

Взаимодействие между базами 1С через COM

В этой статье я расскажу о многих особенностях взаимодействия между базами 1С по COM технологии, накопленных за годы развития и использования интеграционного коммерческого продукта 2iS:Интеграция нашей компании 2iS.

Термины

ОС – операционная система

COM-клиент – процесс(поток) ОС, запросивший создание COM-объекта

COM-сервер – COM-объект, созданный по запросу клиента

Com-соединитель – COM-объект V8X.ComConnector для создания внешних соединений с базами 1С

Automation-сервер – он же OLE-сервер, COM-объект V8X.Application для создания приложений 1С

Издание платформы – подстрока до 2-й точки версии платформы (например 8.2, 8.3)

Сборка платформы – полная строка версии платформы (например 8.2.19.130, 8.2.16.352)

COM классы платформы

Виды COM классов

Для каждого издания платформы в системном реестре регистрируются собственные классы для взаимодействия с базами. Символ “X” обозначает место подстановки номера издания платформы (например 0,1,2,3,4).

1. V8X.ComConnector - Com-соединитель / Менеджер COM-соединений
 - a. С помощью данного объекта выполняется установка внешнего соединения с информационной базой 1С:Предприятия 8 и администрирование кластера серверов. С помощью одного экземпляра объекта может быть установлено неограниченное число соединений.
 - b. Регистрация и отмена регистрации выполняется отдельно для каждой разрядности (32b и 64b)
 - i. Регистрация: Regsvr32 “<ПутьНужнойРазрядности>\bin\comcntr.dll”
 - ii. Отмена регистрации: Regsvr32
“<ПутьНужнойРазрядности>\bin\comcntr.dll” /u
2. V8X.Application - Automation-сервер толстого клиента
 - a. Предоставляет доступ ко всем свойствам и методам глобального контекста клиентского приложения, а также включение и выключение пользовательского интерфейса (главного окна приложения).
 - b. Может запускать как обычное, так и управляемое приложение в зависимости от параметров конфигурации и пользователя базы, но явно задавать тип приложения нельзя.
 - c. Регистрация и отмена регистрации
 - i. Регистрация - 1cv8.exe /regserver
 - ii. Отмена регистрации - 1cv8.exe /unregserver
3. V8Xc.Application - Automation-сервер тонкого клиента
 - a. Предоставляет доступ ко всем свойствам и методам глобального контекста клиентского приложения, а также включение и выключение пользовательского интерфейса (главного окна приложения).
 - b. Регистрация и отмена регистрации
 - i. Регистрация - 1cv8c.exe /regserver
 - ii. Отмена регистрации - 1cv8c.exe /unregserver

Привязка COM классов к dll-файлам

Привязка имени COM класса к конкретному dll-файлу осуществляется через ветку HKEY_CLASSES_ROOT системного реестра, которая собирается [следующим образом](#):

1. Берется содержимое ветки HKEY_CURRENT_USER\Software\Classes
2. Добавляется содержимое ветки HKEY_LOCAL_MACHINE\Software\Classes, но только те ключи, которые отсутствуют в п.1.

Поэтому регистрация класса от имени пользователя имеет приоритет над общей регистрацией.

Для 32-б и 64-б режимов исполнения кода в реестре хранятся отдельные ветки и соответственно отдельные привязки. Поэтому управление COM-классами для них осуществляется независимо. В частности это касается класса V8X.ComConnector (на 64-б операционной системе)

1. 64-б - c:\Program Files\1cv8*\<Версия>\bin\comcntr.dll
2. 32-б - c:\Program Files (x86)\1cv8*\<Версия>\bin\comcntr.dll

Подключение внешнего соединения

Рассмотрим алгоритм подключения внешнего соединения в деталях

1. Создание COM-объекта Com-соединитель (Имя класса - V8X.ComConnector).

Варианты:

i. Внутривпроцессное (inproc)

1. Должен быть зарегистрирован класс V8X.ComConnector той же разрядности, что и COM-клиент
 - a. Если издание платформы COM-клиента совпадает с изданием платформы COM-сервера, то также требуется равенство сборки платформы COM-клиента и COM-сервера.
 - b. Он регистрируется при установке платформы с опцией «Внешнее соединение». В любой момент его можно зарегистрировать командой «Regsvr32 comcntr.dll» в каталоге исполняемых файлов платформы.
2. COM-объект создается в памяти COM-клиента
3. Затраты на создание примерно – 0.01(первое)/0.001(второе) сек
4. Пример

```
КомСоединитель = Новый СОМОбъект("V83.ComConnector");  
ВнешнееСоединение = КомСоединитель.Connect(СтрокаСоединения);
```

ii. Внепроцессное локальное (local)

1. Должен быть зарегистрирован класс V8X.ComConnector любой разрядности и сборки платформы. Необходимо настроить через dcomcnfg.exe COM+ приложение с активацией «серверное приложение»/«Server application». Подробнее рассмотрено далее.
2. COM-объект создается в памяти служебного хост-процесса (dllhost.exe) на компьютере COM-клиента
3. Затраты на создание примерно – 0.06(первое)/0.03(второе) сек
4. Пример

```
КомСоединитель = Новый СОМОбъект("V83.ComConnector");  
ВнешнееСоединение = КомСоединитель.Connect(СтрокаСоединения);
```

iii. Внепроцессное нелокальное (DCOM)

1. На нелокальном компьютере должен быть зарегистрирован класс V8X.ComConnector любой разрядности и сборки платформы. Необходимо

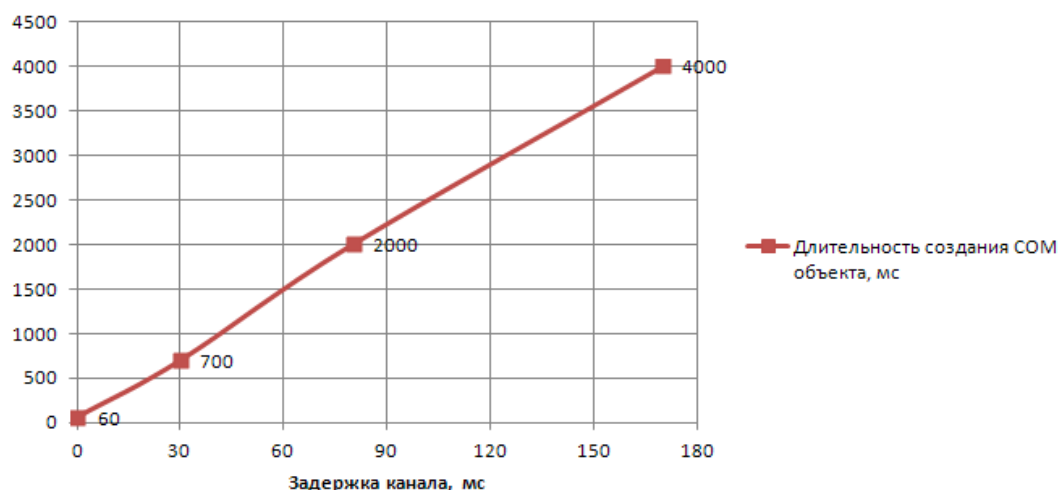
настроить на нелокальном компьютере через dcomcnfg.exe COM+ приложение с активацией «Серверное приложение»/«Server application». Подробнее рассмотрено далее.

