а не её производитель.

В настоящий момент существуют две системы стандартов электронных кодов объектов для использования в RFID. Первые разрабатывает организация EPCglobal GS1. Вторые разрабатывает ISO/IEC. Обе системы частично пересекаются и в вопросах что маркировать, и в вопросах как кодировать. При чтении RFID-считыватель всегда имеет возможность понять, по какому стандарту закодирована метка. Вопрос о том, какой стандарт использовать при кодировании, решается отдельно в каждой конкретной области применения.

Понятие электронного кода продукта (EPC)

Электронный код продукта (EPC, Electronic Product Code) – это способ нумерации **конкретных единиц** товаров, мест хранения, документов и т.д., который используется при маркировке объектов RFID-метками Class 1 Gen 2 по стандарту EPCglobal GS1.

В отличие от кодов (штрихкодов) EAN13 или ISBN, которые обозначают только номенклатуру или артикул товара, EPC в идеале идентифицирует конкретные единицы товара, т.к. в нём есть место под серийный номер каждого экземпляра выпущенного товара. При этом, правда, в качестве серийного номера никто не запрещает хранить нули или номер целой партии товара.

RFID-метками могут помечаться не только сами товары, но и составные части товара (вложенные в коробку аксессуары), упаковки более высокого уровня (блоки, короба) а также тара (поддоны, контейнеры). В кодах EPC всё это уже предусмотрено.

Помимо товаров RFID-метками с EPC могут помечаться основные средства, логистические объекты, скидочные карточки и много другое.

Что в настоящий момент можно кодировать при помощи EPC:

1. Товары с серийным номером (SGTIN)
(включая разные варианты упаковки товара, блочную упаковку, вкладки в коробку, составные части товара, аксессуары и т.п.);
2. Контейнеры, паллеты (SSCC);
3. Места расположения (SGLN);
4. Возвращаемую или оборачиваемую тару (сосуды, поддоны – GRAI);
5. Оборудование, другое имущество (основные средства – GIAI);
6. Скидочные сертификаты, карты лояльности (GSRN);
7. Документы (GDTI);
8. Имущество Министерства Обороны США (USDOD);
9. Что-нибудь другое (GID).

Что кодировать + конкретный способ кодирования в терминах EPC называется «схема».

Помимо конкретных объектов, метки могут наноситься на групповую упаковку, еще более групповую упаковку, либо, наоборот, на составные части объекта. Например, EPC документа может быть наклеена на папку с документами или на ящик с папками.

Что содержит EPC?

Содержимое EPC отличается в зависимости от того, что им кодируется (см. список выше). При этом любой без исключения код EPC содержит 1) заголовок (который и определяет, что кодируем и как, т.е. так называемую схему), 2) так называемое значение *фильтра*, которое определяет уровень упаковки, на которую нанесена метка, и 3) номер компании-производителя или компании-владельца маркируемого товара/объекта. Все возможные заголовки описаны в стандарте GS1 Tag Data Standard (TDS), значение фильтра зависит от схемы, а номера компаниям назначаются международной организацией GS1.

Регистрационный номер компании в ЮНИСКАН/GS1 – обязательный элемент всех схем.

Дополнительно к заголовку, значению фильтра и номеру компании, EPC содержит:

1. Для товаров – номер (артикул, SKU) товара по каталогу и серийный номер конкретного экземпляра;
2. Для контейнеров – серийный номер контейнера, номер компании;
3. Для мест – номер места + дополнительный код;
4. Для возвращаемой и оборачиваемой тары – тип тары и серийный номер экземпляра;
5. Для имущества (основных средств) – номер основного средства;
6. Для скидочных сертификатов и карт лояльности – номер карты или сертификата;
7. Для документов – тип документа и серийный номер экземпляра;
8. Для чего-нибудь другого – тип объекта и серийный номер конкретного экземпляра.

Так же как и большинство стандартных штрихкодов, EPC не содержит информации о количестве, размере, весе или цвете товара, и не предусматривает никакого способа её добавить. Для хранения такой расширенной информации можно либо самостоятельно как-то хитро кодировать в EPC серийные номера, либо использовать пользовательский банк памяти RFID-метки (EPC хранится в отдельном банке памяти под EPC, а под дополнительную информацию отведен отдельный пользовательский банк памяти).

Банк EPC теоретически тоже может позволять хранить пользовательские данные, дописывая их «в хвост» данным самого EPC. Однако на практике редко встречаются RFID-метки, у которых в банке EPC остается свободное место.

Способы кодирования EPC

В RFID-метку EPC записывается при помощи нулей и единиц. Перевод EPC в ноли и единицы называется бинарным кодированием EPC, при этом стандарт на EPC очень подробно описывает, как это должно происходить.

Из метки EPC считывается точно так же в виде нолей и единиц, и чтобы получить из них код компании или серийный номер товара, необходимо произвести декодирование.

Таким образом, один и тот же EPC может быть записан несколькими способами: как число (ноли и единицы) и как осмысленные декодированные данные. При этом самый распространенный способ записи EPC – это строка, представляющая последовательную запись 12-ти байт бинарно закодированного EPC в 16-ричном формате, и именно в таком виде EPC отображают программы, которые идут с RFID-оборудованием по умолчанию.

Данная вставка не обязательна к прочтению

Например, если программа прочитала метку «3024000003320C4063A23312», то это значит, что отдельные байты EPC равны:

	1й байт	2й байт	3й байт	4й байт	5й байт	6й байт	7й байт	8й байт	9й байт	10й байт	11й байт	12й байт
16-ричная запись	30	24	00	00	03	32	0C	40	63	A2	33	12

Если декодировать байты данного EPC, то можно получить следующую информацию:

- Схема кодирования – SGTIN (т.е. закодирован код товара с серийным номером);
- Фильтр – «товар для продажи на кассе»;
- Код компании – 6044 (это регистрационный номер компании-владельца метки или объекта, на который нанесена метка, в реестре международной организации GS1);
- Код товара – 49 (каталожный код номенклатуры в собственной базе компании-владельца метки или объекта, на который нанесена метка. Т.е, например, код товара в базе «1С:Предприятия»);
- Серийный номер – 1671574290 (серийный номер конкретного изделия).

Примеры маркировки товаров при помощи EPC

Пример №1: каждой упаковке молока назначен свой серийный номер:



Считывателю ответили 3 RFID-метки с EPC равными:

3024000003320C4063A23312

3024000003320C4063A23313

3024000003320C4063A23314

итого 3 шт.

Согласно информации, закодированной в этих EPC, перед нами три товара для продажи на кассе, производства компании под номером «6044», каталожный номер «49», с серийными номерами «1671574290», «1671574291» и «1671574292».

Пример №2: у всех упаковок молока одинаковые серийные номера:



Считывателю ответили 3 RFID-метки с идентичными EPC

3024000003320C4063A23312 : 3

итого 3 шт.

Согласно информации, закодированной в этих EPC, перед нами три идентичных товара для продажи на кассе, производства компании под номером «6044», каталожный номер «49», все три с серийным номером «1671574290».

Пример №3: на блоке упаковок молока дополнительно есть своя RFID-метка, у всего одинаковые серийные номера:



Считывателю ответили 7 RFID-меток:

3024000003320C4063A23312 : 6

3084000003320C4063A23312 : 1

итого 6 шт. молока.

Согласно информации, закодированной в этих EPC, перед нами шесть экземпляров идентичного товара для продажи на кассе + упаковка для удобства переноски, производства компании под номером «6044», каталожный номер «49», все с одинаковым серийным номером «1671574290».

Пример №4: ювелирный набор из кулона и двух сережек, на каждом элементе своя метка + отдельная метка на коробочке от набора; проба и другие характеристики ищутся в базе по серийному номеру:



Считывателю ответили 4 RFID-метки:

303000181CE257587E9CA77C

30F000181CE259D87E9CA451

30F000181CE259D87E9CA452

30F000181CE25C587E9CF271

итого 1 шт. товара, 3 шт. вложенных элемента

Согласно информации, закодированной в этих EPC, перед нами один товар для продажи на кассе, внутри которого лежат три вложенных объекта. Всё производства компании под номером «12345», каталожный номер товара – «100701». Вложенные объекты имеют каталожные номера «100711» (2 шт.) и «100721» (1 шт.). У каждого свой серийный номер.

Таким образом из примеров видно, что метки можно наносить:

- как на сам товар, так и на его составные части
и при этом есть возможность отличить, что за метки мы читаем, и не учитывать метки составных частей в сумме чека (но учитывать их при контроле комплектности)
- как на сам товар, так и на оптовые упаковки этого товара
и при этом есть возможность читать только грузовую упаковку, не обращая внимания на много-много индивидуальных меток товара и его составных частей

Понятие уникального кода объекта (UII)

Уникальный код объекта (UII, Unique Item Identifier) – это способ идентификации **конкретных единиц** имущества, библиотечных элементов, грузов, бейджей сотрудников, документов и т.д., который используется при маркировке объектов любыми RFID-метками по стандартам ISO/IEC.

Существует целый ряд стандартов ISO/IEC, рассчитанный каждый на свою область применения.

Примеры того, что в настоящий момент можно кодировать при помощи UII:

1. Авиабалет (IATA Baggage Identification Code);
2. Автомобильные номера (UPU);
3. Библиотечные объекты (код объекта + ISIL);

и, вообще говоря:

4. Что угодно по штрихкоду (EAN.UCC);
5. Любые объекты, под которые предусмотрены идентификаторы данных МН10.8.2 – например, поставщиков и транспортные компании (SCAC), телекоммуникационное оборудование (CLEI), карточки KANBAN и т.п.

Что содержит UИ?

Для каждой конкретной области применения существуют свои правила конструирования UИ, но все они опираются на некий уникальный международный код компании, выданный ей регулирующей организацией, к которому добавляется код объекта, за уникальностью которого в рамках организации отвечает сама организация. Для товаров код компании выдается Юнискан/GS1. Для библиотек – это ISIL библиотеки (в России выдается ГПНТБ).

Установка и обновление компоненты

Установка компоненты автоматически

Дистрибутив компоненты расположен по ссылке <http://www.cleverence.ru/downloads/CleverenceRFID.msi>.

Компонента ставится полностью автоматически и включает файлы документации. Самая последняя версия документации всегда доступна для скачивания с сайта Клеверенс по ссылке: <http://www.cleverence.ru/downloads/CleverenceRFID.pdf>.

Установка компоненты вручную

При желании установку компоненты можно произвести и вручную. Для установки и регистрации компоненты в системе вручную следует:

1. Иметь на ПК установленный .NET Framework 2.0 (<http://www.microsoft.com/downloads/ru-ru/details.aspx?FamilyID=0856eacb-4362-4bod-8edd-aab15c5e04f5>), либо более позднюю версию;
2. Поместить файлы компоненты [Cleverence.RFID.dll](#) (основная библиотека RFID), [Cleverence.RFID.Motorola.dll](#), [Addin.CleverenceRFID.dll](#) (сама внешняя компонента для «1С:Предприятия»), [Symbol.RFID3.Host.dll](#), [RFIDAPI32PC.dll](#) и «Зарегистрировать компоненту [CleverenceRFID.bat](#)» в какую-нибудь папку на целевом ПК;
3. Запустить под правами администратора файл «Зарегистрировать компоненту [CleverenceRFID.bat](#)» (это файл со следующим содержимым):

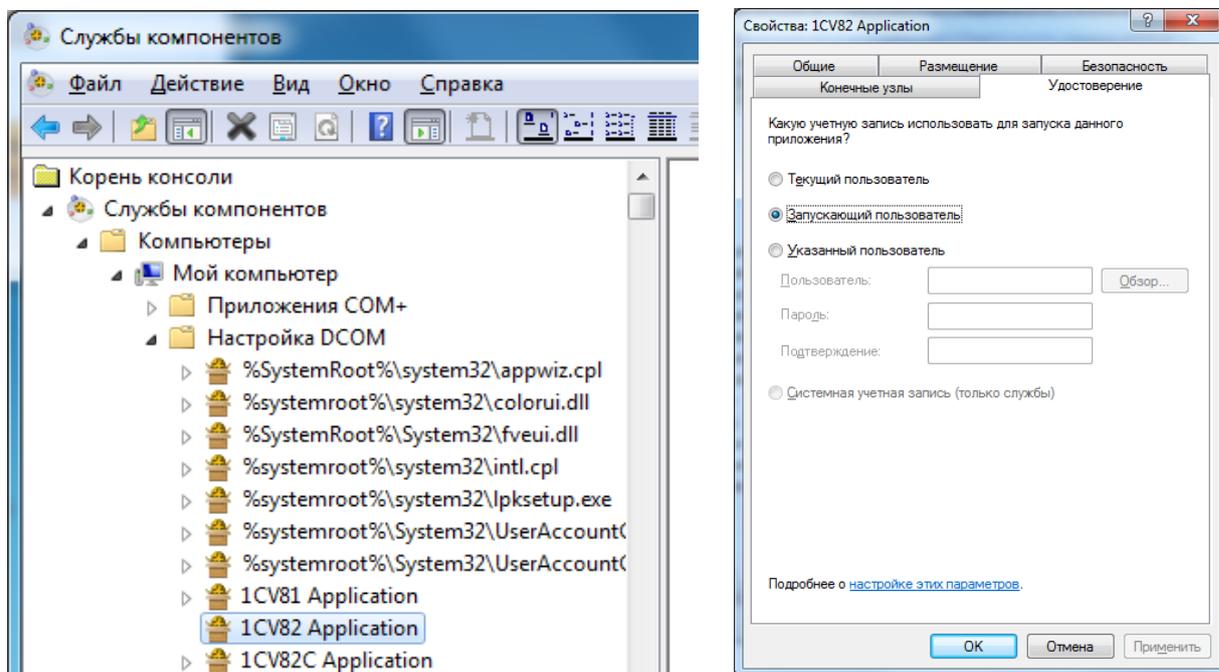
```
%windir%\Microsoft.NET\Framework\v2.0.50727\RegAsm.exe %0..\Cleverence.RFID.dll /codebase  
%windir%\Microsoft.NET\Framework\v2.0.50727\RegAsm.exe %0..\CleverenceRFID.dll /u  
%windir%\Microsoft.NET\Framework\v2.0.50727\RegAsm.exe %0..\CleverenceRFID.dll /codebase /tlb: CleverenceRFID.tlb  
pause
```

в окне командной строки .bat-файла должно появиться "Types Registered Successfully" либо "Типы зарегистрированы успешно".

ПРИМЕЧАНИЕ: если метод «Подключить Внешнюю Компоненту» в 1С (см. ниже «Начало работы с компонентой») выдает ошибку или возвращает Ложь, то следует внимательнее присмотреться к сообщениям, которые выдаются в консоль при выполнении .bat-файла.

Дополнительная настройка компоненты

На операционных системах версий «Home» (Windows 7 Home, Windows 7 Home Premium, Windows XP Home и т.п.) настройки политики безопасности по умолчанию не позволяют компоненте посылать в «1С:Предприятие» события. Из-за этого в версиях «Home» невозможно асинхронное считывание RFID-меток. Попытаться решить эту проблему можно следующим образом:



На некоторых компьютерах узлы «1CV81 Application», «1CV82 Application» и «1CV82C Application» по неизвестной причине могут отсутствовать – в этом случае попробуйте переустановить 1С, либо используйте компоненту на другом компьютере.

Обновление компоненты переустановкой

Для обновления компоненты при помощи установщика достаточно сделать следующее:

1. Зайти панель управления Windows в раздел «Программы и компоненты» (для Windows 7 и Vista) либо «Установка / удаление программ» (для XP);
2. Найти в списке установленных программ запись «Компонента CleverenceRFID для «1С:Предприятия» и нажать «Удалить»;
3. Скачать с сайта www.cleverence.ru новый установщик компоненты по ссылке <http://www.cleverence.ru/downloads/CleverenceRFID.msi> в любую папку на своём ПК;
4. Запустить установщик [CleverenceRFID.msi](#) на том компьютере, на котором проводится обновление, и следовать указаниям мастера установки.

Обновление компоненты копированием файлов

Для обновления компоненты вручную путем переписывания файлов достаточно сделать следующее:

1. Скачать с сайта www.cleverence.ru или взять с другого компьютера новые экземпляры файлов компоненты [Cleverence.RFID.dll](#) (основная библиотека RFID), [Cleverence.RFID.Motorola.dll](#) (считыватели Motorola), [Addin.CleverenceRFID.dll](#) (сама внешняя компонента для

- «1С:Предприятия»), [Symbol.RFID3.Host.dll](#) (библиотека API от Motorola) и [RFIDAPI32PC.dll](#) (библиотека LLRP) в любую папку на своём ПК;
2. Закрывать все окна «1С:Предприятия», которые работали или работают с компонентой на том компьютере, на котором проводится обновление;
 3. Скопировать новые файлы компоненты [Cleverence.RFID.Motorola.dll](#), [Addin.CleverenceRFID.dll](#) и [Symbol.RFID3.Host.dll](#) поверх старых в папке установки компоненты. Скопировать [RFIDAPI32PC.dll](#) поверх старого в папке Windows\system32. Запустить из папки установки компоненты файл [Register.bat](#) под правами администратора.

Лицензирование компоненты

Лицензии на компоненту приобретаются отдельно для каждого конкретного экземпляра считывателя. Например, если у вас 5 (пять) считывателей модели Motorola FX7400, которые вы хотите использовать из компоненты, то вам следует приобрести 5 (пять) лицензий на компоненту CleverenceRFID. Подробнее о процедуре получения лицензии см. раздел «**Получение лицензии на компоненту**».

Установка и настройка RFID считывателей

Если у вас нет под рукой RFID-считывателя – не беда. Компонента позволяет работать в «виртуальном режиме», имитируя считывания и записи несуществующих меток несуществующими считывателями. Пользуясь «виртуальным режимом» можно написать и отладить логику программы без необходимости иметь дело с реальным оборудованием.

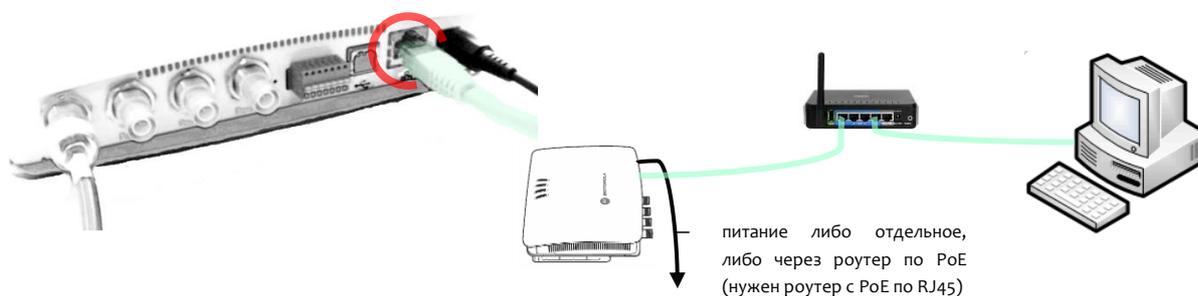
Перед началом работы с RFID считывателем, необходимо произвести его предварительную настройку. В зависимости от модели настройки будут отличаться.

