

**Система управления подключаемым  
оборудованием для 1С:Предприятие 8.2  
по технологии 1С-Рарус**

## Оглавление

<b>1. ЦЕЛЬ ПРОЕКТА.....</b>	<b>3</b>
<b>2. ТРЕБОВАНИЯ ПО ПРОЕКТУ .....</b>	<b>3</b>
<b>3. ОСНОВНЫЕ ПОНЯТИЯ И ТЕРМИНЫ .....</b>	<b>4</b>
<b>4. ПРИНЦИПИАЛЬНЫЕ РЕШЕНИЯ И ОГРАНИЧЕНИЯ .....</b>	<b>6</b>
<b>5. ОПИСАНИЕ СИСТЕМЫ .....</b>	<b>7</b>
5.1. Открытость и расширяемость .....	7
5.2. Типы оборудования .....	8
5.3. Документы описания системы .....	8
5.4. Модели оборудования .....	8
5.5. Общая структура системы.....	8
5.6. Возможные клиентские приложения.....	10
5.7. Демонстрационная макетная конфигурация.....	11
5.8. Построение менеджера управления оборудованием .....	11
5.9. Драйвер оборудования и вызов его методов .....	11
<b>6. ИСПОЛЬЗУЕМЫЕ ТИПЫ ДАННЫХ .....</b>	<b>12</b>
String .....	12
Number.....	12
Boolean.....	12
DateTime .....	13
<b>7. ФУНКЦИИ ОБРАТНОГО ВЫЗОВА.....</b>	<b>13</b>
<b>8. ОСНОВНЫЕ АЛГОРИТМЫ.....</b>	<b>13</b>
8.1. Создание нового экземпляра оборудования .....	13
8.2. Использование экземпляра оборудования.....	14
8.3. Удаление экземпляра оборудования.....	14

# 1. Цель проекта

Разработать систему управления подключаемым оборудованием на основе открытого, публикуемого описания ее технологии. Система должна быть пригодна для использования в любых прикладных решениях, разработанных на платформе 1С:Предприятие 8.2. В том числе как для типовых решений фирмы 1С и так для собственных разработок программных продуктов партнеров.

# 2. Требования по проекту

№	Содержание требования
1	Необходимо обеспечить расширяемость, открытость системы и унификацию ее использования
1.1	Партнеры, конечные пользователи, разработчики или поставщики торгового оборудования должны иметь возможность добавить в систему новую модель оборудования, опираясь лишь на документацию и описания данной технологии
1.2	Внутренний код приложения, использующего данную систему, должен быть полностью абстрагирован от особенностей конкретной модели оборудования. Пользователи приложения, декларирующего поддержку описанного в системе типа оборудования, смогут использовать драйвер любой модели этого типа оборудования, созданный любым производителем согласно этой документации к системе
1.3	Любой драйвер оборудования, созданный в строгом соответствии с данной документацией должен быть работоспособен при его использовании из приложений следующих типов: конфигурация для платформы «1С:Предприятие», Windows-приложение, Windows-сервис
2	Необходимо обеспечить возможность управления с одного рабочего места оборудованием, подключенным к компьютерам других рабочих мест при условии, что компьютеры всех рабочих мест объединены в локальную сеть.
2.1	Для управления удаленным оборудованием должны использоваться только стандартные средства сетевого взаимодействия (протокол TCP-IP и/или его стандартные расширения) и описанные в стандарте методы. Для передачи параметров или данных при взаимодействии с удаленным оборудованием не должна использоваться среда платформы «1С:Предприятие» или другие внутренние механизмы приложения.
2.2	Оборудование, настроенное и используемое на рабочем месте, может быть использовано позднее в удаленном режиме, без необходимости его перенастройки
2.3	Возможность работы с удаленным оборудованием не должна усложнять подключение и настройку оборудования пользователем на рабочем месте
2.4	Основные методы взаимодействия с удаленным оборудованием должны быть полностью аналогичны методам, применяемым для локального оборудования. Клиентское приложение должно взаимодействовать с удаленным оборудованием точно так же, как и с локальным, не различая их, используя единый интерфейс.
3	При работе с оборудованием в различных режимах взаимодействия, должен использоваться единый набор методов
3.1	Должна поддерживаться синхронная работа с драйверами оборудования. В этом режиме возврат управления в клиентское приложение, вызвавшее метод драйвера, производится по завершении выполнения метода.
3.2	Для многопоточных приложений предусматривается возможность работы с оборудованием в асинхронном режиме. При этом клиентское приложение должно самостоятельно обеспечить вызов метода драйвера в отдельном потоке, контролировать его завершение и получить результат. Менеджер оборудования такого приложения должен поддерживать управление очередью команд к оборудованию.
4	Должен быть разработан единый унифицированный механизм установки и настройки системы управления оборудованием на рабочем месте.
4.1	Установка и настройка нового оборудования должна быть доступна пользователю непосредственно из интерфейса клиентского приложения, без необходимости запуска дополнительных приложений
4.2	Создание нового экземпляра оборудования должно производиться с использованием формы удобного пошагового мастера установки
4.3	Настройка оборудования не должна требовать от пользователя дополнительных знаний и навыков, за исключением тех случаев, когда обойтись без этого невозможно
5	Система управления оборудованием должна иметь минимальный набор интерфейсов