2. Необходимо настроить на нелокальном компьютере роль «Сервер приложений»/«Application server» и включить нелокальный доступ к COM+, выбрав опцию «Сетевой доступ к COM+»/«COM+ network access». Подробнее рассмотрено далее.
3. COM-объект создается в памяти служебного хост-процесса (dllhost.exe) на другом компьютере
4. Затраты на создание примерно

а. Экспериментальные данные

Задержка канала, мс	Длительность создания COM-объекта, мс
0	60
30	700
80	2000
170	4000

**Длительность создания COM
объекта от задержки канала**



На графике видно, что длительность создания этого COM-объекта прямо пропорциональна задержке канала и коэффициент пропорциональности составляет примерно 24. Можно предположить, что при создании этого COM-объекта происходит 24 вызова через канал.

5. Пример

```
КомСоединитель = Новый СОМОбъект("V83.ComConnector",  
ИмяКомпьютера);  
ВнешнееСоединение = КомСоединитель.Connect(СтрокаСоединения);
```

2. Выполнение метода Connect (создание внешнего соединения)
 - i. Создание сеанса – 1 сек
 - ii. Инициализация сеанса – [0;?] сек
 1. Подгрузка нужных метаданных (кэш метаданных не используется)
 2. Инициализация параметров сеанса
 3. Событие ПриНачалеРаботыСистемы

Подключение Automation-сервера (OLE)

1. Создание COM-объекта Automation-сервера (Имя класса - V8X.Application).
 - a. Затраты на создание примерно – 0.5 сек
2. Выполнение метода Connect
 - a. Создание сеанса – 1 сек
 - b. Инициализация сеанса – [0;?] сек
 - i. Подгрузка нужных метаданных (кэш метаданных используется, если база есть в списке пользователя)
 - ii. Инициализация параметров сеанса
 - iii. ПередНачаломРаботыСистемы
 - iv. ПриНачалеРаботыСистемы
3. Пример

```
AutomationСервер = Новый СОМОбъект("V83.Application");  
AutomationСервер.Connect(СтрокаСоединения);
```

Затраты на СОМ-вызовы

Каждый раз, когда мы обращаемся во встроенном языке к свойству или методу СОМ-объекта (выполняем СОМ-вызов), происходит ряд дополнительных действий по сравнению с обращением к родному объекту встроенного языка. Если СОМ-сервер находится в другом локальном процессе, то при этом еще сначала выполняется перенос всех параметров из СОМ-клиента в СОМ-сервер и в конце выполняется обратный перенос параметров в СОМ-клиент. Для такого переноса информация естественно должна быть сериализована на стороне отправителя и затем десериализована на стороне получателя, что в совокупности с некоторыми другими действиями получило название маршалинга и демаршалинга. В нелокальном режиме (DCOM) такой перенос будет еще тратить время на сетевые коммуникации и при первом обращении к новому СОМ-объекту на передачу его «скелета», и поэтому задержка канала будет играть решающую роль ([Подробнее о СОМ вызовах](#), [DCOM в википедии](#), [Технический обзор DCOM](#),)

Соотношение затрат на вызов по типам расположения СОМ-сервера

1. Минимальные для внутрипроцессного Com-сервера
3. Средние для локального внепроцессного Com-сервера
4. От средних до больших для нелокального внепроцессного Com-сервера в зависимости от качества канала связи

Минимизация количества СОМ-вызовов

1. Создаем базовую внешнюю обработку с набором общих методов для приема вызовов от СОМ-клиента на стороне СОМ-сервера
 - a. Теперь нам не нужно изменять конфигурацию базы СОМ-сервера, чтобы использовать там сложный функционал.
 - b. В нашем продукте мы назвали ее «Сервисный процессор». На ее передачу и инициализацию уходит около 1 сек
 - c. Пример универсальной передачи файла в СОМ-сервер

```
ДвоичныеДанные = Новый ДвоичныеДанные(ИмяФайлаЛокальное);  
ЗаписьХмл = Новый ЗаписьXML;  
ЗаписьХмл.УстановитьСтроку();  
ЗаписатьXML(ЗаписьХмл, ДвоичныеДанные);  
ЧтениеХмлНаСервере = СОМСервер.NewObject("ЧтениеXML");
```

- ```

ЧтениеХмлНаСервере.УстановитьСтроку(ЗаписьХмл.Заккрыть());
ДвоичныеДанныеНаСервере =
СОМСервер.ПрочитатьХМЛ(ЧтениеХмлНаСервере);
ИмяФайлаНаСОМСервере=
СОМСервер.ПолучитьИмяВременногоФайла(Файл.Расширение);
ДвоичныеДанныеНаСервере.Записать(ИмяФайлаНаСОМСервере);

```
- d. Пример вызова обработки
 

```

СОМОбработка =
СОМСервер.ВнешниеОбработки.Создать(ИмяФайлаНаСОМСервере, Ложь);
Результат = СОМОбработка.МойМетод(ПараметрыХМЛ);

```
  2. Проектируем взаимодействие с СОМ-сервером по принципу "минимум СОМ-вызовов", т.е. готовим все нужное на стороне СОМ-клиента и передаем в качестве параметров "большого" вызова
    - a. Создаем встроенную обработку для выполнения такого вызова в СОМ-клиенте
    - b. Передача агрегатных объектов
      - i. Сериализуем все входные параметры на стороне СОМ-клиента
      - ii. Десериализуем все входные параметры на стороне СОМ-сервера
      - iii. Выполняем действия с параметрами на стороне СОМ-сервера
      - iv. Сериализуем все выходные параметры на стороне СОМ-сервера
      - v. Десериализуем все выходные параметры на стороне СОМ-клиента
    - c. В нашем продукте мы назвали ее «Выполнение сервиса инфобазы»
  5. Для многократных СОМ-вызовов предусматриваем регулируемую частоту
    - a. Например, информационные вызовы могут выполняться с изменяемой частотой