Установка и настройка Motorola FX7400

Считыватель Motorola FX7400 способен работать по сети в двух конфигурациях:

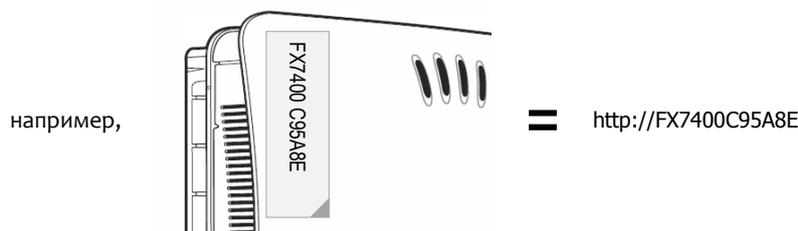
1. через разъем Ethernet (RJ45), путем подключения через роутер либо напрямую к другому ПК;
2. через разъем mini-USB типа A, путем подключения к ПК через драйвер виртуальной сети RNDIS.

Подключение Motorola FX7400 через витую пару (кабель Ethernet, разъем RJ45)



В этой конфигурации считыватель доступен по сети либо:

- a) по IP, который ему должен выдать сам роутер или DHCP сервер, либо
- b) по сетевому имени с наклейки на крышке считывателя (для сетей с DHCP):



Не зная IP или сетевого имени невозможно будет подключиться к считывателю и настроить его. Узнать IP считывателя можно путем поиска считывателей при помощи обработки CleverenceRFID. К сожалению, текущая версия поиска работает только внутри небольших сетей из 10-20 компьютеров и в подсетях 255.255.255.*.

Подключение Motorola FX7400 через кабель USB

Для подключения потребуется кабель с разъемами USB-A («прямоугольник») на mini-USB-A («квадратик»).

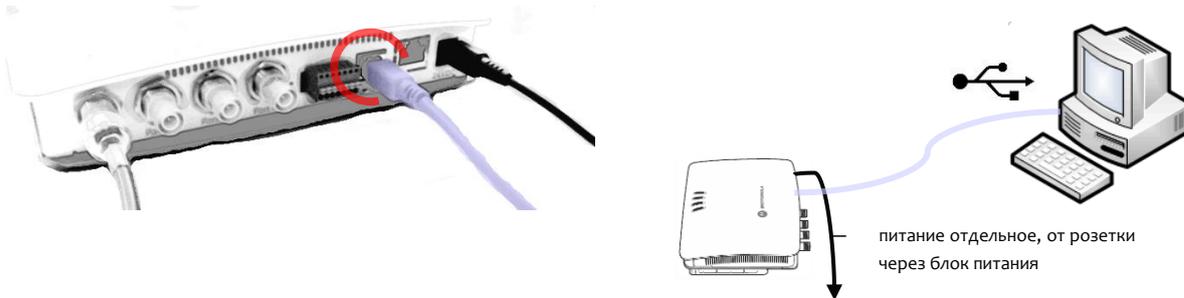
ВНИМАНИЕ!!! Перед подключением кабеля USB следует скачать и установить драйвер виртуальной сети по одной из следующих ссылок:

а) самый новый с сайта Моторола:

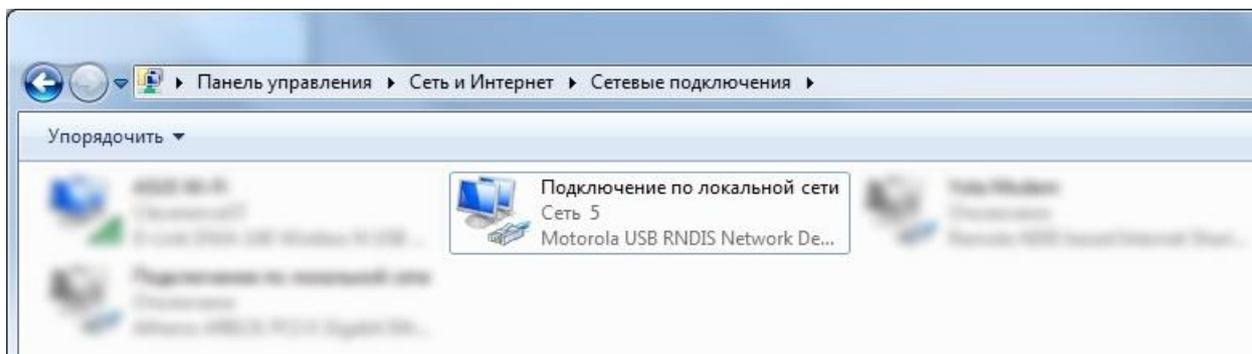
http://support.symbol.com/support/searchEntry.do?rwTarget=%2FrfrPlayerWidget.do&searchMode=GuidedSearch&searchString=rndis&document=DT_SOFTWARE_1_1&cmd=search&contextType=gs (найти, распаковать архив и запустить файл «Motorola RNDIS.msi»)

б) версию 5.02.03 платформа 2205 с сайта Клеверенс:

http://www.cleverence.ru/downloads/rfid/Motorola_RNDIS.msi

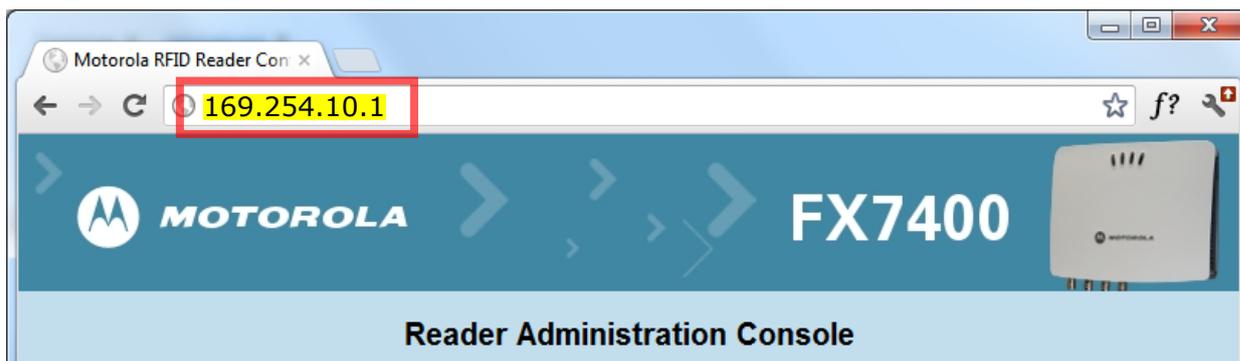


После установки драйвера виртуальной сети и подключения кабеля USB в системе должно появиться новое сетевое подключение с адаптером «Motorola USB RNDIS Network Device»:



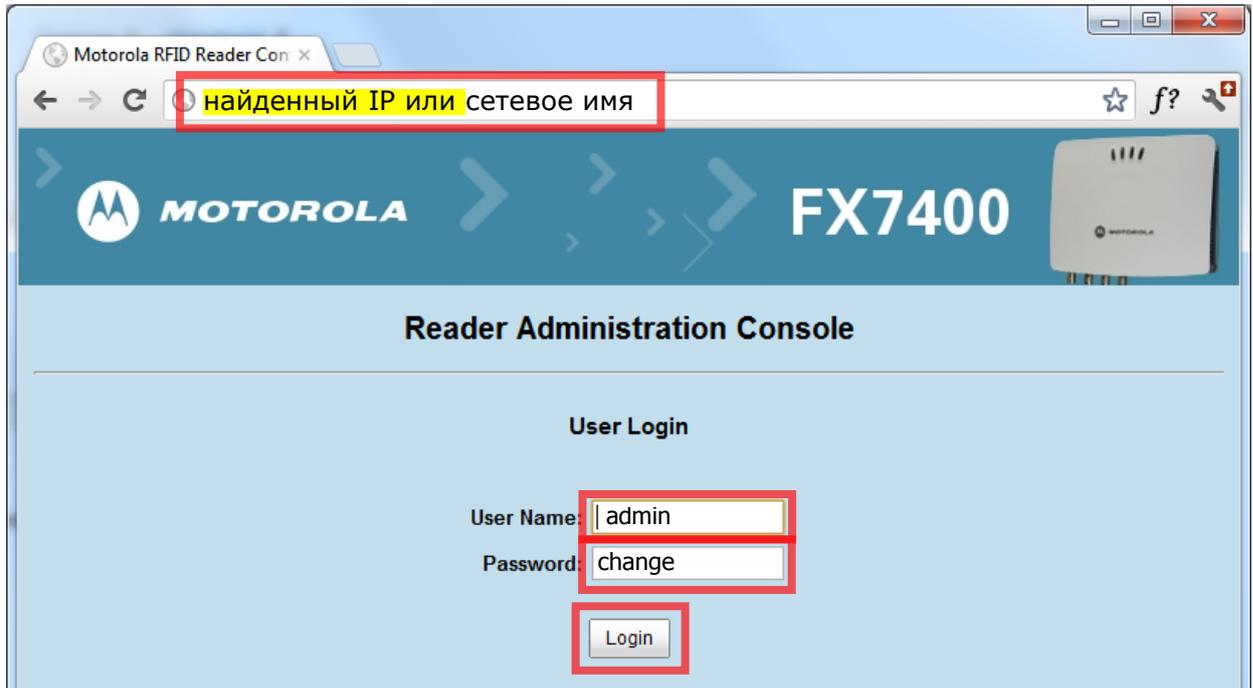
При отключении кабеля USB соединение исчезает из списка. При подключении появляется снова (если не появилось – нажмите F5, чтобы обновить список).

В новой виртуальной сети IP считывателя всегда будет равен «169.254.10.1», его следует ввести в адресной строке вашего браузера:



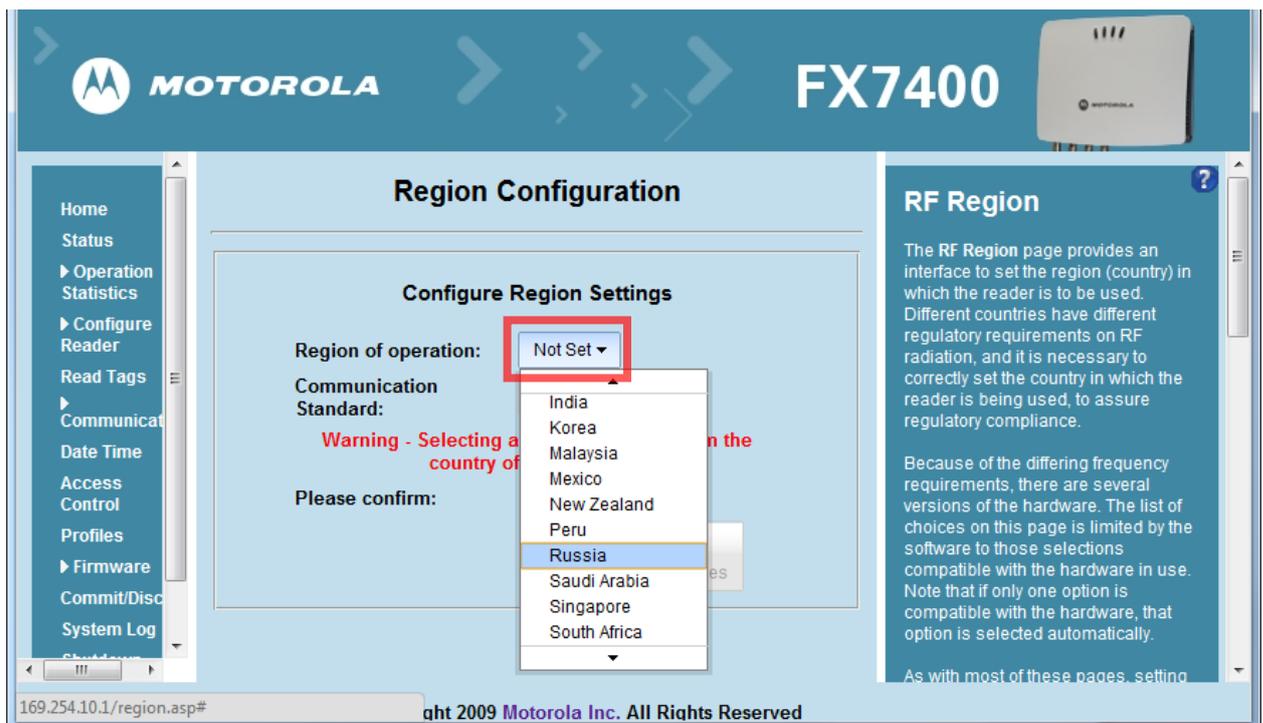
Предварительная настройка Motorola FX7400

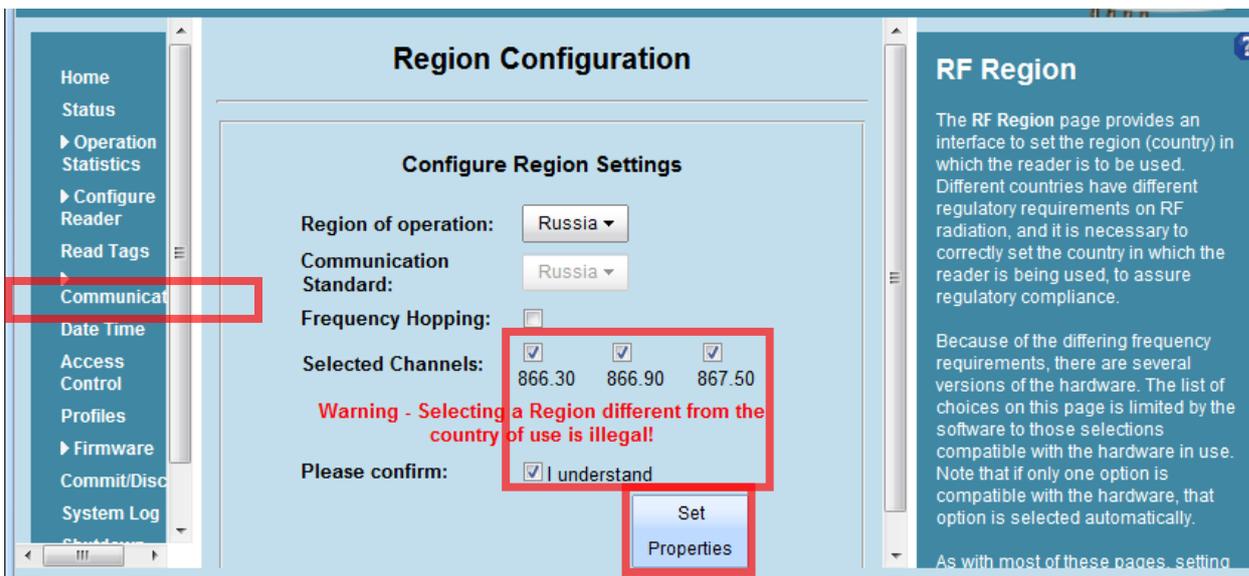
Перед началом работы со считывателем следует залогиниться в административную панель считывателя, используя Интернет-браузер, и произвести настройку региона и диапазона используемых частот:



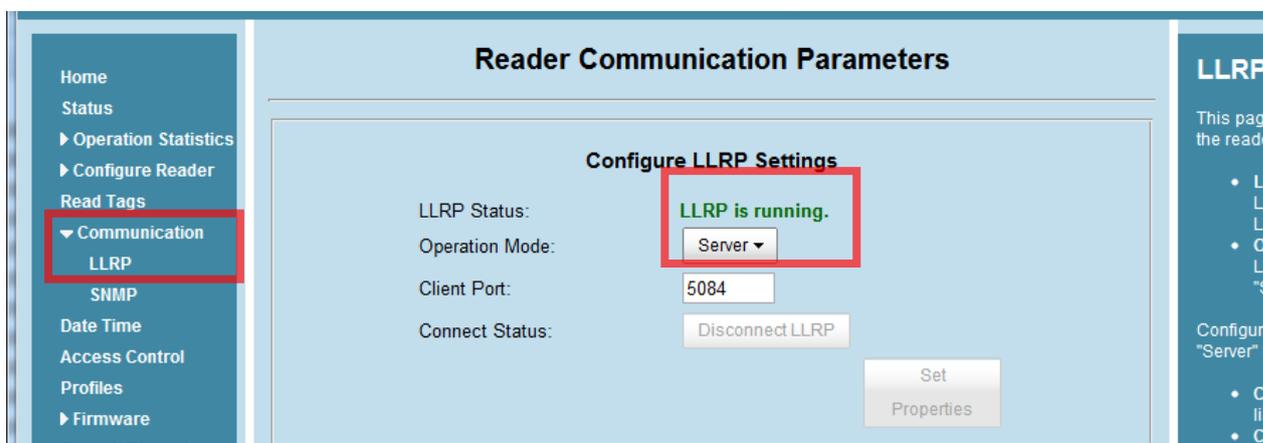
логин и пароль по умолчанию: «admin»/«change» (слово «change» как бы намекает на то, что пароль следует сменить).