	взаимодействия с прикладным решением (т.н. «точку подключения»). Это позволит максимально упростить её интеграцию с любым приложением.
5.1	Система управления оборудованием не должна использовать вызовы любых методов и обращения к любым свойствам приложения, кроме интерфейсных методов, описанных в данном стандарте.
5.2	Ошибки, возникающие при работе устройства или драйвера, не должны приводить к критическим сбоям и нарушению нормальной работы использующего его клиентского приложения
5.3	Приложение должно однозначно идентифицировать все используемые им экземпляры оборудования, для обеспечения привязки устройств к объектам бизнес-логики приложения
6	Система управления оборудованием должна поддерживать совместимость «сверху вниз» с предыдущими версиями данного стандарта на уровне описанных в нем типов оборудования
7	Система не должна вносить ограничений на использование локализованных версий клиентского приложения и операционной системы, установленной на компьютере рабочего места.
8	Все оборудование, подключенное к компьютеру рабочего места, должно быть доступно для использования нескольким экземплярам клиентских приложений, установленными на компьютере. Ограничение: В один момент времени одним устройством можно управлять только из одного приложения. В случае попытки использования этого устройства другими приложениями, система должна выдавать штатную корректную ошибку.
9	В целях поиска и устранения неисправностей и ошибок, драйверы устройств должны иметь средства для записи лога работы
10	В процессе обмена данными между приложением и драйвером устройства, вся текстовая информация передается в кодировке UTF-16. При сохранении настроек оборудования в файл, система также должна использовать эту кодировку.

### 3. Основные понятия и термины

Термин	Описание
Оборудование	Любое торговое или промышленное оборудование, использующееся клиентским приложением
Тип оборудования	Регламентированное стандартом множество, объединяющее устройства с одинаковым функциональным назначением
Драйвер оборудования, Драйвер	Программный компонент с возможностью отдельной установки или обновления в системе и предоставляющий интерфейс управления и взаимодействия с устройством определенной модели. Предоставляемый драйвером интерфейс должен строго соответствовать описанию типа оборудования по данной технологии, но может иметь расширения (дополнительные методы и/или параметры) для выполнения функций, не регламентированных в документах описания системы
Модель, Модель оборудования	Множество устройств одного типа оборудования, одинаковых или близких по своим характеристикам, управляемых драйвером устройства. Модель оборудования определяется драйвером устройства. В случае если один драйвер устройства позволяет единообразно управлять одной из нескольких сходных моделей физических устройств, для системы все эти устройства являются одной и той же моделью. Напротив, если для одного и того же физического устройства существует несколько разных драйверов, для системы они являются разными моделями.
Устройство, Физическое устройство	Физический прибор, аппарат, управляемый драйвером, непосредственно выполняющий некие физические действия (печать текста, считывание штрих-кода, получение веса и пр.)
Экземпляр оборудования	Совокупность файлов и данных о физическом устройстве и драйвере (идентификатор, наименование, список настроек и т. п.), хранимых и используемых для работы с данным устройством системой управления подключаемым оборудованием и приложением-клиентом. Экземпляр оборудования создается на основе модели оборудования
Система управления	Совокупность установленных на компьютере рабочего места

Термин	Описание
подключаемым оборудованием	приложений, библиотек и файлов, предназначенная для управления оборудованием из любого приложения, поддерживающего данный стандарт.
Клиентское приложение	Любое приложение, использующее для управления подключаемым оборудованием описанные в данном стандарте интерфейсы и алгоритмы взаимодействия с драйверами устройств.
Менеджер оборудования	Обособленная часть клиентского приложения или отдельное приложение, в которой сосредоточены сервисные процедуры и регламентированные данным стандартом интерфейсы взаимодействия с оборудованием.
Сетевой транспорт	Дополнительная необязательная часть системы, взаимодействующая с менеджером оборудования и расширяющая его возможности. Предназначен для обеспечения взаимодействия между менеджерами оборудования, функционирующими как на разных компьютерах локальной сети, так и на одном и том же компьютере. Сетевой транспорт предназначен для организации соединений и передачи команд и данных по сети с использованием протокола TCP/IP.
Настройки устройства	Набор именованных значений, простых типов данных: String, Number, DateTime, Boolean, хранящийся вне клиентского приложения в файловом хранилище настроек. Для каждого определенного в системе экземпляра оборудования имеется свой набор настроек. Настройки устройства определяют параметры и режим функционирования устройства. Настройки доступны клиентскому приложению для чтения и записи с помощью специальных методов.
Хранилище настроек	Составная часть системы управления подключаемым оборудованием, которая предназначена для хранения следующей информации: - Списка установленных в системе моделей оборудования - Настройки всех экземпляров локального оборудования (настройки хранятся в файлах формата XML)
Локальное оборудование	Экземпляры оборудования, физически подключенные к данному компьютеру.
Уникальный идентификатор	В контексте этого проекта – строковое представление GUID. Идентификатор содержит 36 символов в верхнем регистре, не заключенных в фигурные скобки. Пример: 8A8910EC-78F5-41A5-9E18-7C28992CF580
Удаленное оборудование	Экземпляры оборудования, подключенные к удаленному по отношению к клиентскому приложению компьютеру в пределах одной локальной сети.
Публикуемый метод, Команда оборудования	Один из определенных в описании интерфейса общих методов, либо методов типа. Все публикуемые методы имеют одинаковую структуру параметров. Входные и возвращаемые методом данные помещаются в два одномерных массива SafeArray. Клиентское приложение для управления оборудованием может использовать только публикуемые методы (команды), транслируемые менеджером оборудования непосредственно драйверу устройства. Служебные методы используются только менеджером оборудования
Входные параметры команды	Набор исходных данных необходимый и достаточный для выполнения команды. Все передаваемые данные помещаются в одномерный массив SafeArray. Допустимо использование вложенных массивов. Также допустимо заполнение не всех полей массива, а только обязательных параметров команды. Если несколько последних параметров являются необязательными, допускается отсутствие их в массиве (размер массива может быть уменьшен на их число). Команды могут не иметь входных параметров. В таком случае, в качестве входных параметров передается пустое значение (Неопределенно / Undefined)
Результаты команды,	Набор данных, возвращаемых после исполнения команды