## Обратная связь

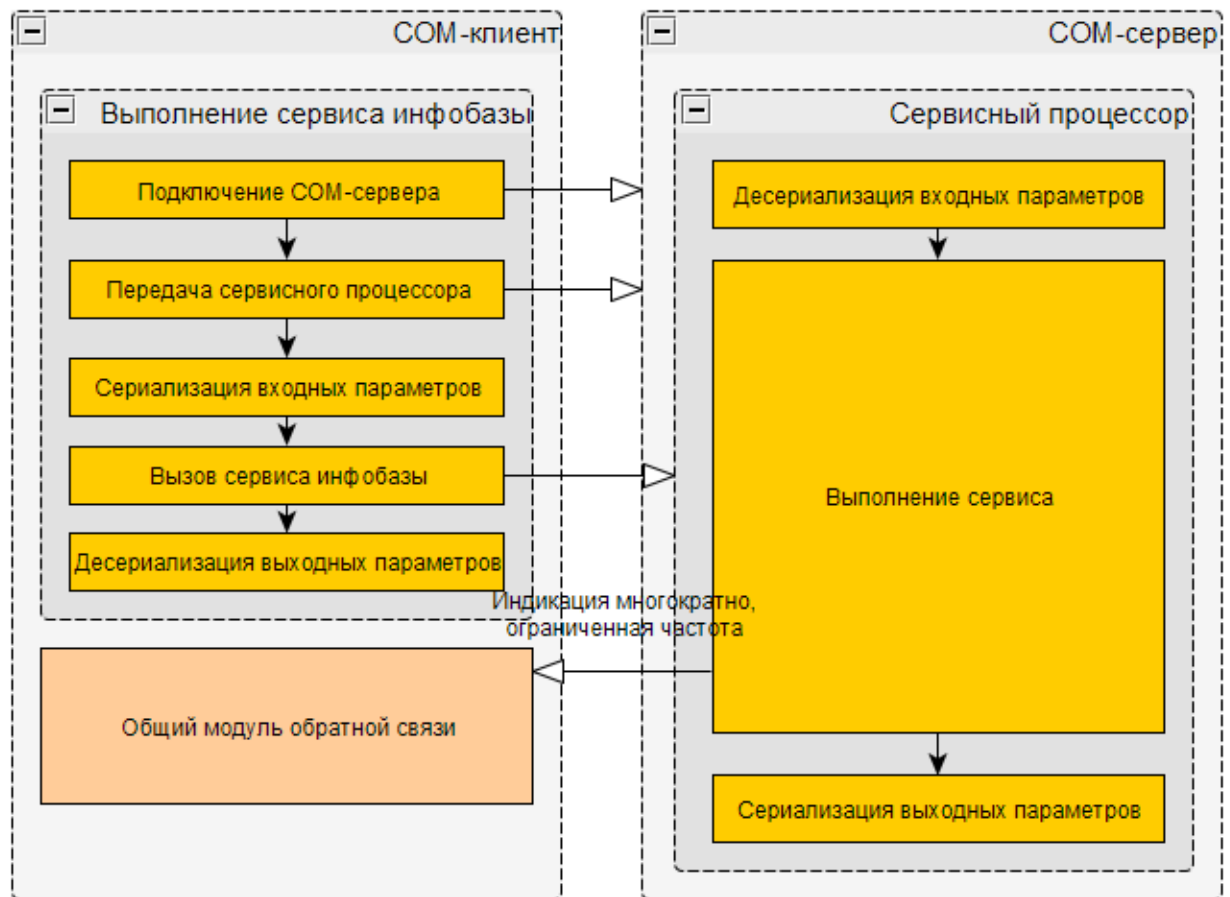
При использовании принципа «минимум СОМ-вызовов» значительную часть времени код будет выполняться в чужом контексте, что серьезно усиливает некоторые неудобства СОМ-вызовов:

1. Нельзя прервать поток СОМ-Клиента и его соединение с сервером приложений 1С, пока не завершится вызов СОМ-сервера
2. У СОМ-Клиента нет информации о прогрессе выполнения вызова в СОМ-сервере

Для борьбы с этими неудобствами можно организовать обратную связь от СОМ-сервера к СОМ-клиенту, передав СОМ-Серверу ссылку на общий модуль СОМ-клиента и периодически вызывая через нее СОМ-клиент. Нужно отметить, что есть ошибки платформы (<https://partners.v8.1c.ru/forum/t/1382465/m/1382465>), проявляющиеся при создании ссылок на один общий модуль более чем в одном СОМ-севере сеанса. Поэтому пока лучше очищать ссылки на общий модуль во всех удерживаемых СОМ-серверах перед помещением в новый. В обратных вызовах можно передавать:

1. информацию о прогрессе выполнения
2. накопленные сообщения пользователю (получать функцией `ПолучитьСообщенияПользователю()`)

## Схема принципа “минимум COM-вызовов”



Такая схема дружелюбна к большим задержкам каналов связи. Поэтому она во многом похожа на вызов веб-сервиса. А следовательно используя ее для COM взаимодействия, вы значительно снижаете затраты на возможные переходы между COM и веб-сервисом в обоих направлениях.

## Отладка

При использовании принципа «минимум COM-вызовов» облегчить отладку выполнения кода на COM-сервере можно, сделав опциональное модальное окно подключения к базе в COM-клиенте. В этом окне должны быть возможности задания одноразовых параметров подключения, выполнения вспомогательных функций и просмотра информации о подключенном сеансе. Ниже в качестве примера показано такое окно из нашего продукта

**Подключение сеанса инфобазы**

Все действия ▾

Способ выполнения:

Инфобазы

Инфобазы:   Способ вызова:

Учетная запись:

Имя пользователя:

Пароль пользователя:

1C

Отладочные сообщения: ☐

Выполнять алгоритмы через внешние обработки: ☒

Сервер DCOM:     Локальный: ☐

Запускать как приложение: ☒ Видимость: ☒ Открыть инструменты разработчика: ☐

Включить отладку внешних соединений: ☐

Выводить сообщения без буферизации: ☐

Режим загрузки системы 2IS:

Удерживаемый сеанс

Использовать: ☒

Номер сеанса:  Начало сеанса:  Пользователь:

Сервер DCOM:    Идентификатор процесса:  Пользователь:

Сервисный процессор:

После создания COM объекты сеансов инфобаз удерживаются в памяти сеанса интеграции.  
Для каждого сочетания инфобазы, типа подключения и сервера DCOM может удерживаться только один сеанс.