Далее следует выставить регион «Russia» и соответствующие частоты:

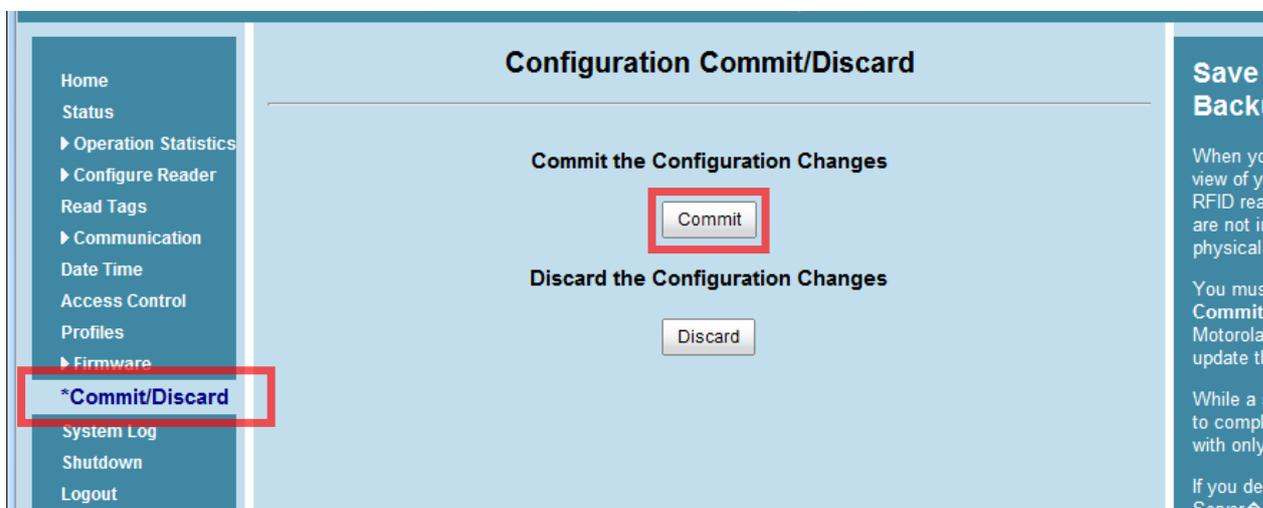




Следует проверить, что LLRP включен (если не включен – включить «Enable LLRP»):

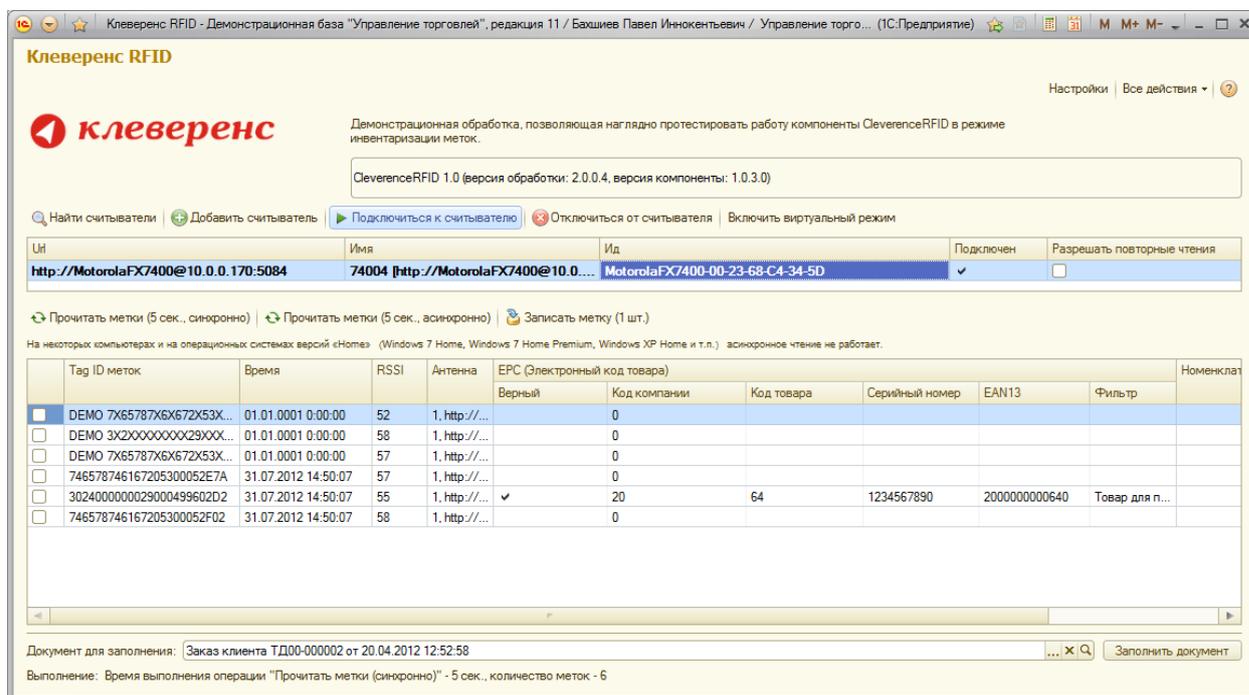


После всех шагов идем в «Commit/Discard» и нажимаем «Commit» (применить изменения):



Демонстрационная обработка

В поставку компоненты входит демонстрационная обработка «Демонстрационная обработка для компоненты CleverenceRFID.epf», которая позволяет оценить функционал компоненты по чтению и записи меток.



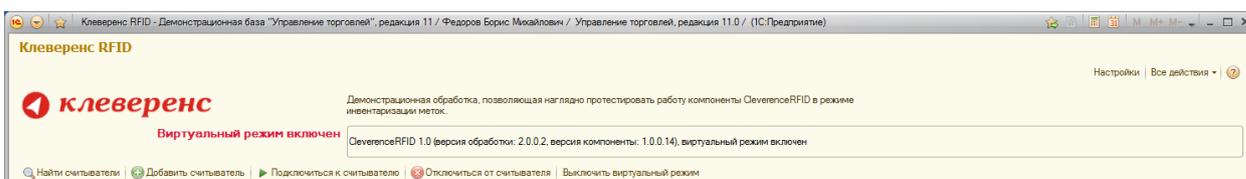
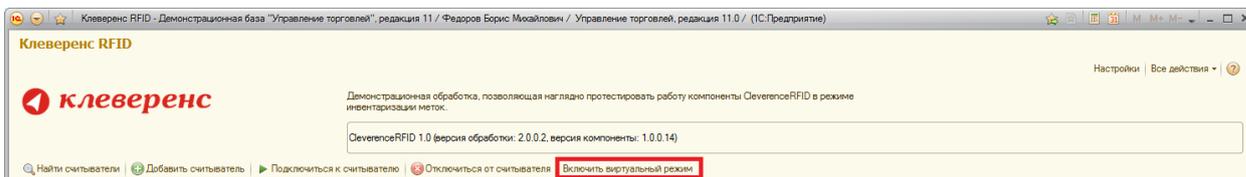
При помощи демонстрационной обработки можно:

1. Поискать RFID-считыватели в локальной сети и подключиться к любому из них;
2. Подключиться к фиктивному «виртуальному» RFID-считывателю, если под рукой нет настоящего;
3. Записать в RFID-метку данные о товаре из базы «1С:Предприятия» (только в «Управлении торговлей 11»);
4. Прочитать RFID-метки, найти в базе «1С:Предприятия» соответствующие им товары (только в «Управлении торговлей 11»);
5. Перенести данные о считанных метках в любой складской документ «1С:Предприятия» (только в «Управлении торговлей 11»).

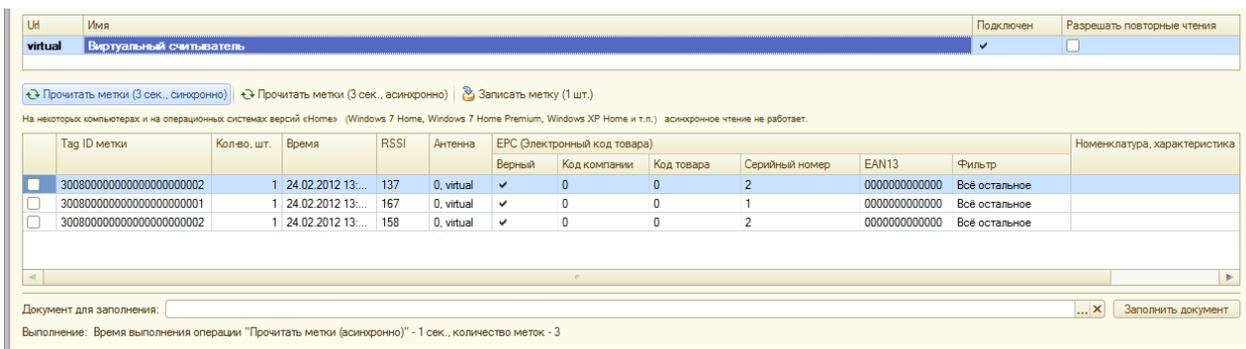
Как видно из перечня, полноценное функционирование обработки возможно только **в конфигурации «Управление торговлей 11»**.

Тестирование компоненты без RFID-считывателей

Для тестирования работы компоненты **без RFID-считывателя** на руках, в ней предусмотрен так называемый «виртуальный режим», в котором компонента подключается к виртуальным считывателям и читает виртуальные метки. Чтобы протестировать работу компоненты без считывателей, в демонстрационной обработке предусмотрена кнопка «Включить виртуальный режим»:

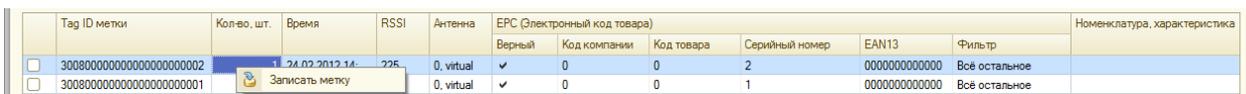


В виртуальном режиме все RFID-считыватели в окне демообработки являются фиктивными и компонента на самом деле ни к одному из них не подключается. Все метки, которые будут якобы читаться компонентой (пока она находится в виртуальном режиме) тоже фиктивные:



В стандартной настройке демонстрационной обработки виртуальная инвентаризация читает 2 (две) конкретные метки всегда + иногда еще от 0 (ноля) до 2 (двух) случайных меток.

В фиктивные метки, прочитанные в виртуальном режиме, даже можно писать (только в «Управлении торговлей 11»):



При этом писаться будут только те две конкретные метки, а в дополнительные случайные метки писать не имеет смысла, т.к. они генерируются случайно и читаются только один раз. Подробнее о записи см. раздел «Тестирование записи».

Тестирование инвентаризации

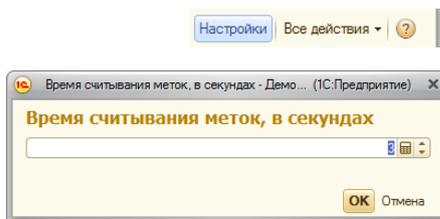
Демонстрационная обработка позволяет провести инвентаризацию RFID-меток, находящихся в поле зрения антенн подключенных RFID-считывателей.



При синхронном чтении «1С:Предприятие» замирает на время чтения, после чего считанные метки появляются сразу все.

При асинхронном чтении «1С:Предприятие» не замирает, а метки появляются одна за другой по мере их считывания. Это более удобный и интересный режим, но он не везде работает (см. «Дополнительная настройка компоненты»).

Время чтения (сколько времени читать метки) можно задать в настройках обработки:



Считанные метки отображаются в таблице, их EPC декодируются:

Tag ID метки	Кол-во, шт.	Время	RSSI	Антенна	EPC (Электронный код товара)				Фильтр	Номенклатура, характеристика	
					Верный	Код компании	Код товара	Серийный номер			
<input type="checkbox"/> DEMO 3X2XXXXXXXXXXXXB369 [в отсутствие лицензии для считывателя '00:23:68:C4:34:5D' код метки был изменен]					0				[DEMO]		
<input checked="" type="checkbox"/> 30240000000470000013483	1	24.02.2012 16:...	36	2, 10.11...	✓	2	28	78979	200000000282	Товар для продажи...	Кроссовки мужские, кожа
<input type="checkbox"/> DEMO 3X2XXXXXXXXXXXXB369 [в отсут...	1	01.01.0001 0.0...	27	2, 10.11...		0	0		[DEMO]		
<input type="checkbox"/> DEMO XXX38XXXXXXXXXXXX [в отсут...	1	01.01.0001 0.0...	71	2, 10.11...		0	0		[DEMO]		
<input checked="" type="checkbox"/> 30240000000490000000041	1	24.02.2012 16:...	39	2, 10.11...	✓	2	36	65	200000000367	Товар для продажи...	Ботинки мужские
<input checked="" type="checkbox"/> 30240000000478000003039	1	24.02.2012 16:...	65	2, 10.11...	✓	2	30	12345	200000000305	Товар для продажи...	Кроссовки "REEBOK"
<input checked="" type="checkbox"/> 30240000000444000089D42	1	24.02.2012 16:...	27	2, 10.11...	✓	2	17	564546	200000000176	Товар для продажи...	Мужские мокасины
<input type="checkbox"/> DEMO 3X2XXXXXXXXXXXXB369 [в отсут...	1	01.01.0001 0.0...	33	2, 10.11...		0	0		[DEMO]		

В отсутствие лицензии на компоненту для того конкретного считывателя, которым прочитана та или иная конкретная метка, Tag ID метки (время от времени в случайном порядке) может быть изменен на строку «DEMO ...» и текст об отсутствии лицензии.

В приведенной таблице колонка «Количество» отображает сколько меток с идентичным Tag ID было прочитано. Колонка «RSSI» – условный уровень сигнала от метки по шкале от 0 до 100. В колонке «Антенна» показан номер антенны считывателя (от 1 до) и url самого считывателя (если читать сразу с нескольких считывателей).

Более детально по составу колонок EPC см. раздел «Понятие электронного кода продукта (EPC)».

В колонке EAN13 показан синтетический штрихкод, сгенерированный по данным EPC метки.

Колонка «Номенклатура» видна только в том случае, если в конфигурации (в которой открыта демонстрационная обработка) присутствует справочник номенклатуры. Товары ищутся либо по коду товара из EPC, либо по синтетическому штрихкоду EAN13 (см. подробнее в разделе «Тестирование записи»).

Если демонстрационная обработка открыта в конфигурации «Управлении торговлей 11», то в нижней части окна обработки отображается раздел, позволяющий использовать данные о считанных метках для заполнения таблицы товаров какого-нибудь документа 1С:



В документ будут переноситься те строки из таблицы считанных меток, которые отмечены галочкой. По умолчанию обработка отмечает галочкой все метки, по которым было найдено соответствие какому-либо товару базы «1С:Предприятия».

Тестирование записи

Если в конфигурации (в которой открыта демонстрационная обработка) присутствует справочник номенклатуры, демонстрационная обработка позволяет записать в метки данные о товарах из базы «1С:Предприятия»:

Tag ID метки	Кол-во, шт.	Время	RSSI	Антенна	EPC (Электронный код товара)				Фильтр	Номенклатура, характеристика	
					Верный	Код компании	Код товара	Серийный номер			
<input checked="" type="checkbox"/> 30240000000470000013483	1	24.02.2012 13:...	36	2, 10.11...	✓	2	28	78979	200000000282	Товар для прода...	Кроссовки мужские, кожа
<input checked="" type="checkbox"/> 302400000004640000B369	1	24.02.2012 13:...	28	2, 10.11...	✓	2	25	857857	2000000002...	Товар для прода...	Мужские туфли nubuk
<input type="checkbox"/> DEMO 3X2XXXXXXXXXXXXB369 [в от...	1	01.01.0001 0.0...	33	2, 10.11...		0	0		[DEMO]		

В отсутствие лицензии на компоненту для конкретного считывателя, которым прочитана интересующая метка, Tag ID метки может быть заменен на строку «DEMO ...» и текст об отсутствии лицензии. Метки с текстом DEMO недоступны для записи. Однако это не значит, что такую метку совсем нельзя записать – метки заменяются на DEMO в случайном порядке, поэтому можно попытаться еще раз нажать одну из кнопок «Чтение меток...», до тех пор пока интересующая метка не будет нормально прочитана.

Запись происходит упрощенно по следующему алгоритму:

1. Обработка генерирует EPC либо на основе штрихкода EAN13, либо по числовому коду 1С товара из базы «1С:Предприятия»;
2. Сгенерированный EPC записывается в банк 01 (EPC-bank) выбранной RFID-метки.

Но если более детально, то запись происходит по следующему алгоритму:

1. Обработка генерирует EPC либо на основе штрихкода EAN13, либо по числовому коду 1С товара из базы «1С:Предприятия»;
2. У выбранной метки запрашивается содержимое банка TID;
3. Сгенерированный EPC записывается в банк 01 (EPC-bank) RFID-метки, у которой банки EPC и TID равны требуемым.

Все эти шаги делаются для того, чтобы записать ровно одну конкретную метку, а не все метки с выбранным EPC. Подробнее см. раздел «Запись банков памяти для меток Class 1 Gen 2».

Сразу после записи обработка выполняет повторное чтение меток, чтобы можно было увидеть результат записи.

Работа с компонентой

Начало работы с компонентой

Полная документация по объектам компоненты (все поля, процедуры и функции) входит в дистрибутив компоненты и ставится вместе с компонентой. Самая последняя версия документации всегда доступна по ссылке: <http://www.cleverence.ru/downloads/CleverenceRFID.pdf>.

Для работы с компонентой следует создать **один** её экземпляр на каждом клиенте (который будет работать с RFID-читывателями со своей локальной машины):

Модуль управляемого приложения:

&НаКлиенте

Перем КлеверенсRFID Экспорт;

Процедура ПриНачалеРаботыСистемы()

Попытка

удачно = ПодключитьВнешнююКомпоненту("AddIn.CleverenceRFID");

КлеверенсRFID = **новый** СОМОбъект("AddIn.CleverenceRFID");

Исключение

Предупреждение("Не удалось подключить компоненту AddIn.CleverenceRFID. Возможно, она не зарегистрирована в системе", 20, "Ошибка подключения компоненты");

Возврат;

КонецПопытки;

КлеверенсRFID.Язык = "Русский";

КонецПроцедуры

Завершение работы с компонентой

Процедура правильного завершения работы с компонентой (при закрытии окна 1С или остановке сервера) должна выглядеть следующим образом:

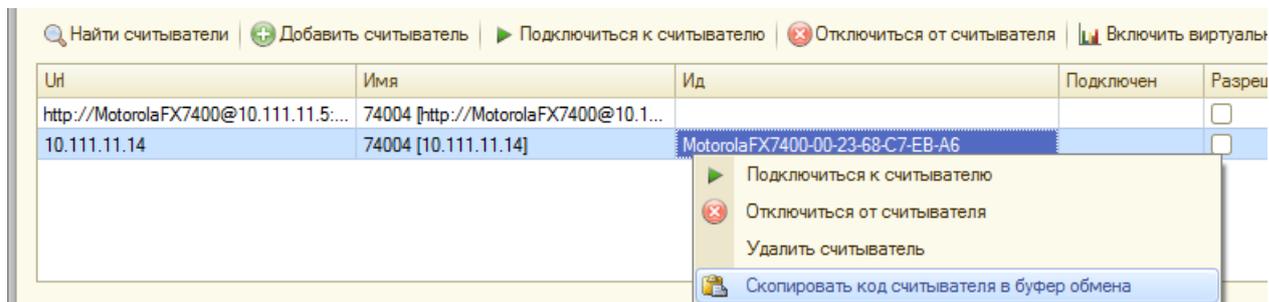
Модуль управляемого приложения:

КлеверенсRFID.ЗавершитьРаботуСКомпонентой();

КлеверенсRFID = **Неопределено**;

Получение лицензии на компоненту

Лицензии на компоненту выдаются отдельно для каждого считывателя в соответствии с его уникальным идентификатором. Чтобы получить лицензию на компоненту для конкретного считывателя, следует в демонстрационной обработке (см. «Демонстрационная обработка») кликнуть правой клавишей мыши на строке со считывателем и выбрать пункт контекстного меню «Скопировать код считывателя в буфер обмена»:



Полученный код (например, «MotorolaFX7400-00-23-68-C7-EB-A6») следует прислать менеджеру, у которого вы приобрели компоненту.

В ответ вы получите файл защиты лицензии с длинным названием, похожим на «license_1C_CleverenceRFID_MotorolaFX7400-00-23-68-C7-EB-A6_(30.07.2012_18-31) (МояФирма, счет).xml».

Этот файл следует просто положить в папку установки компоненты (например, «C:\Program Files\Cleverence Soft\CleverenceRFID») и лицензия тут же должна примениться, 1С перезапускать не нужно.