Термин	Описание
Возвращаемые параметры	оборудованием. Результаты команды помещаются в одномерный массив SafeArray. Допустимо использование вложенных массивов. Также допустимо заполнение не всех полей массива, а только обязательных параметров команды. Если несколько последних параметров являются необязательными, допускается отсутствие их в массиве (размер массива может быть уменьшен на их число). Команды могут не иметь выходных параметров. В таком случае, в качестве выходных параметров передается пустое значение (Неопределенно / Undefined)
Функции обратного вызова	Регламентируемые данным стандартом функции менеджера оборудования, доступные для вызова через его интерфейс IDispatch драйвером оборудования и сетевым транспортом. Функции обратного вызова предназначены для информирования приложения о событиях или передачи данных приложению
Событие	Событие возникает как результат изменения состояния устройства. Событие может быть вызвано внешним воздействием, либо внутренними процессами в устройстве. Для передачи событий приложению драйвер устройства использует функции обратного вызова менеджера оборудования
Объект-абстракт	Объект драйвера устройства, не ассоциированный с экземпляром оборудования. Любой объект драйвера является объектом-абстрактом до выполнения метода Open. У объекта-абстракта допустим вызов только тех методов, которые не требуют взаимодействия с физическим устройством. Объект-абстракт не может использовать функции обратного вызова, поскольку не имеет собственного идентификатора в системе.
Объект драйвера	Объект драйвера устройства, созданный клиентским приложением. Для каждого физического устройства создается отдельный экземпляр объекта драйвера
Идентификатор команды	Уникальный идентификатор команды. Используется при взаимодействии с удаленным оборудованием.

#### 4. Принципиальные решения и ограничения

№	Содержание решения
1	Расширяемость системы будет общедоступной на уровне моделей оборудования. Для создания драйвера новой модели оборудования, достаточно руководствоваться документами стандарта. На уровне типов оборудования расширение стандарта возможно только при условии согласования с фирмой 1С-Рарус. В этом случае обязательна публикация новой версии стандарта. Будет введена процедура сертификации оборудования, разработанного партнерами на соответствие стандарту.
2	Открытость системы будет обеспечиваться публикацией ее документации и свободным распространением. Каждая новая версия системы должна быть совместима с предыдущими реализациями по принципу «снизу-вверх», то есть ранее реализованный функционал будет поддерживаться следующими версиями, а новые возможности системы не должны препятствовать использованию в ней отдельных частей, реализованных по описанию прошлых версий системы
3	Развитие системы происходит в целом, а не отдельными частями. Несмотря на то, что описание системы будет состоять из нескольких документов, не предполагается вести их раздельное версионирования. При изменении любого из них, всегда будет публиковаться полный набор документов описания системы
4	Параметры и результаты выполнения команд должны быть ограничены простыми типами данных - String, Number, DateTime, Boolean. Необходимость такого ограничения обусловлена следующими причинами: <ul style="list-style-type: none"> <li>• Драйверы устройств могут быть созданы с использованием любого языка программирования и платформы разработки. Сложные типы данных могут не поддерживаться языком или платформой.</li> <li>• Управление удаленными устройствами осуществляется через локальную сеть. Передача по сети сложных типов данных затруднена</li> </ul>