Чтобы включить возможность отладки внешних соединений в подкаталоге conf нужно создать файл [comcntrcfg.xml](#) с соответствующим содержимым. В окне подключения можно сделать кнопку/флажок для программного создания такого файла. Однако много версий платформы содержат ошибки, из-за которой предметы отладки внешних соединений могут быть не видны в отладчике даже с этим корректно настроенным файлом. На платформе 8.3 эти проблемы вроде бы решены.

Пример файла comcntrcfg.xml:

```
<config xmlns="http://v8.1c.ru/v8/comcntrcfg">
 <debugconfig debug="true" debuggerURL="tcp://localhost:1560"/>
</config>
```

Также нельзя забывать о необходимости запускать конфигуратор с той же (до регистра букв) строкой соединения, что и предмет отладки. Поэтому в этом диалоге нужно сделать кнопку для запуска конфигуратора COM-сервера с правильной строкой соединения.

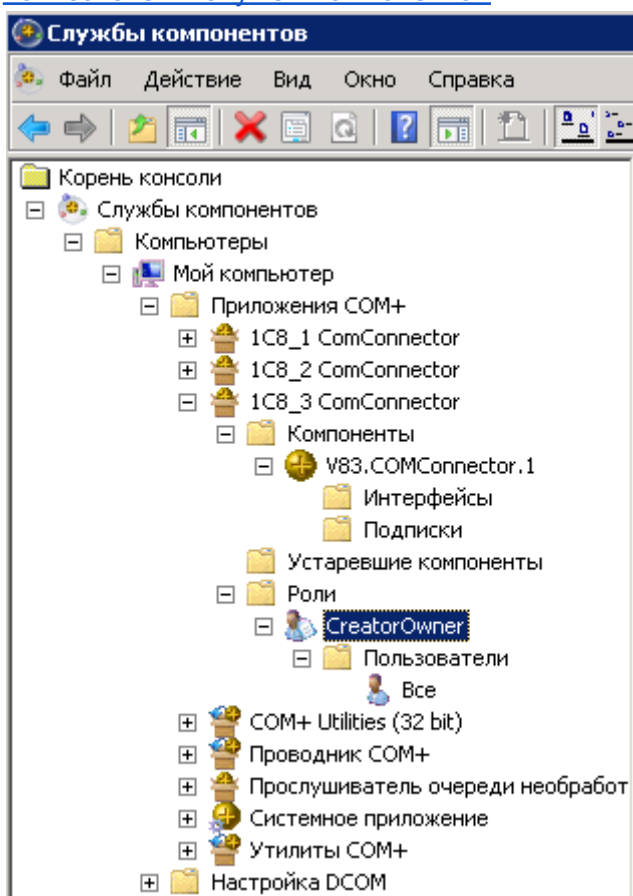
В какой-то мере обойти проблему невидимости предмета отладки внешнего соединения и ряд других неудобств можно путем однократного выполнения кода на толстом клиенте с заменой режима «Внешнее соединение» на «Automation-сервер» (приложение), т.е. используя COM класс V8X.Application и включенную видимость. Поэтому в окне подключения нужны параметры для выбора режима подключения.

После создания подключения на стороне COM-клиента обычно требуется подключение отладчика базы COM-сервера и установка там точки останова в нужном месте. Поэтому в окне подключения нужно расположить кнопку подключения отладчика/конфигуратора

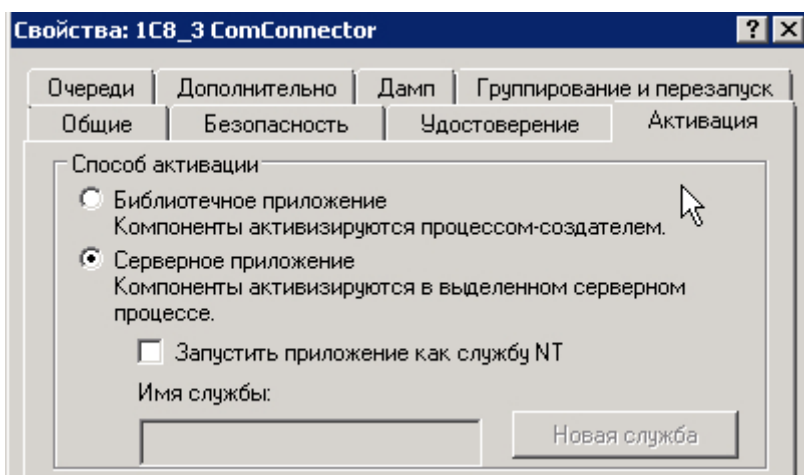
СОМ-сервера. Если СОМ-сервер запущен в режиме видимого приложения, то для подключения отладчика можно воспользоваться асинхронным вызовом исключения в нем. Из диалога с ошибкой в приложении СОМ-сервера через кнопку «Конфигуратор» можно будет открыть конфигуратор и в большинстве случаев сразу подключить отладчик. Если подключение отладчика таким способом недоступно, то оператору поможет отображение информации о сеансе (его номера) в базе СОМ-сервера. Также нужна кнопка открытия в открытом отладчике файла внешней обработки, которая будет выполняться в нем.