Если по какой-то причине лицензия не работает, следует сначала посмотреть содержимое лога ошибок (см. «Ошибки в работе с компонентой») или написать письмо на адрес support@cleverence.ru, вложив в него соответствующий файл лицензии.

Виртуальный режим работы компоненты

Для тестирования работы компоненты **без RFID-считывателя** на руках, в ней предусмотрен так называемый «виртуальный режим», в котором компонента подключается к виртуальным считывателям и читает виртуальные метки. «Виртуальный» в данном случае означает «отсутствующий на самом деле».

Для активации виртуального режима используется следующий код:

Модуль управляемого приложения:

```
// Создать и один раз заполнить глобальную переменную с экземпляром компоненты  
КлеверенсRFID = новый СОМОбъект("AddIn.CleverenceRFID");
```

любой модуль:

```
КлеверенсRFID.ВиртуальныйРежим.Включен = Истина;
```

Настройки виртуального режима позволяют задать параметры работы несуществующих считывателей так, чтобы они удовлетворяли условиям проводимых тестов.

Пример №1 | чтение всегда ровно 6-ти случайных меток

В такой настройке компонента сгенерирует шесть случайных меток и будет их «читать».

любой модуль:

```
КлеверенсRFID.ВиртуальныйРежим.ЧислоМетокМин = 6;  
КлеверенсRFID.ВиртуальныйРежим.ЧислоМетокМакс = 0;  
КлеверенсRFID.ВиртуальныйРежим.ТестовыеМетки.Очистить();
```

Пример №2 | чтение от 6-ти до 10-ти (раз на раз не приходится) случайных меток

В такой настройке компонента будет от инвентаризации к инвентаризации генерировать от шести до десяти случайных меток.

любой модуль:

```
КлеверенсRFID.ВиртуальныйРежим.ЧислоМетокМин = 6;  
КлеверенсRFID.ВиртуальныйРежим.ЧислоМетокМакс = 10;  
КлеверенсRFID.ВиртуальныйРежим.ТестовыеМетки.Очистить();
```

Пример №3 | чтение двух заранее заданных меток

В такой настройке компонента всегда будет «читать» только две указанные метки.

любой модуль:

```
КлеверенсRFID.ВиртуальныйРежим.ЧислоМетокМин = 2;  
КлеверенсRFID.ВиртуальныйРежим.ЧислоМетокМакс = 2;  
КлеверенсRFID.ВиртуальныйРежим.ТестовыеМетки.Очистить();  
КлеверенсRFID.ВиртуальныйРежим.ТестовыеМетки.Добавить("30080000000000000001");  
КлеверенсRFID.ВиртуальныйРежим.ТестовыеМетки.Добавить("30080000000000000002");
```

Пример №4 | чтение двух заранее заданных и одной-двух случайных меток

В такой настройке компонента от инвентаризации к инвентаризации будет «читать» либо две указанные метки + одна случайная, либо две указанные + две случайных.

любой модуль:

```
КлеверенсРФИД.ВиртуальныйРежим.ЧислоМетокМин = 3;  
КлеверенсРФИД.ВиртуальныйРежим.ЧислоМетокМакс = 4;  
КлеверенсРФИД.ВиртуальныйРежим.ТестовыеМетки.Очистить();  
КлеверенсРФИД.ВиртуальныйРежим.ТестовыеМетки.Добавить("3008000000000000000001");  
КлеверенсРФИД.ВиртуальныйРежим.ТестовыеМетки.Добавить("3008000000000000000002");
```

Пример №5 | чтение трех заранее заданных и нескольких случайных меток

В такой настройке компонента от инвентаризации к инвентаризации будет генерировать от нуля до семи случайных меток и «читать» их наряду с тремя заранее заданными.

любой модуль:

```
КлеверенсРФИД.ВиртуальныйРежим.ЧислоМетокМин = 3;  
КлеверенсРФИД.ВиртуальныйРежим.ЧислоМетокМакс = 10;  
КлеверенсРФИД.ВиртуальныйРежим.ТестовыеМетки.Очистить();  
  
// создаем метку по Tag ID.  
tagid1 = "300800000000000000000000";  
метка1 = КлеверенсРФИД.НоваяМетка(tagid1);  
// Атрибут «Счетчик» означает число меток с идентичным ЕРС. Если Счетчик = 2, то при инвентаризации были  
// обнаружены две метки с идентичным ЕРС. В реальной инвентаризации вместо того, чтобы возвратить две  
// одинаковые метки, библиотека объединяет их в одну, и проставляет счетчик = 2.  
метка1.Счетчик = 2;  
КлеверенсРФИД.ВиртуальныйРежим.ТестовыеМетки.Добавить(метка1);  
  
// создаем ЕРС единицы товара с серийным номером «4412», кодом товара «123» от фирмы с кодом «7770».  
// первый ноль означает, что это ЕРС товара для продажи на кассе.  
ерс = КлеверенсРФИД.ЕРСизSGTIN(0, 7770, 123, "4412");  
// создаем метку по ЕРС.  
метка2 = КлеверенсРФИД.НоваяМетка(ерс);  
КлеверенсРФИД.ВиртуальныйРежим.ТестовыеМетки.Добавить(метка2);  
  
// создаем ЕРС единицы товара с серийным номером «332», сам товар задаем по EAN13.  
// первый ноль означает, что это ЕРС товара для продажи на кассе.  
ерс = КлеверенсРФИД.ЕРСизEAN13(0, "4004764390793", "332");  
// создаем метку по ЕРС.  
метка3 = КлеверенсРФИД.НоваяМетка(ерс);  
КлеверенсРФИД.ВиртуальныйРежим.ТестовыеМетки.Добавить(метка3);
```

Пример №6 | сначала какое-то время читается только одна метка, затем только другая

В некоторых ситуациях для тестирования алгоритмов учета может понадобиться управлять сценарием считывания меток. Например, чтобы сразу после запуска инвентаризации читались какие-то одни определенные метки, а спустя пару секунд – другие определенные метки. В приведенной ниже настройке от инвентаризации к инвентаризации компонента будет воспроизводить один и тот же сценарий: сначала «читается» метка "300800000000000000000001", затем она исчезает и начинает «читаться» метка "300800000000000000000002".

любой модуль:

```
КлеверенсРФИД.ВиртуальныйРежим.ЧислоМетокМин = 2;  
КлеверенсРФИД.ВиртуальныйРежим.ЧислоМетокМакс = 2;  
  
ТестовыеМетки = КлеверенсРФИД.ВиртуальныйРежим.ТестовыеМетки;  
  
// добавляем метку по Tag ID. Метка начинает читаться спустя примерно 1 сек. и видна примерно 5 сек.  
ТестовыеМетки.ДобавитьПоВремени("300800000000000000000001", 1, 5);  
// добавляем метку по Tag ID. Метка начинает читаться на 8й сек. и видна примерно 2 сек.  
ТестовыеМетки.ДобавитьПоВремени("300800000000000000000002", 8, 2);
```

Внешние события компоненты

По мере работы компоненты в predetermined procedure «ОбработкаВнешнегоСобытия» основного модуля «1С:Предприятия», а также в procedure «ВнешнееСобытие» формы приходят события.

Событие «НайденСчитыватель»

При **асинхронном** поиске новых считывателей в локальной подсети, компонента посылает внешнее событие «НайденСчитыватель».

```
Источник = "CleverenceRFID"
Событие = "НайденСчитыватель"
Данные = Url найденного считывателя, например «http://XR480@10.10.0.17».
Подключиться к найденному считывателю по полученному url можно позднее,
используя метод компоненты «ПодключитьСчитыватель» (см. «Поиск и подключение
RFID-считывателей»).
```

Пример кода обработки события:

Модуль управляемого приложения:

```
Процедура ОбработкаВнешнегоСобытия(Источник, Событие, Данные) // Предопределенная процедура 1С
// Глобальный обработчик внешнего события
Если Источник = "CleverenceRFID" И Событие = "НайденСчитыватель" Тогда
// Сообщить url найденного RFID-считывателя:
Сообщить("Найден считыватель: " + Данные);
КонецЕсли;
КонецПроцедуры
```

Или, если подписать форму на событие «ВнешнееСобытие»:

ВнешнееСобытие

ВнешнееСобытие

Модуль формы:

```
Процедура ВнешнееСобытие(Источник, Событие, Данные)
Если Источник = "CleverenceRFID" И Событие = "НайденСчитыватель" Тогда
// Сообщить url найденного RFID-считывателя:
Сообщить("Найден считыватель: " + Данные);
КонецЕсли;
КонецПроцедуры
```

Событие «Чтение»

При каждом удачном **асинхронном** чтении RFID-метки (в частности, при **асинхронной** инвентаризации) компонента посылает внешнее событие «Чтение».

```
Источник = "CleverenceRFID"
Событие = "Чтение"
Данные = Tag ID прочитанной метки, например «303000181CE257587E9CA77C».
Более подробную информацию о самой метке можно получить у конкретного
считывателя или у самой компоненты через метод «ВыбратьМетку».
```

В качестве данных в событие приходит только Tag ID метки. Получить более подробные данные можно при помощи метода компоненты «ВыбратьМетку», который принимает Tag ID и возвращает объект компоненты с описанием метки и следующими реквизитами:

Реквизиты объекта компоненты RFID-метка («Cleverence.RFID.RfidTag»)		
Имя реквизита	Имя реквизита англ.	Описание
TagId	TagId	Возвращает Tag ID метки 16-ричным представлении (строка в 24 символа).
Считыватель	Reader	Возвращает считыватель, при помощи которого была считана данная метка.
EPC	EPC	Возвращает значение EPC метки на основании операции инвентаризации окружающих меток RFID-считывателем.
НомерАнтенны	Antennald	Возвращает номер (код) антенны, которая прочла метку с таким Tag ID.
Время	FirstTimeSeen	Возвращает дату/время, в которое метка с таким Tag ID была увидена впервые (по часам компьютера, на котором работает компонента).
Счетчик	SeenCount	Возвращает сколько раз была замечена метка с таким Tag ID. Фактически, для неподвижно лежащих меток это число отражает количество меток с разным номером чипа (TID), но одинаковым Tag ID (одинаковым EPC/UII). Для движущихся меток сюда добавляется количество входов/выходов таких меток за пределы области чтения.
RSSI	PeakRSSI	Возвращает пиковое значение принятого уровня сигнала от метки в произвольных единицах от 0 до 255 (число).

Пример кода обработки события:

```

Модуль управляемого приложения:
Процедура ОбработкаВнешнегоСобытия(Источник, Событие, Данные) // Предопределенная процедура 1С
// Глобальный обработчик внешнего события
Если Источник = "CleverenceRFID" И Событие = "Чтение" Тогда
    Попытка
        // Получить полные данные считанной метки (или одинаковых меток) сразу со всех считывателей:
        метка = КлеверенсRFID.ВыбратьМетку(Данные);
        // Либо получить данные у конкретного считывателя (подробнее о считывателях см. ниже)
        // метка = считыватель.ВыбратьМетку(tagid);

        Сообщить(метка.TagId + ", кол-во: " + метка.Счетчик +
            " шт., время=" + метка.Время.Строка() + ", RSSI=" + метка.RSSI);
        ...
    Исключение
        Сообщить(КлеверенсRFID.ОписаниеОшибки());
    ОкончаниеПопытки;
КонецЕсли;
КонецПроцедуры
    
```

либо, если подписать форму на событие «ВнешнееСобытие»:

```
ВнешнееСобытие 
Модуль формы:
Процедура ВнешнееСобытие(Источник, Событие, Данные)
Если Источник = "CleverenceRFID" И Событие = "Чтение" Тогда
    Попытка
        // Работа с компонентой
        // Получить полные данные считанной метки (или одинаковых меток) сразу со всех считывателей:
        метка = КлеверенсRFID.ВыбратьМетку(Данные);
        // Либо получить данные у конкретного считывателя (подробнее о считывателях см. ниже)
        // метка = считыватель.ВыбратьМетку(tagid);

        Сообщить(метка.TagId + ", кол-во: " + метка.Счетчик +
            " шт., время=" + метка.Время.Строка() + ", RSSI=" + метка.RSSI);
        ...
    Исключение
        Сообщить(КлеверенсRFID.ОписаниеОшибки());
    ОкончаниеПопытки;
КонецЕсли;
КонецПроцедуры
```

Событие «Запись»

При каждой удачной **асинхронной** записи RFID-метки компонента посылает внешнее событие «Запись».

```
Источник = "CleverenceRFID"
Событие = "Запись"
Данные = Tag ID записываемой метки, например «303000181CE257587E9CA77C» (старый Tag ID, т.к. после записи в банк EPC Tag ID метки мог поменяться).
Более подробная информация недоступна, метод «ВыбратьМетку» не применим.
```

В качестве данных в событие приходит только Tag ID метки. Получить более подробные данные можно при помощи метода компоненты «ВыбратьМетку», который принимает Tag ID и возвращает объект компоненты с описанием метки (см. «Событие «Чтение»»).

Пример кода обработки события:

```
Модуль управляемого приложения:
Процедура ОбработкаВнешнегоСобытия(Источник, Событие, Данные) // Предопределенная процедура 1С
// Глобальный обработчик внешнего события
Если Источник = "CleverenceRFID" И Событие = "Запись" Тогда
    // Сообщить Tag ID записанной метки:
    Сообщить("Записана метка: " + Данные);
КонецЕсли;
КонецПроцедуры
```

Поиск и подключение RFID-считывателей

Возможности компоненты позволяют производить поиск RFID-считывателей в локальной подсети (т.е. в диапазонах IP-адресов «192.168.0.1 – 192.168.248.255», «172.16.0.1 - 172.16.240.255» и «10.0.0.1 - 10.255.255.255»). К сожалению, текущая версия поиска работает только внутри небольших сетей из 5-20 компьютеров и в подсетях 255.255.255.* (т.е. если у вас задана слишком широкая подсеть, то поиск скорее всего не работает).

По физическому подключению и настройке RFID-считывателей см. «Установка и настройка RFID считывателей».

Синхронный поиск считывателей

При синхронном поиске окна «1С:Предприятия» замирают на время выполнения процедуры «НайтиСчитыватели» компоненты (примерно 20-30 сек).

Примечание: во время синхронного поиска событие «НайденСчитыватель» не приходит, т.к. такое событие «подвесило» бы 1С.

Пример кода синхронного поиска считывателей (перед этим необходимо начать работу с компонентой, см «Начало работы с компонентой»):

любой модуль:

```
считыватели = КлеверенсRFID.НайтиСчитыватели();
Для индекс = 0 по считыватели.Количество - 1 Цикл
    считыватель = считыватели.Элемент(индекс);
    // Сообщить url найденного RFID-считывателя:
    Сообщить("Найден считыватель: " + считыватель.Url);
КонецЦикла;
```

Асинхронный поиск считывателей

При асинхронном поиске окна «1С:Предприятия» не замирают, т.к. поиск выполняется в фоне. По мере нахождения новых считывателей, компонента посылает внешнее событие «НайденСчитыватель», которое можно обработать в главном модуле.

любой модуль:

```
КлеверенсRFID.НачатьПоискСчитывателей();
```

И далее либо

Модуль управляемого приложения:

```
Процедура ОбработкаВнешнегоСобытия(Источник, Событие, Данные) // Предопределенная процедура 1С
    Если Источник = "CleverenceRFID" И Событие = "НайденСчитыватель" Тогда
        // Сообщить url найденного RFID-считывателя:
        Сообщить("Найден считыватель: " + Данные);
    КонецЕсли;
КонецПроцедуры
```

либо в текущей открытой форме:

ВнешнееСобытие

ВнешнееСобытие  

Модуль формы:

```
&НаКлиенте
Процедура ВнешнееСобытие(Источник, Событие, Данные)
    Если Источник = "CleverenceRFID" И Событие = "НайденСчитыватель" Тогда
        // Передать url найденного RFID-считывателя на добавление в список:
        ПриНахожденииСчитывателя(Данные); // Какая-то процедура обработки найденного считывателя
    КонецЕсли;
КонецПроцедуры
```

Ошибки в работе с компонентой

Если метод «ОбработкаВнешнегоСобытия» в 1С выдает ошибку или возвращает Ложь, то следует внимательнее присмотреться к сообщениям, которые выдаются в консоль при регистрации компоненты

при помощи .bat-файла. Также убедитесь, что ваш компьютер удовлетворяет требованиям, перечисленным в разделе «Установка компоненты».

Если в работе компоненты возникают ошибки (исключительные ситуации), то «1С:Предприятие» в основном интерпретирует их как «Неизвестная ошибка». Чтобы получить более подробную информацию по ошибке, следует использовать метод «ОписаниеОшибки()» компоненты:

любой модуль:

```
Попытка
    // Работа с компонентой
    ...
Исключение
    Сообщить(КлеверенсRFID.ОписаниеОшибки());
ОкончаниеПопытки;
```

Кроме того все ошибки, возникающие при работе самой компоненты (т.е. те, которые возникли после успешного выполнения функции «ПодключитьВнешнююКомпоненту»), пишутся в лог-файл «CleverenceRFID_log.txt», который лежит в папке «Мои документы» того пользователя, под которым сейчас запущено «1С:Предприятие». Некоторые редкие ошибки выполнения функции «ПодключитьВнешнююКомпоненту» тоже пишутся в этот лог.

Известные ошибки в работе компоненты

В работе компоненты выявлены ошибки, которые появляются либо на некоторых конфигурациях машин, либо время от времени на любых конфигурациях.

Чтение меток асинхронно (асинхронная инвентаризация) ничего не читает

Возникает на домашних версиях операционной системы Windows, а также при неправильной настройке политики доступа (прав доступа) компоненты в «1С:Предприятие». Подробнее см. раздел «Дополнительная настройка компоненты».

«Приложение будет закрыто» и обрушение окна «1С:Предприятия»

Возникает при повторных созданиях и уничтожениях компоненты, например при повторных открытиях копий демонстрационной обработки в одном и том же окне «1С:Предприятия». Совет: создавайте не более одного экземпляра компоненты на каждого запущенного клиента 1С.

Операции с метками Class 1 Gen 2

Инвентаризации меток Class 1 Gen 2

Операция инвентаризации поддерживается на уровне радио-протокола обмена между метками и считывателем и возвращает **какие EPC присутствуют в зоне считывания и по сколько раз каждый**.

Например, все метки могут иметь один и тот же EPC/UII, и в этом случае по итогам инвентаризации мы будем знать, что это за EPC, и сколько всего RFID-меток с этим EPC/UII удалось считать ридеру.

Если все метки имеют свой уникальный EPC/UII (не путать с уникальным номером чипа, который безусловно есть у каждой метки Class 1 Gen 2), то операция инвентаризации вернет список этих EPC/UII.

Синхронная инвентаризация меток

Синхронная инвентаризация означает следующее:

1. «1С:Предприятие» дало считывателю команду «считай окружающие метки в течение N секунд» и замерло в ожидании ответа;
2. Считыватель читает метки, «1С:Предприятие» ждет, все формочки замерли. Считыватель закончил через указанное время и вернул результат «1С:Предприятию»;
3. «1С:Предприятие» получило результат, осознала его, формочки «отвисли».