5	Для обеспечения расширяемости списка параметров при развитии системы и упрощения реализации обмена данными по сети, параметры всех публикуемых методов унифицированы, входные и выходные данные помещаются в одномерные массивы SafeArray.
6	Для обеспечения: <ul style="list-style-type: none"> <li>• Совместимости между объектами системы, созданных с использованием разных языков программирования</li> <li>• Независимости от локализации операционной системы</li> <li>• Преимущества со стандартом оборудования UPOS</li> </ul> Решено ограничить наименования настроек и методов, описанных в стандарте латинским алфавитом.
7	Драйвер устройства, реализованный по технологии COM должен использовать модель работы с потоками Apartment (следует из требования 3.2)
8	Для обеспечения простоты установки системы будет использоваться специальный инсталлятор, обеспечивающий автоматическое развертывания и обновления системы на компьютере рабочего места. Инсталлятор позволяет установить систему на локальный компьютер при условии наличия у пользователя прав администратора.
9	Инсталлятор поддерживает установку драйверов оборудования из пакетов, сформированных специальным образом. Для партнеров и разработчиков будет предоставлена методика и программа для создания инсталляционных пакетов драйверов.
10	Все перечислимые значения в стандарте представляются натуральными числами
11	Описание моделей и настройки экземпляров оборудования хранятся в файлах формата xml. Теги и имена атрибутов состоят из символов латинского алфавита. Если имя состоит из нескольких слов, каждое отдельное слово начинается с большой буквы. Пример: TypeName. Исключение: идентификаторы, в которых есть сокращение ID (от Identifier) – обе буквы в верхнем регистре. Примеры: DecicelID , ModelID
12	Для идентификации уникальных объектов в системе используются <a href="#">уникальные идентификаторы</a> – GUID'ы. В текстовом представлении такой идентификатор содержит 36 символов в верхнем регистре, не заключенных в фигурные скобки. Пример: 8A8910EC-78F5-41A5-9E18-7C28992CF580
13	Коды ошибок выполнения команд драйверами представляют собой линейный список, разбитый на диапазоны по типам оборудования. Это обеспечивает возможность обработки приложением ошибок и локализации сообщений об ошибках.
14	Драйвер устройства не должен в процессе работы отображать какие-либо интерфейсные формы и диалоги, за исключением окна настроек, которое выводится при вызове метода ShowSettingsDlg. В случае необходимости, следует использовать функции обратного вызова MessageDlg и InputDataDlg, для организации диалога с пользователем. Это условие продиктовано следующими обстоятельствами: <ul style="list-style-type: none"> <li>• Возможна потеря фокуса окном 1С-Предприятие. Критично для форм АРМ</li> <li>• Необходимо обеспечить единство стиля и эргономики форм в приложении</li> <li>• В случае если менеджер реализован как Windows Service, отображение форм драйверами становится невозможным</li> </ul>

## 5. Описание системы

### 5.1. Открытость и расширяемость

Открытость и расширяемость системы обеспечивается публикацией документации по ее технологии в интернете. Предполагается детальное освещение системы и ее дальнейшее развитие путем предложения ее использования в продуктах партнеров, широким обсуждением и сбором предложений по развитию от партнерской среды и поставщиков оборудования. Детальное документирование всех составных частей системы и публикуемых в комплекте программных объектов позволит разработчикам конфигураций включать в свои продукты поддержку разных типов оборудования и использовать эту технологию в разной степени. Открытость системы на всех уровнях должна поддерживать взаимозаменяемость ее различных программных компонентов их функциональными аналогами, реализованными с использованием альтернативных технологий.

- Пример 1: Использовать только один фискальный регистратор и только на локальном рабочем месте.

- Пример 2: Создать уникальную модель оборудования, и конфигурацию, использующую ее уникальные возможности, наряду с моделями, созданными в строгом соответствии с текущим опубликованным описанием технологии
- Пример 3: Реализовать конфигурацию, использующую стандартные типы и модели оборудования, но имеющую собственный блок настройки оборудования и сетевого взаимодействия.

Планируется расширение стандарта выпуском новых его версий, расширяющих возможности ранее стандартизованных типов оборудования, и добавляющих к стандарту новые типы оборудования. Новые версии планируется выпускать после сбора требований и пожеланий от партнерской среды 1С-Парус и поставщиков оборудования.

## 5.2. Типы оборудования

Типы оборудования, описанные в документации по системе, являются его основным наполнением. Каждый тип оборудования представляет собой описание открытого множества функционально общих по целевому назначению устройств и способов взаимодействия между ним и клиентским приложением.

В целях оптимизации проектирования, каждый тип оборудования вынесен в отдельный проект и будет в дальнейшем развиваться в виде отдельного проекта. В каждом проекте должны быть описаны требования к конкретному типу оборудования, описание его функционирования, набор методов и параметров. При развитии системы, каждый тип оборудования расширяется с учетом обеспечения совместимости снизу вверх, в пределах мажорной версии системы и реализованных в соответствии с документацией драйверов устройств (см. [принципиальные решения](#) 1-3).

## 5.3. Документы описания системы

Проект системы целесообразно разделить на отдельные документы, описывающие его функционально-независимые части. При публикации очередной версии описания системы, будут всегда публиковаться все составляющие ее документы.

### Список документов текущей версии стандарта:

- Основной документ стандарта – Система управления подключаемым оборудованием по технологии 1С-Парус.doc
- Описание системы установки и настройки – Установка и обновление системы управления внешним оборудованием.doc
- Описание типа оборудования Сканер - Сканер.doc
- Описание типа оборудования Дисплей – Дисплей покупателя.doc
- Описание типа оборудования Фискальный регистратор - Фискальный регистратор.doc
- Описание типа Эквайринг-терминал - Эквайринг-терминал.doc
- Регламентированный список кодов ошибок оборудования - Коды ошибок и их диапазоны.xls

## 5.4. Модели оборудования

Каждый тип оборудования расширяется понятием модели оборудования, которое используется для разделения и идентификации драйверов оборудования по различным производителям оборудования одного типа и/или по принципиальным различиям в их реализации. Например, одним и тем же устройством можно управлять через драйверы от разных производителей - они также будут различаться моделями (модель от «Фирмы А» и модель от «Партнера Б»). Для обеспечения совместимости в рамках стандарта, драйвер любой модели должен иметь реализацию всех описанных в описании ее типа настроек и методов. Если какая-либо модель не поддерживает один или несколько необязательных методов своего типа оборудования, при их вызове драйвер должен возвращать соответствующую ошибку (1003 - Команда не поддерживается данной моделью оборудования).

