56,21

Рейтинг

**1С**

Делаем средства разработки бизнес-приложений

PeterG 25 августа 2016 в 11:33

# Способы интеграции с 1С

Блог компании 1С, Анализ и проектирование систем, 1С-Битрикс

Какие важнейшие требования предъявляются к бизнес-приложениям? Одними из самых главных являются следующие задачи:

- Легкость изменения/адаптации логики работы приложения к меняющимся бизнес-задачам.
- Легкость интеграции с другими приложениями.

То, как решается в 1С первая задача, было кратко описано в разделе «Кастомизация и поддержка» этой статьи; мы еще вернемся к этой интересной теме в одной из будущих статей. Сегодня же речь пойдет о второй задаче, об интеграции.

## Интеграционные задачи

Интеграционные задачи могут быть разными. Для решения одних достаточно простого интерактивного обмена данными – например, для передачи в банк списка сотрудников для оформления зарплатных пластиковых карт. Для более сложных задач может быть необходим полностью автоматизированный обмен данными, возможно, с обращением к бизнес-логике внешней системы. Есть задачи, носящие специализированный характер, вроде интеграции с внешним оборудованием (например, торговым оборудованием, мобильными сканерами и т.д.) или с унаследованными или узкоспециализированными системами (например, с системами распознавания RFID-меток). Крайне важно для каждой задачи выбрать наиболее подходящий механизм интеграции.

## Возможности интеграции с 1С

Существуют различные подходы к реализации интеграции с приложениями 1С, какой из них выбрать – зависит от требований задачи.

1. Реализация на основе механизмов интеграции, предоставляемых платформой, своего собственного специализированного API на стороне приложения 1С (например, набора Web- или HTTP-сервисов, которые будут вызывать сторонние приложения для обмена данными с приложением 1С). Плюс этого подхода – устойчивость API к изменению реализации на стороне приложения 1С. Особенность подхода — требуется менять исходный код типового решения 1С, что может потенциально потребовать усилий при слиянии исходных кодов при переходе на новую версию конфигурации. В этом случае может прийти на помощь новая прогрессивная функциональность — расширения конфигурации. Расширения – это, по сути, механизм плагинов, который позволяет создавать дополнения прикладных решений, не меняя самих прикладных решений. Вынос интеграционного API в расширение конфигурации позволит избежать сложностей при слиянии конфигураций при переходе на новую версию типового решения.

2. Использование механизмов интеграции платформы, которые предоставляют доступ снаружи к объектной модели приложения и не требуют доработки приложения или создания расширения. Плюс такого подхода – не нужно менять приложение 1С. Минус – если приложение 1С было доработано, то могут потребоваться доработки в интегрируемом приложении. Пример такого подхода – использование для интеграции протокола OData, реализованного на стороне платформы 1С:Предприятие (подробнее о нем ниже).

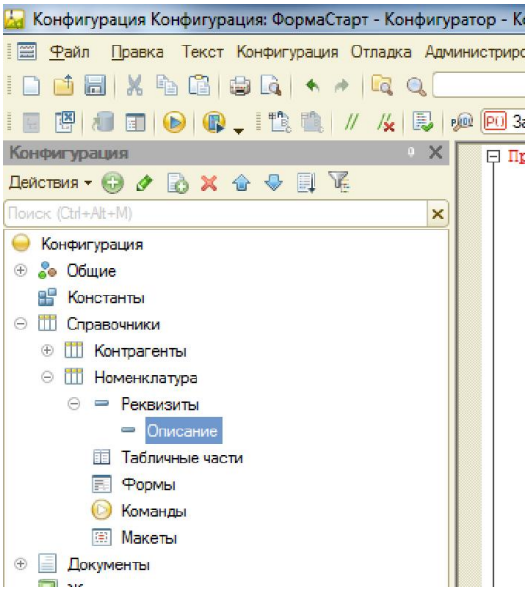
3. Использование готовых прикладных протоколов, реализованных в типовых решениях 1С. Многие типовые решения от 1С и партнеров реализуют на основе механизмов интеграции, предоставляемых платформой, свои собственные прикладные протоколы, ориентированные на конкретные задачи. При использовании этих механизмов не требуется написание кода на

стороне приложения 1С, т.к. мы пользуемся штатными возможностями прикладного решения. На стороне приложения 1С нам нужно лишь выполнить определенные настройки.

## Механизмы интеграции в платформе 1С:Предприятие

### Импорт/экспорт файлов

Предположим, перед нами стоит задача двунаправленного обмена данными между приложением 1С и произвольным приложением. Например, нам нужно синхронизировать список товаров (справочник Номенклатура) между приложением 1С и произвольным приложением.




Для решения такой задачи можно написать расширение, которое выгружает справочник Номенклатура в файл определенного формата (текстовый, XML, JSON, …) и умеет считывать этот формат.

В платформе реализован механизм сериализации прикладных объектов в XML как напрямую, через методы глобального контекста ЗаписатьXML/ЧтениеXML, так и с помощью вспомогательного объекта XDTO (XML Data Transfer Objects).

Любой объект в системе 1С:Предприятие может быть сериализован в XML представление и наоборот.

Эта функция вернет представление объекта в виде XML:

```
Функция Объект_В_XML(Объект)
    ЗаписьXML = Новый ЗаписьXML();
    ЗаписьXML.УстановитьСтроку();
    ЗаписатьXML(ЗаписьXML, Объект);
    Возврат ЗаписьXML.Закрыть();
КонецФункции
```

так будет выглядеть экспорт справочника Номенклатура в XML при помощи XDTO:

```
&НаСервере
Процедура ЭкспортXMLНаСервере()
        НовыйСериализаторXDTO  = СериализаторXDTO;
        НоваяЗаписьXML = Новый ЗаписьXML();
        НоваяЗаписьXML.ОткрытьФайл("C:\Data\Номенклатура.xml", "UTF-8");

        НоваяЗаписьXML.ЗаписатьОбъявлениеXML();
        НоваяЗаписьXML.ЗаписатьНачалоЭлемента("СправочникНоменклатура");

        Выборка = Справочники.Номенклатура.Выбрать();

        Пока Выборка.Следующий() Цикл
                ОбъектНоменклатура = Выборка.ПолучитьОбъект();
```

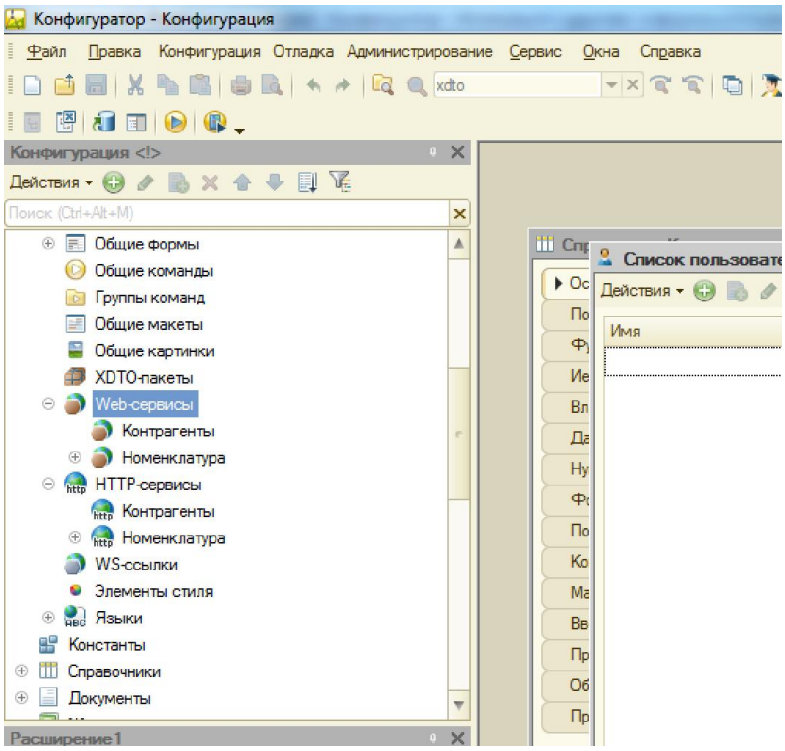
```
                НовыйСериализаторXDTO.ЗаписатьXML(НоваяЗаписьXML, ОбъектНоменклатура, НазначениеТипаXML.Явно
e);
        КонецЦикла;

        НоваяЗаписьXML.ЗаписатьКонецЭлемента();
        НоваяЗаписьXML.Закрыть();
КонецПроцедуры
```

Путем несложной переделки кода экспортируем справочник в JSON. Товары будут записаны в массив; для разнообразия приведем англоязычный вариант синтаксиса:

```
&AtServer
Procedure ExportJSONOnServer()
        NewXDTOSerializer  = XDTOSerializer;
        NewJSONWriter = New JSONWriter();
        NewJSONWriter.OpenFile("C:\Data\Номенклатура.json", "UTF-8");

        NewJSONWriter.WriteStartObject();
        NewJSONWriter.WritePropertyName("СправочникНоменклатура");
        NewJSONWriter.WriteStartArray();

        Selection = Catalogs.Номенклатура.Select();

        While Selection.Next() Do
                NomenclatureObject = Selection.GetObject();

                NewJSONWriter.WriteStartObject();

                NewJSONWriter.WritePropertyName("Номенклатура");
                NewXDTOSerializer.WriteJSON(NewJSONWriter, NomenclatureObject, XMLTypeAssignment.Implicit);

                NewJSONWriter.WriteEndObject();
        EndDo;

        NewJSONWriter.WriteEndArray();
        NewJSONWriter.WriteEndObject();
        NewJSONWriter.Close();
EndProcedure
```

Далее останется только передать данные конечному потребителю. Платформа 1С:Предприятие поддерживает основные интернет-протоколы HTTP, FTP, POP3, SMTP, IMAP, включая их безопасные версии. Также для передачи данных можно использовать HTTP и/или Web-сервисы.