Таким образом, если при синхронной инвентаризации указать считывателю «считай 50 секунд», то окно 1С почти целую минуту не будет доступно для пользователя.

Пример кода для **синхронной** инвентаризации:

Модуль формы:

```
// ----- по нажатии кнопки 1 -----  
// Опрашивать окружающие метки в течение 5000 миллисекунд (5 сек)  
метки = считыватель.ИнвентаризоватьМетки(5000);  
Для индекса = 0 по метки.Количество - 1 Цикл  
метка = метки.Элемент(индекс);  
ОбработатьМетку(метка); // Какая-то процедура обработки метки  
КонецЦикла;
```

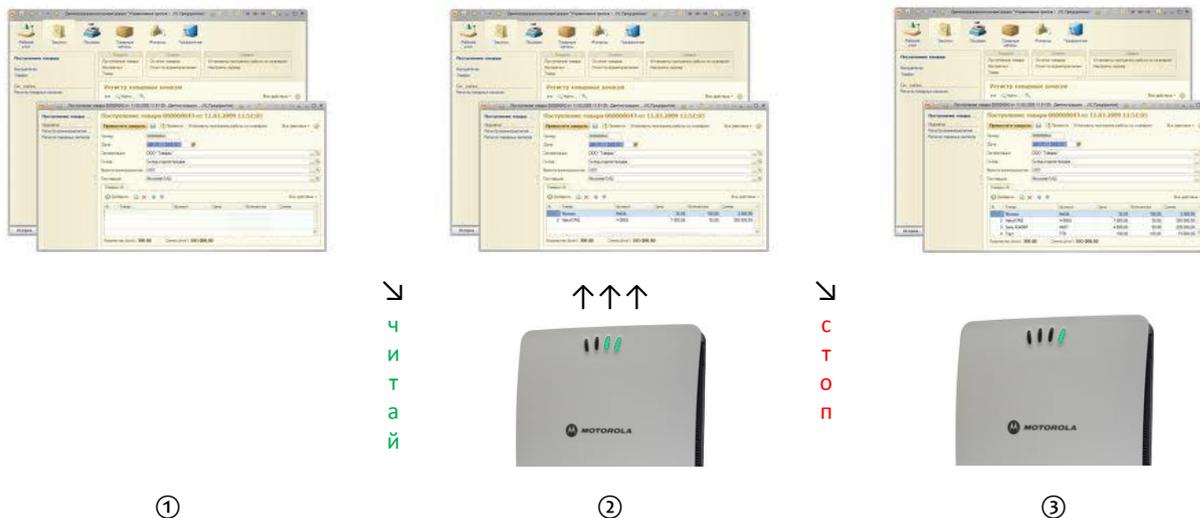
Синхронная инвентаризация не требует обрабатывания внешнего события «Чтение», и поэтому работает во всех конфигурациях «1С:Предприятия 8.2» и всех версиях операционной системы Windows.

Примечание: во время синхронной инвентаризации событие «Чтение» не приходит, т.к. такое событие «подвесило» бы 1С.

Асинхронная инвентаризация меток

Асинхронная инвентаризация означает следующее:

1. «1С:Предприятие» дало считывателю команду «считай окружающие метки в течение N секунд» и продолжило делать свои дела;
2. По мере инвентаризации новых меток считыватель асинхронно посылает «1С:Предприятию» внешние события, в результате чего считанные метки могут интерактивно появляться в окнах и документах «1С:Предприятия»;
3. Считыватель либо закончил через указанное время, либо «1С:Предприятие» дало ему команду закончить инвентаризацию досрочно.



Таким образом, при асинхронной инвентаризации окно 1С всегда остается доступным для взаимодействия с пользователем, а найденные метки могут интерактивно появляться на экране.

Пример кода для асинхронной инвентаризации:

Модуль управляемого приложения:

```
&НаКлиенте
Процедура ОбработкаВнешнегоСобытия(Источник, Событие, Данные) // Предопределенная процедура 1С
    Если Источник = "CleverenceRFID" И Событие = "Чтение" Тогда
        глПриИнвентаризацииМетки(Данные); // какая-то глобальная процедура обработки считывания меток
    КонецЕсли;
КонецПроцедуры
```

Либо подписать форму на внешнее событие и:

ВнешнееСобытие

ВнешнееСобытие

Модуль формы:

```
// ----- по нажатию кнопки 1 -----
&НаКлиенте
Процедура ПоКнопке1(Команда)
    // Опрашивать окружающие метки в течение 5000 миллисекунд (5 сек)
    считыватель.НачатьИнвентаризацию(5000);
КонецПроцедуры

&НаКлиенте
Процедура ВнешнееСобытие(Источник, Событие, Данные) // Предопределенная процедура 1С
    Если Источник = "CleverenceRFID" И Событие = "Чтение" Тогда
        ПриИнвентаризацииМетки(Данные);
    КонецЕсли;
```

КонецПроцедуры

&НаКлиенте

Процедура ПриИнвентаризацииМетки(tagId)

Попытка

// Получить полные данные считанной метки (или одинаковых меток) сразу со всех считывателей:

метка = КлеверенсRFID.ВыбратьМетку(tagId);

// Либо получить данные у конкретного считывателя (подробнее о считывателях см. ниже)

// метка = считыватель.ВыбратьМетку(tagId);

ОбработатьМетку(метка); // Какая-то процедура обработки метки

...

Исключение

Сообщить(КлеверенсRFID.ОписаниеОшибки());

ОкончаниеПопытки;

КонецПроцедуры

// ----- по нажатию кнопки 2 -----

&НаКлиенте

Процедура ПоКнопке2(Команда)

Попытка

// Получить все метки, обнаруженные во время инвентаризации (включая и те, по которым приходили события)

метки = считыватель.ОкончитьИнвентаризацию();

Для индекс = 0 по метки.Количество - 1 Цикл

метка = метки.Элемент(индекс);

ОбработатьМетку(метка);

КонецЦикла;

...

Исключение

Сообщить(КлеверенсRFID.ОписаниеОшибки());

ОкончаниеПопытки;

КонецПроцедуры

Чтение банков памяти меток Class 1 Gen 2

Операция чтения банка поддерживается на уровне радио-протокола обмена между метками и считывателем и возвращает всю или часть информации, содержащейся в интересующих банках RFID-меток.

Чтение банка EPC/UII

Чтение банка EPC/UII происходит во время инвентаризации меток (которая не требует паролей), а также при чтении любых других банков, поэтому отдельно чтением банка EPC/UII озабочиваться необязательно.

Чтение банка USER

Банк USER хранит любую дополнительную информацию в формате ISO 15961 (конкретные упакованные поля со строковыми значениями) либо просто байтами. В зависимости от используемого в метке чипа, банк USER может быть размером от ноля бит до нескольких килобайт.

Пример №1:

Любой модуль:

// Прочсть банки USER всех меток в поле видимости считывателя, в течение 2,5 секунд (2500 миллисекунд)

метки = считыватель.ПрочстьБанкUSER(2500);

Для индекс = 0 по метки.Количество - 1 Цикл

метка = метки.Элемент(индекс);

Сообщить("Прочитано: " + метка.tagID + ", USER = " + Строка(метка.БанкUSER));

КонецЦикла;

(если эту память не прожгли намертво). Текущая реализация компоненты такова, что в рамках одного запроса можно писать только в какой-то один конкретный банк. Считыватель отправляет запрос, а метки, подходящие под условия запроса, каждая по очереди возвращают запрошенную информацию (конкретную часть указанного банка памяти).

Прописывание меток идентификационным кодом EPC/UII

Если метки используются для целей контроля за движением товаров/объектов/документов, то самым главным в RFID-метке будет являться банк EPC. В банке EPC/UII будет содержаться собственно EPC или UII (см. «RFID Class 1 Generation 2 для логистики и розницы»), описывающий, на какой конкретно объект будет нанесена RFID-метка.

Создание новых EPC по данным из базы «1С:Предприятия»

Компонента «Клеверенс RFID» предоставляет много способов создания EPC на основе данных о товарах, упаковках, основных средствах или документах:

Любой модуль:

Попытка

```
// Создание EPC на основе штрихкода EAN13 и уникального серийного номера единицы товара
ерс = КлеверенсRFID.ЕРСизEAN13(
    КлеверенсRFID.ФильтрыEPC.SGTIN_ТоварДляКассы.Значение, EAN13, СерийныйНомер);
```

```
// Создание EPC на основе штрихкода EAN13 и уникального серийного номера паллеты с товаром
ерс = КлеверенсRFID.ЕРСизEAN13(
    КлеверенсRFID.ФильтрыEPC.SGTIN_Контейнер.Значение, EAN13, СерийныйНомер);
```

```
// Создание EPC на основе кода товара и уникального серийного номера единицы товара
// Код компании указан как «2», что означает условно «Наша компания» и, соответственно,
// сгенерированный EPC будет «нашим внутренним EPC», как, например, штрихкоды EAN13 вида «20.....»
ерс = КлеверенсRFID.ЕРСизSGTIN(
    КлеверенсRFID.ФильтрыEPC.SGTIN_ТоварДляКассы.Значение,
    2, НоменклатураКод, СерийныйНомер);
```

```
// Создание EPC для паллеты с товаром. Паллеты маркируются сквозным уникальным номером.
// Код компании указан как «2», что означает условно «Наша компания»...
ерс = КлеверенсRFID.ЕРСизSSCC(
    КлеверенсRFID.ФильтрыEPC.SSCC_Все.Значение, 2, ЧисловойНомерПаллеты);
```

```
// Создание EPC на основе числового кода типа документа и номера конкретного документа.
// Код компании указан как «2», что означает условно «Наша компания» и, соответственно,
// сгенерированный EPC будет «нашим внутренним EPC»
ерс = КлеверенсRFID.ЕРСизGDTI(
    КлеверенсRFID.ФильтрыEPC.GDTI_Все.Значение,
    2, ЧисловойТипДокумента, СерийныйНомерДокумента);
```

Исключение

```
Предупреждение("Ошибка создания EPC: " + КлеверенсRFID.ОписаниеОшибки());
```

КонецПопытки;

Запись сразу в несколько меток

Из 4х банков меток Gen2 для записи доступны три: банк с паролями, банк EPC и пользовательский банк.

Текущая реализация компоненты такова, что записать что-либо в метку можно только зная её Tag ID (чтобы не писать непонятно что в случайные метки). Поэтому прежде чем что-нибудь записать, сначала следует проинвентаризовать метки и получить их Tag ID.

Зная Tag ID, можно записать что-нибудь одновременно во все метки с таким Tag ID.

```
Любой модуль:  
  
Попытка  
    // Создать EPC:  
    ерс = ...  
  
    // Записать EPC:  
    ПодключенныйСчитыватель.ЗаписатьEPCUI(ИнтересуемаяМетка.TagId, ерс, 0);  
    Предупреждение("В метку с tag ID [" + ИнтересуемаяМетка.TagId + "] успешно записан новый EPC [" +  
        ерс.Строка() + "] (" + ерс.БинарноеПредставление + ").");  
  
Исключение  
    Предупреждение("Ошибка записи в метку [" + ИнтересуемаяМетка.TagId + "]: " +  
        КлеверенсRFID.ОписаниеОшибки());  
  
КонецПопытки;
```

Запись только в одну конкретную метку

Запись только в одну конкретную метку опирается на то, что у каждой метки должен быть свой уникальный номер чипа.

Зная EPC, можно прочитать банк TID одной единственной метки и затем записать только в неё:

```
Любой модуль:  
  
Попытка  
    новыйEPC = ...  
  
    // Прочитать номер чипа, пароль на чтение = 0 (нет пароля):  
    tid = ПодключенныйСчитыватель.ПрочитатьБанкTID(ИзвестныйTagID, 0);  
    // Записать новый EPC по номеру чипа, пароль на запись = 0 (нет пароля):  
    ПодключенныйСчитыватель.ЗаписатьEPCпоTID(ИзвестныйTagID, tid, новыйEPC, 0);  
    Предупреждение("В метку с tag ID [" + ИнтересуемаяМетка.TagId + "] успешно записан новый EPC [" +  
        новыйEPC.Строка() + "] (" + новыйEPC.БинарноеПредставление + ").");  
  
Исключение  
    Предупреждение("Ошибка записи в метку [" + ИнтересуемаяМетка.TagId + "]: " +  
        КлеверенсRFID.ОписаниеОшибки());  
  
КонецПопытки;
```

Не зная TID можно просто прочитать банки TID всех меток вокруг и потом записать в нужную:

```
Любой модуль:  
  
Попытка  
    новыйEPC = ...  
  
    // Читать метки и банки TID всех меток вокруг в течение 1,5 сек (1500 миллисекунд)  
    // пароль на чтение = 0 (нет пароля)  
    // возвратится коллекция меток, в каждой из которых будет проставлен реквизит TID  
    метки = ПодключенныйСчитыватель.ПрочитатьБанкиTID(1500, 0);  
    // Записать новый EPC по номеру чипа, пароль на запись = 0 (нет пароля):  
    ПодключенныйСчитыватель.ЗаписатьEPCпоTID(метка[0].TagId, метка[0].TID, новыйEPC, 0);  
    Предупреждение("В метку с tag ID [" + ИнтересуемаяМетка.TagId + "] успешно записан новый EPC [" +  
        новыйEPC.Строка() + "] (" + новыйEPC.БинарноеПредставление + ").");
```

Исключение

```
Предупреждение("Ошибка записи в метку [" + ИнтересуемаяМетка.TagId + "]: " +  
КлеверенсРФИД.ОписаниеОшибки());
```

КонецПопытки;

Использование компоненты для задач розницы

Использование RFID для задач розницы в основном регулируется стандартами международной организации GS1 EPCglobal. С помощью RFID в рамках стандартов можно маркировать и учитывать:

1. Конкретные единицы товаров;
2. Скидочные и бонусные карты покупателей;
3. Выделенные места в торговом зале;
4. Сопроводительные документы;
5. много всего другого...

Что можно записать в RFID-метку

В рамках стандартов EPCglobal в метку можно записать только правильно сформированный EPC, содержащий цифровой регистрационный номер компании в EPCglobal и некий номер маркируемого объекта (тоже цифровой или алфавитно-цифровой, в зависимости от размера памяти EPC/UII банка RFID-чипа, используемого в метке).

Клеверенс поддерживает маркировку следующих объектов:

Обязательный параметр, требуемый ISO 28560:

1. **выпвып** (книги, журнала, диска, читательского билета и т.п.), уникальный в рамках библиотеки;

Необязательные параметры, которые согласно ISO 28560 можно не использовать:

2. **пыпвы** (International Standard Identifier for Libraries and Related Organisations – Международный стандартный идентификатор для библиотек и других связанных организаций, в России ISIL'ы выдаются ГПНТБ);

Следует иметь в виду, что память EPC/UII для записи UII и память USER для записи дополнительных данных у самых бюджетных RFID-меток обычно крайне ограничена. Фактически, в память USER со стандартными 32 бита не поместится ни одно дополнительное поле. Это следует учитывать при разработке системы учета и выборе меток.

Создание новых EPC по данным о товарах

Компонента «Клеверенс RFID» предоставляет много способов создания EPC на основе данных о товарах, упаковках, основных средствах или документах:

Любой модуль:

Попытка

```
// Создание EPC на основе штрихкода EAN13 и уникального серийного номера единицы товара  
erc = КлеверенсRFID.EPCСизEAN13(  
    КлеверенсRFID.ФильтрыEPC.SGTIN_ТоварДляКассы.Значение, EAN13, СерийныйНомер);
```

```
// Создание EPC на основе штрихкода EAN13 и уникального серийного номера паллеты с товаром  
erc = КлеверенсRFID.EPCСизEAN13(  
    КлеверенсRFID.ФильтрыEPC.SGTIN_Контейнер.Значение, EAN13, СерийныйНомер);
```

```
// Создание EPC на основе кода товара и уникального серийного номера единицы товара  
// Код компании указан как «2», что означает условно «Наша компания» и, соответственно,
```

```

// сгенерированный EPC будет «нашим внутренним EPC», как, например, штрихкоды EAN13 вида «20.....»
ерс = КлеверенсRFID.EPCизSGTIN(
    КлеверенсRFID.ФильтрыEPC.SGTIN_ТоварДляКассы.Значение,
    2, НоменклатураКод, СерийныйНомер);

// Создание EPC для паллеты с товаром. Паллеты маркируются сквозным уникальным номером.
// Код компании указан как «2», что означает условно «Наша компания»...
ерс = КлеверенсRFID.EPCизSSCC(
    КлеверенсRFID.ФильтрыEPC.SSCC_Все.Значение, 2, ЧисловойНомерПаллеты);

// Создание EPC на основе числового кода типа документа и номера конкретного документа.
// Код компании указан как «2», что означает условно «Наша компания» и, соответственно,
// сгенерированный EPC будет «нашим внутренним EPC»
ерс = КлеверенсRFID.EPCизGDTI(
    КлеверенсRFID.ФильтрыEPC.GDTI_Все.Значение,
    2, ЧисловойТипДокумента, СерийныйНомерДокумента);

```

Исключение

```

Предупреждение("Ошибка создания EPC: " + КлеверенсRFID.ОписаниеОшибки());
КонецПопытки;

```

Использование компоненты для библиотечных задач

Стандарт ISO 28560 RFID в библиотеках предусматривает RFID-учет всех библиотечных объектов. С помощью RFID в рамках стандарта можно учитывать:

1. Библиотечный фонд – книги, журналы, диски и т.п., выдаваемые абонементам;
2. Читательские билеты (метка либо вклеивается в билет, либо сам билет заменяется RFID-карточкой);
3. Собственное библиотечное имущество, не выдаваемое абонементам (столы, шкафы и т.п.);
4. Товары на продажу;
5. Списанные объекты и объекты, ожидающие утилизации.

Компонента Клеверенс поддерживает все из вышеперечисленного.

Что можно записать в RFID-метку

В рамках стандарта ISO 28560 в метку можно записать 26 полей. Все поля строковые, могут состоять как из цифр, так и из кириллицы или иероглифов.

Из этого списка компонента Клеверенс поддерживает следующие:

Обязательный параметр, требуемый ISO 28560:

3. **Номер библиотечного объекта** (книги, журнала, диска, читательского билета и т.п.), уникальный в рамках библиотеки;

Необязательные параметры, которые согласно ISO 28560 можно не использовать:

4. **ISIL библиотеки** (International Standard Identifier for Libraries and Related Organisations – Международный стандартный идентификатор для библиотек и других связанных организаций, в России ISIL'ы выдаются ГПНТБ);
5. **ISBN, ISSN** или другой штрихкод;
6. **Тип использования объекта** (что это: книга, читательский билет и т.п.);
7. **Заколовок** (поддерживается Юникод – русские буквы, скобки, тире и т.п.).
8. **Адрес полки** (поддерживается Юникод – русские буквы, скобки, тире и т.п.).
9. **Размер набора** (если объект является частью набора, например несколько томов издания).
10. **Номер объекта в наборе** (например «1», если это 1й том).