## 5.5. Общая структура системы

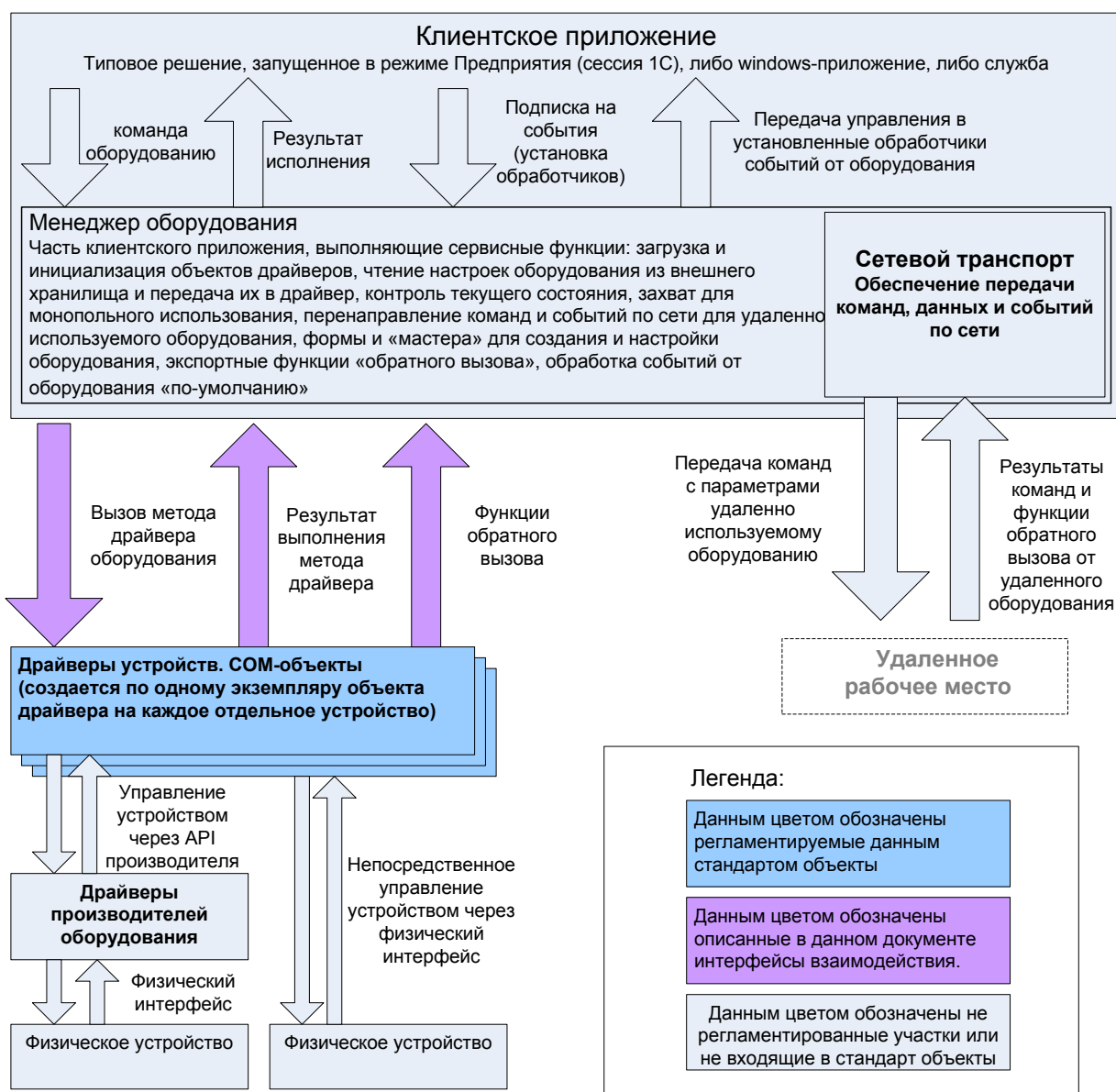
Реализуется 2-х уровневая модель управления, состоящая из:

- Клиентского приложения, использующего для управления подключаемым оборудованием описанные в данном стандарте интерфейсы и алгоритмы взаимодействия с драйверами оборудования
- Драйверов оборудования, каждый из которых индивидуально обслуживает одно устройство. Все методы драйверов оборудования стандартизованы. Так же, драйверы



могут породить события, отправляемые клиентскому приложению с помощью функций обратного вызова.