## HTTP- и веб-сервисы




Приложения 1С могут реализовывать свои HTTP- и веб-сервисы, а также вызывать HTTP- и веб-сервисы, реализованные сторонними приложениями.

## REST интерфейс и протокол OData

Начиная с версии 8.3.5 платформа 1С:Предприятие может автоматически формировать REST интерфейс для всего прикладного решения. Любой объект конфигурации (справочник, документ, регистр сведений и т.д.) можно сделать доступным для получения и модификации данных через REST интерфейс. В качестве протокола доступа платформа использует протокол OData версии 3.0. Публикация сервисов OData выполняется из меню Конфигуратора «Администрирование -> Публикация на веб-сервере», флажок «Публиковать стандартный интерфейс OData» должен быть помечен. Поддерживаются форматы atom/XML и JSON. После того, как прикладное решение опубликовано на веб-сервере, сторонние системы могут обращаться к нему через REST интерфейс с помощью HTTP запросов. Для работы с приложением 1С через протокол OData программирование на стороне 1С не требуется.

Так, URL вида http://<сервер>/<конфигурация>/odata/standard.odata/Catalog_Номенклатура вернет нам содержимое каталога Номенклатура в формате XML — коллекцию элементов entry (заголовок сообщения пропущен для краткости):

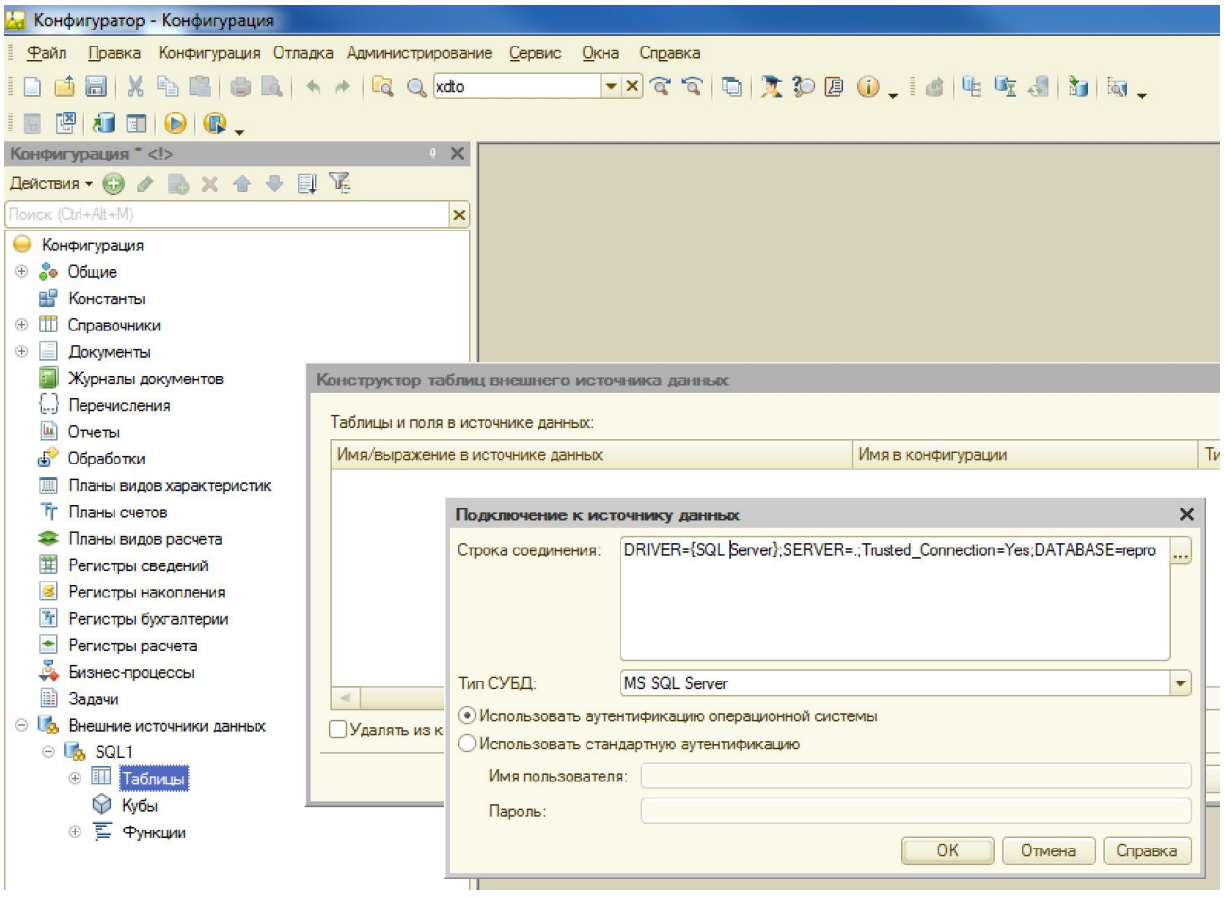
```
<entry>
        <id>http://server/Config/odata/standard.odata/Catalog_Номенклатура(guid'35d1f6e4-289b-11e6-8ba4-e03f4
9b16074')</id>
        <category term="StandardODATA.Catalog_Номенклатура" scheme="http://schemas.microsoft.com/ado/2007/08/
dataservices/scheme"/>
        <title type="text"/>
        <updated>2016-06-06T16:42:17</updated>
        <author/>
        <summary/>
        <link rel="edit" href="Catalog_Номенклатура(guid'35d1f6e4-289b-11e6-8ba4-e03f49b16074')" title="edit-
link"/>
        <content type="application/xml">
                <m:properties xmlns:d="http://schemas.microsoft.com/ado/2007/08/dataservices" xmlns:m="htt
p://schemas.microsoft.com/ado/2007/08/dataservices/metadata">
                        <d:Ref_Key>35d1f6e4-289b-11e6-8ba4-e03f49b16074</d:Ref_Key>
                        <d:DataVersion>AAAAAgAAAAA=</d:DataVersion>
                        <d:DeletionMark>false</d:DeletionMark>
                        <d:Code>000000001</d:Code>
                        <d:Description>Кондиционер Mitsubishi</d:Description>
                        <d:Описание>Мощность 2,5 кВт, режимы работы: тепло/холод</d:Описание>
                </m:properties>
        </content>
</entry>
```

```
<entry>
        <id>http://server/Config/odata/standard.odata/Catalog_Номенклатура(guid'35d1f6e5-289b-11e6-8ba4-e03f4
9b16074')</id>
        <category term="StandardODATA.Catalog_Номенклатура" scheme="http://schemas.microsoft.com/ado/2007/08/
dataservices/scheme"/>
...
```

Прибавляя к URL-у строку «?$format=application/json», получим содержимое каталога Номенклатура в формате JSON (URL вида http://<сервер>/<конфигурация>/odata/standard.odata/Catalog_Номенклатура?$format=application/json ):

```
{
"odata.metadata": "http://server/Config/odata/standard.odata/$metadata#Catalog_Номенклатура",
"value": [{
"Ref_Key": "35d1f6e4-289b-11e6-8ba4-e03f49b16074",
"DataVersion": "AAAAAgAAAAA=",
"DeletionMark": false,
"Code": "000000001",
"Description": "Кондиционер Mitsubishi",
"Описание": "Мощность 2,5 кВт, режимы работы: тепло/холод"
},{
"Ref_Key": "35d1f6e5-289b-11e6-8ba4-e03f49b16074",
"DataVersion": "AAAAAwAAAAA=",
"DeletionMark": false,
"Code": "000000002",
"Description": "Кондиционер Daikin",
"Описание": "Мощность 3 кВт, режимы работы: тепло/холод"
}, …
```

## Внешние источники данных




В некоторых случаях обмен данными через внешние источники данных может оказаться оптимальным решением. Внешние источники данных – это прикладной объект конфигурации 1С, позволяющий взаимодействовать с любой ODBC-совместимой базой данных как на чтение, так и на запись. Внешние источники данных доступны как в Windows, так и на Linux.

## Механизм обмена данными

Механизм обмена данными предназначен как для создания территориально распределенных систем на основе 1С:Предприятия, так и для организации обмена данными с другими информационными системами, не основанными на 1С:Предприятии.

Этот механизм активно используется во внедрениях 1С, и спектр задач, решаемых с его помощью, весьма широк. Это и обмен данными между приложениями 1С, установленными в филиалах организации, и обмен между приложением 1С и сайтом интернет-магазина, и обмен данными между серверным приложением 1С и мобильным клиентом (созданным с помощью мобильной платформы 1С:Предприятия), и многое другое.

Одно из ключевых понятий в механизме обмена данными – это план обмена. План обмена – это особый тип объекта прикладного платформы 1С, определяющий, в частности, состав данных, которые будут участвовать в обмене (какие именно справочники, документы, регистры и т.п.). План обмена содержит также информацию об участниках обмена (так называемых узлах обмена).
Вторая составляющая механизма обмена данными – механизм регистрации изменений. Данный механизм автоматически отслеживает в системе изменения данных, которые должны быть переданы конечным потребителям в рамках плана обмена. С помощью этого механизма платформа отслеживает изменения, произошедшие со времени последней синхронизации, и позволяет минимизировать объем данных, передаваемый в рамках очередного сеанса синхронизации.