## Управление СОМ+ приложениями 1С

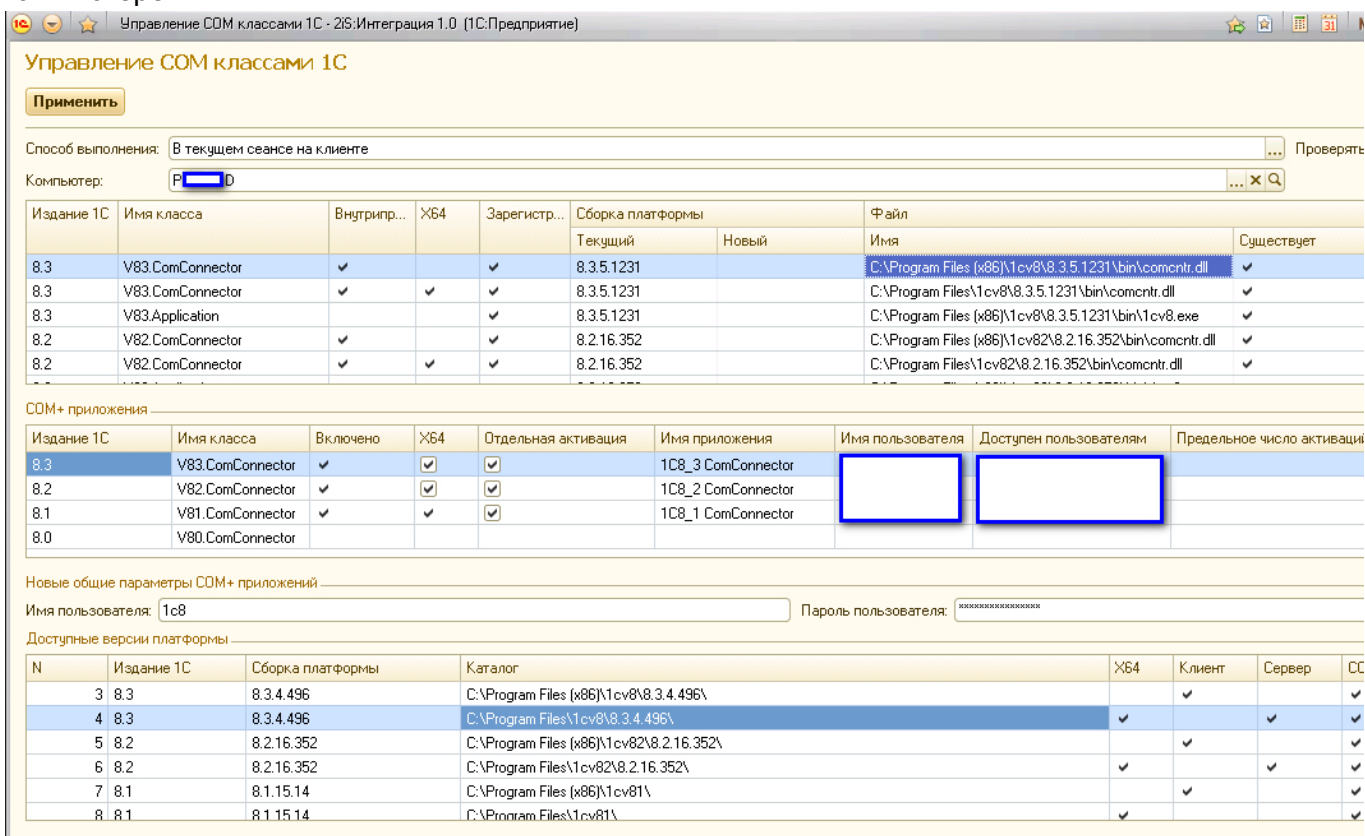
Осуществляется через оснастку dcomcnfg.exe. Для регистрации 1С в качестве СОМ+ приложения необходимо открыть оснастку "dcomcnfg". В ней в ветке "Службы компонентов"- "Компьютеры"- "Мой компьютер"- "Приложения СОМ+" (COM+ Applications) необходимо добавить новое приложение (application). При добавлении можно указать любое имя. Далее необходимо указать пользователя, от имени которого будет работать СОМ+ приложение, на закладке Идентификация (Identity). Также нужно задать доступ к приложению для всех пользователей путем добавления, если отсутствует, роли с любым именем в ветке Роли (Roles), и добавления в ее ветку Пользователи (Users) пользователя Все (Everyone). После добавления приложения необходимо создать новую компоненту. Для этого в ветке "Компоненты" созданного приложения необходимо выбрать пункт меню "Создать" - "Компонент". При создании выбрать импорт уже зарегистрированных компонентов и найдите в списке (желательно 64-битную) компоненту 1С (V8x.COMConnector.1). Подробнее о настройке СОМ+ приложений читайте [Интерфейс пользователя: службы компонентов](#)



Далее можно изменить тип активации (внутрипроцессный/внепроцессный) на закладке Активация/Activation.



В нашем продукте имеется инструмент «Управление COM классами 1С» для просмотра/регистрации/изменения всех COM классов и COM+ приложений 1С на указанном компьютере:



Также управлять COM классами 1С можно через инструмент «Управление COM классами 1С» из подсистемы [Инструменты разработчика](#)

Управление COM классами 1С (ИР) \*

Применить

Файл включения отладки COM соединения

Компьютер:

Сборка платформы:

8.3.18.1201

☐ 64-разрядное

☐ От администратора

☐ Показывать результаты

Пользователь ОС:

CORTEX\Сергей

COM классы

Запустить от администратора

Проверить работу объектов

Издан платфо	Имя класса	Тип класса	X64	Зарегис- трирова	Сборка платформы		Файл		Проверка создания	Проверка подключения
					Текущая	Новая	Имя	Сущес...		
8.3	V83.Application	Application		✓	8.3.18.1201		C:\Program Files (x86)\1cv8\8.3.18...	✓	OK	
8.3	V83.Application	Application	✓	✓	8.3.17.1496		C:\Program Files\1cv8\8.3.17.1496\	✓		
8.3	V83c.Application	CApplication		✓	8.3.18.1201		C:\Program Files (x86)\1cv8\8.3.18...	✓	OK	OK
8.3	V83c.Application	CApplication	✓	✓	8.3.17.1496		C:\Program Files\1cv8\8.3.17.1496\	✓		
8.3	V83.ComConnector	ComConnector		✓	8.3.18.1208	8.3.18.1201	C:\Program Files (x86)\1cv8\8.3.18.1208\bin\comcntr.	✓	OK	Ошибка. Зарегистрированная в COM сборка платформы отличается от текущей
8.3	V83.ComConnector	ComConnector	✓	✓	8.3.17.1496		C:\Program Files\1cv8\8.3.17.1496\	✓		
8.3	V83.ServerAdminScope	ServerAdminSc		✓	8.3.18.1201		C:\Program Files (x86)\1cv8\8.3.18.1201\bin\vdadmin.dl	✓	Ошибка при вызове ...	
8.3	V83.ServerAdminScope	ServerAdminS...	✓							