Общая схема взаимодействия представлена на рисунке:



## 5.6. Возможные клиентские приложения

Клиентское приложение может быть нескольких типов:

- Типовое решение на базе платформы 1С-Предприятие v8 с «толстыми» формами.
- Типовое решение на базе платформы 1С-Предприятие v8 в режиме Управляемого приложения
- Стороннее приложение (Windows Application или Windows Service)

В первом случае менеджер оборудования может быть реализован в виде обработки или внешней компоненты. Во втором случае – в виде общего модуля (исполняемого на клиенте) или внешней компоненты. Во всех случаях можно использовать менеджер оборудования, реализованный как СОМ-сервер. В случае если менеджер реализован на платформе 1С-Предприятие, существует возможность модификации менеджера под нужды конкретной конфигурации. В случае если менеджер создан в виде СОМ-сервера, приложение получает возможность асинхронного, параллельного выполнения команд оборудования в отдельных потоках.

При разработке приложений, в качестве менеджера управления оборудованием может использоваться либо поставляемый в составе стандарта СОМ-сервер, либо самостоятельно реализованный разработчиком приложения аналог, обеспечивающий описанную в данном документе функциональность управления устройствами и передачу команд и событий.

Любое клиентское приложение, использующее драйверы оборудования, соответствующие стандарту, должно обеспечивать регламентированные стандартом интерфейсы и алгоритмы взаимодействия с драйверами и сетевым транспортом (в случае необходимости работы по сети).

## 5.7. Демонстрационная макетная конфигурация

Поставляемая демонстрационная макетная конфигурация, ориентирована на использование управляемых форм. Для обеспечения взаимодействия с системой управления внешним оборудованием, в нее введены вспомогательные объекты. Они не являются неотъемлемой частью системы управления оборудованием, но позволяют оптимальным и показательным образом использовать все блоки стандарта в конфигурации 1С-Предприятие.

- Перечисление. Типы Подключаемого Оборудования\*. Состав элементов данного перечисления определяет состав доступных для клиентского приложения типов оборудования.
- Справочник. Подключаемое Оборудование\*. Необходим для возможности ссылки на конкретный экземпляр оборудования (см. требование 5.3)
- Менеджер управления оборудованием. Состоящий из:
  - Нескольких переменных модуля управляемого приложения, где сохраняются ссылки на созданные объекты драйверов оборудования, объект сетевого транспорта, и пути к хранилищу настроек.
  - Нескольких общих модулей, в которые вынесены служебные функции, реализующие логику работы менеджера управления оборудованием.

\* Данные объекты метаданных реализованы как пример возможной организации блока управления оборудованием клиентского приложения. Их наличие или отсутствие в клиентском приложении никак не влияет на работу системы управления оборудованием, поэтому разработчик клиентского приложения может реализовать любой собственный вариант хранения идентификаторов и определения соответствия оборудования и рабочего места, пользователя и т.п.

## 5.8. Построение менеджера управления оборудованием

Менеджер – это часть клиентского приложения, в которой сосредоточены сервисные процедуры и регламентированные данным стандартом интерфейсы взаимодействия с подключаемым оборудованием. Данным стандартом регламентируется определенный порядок взаимодействия с объектами драйверов оборудования (см. раздел [основные алгоритмы](#)) и сетевым транспортом. Все обращения к оборудованию из других частей клиентского приложения происходят только посредством менеджера оборудования. Для остальных частей клиентского приложения, работа такого менеджера должна быть максимально прозрачной - после того, как экземпляр драйвера оборудования создан и инициализирован, менеджер осуществляет только передачу команд и возврат результата «как есть». Менеджер может передавать драйверу любые команды, массив входных и массив выходных параметров, значение таймаута. При этом команды менеджера транслируются в вызовы соответствующих методов драйверов. При вызове любого метода драйвера менеджером, рекомендуется использовать конструкцию «попытка...исключение» (try...except) При возникновении исключительной ситуации, менеджер должен возвращать соответствующую ошибку (56 – неизвестная ошибка драйвера) с корректным описанием. Алгоритмы взаимодействия менеджера, драйверов и клиентского приложения описаны в разделе «Основные алгоритмы».

Менеджер не должен верифицировать корректность команд приложения и их параметров. Несоблюдение этого принципа приведет к зависимости кода менеджера от версий стандарта.

## 5.9. Драйвер оборудования и вызов его методов

Команды оборудованию от клиентского приложения транслируются менеджером в вызовы публикуемых методов драйвера оборудования. Все публикуемые методы драйвера оборудования являются высокоуровневыми. Под «высокоуровневым методом» здесь подразумевается логически целостное и самодостаточное единичное задание для устройства. Такое задание либо выполняется целиком успешно, либо не выполняется вовсе (например, из-за ошибки или некорректного состояния устройства) - без возможности частичного исполнения метода, без возможности оставить устройство в некоем промежуточном состоянии.

Для обеспечения совместимости, разработчик обязан реализовать в драйвере модели все методы, задекларированные в описании типа оборудования, даже если устройство не поддерживает некоторые из необязательных методов. При вызове таких методов, драйвер

должен возвращать соответствующую ошибку (1003 - «команда не поддерживается устройством»).

Все публикуемые методы драйвера имеют три параметра:

- InputParameters (входные параметры)
- ResultData (параметры результата)
- TimeOut (таймаут ожидания результата).