Обмен данными происходит при помощи XML сообщений определенной структуры. Сообщение содержит данные, изменившиеся со времени последней синхронизации с узлом, и некоторую служебную информацию. Структура сообщений поддерживает нумерацию сообщений и позволяет получать подтверждения от узла-получателя о приеме сообщений. Такое подтверждение содержится в каждом сообщении, приходящем от узла-получателя, в виде номера последнего принятого сообщения. Нумерация сообщений позволяет понять платформе, какие данные уже были успешно переданы на узел-получатель, и избежать повторной передачи, передавая лишь данные, измененные со времени приема узлом-отправителем последнего сообщения с квитанцией о полученных узлом-получателем данных. При такой схеме работы обеспечивается гарантированная доставка даже при ненадежных каналах передачи и потере сообщений.

## Внешние компоненты

В ряде случаев, при решении задач интеграции приходится сталкиваться со специфическими требованиями, например, протоколами взаимодействия, форматами данных, работа с которыми не предусмотрена в платформе 1С:Предприятие. Для такого круга задач в платформе предусмотрена технология внешних компонент, которая позволяет создавать динамически подключаемые модули, расширяющие функционал 1С:Предприятия.

Типичным примером задачи с подобными требованиями, может служить интеграция прикладного решения 1С с торговым оборудованием, начиная от весов и заканчивая кассовыми аппаратами и сканерами штрих-кодов. Внешние компоненты могут быть подключены как на стороне сервера 1С:Предприятия, так и на клиентской части (включая, в том числе, и веб-клиент, а также следующую версию мобильной платформы 1С:Предприятия). Технология внешних компонент предусматривает достаточно простой и понятный программный (C++) интерфейс взаимодействия компоненты с платформой 1С:Предприятие, который должен реализовать разработчик.

Возможности, открывающиеся при использовании внешних компонент, весьма широки. Можно реализовать взаимодействие по специфическому протоколу обмена данными с внешними устройствами и системами, встроить специфические алгоритмы обработки данных и форматов данных и т.п.

## Устаревшие механизмы интеграции

В платформе доступны механизмы интеграции, которые не рекомендуется использовать в новых решениях; они оставлены из соображений обратной совместимости, а также на случай, если другая сторона не может работать с более современными протоколами. Один из них – работа с файлами формата DBF (поддерживается во встроенном языке с помощью объекта XBase).

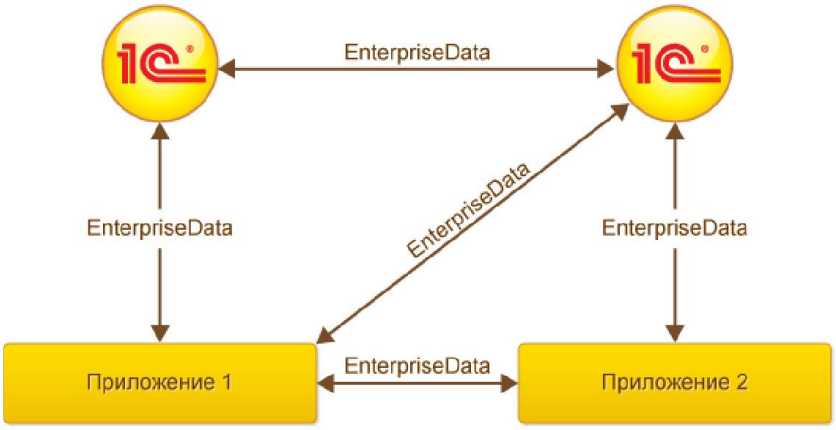
Другой устаревший механизм интеграции – использование технологии COM (доступно только на платформе Windows). Платформа 1С:Предприятие предоставляет два способа интеграции для Windows, использующие технологию COM: Automation-сервер и Внешнее соединение. Они очень похожи, но одним из принципиальных отличий является то, что в случае Automation-сервера запускается полноценное клиентское приложение 1С:Предприятие 8, а в случае внешнего соединения запускается относительно небольшой внутрипроцессный COM-сервер. То есть в случае работы через Automation сервер можно задействовать функционал клиентского приложения, выполнять действия, аналогичные интерактивным действиям пользователя. При использовании внешнего соединения можно использовать только функции бизнес-логики, причем их можно выполнять как на клиентской стороне соединения, где создается внуприпроцессный COM-сервер, так и осуществлять вызов

бизнес-логики на стороне сервера 1С:Предприятия.

Также технологию COM можно использовать для обращения к внешним системам из кода приложения на платформе 1С:Предприятие. В данном случае приложение 1С выступает в качестве COM-клиента. Но следует напомнить, что данные механизмы будут работать только в том случае, если сервер 1С функционирует в среде Windows.

## Механизмы интеграции, реализованные в типовых конфигурациях

### Формат EnterpriseData




В ряде конфигураций 1С (список ниже) на основе описанного выше платформенного механизма обмена данными реализован готовый механизм обмена данными с внешними приложениями, не требующий изменения исходного кода конфигураций (подготовка к обмену данными делается в настройках прикладных решений):

- «1С:ERP Управление предприятием 2.0»
- «Комплексная автоматизация 2»
- «Бухгалтерия предприятия», редакция 3.0
- «Бухгалтерия предприятия КОРП», редакция 3.0
- «Розница», редакция 2.0
- «Управление торговлей базовая», редакция 11
- «Управление торговлей», редакция 11
- «Зарплата и управление персоналом КОРП», редакция 3

Для обмена данными используется формат EnterpriseData, основанный на XML. Формат является бизнес-ориентированным – описанные в нем структуры данных соответствуют бизнес-сущностям (документам и элементам справочников), представленным в программах 1С, например: акт выполненных работ, приходный кассовый ордер, контрагент, номенклатура и т. п.

Обмен данными между приложением 1С и сторонним приложением может происходить:

- через выделенный файловый каталог
- через каталог FTP
- через веб-сервис, развернутый на стороне приложения 1С. Файл с данными передается как параметр веб-методов
- через электронную почту

В случае обмена через веб-сервис стороннее приложение будет инициировать сеанс обмена данными путем вызова соответствующих веб-методов приложения 1С. В остальных случаях инициатором сеанса обмена будет приложение 1С (путем помещения файла с данными в соответствующий каталог или отправкой файла с данными на настроенный почтовый адрес). Также на стороне 1С настраивается, с какой периодичностью будет происходить синхронизация (для вариантов с файловым

обменом через каталог и электронную почту):

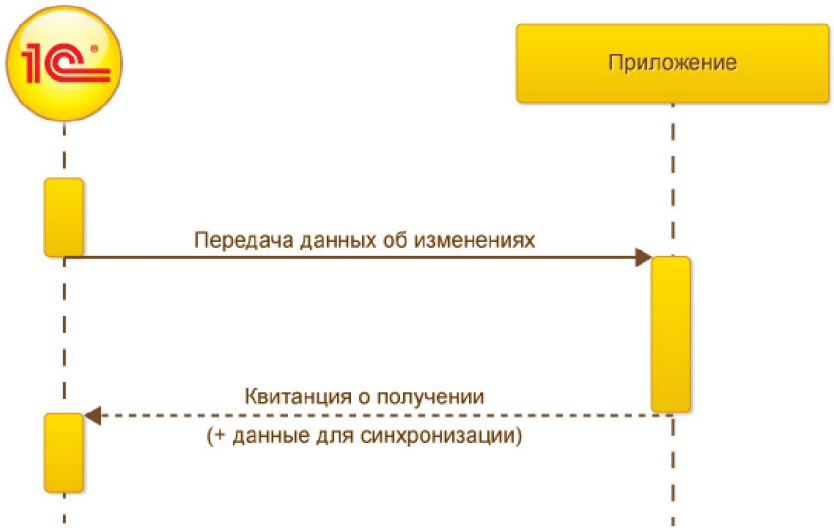
- по расписанию (с заданной периодичностью)
- вручную; пользователь должен будет вручную запускать синхронизацию каждый раз, когда это ему потребуется

### Квитирование сообщений

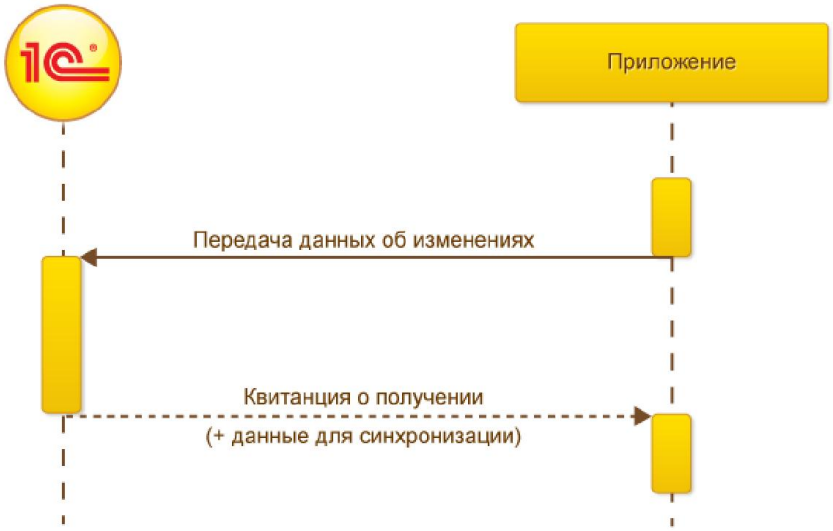
Приложения 1С ведут учет отправленных и полученных сообщений синхронизации и ожидают того же от сторонних приложений. Это позволяет задействовать механизм нумерации сообщений, описанный выше в разделе «Механизм обмена данными».