Следует иметь в виду, что память EPC/UII для записи UII и память USER для записи дополнительных данных у самых бюджетных RFID-меток обычно крайне ограничена. Фактически, в память USER со стандартными 32 бита не поместится ни одно дополнительное поле. Это следует учитывать при разработке системы учета и выборе меток.

Маркировка библиотечных объектов RFID-метками

Для маркировки библиотечного объекта RFID-меткой следует прошить в метку номер объекта, уникальный в рамках библиотеки (см. «Принципы идентификации объектов с помощью меток Class 1 Generation 2» и «

Кратко, EPC хранится в метке и содержит информацию о товаре, компании-производителе (или компании-владельце) и серийном номере конкретной единицы каждого товара или упаковки.

Более подробно о EPC с примерами маркировки см. раздел «Понятие электронного кода продукта (EPC)».

Более подробно использование компоненты RFID от Клеверенс для задач розницы рассмотрено в разделе «Использование компоненты для задач розницы».

RFID Class 1 Generation 2 для библиотек»).

Маркировка библиотечного фонда

Процедура маркировки книг, журналов и т.п. следующая:

1. Обклеиваем интересующие нас объекты «непрошитыми» RFID-метками;
2. По очереди прошиваем каждую метку соответствующим UII объекта.

Если используется «антикражный бит», то в качестве кода применения («антикражного бита») обязательно выставляем «На складе» (подробнее см. «Антикражный механизм для библиотек»).

Пример №1:

```
// если у библиотеки нет ISIL, то можно передать Неопределено
// если используется «антикражный бит», то AFI = КлеверенсRFID.AFI.НаСкладе
uіі = КлеверенсRFID.UІІизБиблиотечногоКода(ISIL, экземпляр.Код, КлеверенсRFID.AFI.НаСкладе);
```

Если вместо «антикражного бита» используется поиск метки по базе библиотеки и «галочка» в карточке объекта, то в качестве кода применения выставляем «Библиотечный» (подробнее см. «Антикражный механизм для библиотек»).

Пример №2:

// если у библиотеки нет ISIL, то можно передать Неопределено
 // если «антикражный бит» не используется, то AFI по умолчанию = КлеверенсРФИД.AFI.Библиотечный
 uii = КлеверенсРФИД.UiiИзБиблиотечногоКода(ISIL, экземпляр.Код);

Пример №3:

// если у библиотеки нет ISIL, то можно передать Неопределено
 uii = КлеверенсРФИД.UiiИзБиблиотечногоКода(ISIL, экземпляр.Код);
 // если память метки позволяет, то можно проставить в Uii тип использования для объекта
 uii.ТипИспользования = КлеверенсРФИД.Библиотеки.ТипыИспользования.ДляВыдачи;

Uii следует записать в банк EPCUii (см. «Если требуется и позволяет память метки, то можно прошить в банк USER некие дополнительные параметры.

Общий алгоритм маркировки»).

Следующая таблица показывает, какой длины могут быть используемые коды в зависимости от размера банка памяти чипа, используемого в RFID-метке:

Поля, используемые при формировании Uii	Тип полей (в каком виде поля представлены в библиотечной системе)	Максимальная длина, в символах		
		Банк EPC/Uii в 128 бит	Банк EPC/Uii в 256 бит	Банк EPC/Uii в 496 бит
Только уникальный код	цифровой, не начинается на «0»	19	40	100
	цифровой, начинается на «0»	16	40	100
	только заглавные латинские буквы	13	32	80
	заглавные латинские буквы и цифры, тире и пр.	10	26	66
	любые латинские буквы и цифры, тире и пр.	9	23	57
	русские буквы, иероглифы, пунктуация и т.п.	5	16	35
Уникальный код + ISIL	код цифровой, ISIL российский*	-	20 + 11	80 + 11
	код заглавные латинские буквы и цифры, ISIL российский	-	13 + 11	53 + 11
	код любые латинские буквы и цифры, ISIL российский	-	11 + 11	45 + 11
	код любой: русские буквы, иероглифы, пунктуация и т.п., ISIL российский	-	7 + 11	30 + 11
	код цифровой, не начинается на «0», ISIL международный**	-	14 + 16	72 + 16
	код цифровой, начинается на «0», ISIL международный	-	12 + 16	72 + 16
	код заглавные латинские буквы и цифры, тире, ISIL международный	-	8 + 16	48 + 16
	код любые латинские буквы и цифры, тире, ISIL международный	-	6 + 16	41 + 16
	код любой: русские буквы, иероглифы, знаки, ISIL международный	-	3 + 16	28 + 16

Уникальный код + код применения	код цифровой , не начинается на «0»	12	34	96
	код цифровой , начинается на «0»	10	34	96
	код только заглавные латинские буквы	8	27	76
	код заглавные латинские буквы и цифры	6	22	64
	код любые латинские буквы и цифры	6	19	54
	русские буквы, иероглифы , пунктуация и т.п.,	2	14	34
Уникальный код+ ISIL + код применения	код цифровой , ISIL российский*	-	19 + 11	74 + 11
	код заглавные латинские буквы и цифры , ISIL российский	-	10 + 11	49 + 11
	код любые латинские буквы и цифры , ISIL российский	-	9 + 11	42 + 11
	код любой: русские буквы, иероглифы , пунктуация и т.п., ISIL российский	-	5 + 11	29 + 11
	код цифровой , не начинается на «0», ISIL международный**	-	9 + 16	66 + 16
	код цифровой , начинается на «0», ISIL международный	-	8 + 16	66 + 16
	код заглавные латинские буквы и цифры , тире, ISIL международный	-	5 + 16	44 + 16
	код любые латинские буквы и цифры , тире, ISIL международный	-	4 + 16	37 + 16
	код любой: русские буквы, иероглифы , знаки , ISIL международный	-	2 + 16	27 + 16

* ISIL по стандарту ISO 15551 российский, 11 символов вида «RU-12345678»

** ISIL по стандарту ISO 15551 международный, 16 символов вида «US-InU-Mu1233456»

Приведенную таблицу следует читать так:

например, **для самой верхней строки** таблицы цифры справа означают, что:

- если библиотечный код состоит только из цифр и не начинается на «0», то в метке с банком UPC/UII в 128 бит поместится код длиной максимум 19 цифр;
- в метке с банком UPC/UII в 256 бит поместится максимум 40 цифр такого кода;
- в метке с банком UPC/UII в 496 бит поместится аж 100 цифр такого кода.

для самой нижней строки таблицы цифры справа означают, что:

- в метке с банком UPC/UII в 128 бит библиотечный код с ISIL и кодом применения не поместятся вообще;
- в метку с банком UPC/UII в 256 бит поместится 2 символа библиотечного кода в Юникоде + международный ISIL из 16 символов и код применения;

- в метку с банком UPC/UII в 496 бит поместится 27 символов библиотечного кода в Юникоде + международный ISIL из 16 символов и код применения.

Если требуется и позволяет память метки, то можно прошить уже в банк USER некие дополнительные параметры из списка, приведенного в самом начале раздела (заголовок книги, место расположение на полке и т.п.).

Пример №4. Прошивка в банк USER наименования книги, номера тома и места на полке:

```
// получить с сервера используемый пароль на доступ к RFID-меткам
парольНаДоступ = ПолучитьПарольНаДоступRFID();

бо = КлеверенсRFID.СоздатьБиблиотечныйОбъект();
бо.Наименование = "Л. Н. Толстой. Война и Мир, том 1й";
бо.РазмерНабора = 4; // 4 тома
бо.НомерВНаборе = 1; // 1й том
бо.МестоНаПолке = "А-14-21";

банк = бо.СформироватьUSERБанк();
// заполненные выше данные займут ровно 74 байта памяти банка USER
// метки с банком памяти USER < 74 бита не смогут быть прошитыми
считыватель.ЗаписатьUSER(метка.TagId, банк, парольНаДоступ);
```

Данные из приведенного примера займут ровно 74 байта банка памяти USER. Самые бюджетные метки в настоящий момент имеют всего 32 бита памяти USER и, соответственно, не смогут быть использованы в таком сценарии.

Маркировка читательских билетов (и RFID-карточек)

Читательские билеты можно промаркировать RFID-метками (вклеить в билет), либо полностью заменить билеты RFID-карточками.

Процедура маркировки читательских билетов следующая:

1. Вклеиваем во все читательские билеты «непрошитые» RFID-метками;
2. По очереди прошиваем каждую метку соответствующим UII объекта.

Процедура выдачи RFID-карточек следующая:

1. Вставляем чистую RFID-карточку в специализированный карточный принтер и печатаем на ней фотографию и другую информацию о владельце и библиотеке;
2. Кладем RFID-карточку на антенну считывателя и прошиваем соответствующим UII читателя.



В UII метки для читательского билета желательно указать, что это не книга, а именно читательский билет. Иначе при выдаче и возврате книг об этом придется догадываться по коду билета, читать другие банки памяти и пр.

Код AFI для читательского билета всегда должен быть равен «Библиотечный», чтобы не «звенеть» на воротах библиотеки и в магазинах.

Пример №1:

```
// если у библиотеки нет ISIL, то можно передать Неопределено
uii = КлеверенсRFID.UIIизБиблиотечногоКода(ISIL, читатель.Код, КлеверенсRFID.AFI.Библиотечный);
uii.ТипИспользования = КлеверенсRFID.Библиотеки.ТипыИспользования.ЧитательскийБилет;
```

UII следует записать в банк EPCUII (см. «Если требуется и позволяет память метки, то можно прошить в банк USER некие дополнительные параметры.

Общий алгоритм маркировки»).

Предусмотрены следующие типы использования для читательских билетов:

- КлеверенсRFID.Библиотеки.ТипыИспользования.ЧитательскийБилет; (любой)
- КлеверенсRFID.Библиотеки.ТипыИспользования.ЧитательскийБилет_Взрослый;
- КлеверенсRFID.Библиотеки.ТипыИспользования.ЧитательскийБилет_Подростковый;
- КлеверенсRFID.Библиотеки.ТипыИспользования.ЧитательскийБилет_Детский;

Пример №2. Проверка, что тип использования у метки – любой читательский билет:

```
Если метка.Объект.Тип() = "БиблиотечныйКод" И метка.Объект.ТипИспользования <> Неопределено И  
метка.Объект.ТипИспользования.КодКласса = КлеверенсRFID.Библиотеки.ТипыИспользования.ЧитательскийБилет.КодКласса  
Тогда
```

Если требуется по логике и позволяет память метки, то можно прошить в банк USER некие дополнительные параметры.

Пример №3. Прошивка в банк USER имени владельца билета:

```
// получить с сервера используемый пароль на доступ к RFID-меткам  
парольНаДоступ = ПолучитьПарольНаДоступRFID();  
  
бо = КлеверенсRFID.СоздатьБиблиотечныйОбъект();  
бо.Наименование = читатель.ФИО;  
  
банк = бо.СформироватьUSERБанк();  
считыватель.ЗаписатьUSER(метка.TagId, банк, парольНаДоступ);
```

Маркировка библиотечного имущества (столы и стулья)

Библиотечное имущество может потребовать два типа RFID-меток: гибкие (для дерева/пластика) и корпусные (для металла). На гибкие метки можно печатать информацию на специализированном этикеточном принтере. На корпусных RFID-метках можно просто писать маркером или использовать самоклеющуюся этикетку (а на этикетку распечатать при помощи того же специализированного этикеточного принтера).

Процедура маркировки собственного имущества следующая:

1. Печатаем/пишем/наклеиваем на «непрошитые» RFID-метки наименование, инвентарный номер, штрихкод и т.п.;
2. По очереди прошиваем каждую метку соответствующим UII объекта;
3. Обклеиваем имущество прошитыми RFID-метками.

В UII метки для имущества желательно указать, что это не книга, а именно библиотечное имущество. Если используется «антикражный бит», то в качестве кода применения («антикражного бита») обязательно выставляем «На складе» (подробнее см. «Антикражный механизм для библиотек»).

Пример №1:

```
// если у библиотеки нет ISIL, то можно передать Неопределено  
uii = КлеверенсRFID.UIIизБиблиотечногоКода(ISIL, имущество.Код, КлеверенсRFID.AFI.НаСкладе);
```

```
uII.ТипИспользования = КлеверенсРФИД.Библиотеки.ТипыИспользования.Имущество;
```

UII следует записать в банк EPCUII (см. «Если требуется и позволяет память метки, то можно прошить в банк USER некие дополнительные параметры.

Общий алгоритм маркировки»).

Предусмотрены следующие типы использования для имущества:

- КлеверенсРФИД.Библиотеки.ТипыИспользования.Имущество; (любое)
- КлеверенсРФИД.Библиотеки.ТипыИспользования.Имущество_Компьютер;
- КлеверенсРФИД.Библиотеки.ТипыИспользования.Имущество_Видеопроектор;
- КлеверенсРФИД.Библиотеки.ТипыИспользования.Имущество_Кинопроектор;
- КлеверенсРФИД.Библиотеки.ТипыИспользования.Имущество_Доска;
- КлеверенсРФИД.Библиотеки.ТипыИспользования.НеДляВыдачи;

Пример №2. Проверка, что тип использования у метки – любое имущество:

```
Если метка.Объект.Тип() = "БиблиотечныйКод" И метка.Объект.ТипИспользования <> Неопределено И  
(метка.Объект.ТипИспользования.КодКласса = КлеверенсРФИД.Библиотеки.ТипыИспользования.Имущество.КодКласса или  
метка.Объект.ТипИспользования.КодКласса = КлеверенсРФИД.Библиотеки.ТипыИспользования.НеДляВыдачи.КодКласса) Тогда
```

Если требуется и позволяет память метки, то можно прошить в банк USER некие дополнительные параметры.

Общий алгоритм маркировки

Поскольку метки прошиваются конкретным библиотечным кодом, все их следует прошивать по очереди. Наиболее удобный способ – сначала оптом обклеить интересующие объекты «непрошитыми» метками, а затем по одному прошить уникальным кодом.

Алгоритм следующий:

1. По одному кладем объекты на антенну RFID-считывателя;
2. Выбираем из базы, что это такое;
3. Формируем UII на основе некоего уникального используемого в библиотеке кода;
4. Программа должна убедиться, что метка в поле чтения присутствует только одна. Если меток больше – выдать предупреждение.

Иногда невозможно организовать работу так, чтобы читалась только одна метка. В этом случае программа может опираться на значение RSSI (уровень сигнала от метки) и проверять, что метка с большим RSSI в поле зрения только одна.

5. Прописать сформированный UII в метку. Затем сразу же прочитать метку и проверить, что всё записалось корректно.

Пример:

```
// получить с сервера используемый пароль на доступ к RFID-меткам  
парольНаДоступ = ПолучитьПарольНаДоступRFID();  
Пока Истина Цикл  
    // Заставить пользователя выбрать из базы конкретный объект фонда, читательский билет и т.п.  
    // если выбранному объекту уже сопоставлена метка – переспросить пользователя
```

```

// (например, метка могла выйти из строя и действительно требуется перемаркировка)
маркируемыйОбъект = ВыбратьЭкземпляр();
Если маркируемыйОбъект = Неопределено Тогда
    Возврат;
КонецЕсли;

режим = РежимДиалогаВопрос.ОКОтмена;
выбраннаяМетка = Неопределено;
Пока выбраннаяМетка = Неопределено Цикл
    ответ = Неопределено;
    метки = Неопределено;
    // Поискать вокруг антенны RFID-метки в течение 1й секунды (1000 миллисекунд)
    Попытка
        метки = считыватель.ИнвентаризоватьМетки(5000);
    Исключение
        Вопрос("Ошибка поиска меток! " + КлеверенсRFID.ОписаниеОшибки(), РежимДиалогаВопрос.ОК);
        Продолжить;
    КонецПопытки;

    Если метки.Количество = 0 Тогда
        ответ = Вопрос("Положите маркируемый объект на антенну!", режим);
    Иначе
        Если метки.Количество > 0 Тогда
            ответ = Вопрос("Уберите от антенны посторонние предметы!", режим);
        Иначе
            // Выбрать единственную метку
            выбраннаяМетка = метки.Элемент(0);
        КонецЕсли;
    КонецЕсли;

    Если ответ = КодВозвратаДиалога.Отмена Тогда
        Прервать;
    КонецЕсли;
КонецЦикла;

Попытка
    // Создать UII в соответствии с тем, какой объект выбрали, и с правильным AFI:
    ui1 = СоздатьПравильныйUII(маркируемыйОбъект);
    // Записать UII
    считыватель.ЗаписатьEPCUII(выбраннаяМетка.TagId, ui1, парольНаДоступ);
    // Проверить
    записанное = считыватель.ПрочитатьEPCUII(выбраннаяМетка.TagId, парольНаДоступ);
    Если записанное.EPCUII.AFI <> ui1.AFI или записанное.EPCUII.УникальныйКод <> ui1.УникальныйКод Тогда
        Вопрос("Ошибка записи метки!", РежимДиалогаВопрос.ОК);
        Продолжить;
    КонецЕсли;

    Сообщить("В метку с Tag ID [" + выбраннаяМетка.TagId + "] успешно записан новый UII [" +
        ui1.Строка() + "] (" + ui1.БинарноеПредставление + ").");
Исключение
    Предупреждение("Ошибка записи в метку! " + КлеверенсRFID.ОписаниеОшибки());
КонецПопытки;
КонецЦикла;

```

Антикражный механизм для библиотек

Стандарт ISO 28560 предлагает на выбор три варианта реализации антикражной системы для библиотеки:

1. Использовать коды применения (AFI)

У всего, что можно выносить, используется код применения «Библиотечный». У всего, что нельзя выносить, – код применения «НаСкладе».

При выдаче/возврате коды применения в метках перепрошиваются.

Это наиболее предпочтительный метод, т.к. позволяет RFID-считывателю на антикражных воротах работать автономно без подключения к библиотечной системе, не требует дополнительного оборудования на выдаче/возврате.

2. Использовать поиск по базе данных

В этом случае RFID-считывателю на антикражных воротах требуется постоянное подключение к библиотечной базе, чтобы искать в ней по UИИ и смотреть, что выносятся.

3. Использовать гибридные UHF/EAS или HF/EAS метки

Т.е. использовать метки, в которых помимо RFID есть антикражная полоска. Метки будут дороже обычных, но это позволяет задействовать обычные магазинные антикражные ворота. Однако, EAS работает на других частотах и, соответственно, для выдачи/возврата потребуются либо гибридное RFID/EAS оборудование (дорогое), либо два набора оборудования и лишние действия при выдаче/возврате.

Решение, какой метод подходит лучше, принимает сама библиотека.