Сборки платформы

Толстый

Тонкий

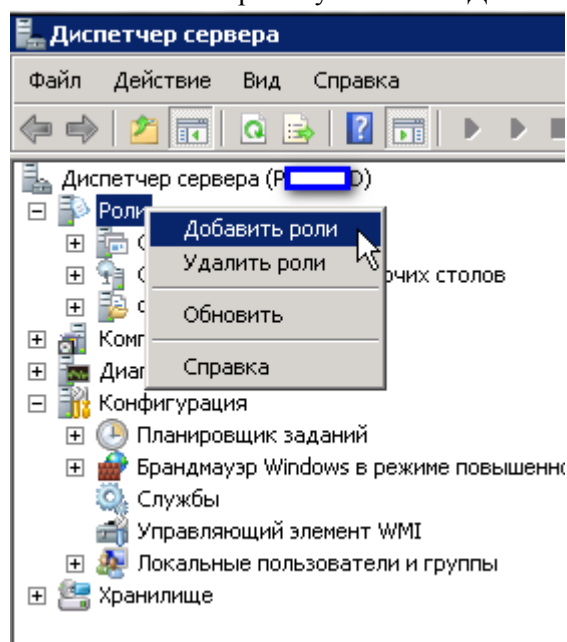
Установить новой

Открыть в проводнике

Издани...	Сборка пла...	X64	Каталог	Файлы существуют	Толстый клиент (Appli...	Тонкий клиент (CAppli...	Внешнее соединение ...	Консоль серверов (S...
8.3	8.3.18.1208		C:\Program Files (x86)\1cv8\8.3.18.1208\	✓	✓	✓	✓	✓
8.3	8.3.18.1201		C:\Program Files (x86)\1cv8\8.3.18.1201\	✓	✓	✓	✓	✓
8.3	8.3.18.1128		C:\Program Files (x86)\1cv8\8.3.18.1128\	✓				
8.3	8.3.17.1549		C:\Program Files (x86)\1cv8\8.3.17.1549\	✓	✓	✓	✓	✓
8.3	8.3.17.1496	✓	C:\Program Files\1cv8\8.3.17.1496\	✓	✓	✓	✓	✓
8.3	8.3.16.1224		C:\Program Files (x86)\1cv8\8.3.16.1224\	✓	✓	✓	✓	✓
8.3	8.3.15.1958		C:\Program Files (x86)\1cv8\8.3.15.1958\	✓	✓	✓	✓	✓

## Настройка DCOM-сервера

Для Windows 2008 Server нужно добавить роль «Сервер приложений» (Application server). Для этого необходимо открыть панель управления компьютером, щелкнуть правой кнопкой мыши в ветки "Роли" и выбрать пункт меню "Добавить роль".



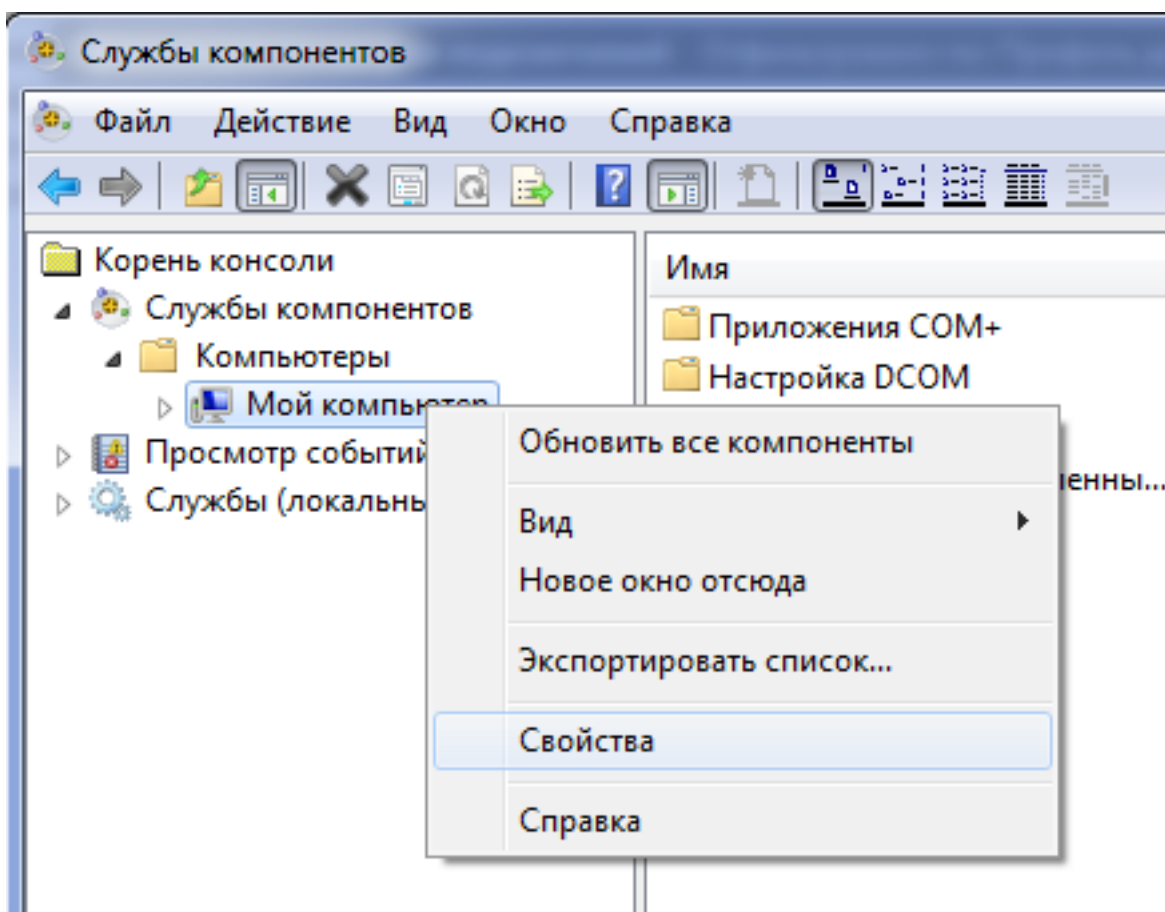
Появится око в котором необходимо выбрать роль "Сервер приложений"/"Application server". На этапе выбора ролей сервера приложений необходимо указать пункт "Сетевой доступ к COM+"/"COM+ network access". Если это не сделать, то при попытке создания COM-объекта вы будете получать ошибку "Ошибка при вызове конструктора (COMОбъект): The component or application containing the component has been disabled".