Приложения 1С в ходе синхронизации передают только информацию об изменениях, произошедших с бизнес-сущностями со времени последней синхронизации (чтобы минимизировать объем передаваемой информации). При первой синхронизации приложение 1С выгрузит все бизнес-сущности (например, элементы справочника номенклатуры) в формате EnterpriseData в XML-файл (поскольку все они являются «новыми» для внешнего приложения). Стороннее приложение должно обработать информацию из пришедшего от 1С XML-файла и при следующем сеансе синхронизации поместить в файл, отправляемый в 1С, в специальную секцию XML, информацию, что сообщение от 1С за определенным номером успешно принято. Сообщение-квитанция является для приложения 1С сигналом, что все бизнес-сущности успешно обработаны внешним приложением и информацию о них передавать больше не нужно. Помимо квитанции XML-файл от стороннего приложения также может содержать данные для синхронизации со стороны приложения (например, документы реализации товаров и услуг).

После получения сообщения-квитанции приложение 1С помечает все изменения, переданные в предыдущем сообщении, как успешно синхронизированные. Лишь несинхронизированные изменения в бизнес-сущностях (создание новых сущностей, изменение и удаление существующих) будут отправлены во внешнее приложение при следующем сеансе синхронизации.




При передаче данных от внешнего приложения в приложение 1С картина меняется на обратную. Внешнее приложение должно заполнить секцию-квитанцию в XML файле соответствующим образом и поместить бизнес-данные для синхронизации со своей стороны в формате EnterpriseData.




### Упрощенный обмен данными без квитирования

Для случаев простой интеграции, когда достаточно только передавать информацию от стороннего приложения в приложение 1С и обратной передачи данных из приложения 1С в стороннее приложение не требуется (например, интеграция онлайн-магазина, передающего информацию о продажах в «1С:Бухгалтерию»), есть упрощенный вариант работы через веб-сервис (без квитирования), не требующий настроек на стороне приложения 1С.

### Специализированные интеграционные решения

Существует типовое решение «1С:Конвертация данных», которое использует механизмы платформы для конвертации и обмена данными между типовыми конфигурациями 1С, но может быть также использовано для интеграции со сторонними приложениями.

### Интеграция с банковскими решениями

Стандарт «Клиент банк», разработанный специалистами 1С более 10 лет назад, фактически стал стандартом индустрии в России. Следующий шаг в этом направлении – технология DirectBank, позволяющая отправлять платежные документы в банк и получать выписки из банка непосредственно из программ системы «1С:Предприятия» нажатием одной кнопки в программе «1С»; при этом не требуется установка и запуск дополнительных программ на клиентский компьютер.

Есть также стандарт по обмену данными в зарплатных проектах.

### Прочее

Заслуживают упоминания протокол обмена между системой 1С:Предприятие и сайтом, стандарт обмена коммерческой информацией CommerceML (разработанный совместно с Microsoft, Intel, Price.ru и другими компаниями), стандарт обмена данными по операциям эквайринга.

**Теги:** 1С, erp системы, разработка приложений, интеграция систем

+17 161 109k 92

1С 56,21
Делаем средства разработки бизнес-приложений

27,0 Карма –0,8 Рейтинг 111 Подписчики 0 Подписки

@PeterG
Пользователь

Поделиться публикацией

ПОХОЖИЕ ПУБЛИКАЦИИ

13 октября 2016 в 09:55

**1С в облаках**

+14 32,7k 47 57

27 октября 2015 в 12:20

**Платформа «1С: Предприятие» — что под капотом?**

+17 60k 180 160

21 сентября 2015 в 13:43

**И снова про 1С**

+8 48k 123 160

## Комментарии 92

**Serginio1** 25 августа 2016 в 12:53 0

Добрый день. В технологии Внешних компонент заложен возврат объектов и передача объектов в параметрах ВК. Но пока этого сделать нельзя. Приходится возвращать строку и из неё создавать ВК

```
//1С при передаче по ссылке свойства ВК Список.Current

// при выходе из метода присваивает  Список.Current значение переданное изначально
// Поэтому помечаем входной параметр как Знач

//Или же делать так, если методы изменить нельзя
// То нужно присвоить значение переменной и вызвать метод передав в параметрах эту переменную

//Стр=Список.Current;
//Зазача=ъ(Стр);

Функция Ъ(знач Ссылка)

    // Создаем объект по ссылке полученной из методов .Net классов

    //Физически это строка ёЁ<Ьъ>№_%)Э?&2 содержащее 12 символов для отделения их от других строк

    //и индекс в спике исполуемых объектов на стороне .Net


    рез = Новый("AddIn.NetObjectToNative.NetObjectToNative");
    // И установим ссылку

    рез.УстановитьСсылку(Ссылка);
    возврат  рез
КонецФункции// СоздатьОбъектПоСсылке()
```

Так при передаче объекта в параметрах приходится получать строковую ссылку

```
СборкаSqlClient=ъ(Врап.Сборка("System.Data.SqlClient.dll"));
    SqlConnection=ъ(СборкаSqlClient.GetType("System.Data.SqlClient.SqlConnection"));
    SqlCommand=ъ(СборкаSqlClient.GetType("System.Data.SqlClient.SqlCommand"));

    connection =ъ(Врап.Новый(SqlConnection.ПолучитьСсылку(),ConnectionString));
    connection.Open();
```

```
    ТекстЗапроса = "Select Номенклатура.DESCR Наименование  From sc84 Номенклатура where DESCR Like '%'+@Стр
ока+'%'
    |order by Номенклатура.DESCR";


    command = ъ(Врап.Новый(SqlCommand.ПолучитьСсылку(),ТекстЗапроса,connection.ПолучитьСсылку()));

    Parameters=ъ(command.Parameters);
    Parameters.AddWithValue("@Строка", "ДСП");
    dr = ъ(command.ExecuteReader());

    Пока dr.Read() Цикл
        Сообщить(dr.get_Item("Наименование"));
    КонецЦикла;
```

**alexey-lustin** 25 августа 2016 в 17:19 0

Я так понимаю вы имеете ввиду вашу реализацию работы с внешними компонентами, причем если мне не изменяет память на C#, что как бы нам напоминает что технология внешних компонент на NativeAPI ориентирована на C++, причем на кроссплатформенный C++.

```
int GetIntegerFromVariant(tVariant* arg)
{
    if (TV_VT(arg) == VTYPE_I2)
        return TV_I2(arg);
    if (TV_VT(arg) == VTYPE_I4)
        return TV_I4(arg);

    throw ArgumentTypeException(std::string("Invalid argument type"));
}
```

я бы ссылался на официальную документацию

- для COM компонент — у нас есть тип VTYPE_DISPATCH
- для кроссплаторфменной разработки мы несколько ограничены в типах tVariant

https://its.1c.ru/db/metodtorg#content:3221:hdoc:_toc235443219

Но я все также не пойму что вы хотите ?

Вот так например ?

```
ОбъектМойКомпоненты = ИнициализироватьОбъектКомпоненты();
_справочникСсылка = ПолучитьСсылкуНаСпровочникВВидеОбъекта();
ОбъектМойКомпоненты.ОбработатьСсылкуНаСправочник(_справочникСсылка);
```

P.S. sc84 — говорит мне об использовании интеграции с 1С 7.7 прямыми запросами ;-). Всем известно что такой способ интеграции должен писаться с помощью других средств. ;-) 1С++ называется.

**Serginio1** 25 августа 2016 в 18:09 +1

Из 8 ки и из под линукса? 1С, Linux, Excel, Word, OpenXML и Net Core

Да это не суть. В Native API есть структура

```
struct _tVariant
        {
            _ANONYMOUS_UNION union
            {
               .......
            IID IDVal;
            struct _tVariant *pvarVal;
            struct tm      tmVal;
            _ANONYMOUS_STRUCT struct
```

```
            {
                void* pInterfaceVal;
               IID InterfaceID;
        }
        __VARIANT_NAME_2/*iface*/;
            _ANONYMOUS_STRUCT struct
            {
                char* pstrVal;
        uint32_t strLen; //count of bytes
    }
    __VARIANT_NAME_3/*str*/;
            _ANONYMOUS_STRUCT struct
            {
                WCHAR_T* pwstrVal;
    uint32_t wstrLen; //count of symbol
    /         } __VARIANT_NAME_4/*wstr*/;
        } __VARIANT_NAME_1;
        uint32_t cbElements;    //Dimension for an one-dimensional array in pvarVal
    TYPEVAR vt;
    };
```

В которой можно использовать
void* pInterfaceVal;
IID InterfaceID;

А в vt; указать, что это ВК.