Первые два параметра представляют собой одномерные массивы SafeArray со структурой, описанной в стандарте соответствующего типа оборудования, либо имеют значение Неопределено (Undefined), если метод не имеет входных или/и выходных параметров.

Последний параметр имеет тип [Number](#) и определяет максимальное время ожидания выполнения команды драйвером в секундах.

## 6. Используемые типы данных

Типы данных, используемые в стандарте, являются подмножеством вариантного типа данных. Это позволяет сделать систему максимально независимой от используемых при разработке драйверов и менеджера сред и языков программирования.

Во входных параметрах команд и в данных результата используются только простые типы: [String](#), [Number](#), [DateTime](#), [Boolean](#). При необходимости передачи большого количества параметров или сложных объемных структур данных, они могут помещаться в массивы SafeArray произвольной вложенности и размерности. Целесообразность такого подхода продиктована требованием совместимости и единообразия поведения объектов, реализованных при помощи разных программных технологий, необходимостью реализации общего менеджера управления, прозрачно передающего любые команды. Следует обратить внимание, что в функциях Windows API, работающих с SafeArray старшее измерение массива имеет больший номер. В методах для работы с SafeArray платформы 1С:Предприятие старшее измерение напротив - имеет меньший номер, как это принято во многих языках программирования.

Примечание: В описании данного стандарта принят порядок измерений SafeArray, соответствующий используемому в платформе 1С:Предприятие

Описание простых типов:

### String

Тип предназначен для передачи и хранения текстовой информации. Соответствует типу vt\_bstr. Допустимо использование многострочных строк. При этом данным стандартом регламентируется использование в качестве разделителя строк последовательности из двух символов с десятичными кодами 13 и 10 (CR/LF – стандартный разделитель строк для ОС Windows). При конвертации данных этого типа в формат XML, разделители строк заменяются на пару символов «\n». Если же в тексте встречается данная последовательность символов, то она преобразуется к строке «\n». При чтении XML-формата происходит обратное преобразование. Если максимальная длина строкового параметра или настройки явно не ограничена данным стандартом, драйвер должен корректно обрабатывать строку любой допустимой в операционной системе длины, не возвращая при этом ошибок. При выполнении методов, драйвер может обрезать или переносить слишком длинные строки – на усмотрение разработчика.

### Number

Тип предназначен для передачи числовой информации. Если значение параметра представляет собой целое число и попадает в интервал (-2 147 483 648..2 147 483 647), то при его передаче из среды 1С-Предприятие, драйвер получает вариантный тип vt\_i4, иначе - это вещественное число двойной точности соответствующее типу vt\_r8. При конвертации в XML, в качестве десятичного разделителя используется точка.

Существует некоторая проблема с точностью при работе с вещественными числами vt\_r8.

Связано это с особенностями представления вещественных типов в памяти. Подробности на <http://www.delphikingdom.com/asp/viewitem.asp?catalogid=374> Обход данной проблемы, при ее потенциальном возникновении, целиком ложиться на разработчика драйвера и может быть осуществлен при помощи конвертации значений в другие типы на стороне драйвера.

### Boolean

Логический тип. При конвертации в XML используются следующие текстовые представления значений этого типа: «False» - ложь, «True» - истина. При передаче из среды 1С-Предприятие значения данного типа, драйвер получает значение типа vt\_bool

## DateTime

Этот тип предназначен для хранения даты и времени. При передаче из среды 1С-Предприятие значения данного типа, драйвер получает значение типа vt\_date. При конвертации значений в файлы XML, используется следующий формат:

DD.MM.YYYY HH:NN:SS где:

DD – номер дня месяца

MM – номер месяца

YYYY – год

HH – час дня в формате 24h

NN – минуты

SS – секунды

## 7. Функции обратного вызова

Данные функции предназначены для вызова их драйверами устройств.

При инициализации любого драйвера оборудования, ему передается в качестве входного параметра метода Init ссылка на интерфейс IDispatch менеджера оборудования. Используя эту ссылку, драйвер может вызывать функции обратного вызова, реализованные в менеджере. Для удаленно используемого оборудования вызовы функций будут перенаправляться удаленным клиентским приложениям, через сетевой транспорт. Клиентское приложение исполняет все методы, связанные с обработкой событий всегда синхронно. В случае реализации клиентского приложения и менеджера на платформе 1С-Предприятие, это может приводить к блокировке работы всей системы управления оборудованием на время обработки события. Любое другое событие, от другого устройства или от удаленного клиентского приложения не может быть обработано до возврата управления из функции обрабатывающей предыдущее событие. Поэтому, при разработке клиентского приложения, рекомендуется возвращать управление из процедуры обработки обратного вызова максимально быстро, не используя операции с большим временем выполнения (поиск элемента в базе данных и пр.).

## 8. Основные алгоритмы

В данном документе алгоритмы взаимодействия приложения с драйверами устройств описаны со стороны клиентского приложения. Для лучшего понимания работы системы в целом, рекомендуется изучить также алгоритмы взаимодействия со стороны драйвера устройства (см. разделы «Служебные методы», «Общие методы» и «Настройки» в описаниях типов оборудования).

Примечание: в описаниях ниже предполагается, что драйвер устройства был предварительно установлен на компьютере рабочего места (см. документ «Установка и обновление системы управления оборудованием.doc»).