Выдача и возврат книг

При выдаче и возврате объектов фонда программа должна следовать следующему алгоритму:

1. Просканировать пространство вокруг антенны RFID-считывателя на наличие меток;
2. Посмотреть, нет ли среди прочитанных UИИ читательских билетов. Если их больше одного – попросить убрать лишние и снова просканировать пространство. Если нет ни одного – попросить положить (или заставить библиотекаря выбрать читателя вручную);
3. На основе остальных UИИ сформировать список выдаваемого/возвращаемого;
4. Если используется «антикражный бит», то перепрошить его в метках объектов фонда (но не трогать его в читательских билетах, имуществе и прочем вокруг!).

Пример кода для перепрошивки AFI (см. «Антикражный механизм для библиотек»)

```
// получить с сервера используемый пароль на доступ к RFID-меткам
парольНаДоступ = ПолучитьПарольНаДоступRFID();
типДляВыдачи = КлеверенсRFИД.Библиотеки.ТипыИспользования.ДляВыдачи;
// цикл по всем считанным меткам
Для индекс = 0 по метки.Количество - 1 Цикл
    метка = метки.Элемент(индекс);
    Если метка.Объект.Тип() <> "БиблиотечныйКод" или
        (метка.Объект.ТипИспользования <> Неопределено И метка.Объект.ТипИспользования <> типДляВыдачи)
        Тогда
            Продолжить;
        КонецЕсли;

    Попытка
        // Проставить код применения «Библиотечный» (при выдаче) или «НаСкладе» (при возврате):
        uii.AFI = КлеверенсRFИД.AFI.Библиотечный;
        // Записать UИИ
        считыватель.ЗаписатьEPCUИИ(метка.TagId, uii, парольНаДоступ);
    Исключение
        Предупреждение("Ошибка записи в метку!" + КлеверенсRFИД.ОписаниеОшибки());
    КонецПопытки;
КонецЦикла;
```


Справочник разработчика

Cleverence.RFID.Api

Предоставляет API для работы со стационарными RFID-считывателями.

UИизБиблиотечногоКода (UИfromISIL)

Создает экземпляр библиотечный вариант UИ на основе переданных аргументов.

Синтаксис: **UИизБиблиотечногоКода** (<isil>, <itemIdentifier>)

Имя параметра	Описание
isil	ISIL библиотеки-владельца или null.
itemIdentifier	Номер библиотечного объекта, уникальный в рамках конкретной библиотеки.

UИизБиблиотечногоКода (UИfromISIL)

Создает экземпляр библиотечный вариант UИ на основе переданных аргументов.

Синтаксис: **UИизБиблиотечногоКода** (<isil>, <itemIdentifier>, <afi>)

Имя параметра	Описание
isil	ISIL библиотеки-владельца или null.
itemIdentifier	Номер библиотечного объекта, уникальный в рамках конкретной библиотеки.
afi	Код применения для объекта (важен для учета выдачи/возврата).

СоздатьБиблиотечныйОбъект (CreateLibraryItem)

Создает пустой экземпляр набора записей о библиотечном объекте.

Метод не принимает аргументов.

ОтключитьВсеСчитыватели (DisconnectAllReaders)

Освобождает все существующие подключения в рамках библиотеки.

Метод не принимает аргументов.

НайтиСчитыватели (LookupReaders)

Производит поиск и возвращает список RFID-считывателей в локальной подсети.

Функция производит поиск считывателей только в локальных подсетях, т.е. в диапазонах IP-адресов: (192.168.0.1 – 192.168.248.255), (172.16.0.1 - 172.16.240.255) и (10.0.0.1 - 10.255.255.255).

Метод не принимает аргументов.

ПолучитьСчитыватель (GetReader)

Получает существующий или создает новый RFID-считыватель по указанному URL.

Синтаксис: **ПолучитьСчитыватель** (<url>)

Имя параметра	Описание
url	URL считывателя с указанием типа подключения, адреса и порта.

ПодключитьСчитыватель (ConnectToReader)

Выполняет подключение к RFID-считывателю по указанному URL (с возможностью чтения/записи, но без возможности управления параметрами считывателя, см. «ПодключитьСчитыватель (ConnectToReader)»).

Синтаксис: **ПодключитьСчитыватель** (<url>)

Имя параметра	Описание
url	URL для подключения с указанием типа подключения, адреса и порта.

ПодключитьСчитывательПодПаролем (ConnectToReaderWithPassword)

Выполняет подключение к RFID-считывателю по указанному URL (с возможностью как чтения/записи, так и управления параметрами считывателя).

Синтаксис: **ПодключитьСчитывательПодПаролем** (<url>, <userName>, <password>)

Имя параметра	Описание
url	URL для подключения с указанием типа подключения, адреса и порта.
userName	Имя пользователя для подключения.
password	Пароль пользователя.

ВыбратьМетку (FetchTag)

Внимает из очереди считанных меток данные метки (меток) с указанным Tag ID сразу со всех RFID-считывателей, на которых в рамках библиотеки было запущено чтение меток.

Т.к. метки вынимаются из очереди, второй и последующие вызовы метода с одним и тем же Tag ID могут вернуть пустой результат.

Синтаксис: **ВыбратьМетку** (<tagid>)

Имя параметра

Описание

tagid

Tag ID интересующих меток.

ВыбратьМетки (FetchTags)

Вынимает из очереди считанных меток данные всех меток сразу со всех RFID-считывателей, на которых в рамках библиотеки было запущено чтение меток.

Т.к. метки вынимаются из очереди, второй и последующие вызовы метода могут вернуть пустой результат.

Метод не принимает аргументов.

EPСизSGTIN (EPCfromSGTIN)

Создает экземпляра SGTIN-варианта EPC на основе переданных аргументов.

Синтаксис: **EPСизSGTIN** (<filterValue>, <company>, <item>, <serial>)

Имя параметра

Описание

filterValue

Filter Value кода для указания типа упаковки, для которой предназначен данный EPC.

company

Код компании, зарегистрированной в GS1.

item

Код товара согласно каталога компании.

serial

Серийный номер экземпляра товара.

EPСизEAN13 (EPCfromEAN13)

Создает экземпляр SGTIN-варианта EPC на основе переданных аргументов.

Синтаксис: **EPСизEAN13** (<filterValue>, <ean13>, <serial>)

Имя параметра

Описание

filterValue

Filter Value кода для указания типа упаковки, для которой предназначен данный EPC.

ean13

Строка со штрихкодом EAN13.

НоваяМетка (CreateTag)

Создает экземпляр метки на основе указанного EPC.

Синтаксис: **НоваяМетка (<tagId>)**

Имя параметра	Описание
епс	EPC метки.

НоваяМетка (CreateTag)

Создает экземпляр метки на основе указанного UII.

Синтаксис: **НоваяМетка (<tagId>)**

Имя параметра	Описание
uii	UII метки.

СоздатьБиблиотечныйОбъект (CreateLibraryItem)

Создает и заполняет набор записей о библиотечном объекте на основе данных из USER-банка памяти RFID-метки.

Метод не принимает аргументов.

Имя параметра	Описание
bank	Экземпляр USER-банка памяти RFID-метки

Библиотеки (Libraries)

Содержит методы и перечисления, специфичные для библиотечного применения.

Версия (Version)

Возвращает версию компоненты.

Язык (Culture)

Возвращает или устанавливает текущую локализацию компоненты.

ВиртуальныйРежим (VirtualMode)

Возвращает настройки виртуального режима работы.

LookupTagParams (LookupTagParams)

Возвращает настройки проведения инвентаризации (слежения за метками) для всех RFID-считывателей. При этом собственные настройки конкретных считывателей могут добавлять/перекрывать общие настройки.

ФильтрыEPC (EpcFilterValues)

Возвращает коды фильтров EPC.

AFI (AFI)

Возвращает коды AFI/ASF (Application Family Identifier и Application Sub Family коды [ISO15961]), отражающее сферу применения (индустрию) объекта, на который нанесена метка.

Cleverence.GS1.AttributeBits

Задаёт дополнительные флаги для сфер применения меток UHF, кодированных по EPCglobal-совместимому стандарту.

Unassigned	Не назначены.
------------	---------------

Hazardous	Опасно.
-----------	---------

Cleverence.GS1.Epc

Реализация стандарта GS1 на электронный код товара EPC (Electronic Product Code <http://www.gs1.org/aboutepc/essential>). (см. GS1 Tag Data Standard (TDS) v 1.5, http://www.gs1.org/gsmf/kc/epcglobal/tds/tds_1_5-standard-20100818.pdf).

Строка (ToString)

Возвращает строковое представление данного EPC (URI либо описание ошибки).

Метод не принимает аргументов.

Схема (EpcScheme)

Возвращает вариант схемы, которой соответствует данный EPC.

(пока поддерживаются только схемы кодирования SGTIN-96, GDTI-96 и SSCC96).

AttributeBits (AttributeBits)

Возвращает дополнительные флаги (см. «AttributeBits (AttributeBits)»)

КодКомпании (Company)

Возвращает код компании, зарегистрированной в GS1.

Ссылка (Reference)

Возвращает ссылку на объект в каталоге. В зависимости от схемы кодирования EPC эта ссылка будет означать либо код товара согласно каталога компании, либо тип документа, либо код вида возвратной тары, либо тип места назначения и т.п.

СерийныйНомер (Serial)

Возвращает серийный номер конкретного объекта.

Фильтр (FilterValue)

Возвращает значение Filter Value из бинарного кодирования.

URI (URI)

Возвращает EPC URI (EPC pure identity URI) согласно стандарта GS1.

SGTIN (SGTIN)

Возвращает объект, содержащий поля SGTIN, либо "Неопределено", если данный EPC кодирует не SGTIN (т.е., не товар, не аксессуар и не упаковка для товаров).

GDTI (GDTI)

Возвращает объект, содержащий поля GDTI, либо "Неопределено", если данный EPC кодирует не GDTI (т.е. это не документ и не контейнер для документов).

SSCC (SSCC)

Возвращает объект, содержащий поля SSCC, либо "Неопределено", если данный EPC кодирует не SSCC (т.е. это не паллета и не контейнер).

EPCHeaderValueOffset	Смещение до заголовка EPC в битах.
----------------------	------------------------------------

EPCHeaderValueLength	Длина заголовка EPC в битах.
----------------------	------------------------------

FilterOffset	Смещение до поля Filter в битах.
--------------	----------------------------------

FilterLength	Длина поля Filter в битах.
--------------	----------------------------

PartitionOffset	Смещение до поля Partition в битах.
-----------------	-------------------------------------

PartitionLength	Длина поля Partition в битах.
-----------------	-------------------------------

Cleverage.GS1.EpcFilterValue

Значения Filter Value для кодов EPC согласно стандарта GS1 на фильтрацию меток на уровне радиопrotocol обмена.

SGTIN_ВсеОстальное	SGTIN Всё остальное.
--------------------	------------------------

SGTIN_ТоварДляКассы	SGTIN Товар для продажи на кассе.
---------------------	-------------------------------------

SGTIN_УпаковкаДляТранспортировки	SGTIN Упаковка для транспортировки.
----------------------------------	---------------------------------------

SGTIN_Зарезервировано1	SGTIN Зарезервировано (см. раздел 10.1 стандарта GS1 TDS).
------------------------	--

SGTIN_УпаковкаДляПереноски	SGTIN Упаковка для удобства переноски.
----------------------------	--

SGTIN_Зарезервировано2	SGTIN Зарезервировано (см. раздел 10.1 стандарта GS1 TDS).
------------------------	--

SGTIN_Контейнер	SGTIN Метка на большом контейнере (паллете).
-----------------	--

SGTIN_ВложениеИлиАксессуар	SGTIN Составная часть товара или аксессуар, вложенный в коробку и не предназначенный для продажи отдельно от товара.
----------------------------	--

GDTI_Все	GDTI Всё остальное.
----------	-----------------------

GDTI_Зарезервировано	GDTI Зарезервировано (см. раздел 10.1 стандарта GS1 TDS).
----------------------	---

SSCC_ВсеОстальное	SSCC Всё остальное.
SSCC_Reserved	SSCC Зарезервировано (см. раздел 10.1 стандарта GS1 TDS).
SSCC_FullCaseForTransport	SSCC Упаковка для транспортировки.
SSCC_UnitLoad	SSCC Контейнер (паллета).

Схема (EpcScheme)

Возвращает EPC схему, с которой связано данное значение Filter Value.

Значение (Value)

Возвращает числовое значение Filter Value.

Тип (Type)

Возвращает международное строковое описание числового значения Filter Value для данной EPC схемы.

Представление (DisplayName)

Возвращает локализованное строковое описание числового значения Filter Value для данной EPC схемы.

Cleverence.GS1.EpcScheme

Перечисление вариантов EPC.

Unknown	Неизвестный.
SGTIN	Serialized Global Trade Item Number (код товара в 14 цифр + серийный номер).
SSCC	Serial Shipping Container Code (код контейнера).
SGLN	Serialized Global Location Number (код местоположения).
GRAI	Global Returnable Asset Identifier (код возвращаемой единицы или тары).
GIAI	Global Individual Asset Identifier (код единицы имущества, такого как оборудование или основное средство).
GSRN	Global Service Relation Number (код сертификата или карты лояльности).
GDTI	Global Document Type Identifier (код экземпляра документа).
GID	General Identifier (код «вообще»).

USDOD

US Department of Defense Identifier (специальный код для идентификации имущества Министерства обороны США).

Cleverence.ISO.Afi

AF+ASF коды. Application Family и Application Sub Family коды [ISO 15961] отражают сферу применения (индустрию) объекта, на который нанесена метка.

Unknown

Неизвестно.

HaСкладе

Товар на складе.

Авиабагаж

Авиабагаж.

Библиотечный

Библиотечный объект.

Cleverence.ISO.AfiCollection

Коллекция объектов типа «Afi (Afi)».

Количество (Count)

Возвращает количество элементов в списке.

Элемент (get_Item)

Возвращает элемент по указанному индексу.

Синтаксис: Элемент(<Индекс>);

Добавить (Add)

Добавляет в список новый элемент.

Синтаксис: Добавить(<Элемент>);

Удалить (Remove)

Удаляет из списка указанный элемент.

Синтаксис: Удалить(<Элемент>);

УдалитьПоИндексу (RemoveAt)

Удаляет из списка элемент по указанному индексу.

Синтаксис: УдалитьПоИндексу(<Индекс>);

Имя параметра

Описание

Индекс

Индекс элемента в списке, от 0 до (Количество - 1).

Добавить (Add)

Добавляет в коллекцию указанный код применения.

Синтаксис: **Добавить** (<afi>)

Имя параметра

Описание

afi

AFI (Application Family Identifier, код применения).

Cleverence.ISO.LibraryItem

Содержит записи о библиотечном объекте, пригодные для записи в USER банк RFID-метки согласно стандарту ISO 28560.

СформироватьUSERБанк (ToUSER_BANK)

Создает и заполняет экземпляр USER-банка памяти RFID-метки на основе данного набор записей о библиотечном объекте.

Метод не принимает аргументов.

УникальныйКод (PrimaryItemIdentifier)

Возвращает или устанавливает строковой код библиотечного объекта, уникальный в рамках конкретной библиотеки. Устанавливать не обязательно, т.к. этот же самый код уже должен храниться в банке EPC/UII используемой RFID-метки.

ISIL (ISIL)

Возвращает или устанавливает ISIL библиотеки, которой принадлежит объект. Национальным агентством по присвоению кодов ISIL в России является ГПНТБ. Строка ISIL должна соответствовать ISO 15511.

Наименование (Title)

Возвращает или устанавливает заголовок (название) библиотечного объекта. Можно использовать все символы Юникод.

МестоНаПолке (ShelfLocation)

Возвращает или устанавливает номер полки для хранения библиотечного объекта (строка).

РазмерНабора (SetSize)

Возвращает или устанавливает количество объектов в библиотечном наборе (например, общее число томов в издании). Если это не набор или размер набора неизвестен, то равно нулю.

ИндексВНаборе (SetIndex)

Возвращает или устанавливает номер объекта в библиотечном наборе (например, номер тома). Если это не набор, то равно нулю.

Если это первый элемент набора и при этом известно, что в наборе присутствуют объекты без RFID-меток, то тоже должно быть равно нулю (соотв. у следующего объекта в наборе номер должен быть не 1, а сразу 2).

Cleverence.ISO.LibraryUii

Представляет собой (UII, Unique Item Identifier, уникальный идентификатор) для библиотечного объекта согласно стандарту ISO 28560.

Закодировать (Encode)

Выполняет кодирование UII библиотечного объекта в байты в соответствии со стандартом ISO 28560.

Синтаксис: **Закодировать (<alignToWords>)**

УникальныйКод (PrimaryItemIdentifier)

Возвращает или устанавливает номер библиотечного объекта, уникальный в рамках конкретной библиотеки.

ISIL (ISIL)

Возвращает или устанавливает ISIL библиотеки-владельца объекта. Необязательный параметр. Строку ISIL можно хранить в USER-банке RFID-метки. Национальным агентством по присвоению кодов ISIL в России является ГПНТБ. Строка ISIL должна соответствовать ISO 15511.

ТипИспользования (TypeOfUsage)

Возвращает или устанавливает тип использования объекта. Тип использования определяет, что это: объект фонда, читательский билет или собственное имущество библиотеки (стол, принтер).

IsValid (IsValid)

Возвращает true, если данные банка памяти корректны и соответствуют стандарту.

ErrorString (ErrorString)

Возвращает описание ошибки для некорректных данных банка памяти.

DataElements (DataElements)

Возвращает коллекцию data elements, который были закодированы или требуется закодировать в данном банке.

ApplicationDefined	Доморощенная схема кодирования, определяемая самим приложением (само приложение декодирует).
Integer	Просто число, декодируемое из столько байт, сколько указано в параметре Length.
NumericString	Квартеты бит, интерпретируемые каждый как цифра от 0 до 9.
UppercaseAlphabetic	5-битные латинские символы в верхнем регистре.
UppercaseAlphaNumeric	6-битные латинские символы в верхнем регистре + цифры + еще.
ASCII	7-битный ASCII.
Unaltered8Bit	8-битная кодировка, по умолчанию ISO/IEC 8859-1.
UTF8	Упакованный Unicode, согласно ISO/IEC 10646.

Cleverence.ISO.TypeOfUsage

Представляет собой тип использования библиотечного объекта, согласно ISO 28560.

ДляПродажи	Экземпляр для продажи, любой.
ДляПродажи_Автоматической	Экземпляр для автоматизированной продажи (например, при помощи автомата по продаже).
ДляПродажи_Ручной	Экземпляр для ручной продажи.