**Serginio1** 26 августа 2016 в 11:19 0

Кстати в версии 8.3.10 планируется передача в параметрах двоичных данных

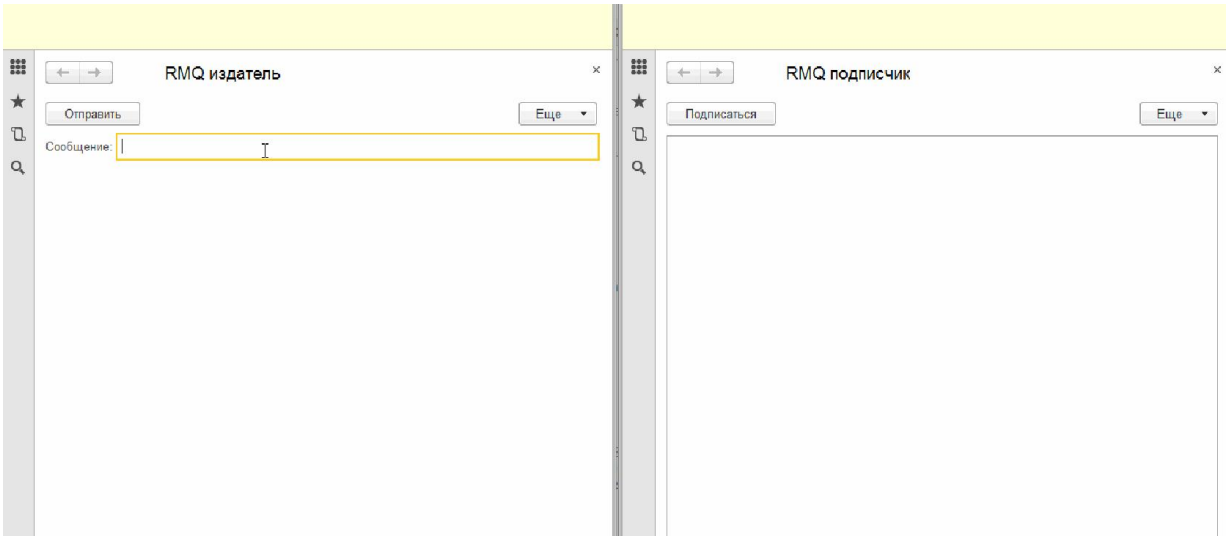
> Раньше при передаче двоичных данных между 1С: Предприятием и внешней компонентой существовал ряд ограничений. Например, передать двоичные данные во внешнюю компоненту было нельзя, а при работе в веб-клиенте обмен двоичными данными был вообще невозможен.
>
> Теперь мы убираем все эти ограничения. Обмен двоичными данными вы можете осуществлять в обе стороны и даже в веб-клиенте.
>
> На работе существующих внешних компонент это никак не отразится. Они будут работать, как и раньше. А вот во вновь создаваемых компонентах вы сможете теперь в качестве параметров передавать объекты ДвоичныеДанные

https://wonderland.v8.1c.ru/blog/rasshirenie-funktsionalnosti-raboty-s-dvoichnymi-dannymi/

**alexey-lustin** 31 августа 2016 в 19:11 0

Я подписан — видел.

Мы сейчас сделали (со всякими BDD и CI-CD) вот такое.



И вроде как сложные типы коллеги победили. Но там CMake и C(C++). Естественно кросс-платформенное — это кстати к вопросу о интеграции.через NativeAPI. На GIF'ке — автоматический сценарий по Gherkin с кнопконажималкой.

Теперь уже и мне стало интересно — как они это сделали, уточню напишу.

**Serginio1** 1 сентября 2016 в 07:41 0

Могу посмотреть какие есть в .Net Core. В большом есть. У меня к сожалению сейчас нет Linux но должно компилироваться. Но например взял 8.3.9 и скомпилировал по 64. Там вылетает ошибка, хотя если сделать свое 64 приложение то все нормально работает. Попробую откомпилировать по VS 2013
http://rsdn.org/forum/dotnet/6538046.1

**Serginio1** 1 сентября 2016 в 09:54 0

Что касается WebSocket то в .Net Core они появятся осенью в 1.1
.NET Core Roadmap
Там хорошая технология SignalR

1C Messenger для отправки сообщений, файлов и обмена данными между пользователями 1С, вэб страницы, мобильными приложениями а ля Skype, WhatsApp

.NET Core Roadmap

Уже сейчас много кроссплатформенного под .Net Core
Например RSS Каналы

**Serginio1** 1 сентября 2016 в 14:19 0

Вот кстати на взрослом .Net 7 причин, почему интеграцию необходимо строить на очередях. Практика RabbitMQ. Отказ от Zato ESB и OData в 1С

А вот клиент для .Net Core RabbitMQ .NET client — CoreCLR VERSION https://www.rabbitmq.com/

**alexey-lustin** 1 сентября 2016 в 15:28 0

Что-то аж прям обидно стало… Оказывается Петро это взрослому, а мы ту погулять вышли. Я в курсе этой темы — причем Петро там многое закостылил, но в целях бизнес-эффекта для пользователей.

Посмотрите комментарий мой что-ли в указанной статье — мы начинали с C# и в продуктиве отказались.

Начиналось все с вот таких экспериментов https://github.com/allustin/one-c-connectors

Затем были статьи mini_root

http://infostart.ru/public/21077/
http://infostart.ru/public/21506/
http://infostart.ru/public/22184/
http://infostart.ru/public/59961/

Потом была история с курсом на 8 часов для 1С-ников https://silverbulleters.timepad.ru/event/131122/

и игра в кролика между любыми приложениями https://bitbucket.org/alexey_yakimovich/

и доклад для архитекторов https://www.youtube.com/watch?v=R1HQgOHEiME
чтобы не продолбать Топологию роутинга.

И после этого мы отказались от C# перейдя на C++ в продуктиве — это эффективней. Естественно с плагинами Shovel и Federate на самом RabbitMQ.

Бизнес-эффект если кто не в курсе, это когда любое событие в федеральное сети доставляется в центральные BI и остальные системы в течении 15 секунд и не более.

**Serginio1** 1 сентября 2016 в 18:06 0

А продуктивней в чем? Там по сути то задержки это 1С да посылка сообщений на сам сервер. При этом до сих пор нет асинхронных методов.

**alexey-lustin** 1 сентября 2016 в 15:30 0

Что касается возможности NET в части кроссплатформенности — меня если честно подзадолбала его поддержка в oscript.io, поэтому C++

**Serginio1** 1 сентября 2016 в 18:10 0

Ну он вышел то совсем недавно. Но C++ нужно писать ВК. Здесь можно использовать код. Мало того скоро напишу для Net Core аналог .NET(C#) для 1С. Динамическая компиляция класса обертки для использования .Net событий в 1С через ДобавитьОбработчик или ОбработкаВнешнегоСобытия

Сейчас проверил такой код

```
var compilation = Microsoft.CodeAnalysis.CSharp.CSharpCompilation.Create("a")
    .WithOptions(new Microsoft.CodeAnalysis.CSharp.CSharpCompilationOptions(Microsoft.CodeA
nalysis.OutputKind.DynamicallyLinkedLibrary))
    .AddReferences(
        Microsoft.CodeAnalysis.MetadataReference.CreateFromFile(typeof(object).GetTypeInfo
().Assembly.Location))
        .AddSyntaxTrees(Microsoft.CodeAnalysis.CSharp.CSharpSyntaxTree.ParseText(
        @"
using System;

public  class C
{

     public  C(){}
    public string  M()
    {
        return ""Hello Roslyn."";
    }
}"));

            var fileName = @"d:\NetStandart\TestCoreNetApp\src\TestCoreNetApp\bin\Debug\net
coreapp1.0\a.dll";

            compilation.Emit(fileName);

            var a = System.Runtime.Loader.AssemblyLoadContext.Default.LoadFromAssemblyPath
(fileName);


            Type тип = a.GetType("C");
            var obj = Activator.CreateInstance(тип);

            var res = тип.GetMethod("M").Invoke(obj, null);
            Console.WriteLine(res.ToString());
```

**AndreyKu** 25 августа 2016 в 12:59 +1

Очень хочу когда-нибудь увидеть встроенный HTTP сервер как класс в технологической платформе.
Текущая необходимость использовать сторонние HTTP сервере (Apache, IIS) для публикации HTTP сервисов мне не нравится. Есть очень простые задачи по интеграции при реализации которых уходит на порядок больше времени для настройки окружения, чем на саму реализацию. Да и с точки зрения секьюрости открывать дефолтно настроенный ( а большинстве случаев так и есть, настраивается по мануалам без особого понимания) HTTP сервер — опасно.

**arabo_xv** 25 августа 2016 в 13:20 +1

Ваша мечта сбылась http://v8.1c.ru/overview/release_8_4_1/

**AndreyKu** 25 августа 2016 в 14:24 0

Как я понял из описания встроенный веб сервер это новый компонент «кластера». Кластер серверов же используется только для клиент-серверных ИБ, и на его лицензирование нужен дорогостоящий ключ. Если это так, то мне эта новая фича совсем не нравится. Я подразумевал именно как класс технологической платформы. Или я что-то путаю?

**mihmig** 25 августа 2016 в 14:50 0

А как Вы хотели? Главбух МарьВанна запускает на своём рабочем месте 1с-ку (желательно базовую, за 3 тыщи), а все остальные менеджеры забивают данные через браузер? Не покупая при этом лицензии даже для сетевой версии?

Нет 1С никогда на это не пойдёт…

**AndreyKu** 25 августа 2016 в 15:21 0

Для базовых версий вы не можете менять конфигурацию.
А если говорим о профессиональных версиях то сейчас же может, только нужно дополнительного окружение в виде стороннего веб сервера.