### 8.1. Создание нового экземпляра оборудования

- У пользователя запрашивается тип создаваемого оборудования
- Из хранилища настроек (см. описание файла Models.xml в документе «Установка и обновление системы управления оборудованием») извлекается список установленных в системе моделей оборудования выбранного типа. Пользователь выбирает модель оборудования из полученного списка
- Менеджер оборудования создает объект драйвера выбранной модели
- У созданного объекта вызывается метод Init, инициализирующий драйвер. После выполнения данного метода, объект драйвера считается [объектом-абстрактом](#)
- У объекта вызывается метод драйвера GetDeviceInfo для получения статической информации о драйвере.
- Для получения значений настроек по умолчанию, у драйвера вызывается метод GetSettings
- Если драйвер имеет собственную форму для настройки оборудования (элемент IsSettingsDlgExist из структуры DeviceInfo), у него вызывается метод ShowSettingsDlg для её отображения. Если форма настроек отсутствует, приложение отображает свою универсальную форму, заполнив её значениями настроек по умолчанию.
- Пользователь редактирует значения в форме настроек устройства и закрывает её.
- Если пользователь отказался от сохранения настроек, нажав на форме кнопку «Отмена», считается, что он отказался от создания устройства. У драйвера вызывается метод Close и объект драйвера уничтожается. Процедура создания устройства прерывается.

- Если использовалась форма настройки приложения, то набор настроек передается в драйвер через метод `SetSettings` для проверки корректности их заполнения (валидации)
- Менеджер генерирует уникальный идентификатор для нового экземпляра оборудования и создает соответствующий ему индивидуальный каталог в хранилище настроек
- У драйвера запрашивается набор его текущих настроек с помощью метода `GetSettings`. Набор настроек данного экземпляра оборудования сохраняется в индивидуальном каталоге хранилища настроек в файле `Settings.xml` (см. документ «Установка и обновление системы управления оборудованием»)
- У данного объекта драйвера вызывается метод `Close`, после чего он выгружается из памяти приложения
- Клиентское приложение сохраняет идентификатор созданного экземпляра оборудования для последующего его использования (см. [Использование экземпляра оборудования](#))

## 8.2. Использование экземпляра оборудования

В этом разделе описывается алгоритм работы с ранее созданным экземпляром оборудования.

- Перед началом работы с оборудованием клиентскому приложению необходимо выполнить следующую последовательность действий:
  - По известному ему идентификатору устройства и при помощи информации из хранилища настроек системы определить модель устройства
  - По описанию модели определить и создать объект драйвера устройства
  - Выполнить первичную инициализацию драйвера вызвав у него метод `Init`. При этом в параметрах метода драйверу передается ссылка на интерфейс `IDispatch` менеджера оборудования. Объект драйвера в этом состоянии считается [объектом-абстрактом](#) (т.е. не установлено соответствие с каким-либо экземпляром оборудования)
  - Считать из хранилища настроек набор настроек данного экземпляра оборудования и передать его в объект драйвера при помощи метода `SetSettings`
  - Запомнить значение настройки `DefaultTimeOut` для использования в качестве таймаута при последующих обращениях к устройству
  - Вызвать метод драйвера `Open`. В параметрах метода необходимо передать идентификатор устройства.
  - Начальным состоянием любого устройства является «Выключено». В этом состоянии допускается вызов следующих методов устройства: `GetDeviceInfo`, `GetSettings`, `SetSettings`, `ShowSettingsDlg`, `CheckHealth`, `Enable`, `Disable`, `Close`. Все остальные методы будут доступны после включения устройства.
- Вызвать метод драйвера `Enable`. При выполнении метода `Enable` драйвер должен выполнить все необходимые операции по подготовке и проверке готовности устройства к работе. Если метод `Enable` завершается успешно, то устройство переходит в состояние «Включено».
- Клиентское приложение взаимодействует с устройством, вызывая у драйвера специфические методы, определенные для данного типа оборудования.
- Драйвер устройства может порождать различные события посредством вызова у менеджера функций обратного вызова. Обработка событий приложением происходит в синхронном режиме
- По окончании работы с драйвером устройства, приложение должно отключить его, вызвав метод `Disable`. При исполнении данного метода драйвер должен привести физическое устройство в исходное состояние, высвободить все захваченные им в процессе работы ресурсы (закрывать открытые им файлы, коммуникационные порты), выгрузить использовавшиеся сторонние библиотеки и т.п.
- Перед уничтожением объекта драйвера устройства, приложение вызывает метод-деструктор `Close`. Драйвер очищает все свои служебные структуры данных, в том числе очищает ссылку на интерфейс `IDispatch` менеджера оборудования, переданную ему при вызове метода `Init`. Последующее использование объекта драйвера после вызова `Close` не предусматривается. Менеджер оборудования должен очистить все ссылки на объект драйвера для высвобождения его из памяти

## 8.3. Удаление экземпляра оборудования

Для удаления экземпляра оборудования из системы, достаточно удалить индивидуальный каталог экземпляра оборудования из хранилища настроек (см. описание в документе «Программа установки системы управления оборудованием.doc»). Далее следует очистить в настройках клиентских приложений, использовавших данный экземпляр, все ссылки на его идентификатор