## Настройка сетевого экрана

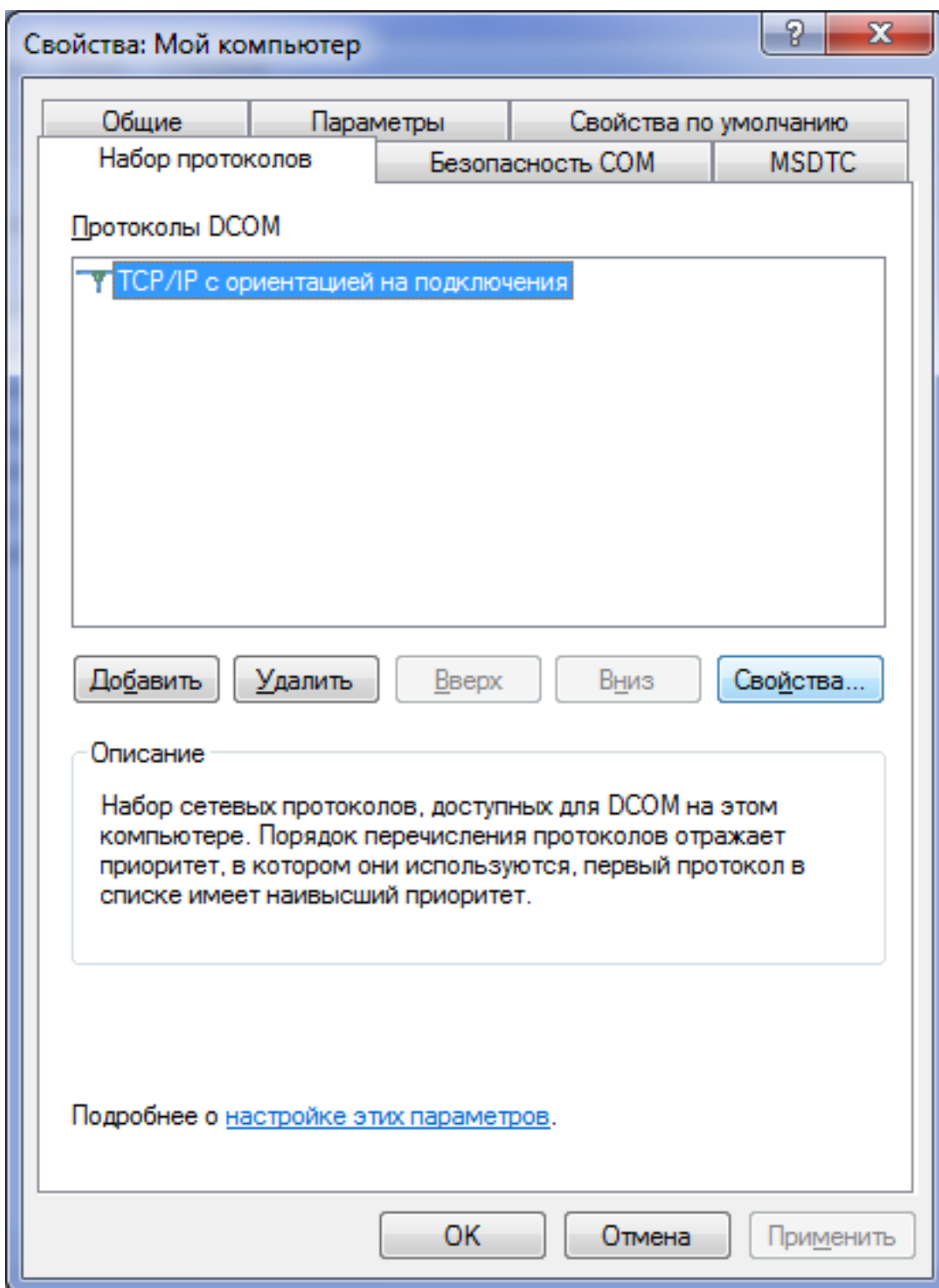
DCOM процессам для связи между собой динамически выдаются порты из диапазона (по умолчанию) от 1024 до 65535. Поэтому в сетевых экранах обоих компьютеров (com-клиента

и com-сервера) должны быть открыты порты RPC (135, 139, 445, 593) и диапазон 1024-65535 (TCP/UDP).

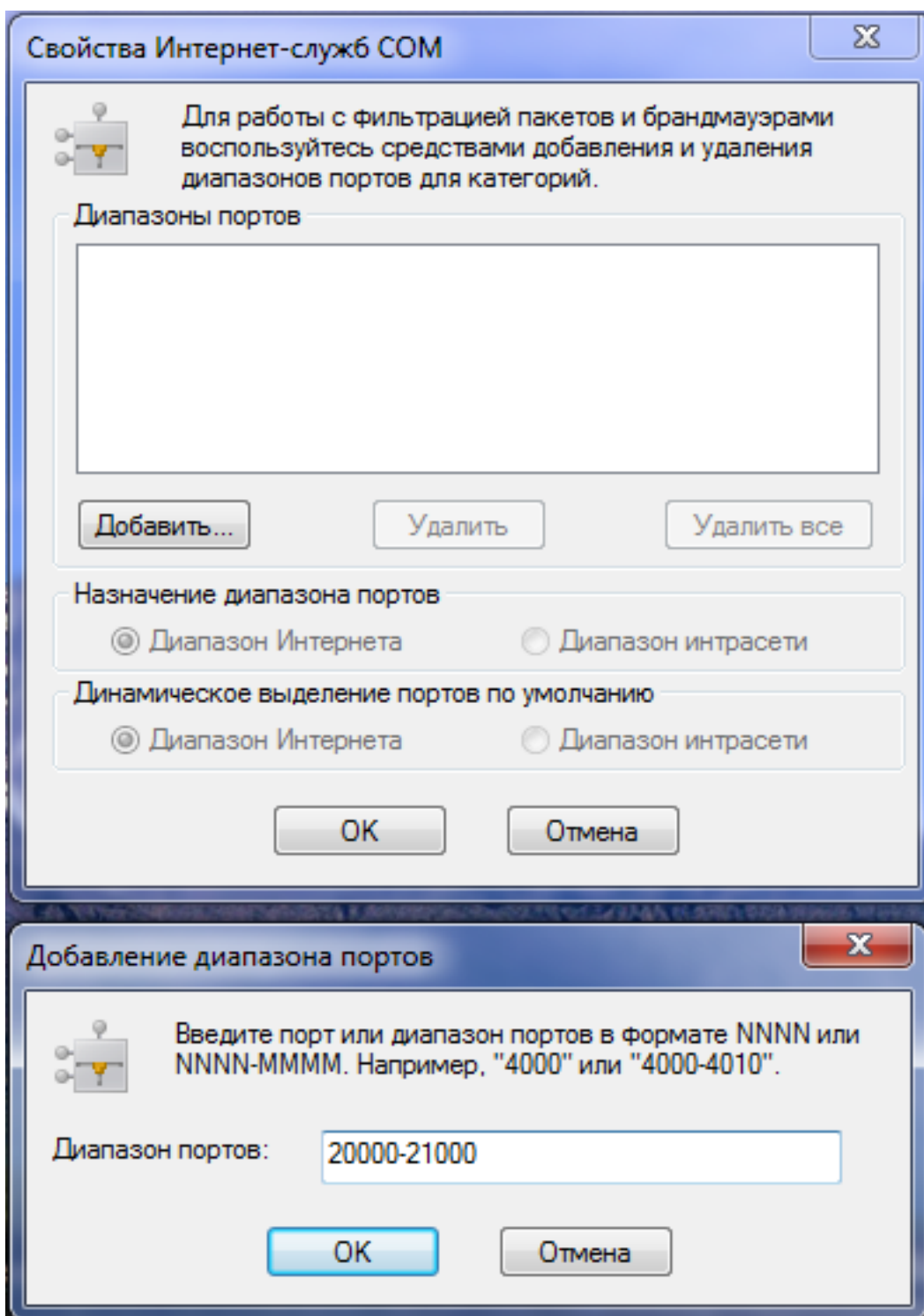
Можно ограничить этот диапазон портов. Для этого открываем Администрирование – Службы компонентов (или в командной строке набираем dcomcnfg.exe). В открывшейся программе находим ветку «Мой компьютер» и открываем её свойства (см. скриншот).