**marzhin** 25 августа 2016 в 13:20 0

Начиная с версии 8.4.1: https://its.1c.ru/db/v841doc#bookmark:v8:TI000000005

**Arch0n** 25 августа 2016 в 13:20 0

В 8.4 же обещали

**marzhin** 25 августа 2016 в 13:24 0

8.4.1.X — будет первым релизом версии 8.4, все честно! :)

**mihmig** 25 августа 2016 в 14:52 +1

Автор статьи лукавит — для получения данных ИЗ 1С требуется настройка дополнительного софта (web-сервера apache).

**Serginio1** 25 августа 2016 в 15:10 0

А у меня IIS/ Но тогда еще и MS SQL и прочие базы. Много чего нужно настраивать. Для этого существуют сис админы

**OLEG4120** 25 августа 2016 в 16:46 0

Админы… Вот в этом и есть ~~все~~ почти все программисты 1С. Мало кто из них может настроить что-нибудь за пределами платформы и конфигурации.

**Serginio1** 25 августа 2016 в 18:12 0

К сожалению приходится не только программировать (на разных языках), но еще и заниматься администрированием. Это не правильно.
Основной закон развития капитализма (эволюции), это увеличение специализации.

**Arch0n** 6 сентября 2016 в 12:39 0

Как раз наоборот. Они даже основы бухучета знают.
Ну и по моей практике как раз наоборот сисадмины почему-то (может в силу лени) часто не могут настроить IIS приходится самому делать. А мне за это не платят :(

**fishca** 7 сентября 2016 в 08:44 0

чаще не хотят настраивать, апач — это гуд :), иис — масдай :)

**denegny** 25 августа 2016 в 16:04 0

да-да, 1С это мало **OpenSource**, это скорее просто **OpenCode**

**PeterG** 25 августа 2016 в 16:16 +1

Да, вы правы — я забыл упомянуть, что при работе через OData (ведь вы о нем?) нужно настроить Apache или IIS (правда, настройки минимальны).

**mihmig** 26 августа 2016 в 11:26 0

Тем не менее, быть может опишете процесс настройки? Интересует связка Linux(Ubuntu/Centos) и apache?
Если конечно SMM-бюджет для хабра уже не исчерпан…

**o4karek** 26 августа 2016 в 11:49 0

А штатные пути не работают?
http://its.1c.ru/db/v83doc#bookmark:adm:TI000000485

**fishca** 25 августа 2016 в 16:04 +1

Когда же появятся расширения(плагины) не для конфигураций, а для конфигуратора?
Сколько же можно уже игнорировать просьбы разработчиков?
Сделайте уже нормальный OpenConf/Снегопат.

**PeterG** 25 августа 2016 в 16:09 0

Мы двигаемся в этом направлении: Enterprise Development Tools

> В новую среду разработки заложена потенциальная возможность расширения. Используя прикладной интерфейс и технологию подключаемых модулей (плагинов) мы планируем предоставить вам возможность расширять имеющиеся инструменты разработки.

Это, правда, не конфигуратор, но его замена в будущем.

**fishca** 25 августа 2016 в 16:31 0

Прекрасное далёко, не будь ко мне жестоко,
Не будь ко мне жестоко, жестоко не будь.
От чистого истока в прекрасное далёко,
В прекрасное далёко я начинаю путь

**PQR** 25 августа 2016 в 16:59 0

Недавно узнал про строенные возможности доступа по REST — поначалу идея выглядела очень удобной!

Но на деле при выборке документов (например счёта на оплату) повсюду вижу ссылки на другие сущности типа: 35d1f6e5-289b-11e6-8ba4-e03f49b16074. Приходится делать много дополнительных REST запросов, чтобы собрать полную информацию о документе. Возможно, есть какая-то опция «загрузить все связанные объекты по ссылкам на глубину N»?

А вообще, REST уже не модно, даёшь в следующей версии 1С встроенный GraphQL!

**alexey-lustin** 25 августа 2016 в 17:29 0

А вот не надо это называть REST — это OData протокол, что как бы несколько шире чем REST.

А вы хотите сделать SELECT *FROM* TOJSON? Зачем ?

Если вам понадобились ссылки на связанные объекты — явно вы интегрируете данные в свою систему, где также хотите реализовать бизнес-логику. То есть "сделать странное"...

Что касается ODATA и автоматического REST интерфейса по метаданным — напомню, уже в публичном доступе новая функциональность http://v8.1c.ru/o7/201508odata/index.htm

**PQR** 25 августа 2016 в 19:12 0

> явно вы интегрируете данные в свою систему, где также хотите реализовать бизнес-логику. То есть «сделать странное»...

Так и есть! Вся бизнес логика и управленческий учёт на стороне «своей системы», а из 1С (Бухгалтерии) делаем печать документов и различную регламентированную отчётность, которая часто меняется и которую нет смысла программировать самостоятельно.

Странно это или нет? Да, можно было бы программировать бизнес логику непосредственно внутри 1С (платформа гибкая, мощная), но мы решили «снаружи». 1С у нас выступает как «микросервис» для бухгалтерской отчётности. Вполне рабочий вариант, я вам скажу! И по технологиям более привлекательный, т.к. мы не ограничены одним языком программирования или платформой.

**PeterG** 25 августа 2016 в 17:34 0

Запрашиваемая вами функциональность реализована в платформе 8.3.9.

**PeterG** 25 августа 2016 в 17:42 +2

Из документации по 8.3.9 (8.3.9. уже доступна в тестовом варианте):

> $expand
> Данный параметр позволяет вместе с результатами основного запроса получать значения связанных сущностей, что позволит не запрашивать каждую сущность отдельно.

**PQR** 25 августа 2016 в 19:04 0

Это то что нужно! Спасибо, попробую.

**Serginio1** 25 августа 2016 в 22:53 0

Молодцы! Но год назад не работало

Linq To ODATA

Да есть возможность в ODATA сразу подгружать нужные свойства

```
var result = (from Тестовый in context.Catalog_Тестовый.Expand("РеквизитСправочник")
                          where Тестовый.Ref_Key == new Guid("aada18ad-5308-11e5-8e05-c86000c70663")
                          select Тестовый).SingleOrDefault();
```

Но пока в 1С Параметр $expand не поддерживается"

**VasiliyKudryavtsev** 25 августа 2016 в 23:32 0

Год назад 8.3.9 не было)

**LuckyRaul** 25 августа 2016 в 18:41 0

Пытались пользоваться odata через Json. Вообще не пашет.

И хотя написано в заголовках написана одна версия odata, ответы присылает по другой.

У кого нибудь получилось состыковаться и изменить данные в 1с

**Serginio1** 25 августа 2016 в 22:56 0

Вот здесь Linq To ODATA я работал с JSON. Можно посмотреть фиддлером запросы

**LuckyRaul** 26 августа 2016 в 11:36 0

Нас больше интересует интеграция с web языками, с линукса.

**Serginio1** 26 августа 2016 в 11:41 0

Ну можно и с линукса 1С, Linux, Excel, Word, OpenXML,ADO и Net Core
Там суть в другом. Когда выставляешь context.Format.UseJson(); то идут заголовки и 1С возвращает данные в формате JSON.

**LuckyRaul** 26 августа 2016 в 11:47 0

Мы пытались адаптировать готовую либу для odata под 1С. Сразу оно естественно не заработало.

Столкнулись с тем, что когда кидаешь в 1С запрос в формате json с заголовками «хочу-json» получаешь ответ в xml. Добавив url $format=application/json получили наконец json.

На этом этапе мы смогли получать данные от туда норм.

Дальше встал вопрос сохранения в json. Тут сколько не бились, ничего не получилось. Потом я заметил расхождения в версиях Odata смотря на документацию на сайте odata и мы отложили это дело.

**unpete** 29 августа 2016 в 18:41 0

> интеграция с web языками, с линукса

В metadata.js есть двунаправленная шина данных — хоть в браузере, хоть под nodejs. К ней можно подключить 1С-ные и не 1С-ные базы разной структуры. Вместе с библиотекой интеграции, получается почти честная CRDT для 1С.

**zuek** 26 августа 2016 в 09:24 0

> Внешние источники данных – это прикладной объект конфигурации 1С, позволяющий взаимодействовать с любой ODBC-совместимой базой данных как на чтение, так и на запись.

Так сложилось, что последние лет 6 я занимаюсь сопровождением 1С под Linux (в качестве скорее сисадмина, чем разработчика), и постоянно сталкиваюсь с проблемами, что в стандартных конфигурациях вроде всё замечательно работает, но вдруг вылезает захардкоженное обращение к сущностям, отсутствующим у моих подопечных как класс — то штрихкод формируется прямым обращением к .dll (уже нет), то для формирования файлов отчётности в xml (!) требуется наличие Internet Explorer (по крайней мере было справедливо для УПП годичной давности), то путь к [файловому] хранилищу обязательно должен оканчиваться символом "\" (а у меня все пути оканчиваются "/")… вот теперь ещё и про «ODBC-совместимые базы данных» напомнили… с публикацией на web-сервере из Linux — вообще отдельная боль.
Хотелось бы узнать — есть ли, и если есть, то насколько приоритетна задача создания платформы полностью работоспособной без Windows?

**PeterG** 26 августа 2016 в 09:30 0

> Внешние источники данных – это прикладной объект конфигурации 1С, позволяющий взаимодействовать с любой ODBC-совместимой базой данных как на чтение, так и на запись.