ДляПродажи_Зарезервировано	Экземпляр для продажи, конкретное применение не определено, код зарезервирован для использования в будущих стандартах.
ДляВыдачи	Объект фонда, для выдачи на руки, любой.
ДляВыдачи_Автоматической	Объект фонда, для выдачи на руки, подлежит автоматической сортировки.
ДляВыдачи_Ручной	Объект фонда, для выдачи на руки, не подлежит автоматической сортировки.
ДляВыдачи_Зарезервировано	Объект фонда, для выдачи на руки, конкретное применение не определено, код зарезервирован для использования в будущих стандартах.
НеДляВыдачи	Объект фонда, не подлежит выдаче на руки, любой.
НеДляВыдачи_Зарезервировано	Объект фонда, не подлежит выдаче на руки, конкретное применение не определено, код зарезервирован для использования в будущих стандартах.
ВнутреннееИспользование	Для внутреннего использования библиотекой, любое.
ВнутреннееИспользование_Зарезервировано	Для внутреннего использования библиотекой, конкретное применение не определено, код зарезервирован для использования в будущих стандартах.
ВнутреннееИспользование2	Для внутреннего использования библиотекой, любое.
ВнутреннееИспользование2_Зарезервировано	Для внутреннего использования библиотекой, конкретное применение не определено, код зарезервирован для использования в будущих стандартах.
ForFutureUse_Unspecified	Конкретное применение не определено, код зарезервирован для использования в будущих стандартах.
ForFutureUse_FutureUse	Конкретное применение не определено, код зарезервирован для использования в будущих стандартах.
НетИнформации	Информация о типе использования отсутствует.

Списано	Списанный объект, любой.
Списано_ПодлежитПродаже	Списанный объект, подлежит продаже.
Списано_Продано	Списанный объект, продан.
Списано_ПодлежитУтилизации	Списанный объект, для утилизации.
Списано_Зарезервировано	Списанный объект, конкретное применение не определено, код зарезервирован для использования в будущих стандартах.
ЧитательскийБилет	Читательский билет, любой.
ЧитательскийБилет_Взрослый	Читательский билет, взрослый читатель.
ЧитательскийБилет_Подростковый	Читательский билет, подростковый читатель.
ЧитательскийБилет_Детский	Читательский билет для ребенка.
ЧитательскийБилет_Зарезервировано	Читательский билет, конкретное применение не определено, код зарезервирован для использования в будущих стандартах.
Имущество	Библиотечное оборудование, любое.
Имущество_Компьютер	Библиотечное оборудование - компьютер.
Имущество_Видеопроектор	Библиотечное оборудование - видеопроектор.
Имущество_Кинопроектор	Библиотечное оборудование - кинопроектор.
Имущество_Доска	Библиотечное оборудование - доска.
Имущество_Зарезервировано	Библиотечное оборудование, конкретное применение не определено, код зарезервирован для использования в будущих стандартах.

КодКласса (ClassCode)

Возвращает код класса использования библиотечного объекта.

ИмяКласса (ClassName)

Возвращает наименование класса использования библиотечного объекта.

КодИспользования (UsageCode)

Возвращает код типа использования библиотечного объекта.

Описание (UsageDescription)

Возвращает описание типа использования библиотечного объекта.

Cleverence.ISO.Uii

Представляет собой уникальный идентификатор объекта (UII, Unique Item Identifier), в рамках определенной сферы применения, согласно стандарту ISO 15961 (Radio frequency identification (RFID) for item management - Data protocol: application interface).

AFI (AFI)

Возвращает значение кода применения (см. «Afi (Afi)»)

Value (Value)

Возвращает синтетический уникальный код, сформированный на основе данных UII.

Cleverence.RFID.LookupTagParams

Содержит настройки режима инвентаризации, такие как фильтры для меток и AFI.

IsValid (IsValid)

Возвращает true, если данный EPC/UII корректен и соответствует стандарту.

ErrorString (ErrorString)

Возвращает описание ошибки для некорректных EPC/EPC или "Неопределено" для корректных EPC/EPC.

БинарноеПредставление (BinaryString)

Возвращает строку 16-ричного представления бинарного кодирования данного EPC/UII.

Cleverence.RFID.RfidAntennaInfo

Содержит информацию об антенне RFID-считывателя.

Ид (Id)

Возвращает идентификационный номер (код) антенны.

Разрешена (IsEnabled)

Возвращает флаг того, что порт антенны с таким номером активен (используется). Или "Неопределено", если подключение к считывателю происходило без логина/пароля и состояние антенны неопределено.

Подсоединена (IsConnected)

Возвращает флаг того, что антенна с таким номером физически подключена к порту. Или "Неопределено", если подключение к считывателю происходило без логина/пароля и состояние антенны неопределено.

КоэффициентНаправленности (Gain)

Возвращает коэффициент направленного действия для подключенной антенны в dBi (число).

ВходнаяМощность (TransmitPower)

Возвращает входную мощность, подаваемую на данную антенну, в dBm (число).

Cleverence.RFID.RfidAntennaInfoCollection

Коллекция объектов типа «RfidAntennaInfo (RfidAntennaInfo)».

Количество (Count)

Возвращает количество элементов в списке.

Элемент (get_Item)

Возвращает элемент по указанному индексу.

Синтаксис: Элемент(<Индекс>);

Добавить (Add)

Добавляет в список новый элемент.

Синтаксис: Добавить(<Элемент>);

Удалить (Remove)

Удаляет из списка указанный элемент.

Синтаксис: Удалить(<Элемент>);

УдалитьПоИндексу (RemoveAt)

Удаляет из списка элемент по указанному индексу.

Синтаксис: УдалитьПоИндексу(<Индекс>);

Имя параметра

Описание

Индекс

Индекс элемента в списке, от 0 до (Количество - 1).

Cleverence.RFID.RfidChip

Содержит описание функциональных возможностей конкретной модели чипа RFID-метки.

Cleverence.RFID.RfidReader

Содержит методы по работе со стационарным RFID-считывателем.

Подключить (Connect)

Выполняет подключение к RFID-считывателю (с возможностью чтения/записи, но без возможности управления параметрами считывателя, см. «ПодключитьПодПаролем (ConnectWithPassword)»).

Метод не принимает аргументов.

ПодключитьПодПаролем (ConnectWithPassword)

Выполняет подключение к RFID-считывателю с возможностью как чтения/записи, так и управления параметрами считывателя).

Синтаксис: ПодключитьПодПаролем (<userName>, <password>)

Имя параметра

Описание

userName

Имя пользователя для подключения.

password

Пароль пользователя.

ПолучитьВозможности (GetCapabilities)

Получает и возвращает конфигурацию RFID-считывателя.

Метод не принимает аргументов.

ИнвентаризоватьМетки (LookupTags)

Выполняет поиск и чтение Tag ID меток в радиусе видимости считывателя в течение определенного времени.

Очищает очередь считанных меток.

Синтаксис: **ИнвентаризоватьМетки** (<readTime>)

Имя параметра

Описание

readTime

Количество времени в миллисекундах, в течение которого считывателю следует искать метки.

НачатьИнвентаризацию (LookupTagsBegin)

Начинает поиск и чтение Tag ID меток в радиусе видимости считывателя в течение определенного времени. Функция выполняется асинхронно и возвращает управление вызывающей стороне сразу после начала чтения. Считанные метки приходят в событии «TagRead (TagRead)».

Очищает очередь считанных меток.

Синтаксис: **НачатьИнвентаризацию** (<readTime>)

Имя параметра

Описание

readTime

Количество времени в миллисекундах, в течение которого считывателю следует искать метки.

ОкончитьИнвентаризацию (LookupTagsEnd)

Прерывает работу по чтению меток, инициированную вызовом функции «НачатьИнвентаризацию (LookupTagsBegin)».

Возвращает всё, что было считано (не только из очереди считанных меток, а вообще все метки), и очищает очередь считанных меток.

Метод не принимает аргументов.

ВыбратьМетку (FetchTag)

Вынимает из очереди считанных меток данные метки (меток) с указанным Tag ID.

Т.к. метки вынимаются из очереди, второй и последующие вызовы метода с одним и тем же Tag ID могут вернуть пустой результат.

Поведение зависит от флага, разрешающего повторные чтения. При повторных чтениях метки могут снова оказаться в очереди.

Синтаксис: **ВыбратьМетку** (<tagid>)

Имя параметра

Описание

tagId

Tag ID интересующих меток.

Выбрать Метки (FetchTags)

Вынимает из очереди считанных меток данные всех меток.

Т.к. метки вынимаются из очереди, второй и последующие вызовы метода могут вернуть пустой результат.

Поведение зависит от флага, разрешающего повторные чтения. При повторных чтениях метки могут снова оказаться в очереди.

Метод не принимает аргументов.

Прочитать Банк EPCUII (ReadEPCUII)

Производит чтение EPC/UII-банка (банка 01) первой попавшейся RFID-метки с указанным значением Tag ID, с указанием пароля на доступ (Access Password, хранящийся в банке 00 RFID-метки). Если пароля нет, то следует указать 0.

Синтаксис: **ПрочитатьБанкEPCUII (<tagId>, <accessPassword>)**

Имя параметра

Описание

tagId

Tag ID метки для чтения.

accessPassword

Число в 32 бита, задающее пароль на доступ к метке. Если пароля нет, то следует указать 0.

Записать EPCUII (WriteEPCUII)

Производит запись в EPC/UII-банк (банк 01) первой попавшейся RFID-метки с указанным значением Tag ID, с указанием пароля на доступ (Access Password, хранящийся в банке 00 RFID-метки). Если пароля нет, то следует указать 0.

Синтаксис: **ЗаписатьEPCUII (<tagId>, <epcuii>, <accessPassword>)**

Имя параметра

Описание

tagId

Tag ID метки для чтения.

epcuii

Записываемый EPC/UII - электронный код товара или уникальный идентификатор объекта.

accessPassword

Число в 32 бита, задающее пароль на доступ к метке. Если пароля нет, то следует указать 0.

ЗаписатьEPCUIIпоTID (WriteEPCUIIforTID)

Производит запись в EPC-банк (банк 01) первой попавшейся RFID-метки с указанным значением Tag ID и содержимым банка TID (уникальный номер чипа, который, в отличие от Tag ID действительно всегда уникален). С указанием пароля на доступ (Access Password, хранящийся в банке 00 RFID-метки). Если пароля нет, то следует указать 0.

Синтаксис: **ЗаписатьEPCUIIпоTID (<tagId>, <tid>, <epcuii>, <accessPassword>)**

Имя параметра	Описание
tagId	Tag ID нужной метки для записи.
tid	Содержимое банка TID нужной метки для записи.
epcuii	Записываемый EPC/UII - электронный код товара или уникальный идентификатор объекта.
accessPassword	Число в 32 бита, задающее пароль на доступ к метке. Если пароля нет, то следует указать 0.

ПрочетьБанкRESERVED (ReadRESERVED)

Производит чтение RESERVED-банка (банка 00) первой попавшейся RFID-метки с указанным значением Tag ID, с указанием пароля на доступ (Access Password, хранящийся в банке 00 RFID-метки). Если пароля нет, то следует указать 0.

Синтаксис: **ПрочетьБанкRESERVED (<tagId>, <accessPassword>)**

Имя параметра	Описание
tagId	Tag ID метки для чтения.
accessPassword	Число в 32 бита, задающее пароль на доступ к метке. Если пароля нет, то следует указать 0.

ПрочетьБанкTID (ReadTID)

Производит чтение TID-банка (банка 10) первой попавшейся RFID-метки с указанным значением Tag ID, с указанием пароля на доступ (Access Password, хранящийся в банке 10 RFID-метки). Если пароля нет, то следует указать 0.

Синтаксис: **ПрочетьБанкTID (<tagId>, <accessPassword>)**

Имя параметра	Описание
tagId	Tag ID метки для чтения.

accessPassword	Число в 32 бита, задающее пароль на доступ к метке. Если пароля нет, то следует указать 0.
----------------	--

ReadTIDs (ReadTIDs)

Производит чтение TID-банка (банка 10) всех RFID-меток в поле видимости считывателя, с указанием пароля на доступ (Access Password, хранящийся в банке 10 RFID-метки). Если пароля нет, то следует указать 0.

Синтаксис: **ReadTIDs** (<readTime>, <accessPassword>)

Имя параметра	Описание
readTime	Количество времени в миллисекундах, в течение которого считывателю следует искать метки.
accessPassword	Число в 32 бита, задающее пароль на доступ к метке. Если пароля нет, то следует указать 0.

ПрочетьБанкUSER (ReadUSER)

Производит чтение USER-банка (банка 11) первой попавшейся RFID-метки с указанным значением Tag ID, с указанием пароля на доступ (Access Password, хранящийся в банке 11 RFID-метки). Если пароля нет, то следует указать 0.

Синтаксис: **ПрочетьБанкUSER** (<tagId>, <accessPassword>)

Имя параметра	Описание
tagId	Tag ID метки для чтения.
accessPassword	Число в 32 бита, задающее пароль на доступ к метке. Если пароля нет, то следует указать 0.

ReadUSERS (ReadUSERS)

Производит чтение USER-банка (банка 11) всех RFID-меток в поле видимости считывателя, с указанием пароля на доступ (Access Password, хранящийся в банке 11 RFID-метки). Если пароля нет, то следует указать 0.

Синтаксис: **ReadUSERS** (<readTime>, <accessPassword>)

Имя параметра	Описание
readTime	Количество времени в миллисекундах, в течение которого

считывателю следует искать метки.

accessPassword

Число в 32 бита, задающее пароль на доступ к метке. Если пароля нет, то следует указать 0.

ВключитьАнтенну (EnableAntenna)

Активирует (включает) использование антенны с указанным номером (кодом).

Синтаксис: **ВключитьАнтенну (<antennaId>)**

Имя параметра

Описание

antennaId

Номер (код) антенны согласно Cleverence.RFID.RfidAntennaInfo.

ВыключитьАнтенну (DisableAntenna)

Деактивирует (выключает) использование антенны с указанным номером (кодом).

Синтаксис: **ВыключитьАнтенну (<antennaId>)**

Имя параметра

Описание

antennaId

Номер (код) антенны согласно Cleverence.RFID.RfidAntennaInfo.

УстановитьВходнуюМощностьДляАнтенны (SetTransmitPower)

Устанавливает входную мощность для антенны с указанным номером (кодом).

Синтаксис: **УстановитьВходнуюМощностьДляАнтенны (<antennaId>, <powerLevel>)**

Имя параметра

Описание

antennaId

Номер (код) антенны согласно Cleverence.RFID.RfidAntennaInfo.

powerLevel

Требуемая мощность в процентах от максимальной (от 1 до 100).

Отключить (Disconnect)

Выполняет отключение от считывателя. Ничего не принимает и ничего не возвращает.

Метод не принимает аргументов.

LookupTagParams (LookupTagParams)

Возвращает настройки проведения инвентаризации (слежения за метками) для данного RFID-считывателя. Они могут добавлять/перекрывать общие настройки для всех считывателей, указанные в «LookupTagParams (LookupTagParams)».

Имя (DisplayName)

Возвращает отображаемое имя RFID-считывателя.

Ид (Id)

Возвращает идентификатор RFID-считывателя.

Url (Url)

Возвращает URL до RFID-считывателя.

Модель (Model)

Возвращает модель RFID-считывателя.

Подключен (IsConnected)

Возвращает true (Истина), если подключение к RFID-считывателю активно.

ИдетИнвентаризация (LookupTagsInProgress)

Возвращает true (Истина), если считыватель в настоящее время выполняет инвентаризацию.

РазрешатьПовторныеЧтения (AllowRepetitiveReads)

Возвращает или устанавливает флаг, указывающий библиотеке, следует ли ему при чтении возвращать повторные считывания. Если флаг не выставлен - все события и данные об индивидуальных считываний приходят отдельно. Если флаг выставлен - все события и данные группируются/объединяются по Tag ID.

Cleverence.RFID.RfidReaderCapabilities

Содержит информацию об оснащении и возможностях RFID-считывателю.

Антенны (Antennas)

Возвращает коллекцию описаний антенн RFID-считывателя.

Cleverence.RFID.RfidReaderCollection

Коллекция объектов типа «RfidReader (RfidReader)».

Количество (Count)

Возвращает количество элементов в списке.

Элемент (get_Item)

Возвращает элемент по указанному индексу.

Синтаксис: Элемент(<Индекс>);

Добавить (Add)

Добавляет в список новый элемент.

Синтаксис: Добавить(<Элемент>);

Удалить (Remove)

Удаляет из списка указанный элемент.

Синтаксис: Удалить(<Элемент>);

УдалитьПоИндексу (RemoveAt)

Удаляет из списка элемент по указанному индексу.

Синтаксис: УдалитьПоИндексу(<Индекс>);

Имя параметра	Описание
Индекс	Индекс элемента в вписке, от 0 до (Количество - 1).
ДобавитьПоUrl (AddByUrl)	
	Добавляет в коллекцию RFID-считыватель с указанным Url.

Синтаксис: **ДобавитьПоUrl** (<url>)

Имя параметра	Описание
url	Url для подключения к RFID-считывателю.

Cleverence.RFID.RfidTag

Содержит данные о RFID-метке на основании операции инвентаризации окружающих меток RFID-считывателем.

TagId (TagId)

Возвращает Tag ID метки 16-ричном представлении (строка в 24 символа).

Считыватель (Reader)

Возвращает считыватель, при помощи которого была считана данная метка.

Объект (Identity)

Возвращает значение EPC/UII метки (если метка закодирована в соответствии со стандартом EPCglobal или ISO), полученный на основании операции инвентаризации окружающих меток RFID-считывателем.

НомерАнтенны (AntennaId)

Возвращает номер (код) антенны, которая прочла метку с таким Tag ID.

Время (FirstTimeSeen)

Возвращает дату/время, в которое метка с таким Tag ID была увидена впервые (по часам компьютера, на котором работает Api)

Счетчик (SeenCount)

Возвращает сколько раз была замечена метка с таким Tag ID.

Фактически, для неподвижно лежащих меток это число отражает количество меток с разным номером чипа (TID), но одинаковым Tag ID (одинаковым EPC). Для движущихся меток сюда добавляется количество входов/выходов таких меток за пределы области чтения.

RSSI (PeakRSSI)

Возвращает пиковое значение принятого уровня сигнала от метки в произвольных единицах от 0 до 255 (число).

Cleverence.RFID.RfidTagCollection

Коллекция объектов типа «RfidTag (RfidTag)».

Количество (Count)

Возвращает количество элементов в списке.

Элемент (get_Item)

Возвращает элемент по указанному индексу.

Синтаксис: Элемент(<Индекс>);

Добавить (Add)

Добавляет в список новый элемент.

Синтаксис: Добавить(<Элемент>);

Удалить (Remove)

Удаляет из списка указанный элемент.

Синтаксис: Удалить(<Элемент>);

УдалитьПоИндексу (RemoveAt)

Удаляет из списка элемент по указанному индексу.

Синтаксис: УдалитьПоИндексу(<Индекс>);

Имя параметра

Описание

Индекс

Индекс элемента в списке, от 0 до (Количество - 1).

Добавить (Add)

Добавляет в коллекцию метку с указанным Tag ID.

Синтаксис: **Добавить (<tagId>)**

Имя параметра

Описание

tagId

Tag ID метки в виде строки в 16-ричном формате.

Cleverence.RFID.VirtualRfidTag

Содержит данные о RFID-метке на основании операции инвентаризации окружающих меток RFID-считывателем.