В свойствах открываем вкладку «Набор протоколов», там, в свою очередь, открываем свойства «TCP/IP с ориентацией на подключения»



В свойствах TCP/IP по умолчанию – пустой список. В него можно добавить конкретный диапазон портов, которые будут использоваться DCOM. Какие именно порты выбрать – решать вам. Главное, чтобы не было других стандартных служб, уже занявших выбранные порты. Мне встречались рекомендации открывать не менее 1000 портов, так как очень многие системные программы используют DCOM. Так на скриншоте ниже я указываю, что DCOM должен работать по диапазону портов 20000-21000.



После этого нужно перезагрузить компьютер (перезапустить отдельно DCOM нельзя, слишком много на него завязано), и DCOM станет работать только по указанным портам.

## Проблемы

1. Высокие затраты на подключение/инициализацию
  - a. Удержание com-серверов – эффективное средство борьбы. Варианты:
    - i. Временное хранилище. Производитель не рекомендует, но деваться больше некуда. <http://infostart.ru/public/331683/>
    - ii. Повторное использование возвращаемых значений. Очищаются каждые 20 мин

iii. COM+ пулинг. [v82Server.dll - Технология Microsoft COM+ для доступа к 1C82](#)

2. Разные сборки платформы клиента и сервера при одинаковом издании платформы
  - a. Для обхода для каждой отличной от COM-клиента сборки платформы назначаем и настраиваем DCOM сервер
3. Отсутствие информации о типах вложенных в COM-объекты объектов 1С (вместо типа объекта 1С везде получаем тип «СОМОбъект») и отсутствие контекстной подсказки по ним
  - a. Для обхода применяем принцип «минимум COM-вызовов»
4. При передаче исключения из COM-сервера оно принимает краткую форму из-за ошибки платформы
  - a. Для обхода оборачиваем каждый вызов на стороне COM-сервера в попытку и в исключении перевызываем исключение с полным описанием ошибки
5. При создании COM-объекта ошибка «Версия компоненты 'comcntr' (8.X.XX.XXX) отличается от версии корневого модуля 'core82' (8.X.YY.YYY)»
  - a. Для обхода создаем COM+ приложение с внепроцессной активацией. Тогда COM-сервер будет создаваться в отдельном процессе и совпадения версий этих модулей не потребуются.
6. Медленные каналы (большие задержки)
  - a. Каждой подсети за отдельным медленным каналом назначаем и настраиваем свой DCOM сервер и при обращении к базам за этим каналом COM-серверы создаем на этом компьютере, минимизируя количество и объемы вызовов Com-клиент<->Com-сервер
  - b. Кешируем (повторно используем) COM-серверы
7. Невозможно разорвать соединение рабочего процесса, в котором ожидается завершение вызова COM-сервера
  - a. Для обхода можно использовать создание COM-объектов в выделенных хост-процессах, настроив соответствующим образом COM+ приложение, и регистрировать идентификаторы этих процессов в привязке к сеансам COM-клиентов. Тогда при необходимости завершить серверное соединение такого сеанса можно сначала убить процесс COM-сервера, а потом и сам сеанс.
8. При обращении к COM-объекту может возникать ошибка «Неизвестная ошибка». Возможные причины:
  - a. Com-сервер перестал существовать. Возможные причины:
    - i. Переполнение стека
    - ii. Невосстановимая ошибка
    - iii. Хост-процесс завершился
  - b. Потеря связи с нелокальным COM-сервером
9. Совпадение имен свойств и методов без параметров объекта создает неоднозначность обращения к ним через COM
  - a. Например, если у справочника есть реквизит ПолноеНаименование, то из-за наличия метода ПолноеНаименование() обращение к реквизиту ПолноеНаименование через COM может привести к вызову метода ПолноеНаименование()
  - b. Для обхода используем принцип «минимум COM-вызовов»
10. При создании COM-объекта через DCOM может возникать ошибка "Сервер RPC недоступен"
  - a. Проверьте работу службы «Remote Procedure Call (RPC)» на обоих компьютерах
  - b. Проверьте настройки сетевого экрана на обоих компьютерах

# Com-технология VS веб-сервисы в локальной сети

## Недостатки

1. Рассмотренные выше сложности с взаимодействием разных сборок платформы 1С, требующие для своего решения сервисный механизм
2. Высокие затраты на подключение/инициализацию, поэтому при отсутствии кеширования неэффективно для частых мелких вызовов
3. Привязка к семейству операционных систем (платформе) Windows
4. Требуется закрытая сеть (LAN, VPN)

## Преимущества

1. Отсутствие необходимости изменять конфигурацию баз, к которым подключаемся
2. Широкая функциональность
3. Проще отладка
4. Не требуется разворачивать и поддерживать веб-сервер
5. Программно доступен интерактивный режим подключенной базы

## Захват клиентских лицензий

В большинстве случаев всем клиентским приложениям, частным случаем которого являются Automation-сервер и внешнее соединение, в одном сеансе ОС требуется всего лишь одна на всех лицензия. Однако если клиентскому приложению не удалось получить аппаратную лицензию (от локального HASP или сетевого HASP) и программную лицензию (платформы или базовой конфигурации), то предпринимается попытка получить отдельную лицензию с сервера 1С:Предприятия. Сервер 1С:Предприятия может выдать лицензию только, если в свойствах базы разрешена выдача лицензий сервером 1С:Предприятия. В таком случае каждое клиентское приложение и внешнее соединение может захватывать свою собственную лицензию.

## Заключение

Если вы хотите подробнее узнать об описанных особенностях, вы можете [бесплатно](#) скачать конфигурацию и демонстрационную базу продукта, для чего необходимо оформить предварительный заказ на Инфостарте на продукт [2iS:Интеграция](#). Несмотря на наличие в конфигурации защищенного модуля, 98% программного кода конфигурации открыто, то есть Вы можете изучить реализацию описанных приемов.

Полезные ссылки на its.1c.ru:

- [Особенности использования COM-объектов](#)
- [Получение значений перечислений через внешнее соединение или Automation-сервер](#)
- [Передача объектов 1С:Предприятия через COM](#)