ODBC-драйвера есть и под Linux. Соответственно, внешние источники данных работают и под Linux.

> Хотелось бы узнать — есть ли, и если есть, то насколько приоритетна задача создания платформы полностью работоспособной без Windows?

Задача в общем-то решена 1С. Есть много внедрений 1С на Linux. В нашем облачном сервере 1cfresh.com десятки серверов на Linux.
Если в каких-то конфигурациях что-то не работает на Linux — чиним.

**zuek** 26 августа 2016 в 10:56 +1

> Задача в общем-то решена 1С. Есть много внедрений 1С на Linux. В нашем облачном сервере 1cfresh.com десятки серверов на Linux.
> Если в каких-то конфигурациях что-то не работает на Linux — чиним.

Не совсем так. У меня тоже есть несколько внедрений 1С на Linux, и тоже вполне работоспособных, по крайней мере, если используются *последние типовые* конфигурации. По моим наблюдениям, главные недоработки в этом направлении на сегодня:

1. Нет GUI-аналога оснастки "Администрирование серверов 1С Предприятия" — мне пришлось костылить собственный web-интерфейс для дёрганья командной строки на 1С-сервере.
2. УПП содержит много элементов, жёстко ссылающихся на Windows-сущности. Я понимаю, что конфигурация устаревшая, что надо бы на ERP мигрировать, но вот миграция эта, в отличии от Бухгалтерии 2.0->3.0 или "Кадров" оценена только по лицензии в весьма ощутимую сумму денег, а не стоит забывать, что ещё и настройки все надо с нуля переписывать, что очень сильно демотивирует потребителя от такого решения (по сути — новое внедрение). Вот и приходится "грызть кактус", вынимая из сейфа ноутбук с Windows Home, когда приходит пора сдавать отчётность. Благо ещё не пришлось интегрировать в эту конфигурацию складское или производственное оборудование — обошлись обменами с внешними источниками.

**Serginio1** 26 августа 2016 в 11:23 0

Я написал два топика
1С Linux. Что не хватает по сравнению с Windows? InfoStart
1С Linux. Что не хватает по сравнению с Windows? Mista.ru

Но так и не получил внятного ответа. Меня прежде всего интересует что мне добавить, что бы моя разработка заинтересовала людей.
Чего не хватает в Linux по сравнению с Windows

**VolCh** 26 августа 2016 в 23:41 0

Разработчиков 1С, не зацикленных на винде.

**Dementor** 30 августа 2016 в 19:43 0

> главные недоработки в этом направлении на сегодня

У них есть недоработки, но вовсе не те, о которых вы написали.

1) Это не unix-way! Только командная строка, только хардкор! И не просите больше такую ересь :)

2) С 2006 года я стал участвовать в проектах на УПП на платформе Предприятие 8.0 и четко помню, что в то время, когда создавались нелюбимые вами связанные с Windows механизмы, никакого Linux-сервера даже в планах не было, не говоря уже про нативные клиенты, а следовательно не было необходимости что-либо делать кроссплатформенным. Рабочий (а не wine-симуляция) клиент появился относительно недавно, но слишком поздно. Не помню где точно я это прочитал (в инфописьме или в какой-то статье) и не могу привести ссылочку, но компания 1С после релиза рабочей версии ERP отказались от дальнейшего развития функциональности в УПП, только поддержка существующих наработок. Следовательно никакого глобального рефакторинга не будет.

**zuek** 31 августа 2016 в 09:43 0

Про неупомянутые мной недоработки можно почитать в официальных списках — их там сотни, но о них уже сказано и обещано поправить, а вот по поводу:

1. Ну не могу я дежурному специалисту техподдержки группы производственных предприятий давать рутовый (даже судоэрский) доступ к командной строке продакшн-сервера. Да и "не по Ваньке шапка" — тут нужен какой-то доступный и удобный механизм, позволяющий завершить зависшие сеансы (сеансы-зомби, которые проявляют фоновую активность, но клиентский интерфейс уже отвалился от графической оболочки) или выявить источник повышенной нагрузки на сервер. Пока для себя решил эту задачу как раз написанием собственного web-интерфейса, дёргающего команды на сервере, но это как-то по-колхозному смотрится.
2. То, что УПП "похоронили" стало известно примерно через 6 месяцев после начала моего самого крупного внедрения и примерно через месяц после начала его ввода в эксплуатацию. Платформа 8.3 на тот момент обладала отвратительной стабильностью, а ERP была доступна только "для ознакомления", хотя на неё пообещали скидку при условии подписания контракта на внедрение с публичной оглаской. Да, понимаю, что стоило бы на раннем этапе "броситься в омут", но слишком уж высоки были риски и бизнес не согласился.

**VasiliyKudryavtsev** 31 августа 2016 в 10:56 0

Все-таки не похоронили. УПП поддерживается, делается в т.ч. достаточно крупные доработки типа ЕГАИС и пилотного проекта ФСС

**Dementor** 31 августа 2016 в 11:30 0

Для чего нужен рутовский пароль или право на sudo для запуска RAC-утилиты консольного администрирования кластера? Даже не нужно прямого доступа к продакшену. Главное, что бы RAS был поднят и нужные порты открыты.

**zuek** 31 августа 2016 в 12:16 0

Повторюсь — "… нужен какой-то доступный и удобный механизм..." — дежурный специалист (круглосуточный) имеет наименьший необходимый набор навыков для устранения текущих сбоев — если этих навыков не хватает, то он фиксирует внештатную ситуацию и дёргает следующий уровень, что стоит дополнительных денег и нервов. Способный продраться сквозь последовательность команд, позволяющих увидеть картину на сервере/кластере специалист будет стоить значительно дороже, чем специалист, способный в графическом интерфейсе посмотреть на основные показатели, сравнить их с "эталонными" и предпринять соответствующие действия для разрешения проблемы. Почему для Windows такой инструмент есть "из коробки" (к нему, правда, тоже есть немало претензий), а для Linux — нет?
Прям возникает желание привести мою вэб-морду к более-менее унифицированному виду и опубликовать для общественности...

**Dementor** 31 августа 2016 в 12:22 0

Я уже понял, что вам нужно что-то типа такого — http://infostart.ru/public/414870/

Как альтернативу, можно сделать сценарий, который проверит состояние кластера и даст пользователю варианты исправления ошибок. Это даже еще быстрее и удобнее чем в gui-интерфейсе кликать мышкой между базами различных серверов в поиске зависших сеансов.

**pelenur1** 26 августа 2016 в 12:33 +4

Я читаю на хабре про 1С, и никто не пишет свои эмоции при виде кода на русском языке. Неужели свершилось?

**Serginio1** 26 августа 2016 в 14:24 +1

Они на мне все выплеснулись

**dsdred** 27 августа 2016 в 15:11 0

Наверное дочитали до примера англоязычного на JSON…

**cyberyak** 2 мая 2017 в 14:25 0

мыши давно уже превратились в кактус.

**projectsolution** 26 августа 2016 в 12:34 0

Как жаль что у 1С нет реальных конкурентов в РФ по доступной цене. Всем принудительно нужна подписка на ИТС, без сдачи за деньги экзамена нельзя внедрять ERP, открытость почти нулевая, игнорирование просьб разработчиков. Хорошо пока за объем воздуха в желтой коробке счет не выставляют.

**dsdred** 27 августа 2016 в 15:10 0

На счет подписки ИТС, это не огромные деньги. Если брать другие системы там абаненка за тех поддержку куда больше, да и стоимость лицензий… Почему это без здачи экзамена нельзя внедрять ERP? Если ты не франч, то внедряй на здоровье. По игнорированию просьб, по крайней мере для Документооборота есть группа в ВК где общаешься напрямую с разработчиками, там довольно таки оперативно отрабатывают вопросы, предлож

**Arch0n** 6 сентября 2016 в 12:42 0

Вы еще SAP не видели. Там например в шине интеграции между ихними модулями заставят платить за трафик (!) причем на вами же купленных серверах.

**fishca** 7 сентября 2016 в 08:44 0

Бабло пилит бабло, все логично

**VolCh** 26 августа 2016 в 23:39 0

Подскажите, пожалуйста, насколько трудозатратно реализовать со стороны 1С обращение к веб-сервису а-ля рестфул в локальной сети для импорта данных по требованию иили расписанию о финансовых операциях от контрагентов до движений реальных средств и начислений по счетам учёта? Почему-то со стороны специалистов по 1С встречается ярое противодействие. Плюс такое же ярое против выгрузки данных в Постгрес-базу под Линуксом, причём в условиях когда сама 1С крутится на Постгресе под Линуксом. Такое ощущение, что сам этот факт для них шоком является, уверены, что всё под виндой и мсскулем, к постргресу пытаются подрубиться через сколь сервер студию или как там её даже без odbc.

**MinimaJack** 29 августа 2016 в 09:47 0

обращение сделать недолго… проблемно реализовать корректный импорт данных. Так как даже одни и те же данные(цифрово-визуально) должны быть введены по разному в 1С ( пример: покупка-продажа валюты и списание-поступление на расчетный счет) …
Для денежных средств есть экспорт-импорт из КБ — работает нормально.

**Falseclock** 16 января 2017 в 17:00 0

Черт дернул обновиться с 8.3.8.2197 на 8.3.9.2033

Плоды массового обновления на праздниках коту под хвост. Придется везде откатываться.

Теперь ODATA не работает на Linux серверах, сразу и не заметил, пока пользователи не пришли и не стали жаловаться, что CRM не работает. По HTTP даже не запрашивает пароль и вылетает с ошибкой 408/409

1C:Enterprise 8 application error:

Error executing query
by reason:
server_addr=tcp://1c:11560 descr=recv returns zero, disconnected line=1570 file=./src/DataExchangeTcpClientImpl.cpp

**PeterG** 17 января 2017 в 18:04 0

На v8@1c.ru написали?

**vistoyn** 21 февраля 2017 в 18:10 0

Я так понял для типовых базовых конфигураций odata не работает.
Какой тип конфигурации и лицензии нужно покупать, чтобы odata работал?

**Falseclock** 22 февраля 2017 в 08:57 0

на сколько я знаю ODATA не зависит от типа конфигурации. Просто в последних релизах конфигураций ODATA для каждого документа надо включать отдельно

**vistoyn** 22 февраля 2017 в 11:22 0

как включить ODATA для всех документов и справочников?

Я нашел в документации:
Механизм установки состава объектов, доступных с помощью стандартного интерфейса OData, можно выполнить в виде внешней обработки. Для этого не требуется модифицировать прикладное решение.

Как запустить внешнюю обработку?

**Falseclock** 22 февраля 2017 в 11:26 +1

https://drive.google.com/open?id=0B2lAfZOYZcTxMXFEMlVDSUJSMFk

вот обработка внешняя. Возможно, для вашей конфигурации надо будет чуток подшаманить в конфигураторе.

**vistoyn** 22 февраля 2017 в 12:32 0

Спасибо! Помогло!

Нужно было создать функцию "СведенияОВнешнейОбработке", как написано по ссылке:
http://1c-pro.ru/threads/stati-po-razrabotke-k-rassmotreniju.47563/

и интерфейс OData заработал!

**Falseclock** 22 февраля 2017 в 12:37 0

Класс! Тогда ловите еще одну удобную штуку, которую давеча сделал

http://forum.infostart.ru/forum9/topic166045/

Реальный пример как вытаскивать отчеты в виде HTML, JSON или XLS.
Можно и XML прикрутить по желанию

**Falseclock** 22 февраля 2017 в 09:18 0

не туда написал

**vistoyn** 22 февраля 2017 в 14:40 0

В типовой конфигурации базовой лицензии есть ограничение на одновременную работу в один пользователь.
т.е. 1С OData считает за пользователя.
Даже если обращения к OData не идут, то предприятие не запускается, говорит есть подключения.
Если перезагрузить IIS, то он пустит. Но кто будет каждый раз перезапускать IIS?
Я так понимаю, где-то должен быть параметр в IIS или в 1С, который отвечает за таймаут, чтобы по истечении этого времени IIS отпустил базу.
Кто нибудь сталкивался с этим?

**vistoyn** 22 февраля 2017 в 15:53 0

Нашел параметр при публикации веб клиента "Время жизни соединения", поставил 120сек. Но он видимо не влияет. После того как закрывается http соединение к odata, через 60сек становится возможным редактировать данные в 1С. Главное, чтобы все http соединения были закрыты.
Я проверял в браузере хром и через wget. 60с таймаут и там и там.
Вобщем odata работает :)

**Falseclock** 22 февраля 2017 в 21:16 0

в последних версиях платформы есть параметр использования сессий. Поставьте без сессий и будет счастье

```
<standardOdata enable="true"
               reuseSessions="dontuse"
               sessionMaxAge="20"
               poolSize="10"
               poolTimeout="5"/>
```

**Falseclock** 22 февраля 2017 в 09:19 0

@1С, вот у меня наболевший вопрос. В платформе найдены баги. Я как конечный пользователь не могу никуда написать о них. Пишу партнерам по франшизе, а они отвечают, что они могут только сообщать об ошибках в конфигурациях.

на https://partners.v8.1c.ru доступ никто не дает.

Вот такой вот замкнутый круг.

**fishca** 22 февраля 2017 в 11:04 0

Если найдена ошибка в платформе, пишете письмо на v8@1c.ru

http://1c.ru/rus/support/support.htm#8

**o4karek** 22 февраля 2017 в 15:19 0

Если вы находите ошибку в программном продукте 1С (конфигурация или платформа — неважно), и вы являетесь зарегистрированным пользователем этого программного продукта, для сообщений об ошибках существует электронная почта линии консультации: v8@1c.ru

> Пишу партнерам по франшизе, а они отвечают, что они могут только сообщать об ошибках в конфигурациях.

Вот это похоже на отмазку франча :) Если я правильно понял фразу «партнерам по франшизе».

**Falseclock** 8 марта 2017 в 13:02 0

ну я написал одному из крупнейших партнеров, на что получил вот такой вот ответ:

> Добрый день!
> Мы оказываем консультационные услуги только по использованию типовых решений для Казахстана и отраслевых решений компании «1С-Рейтинг».
> Вопросы по использованию возможностей платформы «1С: Предприятие» при доработке/разработке конфигураций Вы можете задавать на специализированном форуме участников-партнеров https://partners.v8.1c.ru/.
>
> В частности, по Вашему вопросу есть похожие сообщения https://partners.v8.1c.ru/forum/topic/1560477#m_1560477, https://partners.v8.1c.ru/forum/message/1561229#m_1561229, в которых данная ситуация признана ошибкой платформы.

**Falseclock** 24 февраля 2017 в 20:44 0

А можно ли с помощью расширения конфигурации отлавливать допустим создание, изменение, отмену проводки, пометки на удаление и удаление какого-либо документа и инициировать http запрос на сторонний сервер с указанием данных, например uuid документа и тип операции?

Почему с помощью расширения? — Не хочется снимать с поддержки конфигурацию.
Задача информировать внешние интеграционные системы об изменениях в базе, нежели регулярно опрашивать.

**PeterG** 1 марта 2017 в 17:03 0

Да, можно. Можно в соответствующем модуле сделать обработчик «До» или «После», который выполнит необходимое вам действие.

Архитектурно правильнее не вызывать веб-сервис (да и вообще внешние ресурсы) из обработчиков, т.к. время их отклика труднопредсказуемо и может существенно замедлить работу системы. Правильнее будет оформить вызовы веб-сервиса в фоновом задании, а из обработчика инициализировать их выполнение.

**Falseclock** 8 марта 2017 в 12:58 0

Спасибо. Не дожидаясь ответа реализовал еще пару недель назад.
Вызов веб сервера осуществляется локально, то есть в одной подсети виртуальных машин на гипервизоре.
Сам веб сервер только лишь принимает данные и транслирует их в сокет и отвечает кодом 200. То есть по сути является прокси.

А сокеты уже сами куда надо отправляют и обрабатывают данные. Так что не вижу смысла формировать фоновые задания.

Единственное, в расширениях нельзя иметь константы, обработчики и чего-то еще… не помню… пришлось снимать с поддержки.

**cyberyak** 2 мая 2017 в 14:22 0

Спасибо за статью. Из статьи я понял, что можно обращаться к 1С приложениям через REST из внешних приложений. Подскажите, пожалуйста, можно ли использовать внешние RESTful сервисы и обращаться к ним из 1С? Есть ли способы интеграции с RESTful старым клиентам, на 8.2, предыдущих 8.3 платформах?

**PeterG** 2 мая 2017 в 15:56 0

> можно ли использовать внешние RESTful сервисы и обращаться к ним из 1С?

Да, начиная с версии платформы 1С: Предприятие 8.3.6.

> Есть ли способы интеграции с RESTful старым клиентам, на 8.2, предыдущих 8.3 платформах?

Насколько знаю — можно, через объект HTTPСоединение. Объект HTTPСоединение появился в платформе достаточно давно, чуть ли не с 8.3.1. В общем — если в коде доступен объект HTTPСоединение, то, скорее всего, интеграция возможна (если вызываемый сервис не использует какие-то экзотические атрибуты).

**cyberyak** 2 мая 2017 в 16:50 0

Спасибо за оперативный ответ. Надо пробовать, жаль, многие клиенты на 8.2…

**PeterG** 3 мая 2017 в 12:56 0

Прошу прощения, ошибся. Это:

> Объект HTTPСоединение появился в платформе достаточно давно, чуть ли не с 8.3.1.

надо читать как:

> Объект HTTPСоединение появился в платформе достаточно давно, чуть ли не с **8.1**

могу ошибаться, но вроде бы HTTPСоединение появилось аж в 8.1

**VasiliyKudryavtsev** 4 мая 2017 в 11:16 0

Не совсем так.
Автоматическое REST API и http-сервисы, создаваемые разработчиком появились в 8.3.5

А HTTPСоединение было давным давно, но большинство возможность были добавлены в 8.2.18

Только полноправные пользователи могут оставлять комментарии. Войдите, пожалуйста.

САМОЕ ЧИТАЕМОЕ

Сутки | Неделя | Месяц

**Как мы запустили роботов в маленький Чернобыль**

+110 16,4k 86 84

**NVIDIA Jetson Nano: тесты и первые впечатления**

+32 17,8k 51 22

**Вот это поворот: почему Apple изменила требования к разработчикам приложений**

+14 11,4k 20 15

**Самодокументируемый код – это (как правило) чушь**

+13 15,2k 67 99

**Мы пришли с миром от всего человечества**

+21 10,6k 32 41



© 2006 – 2019 «ТМ»