

Понятие контекста программного кода

Контекст – это определенное окружение доступных свойств и методов в данном конкретном программном модуле, в данной конкретной строке. Т.е. это те процедуры и функции, те свойства и методы, которые доступны в данной конкретной точке.

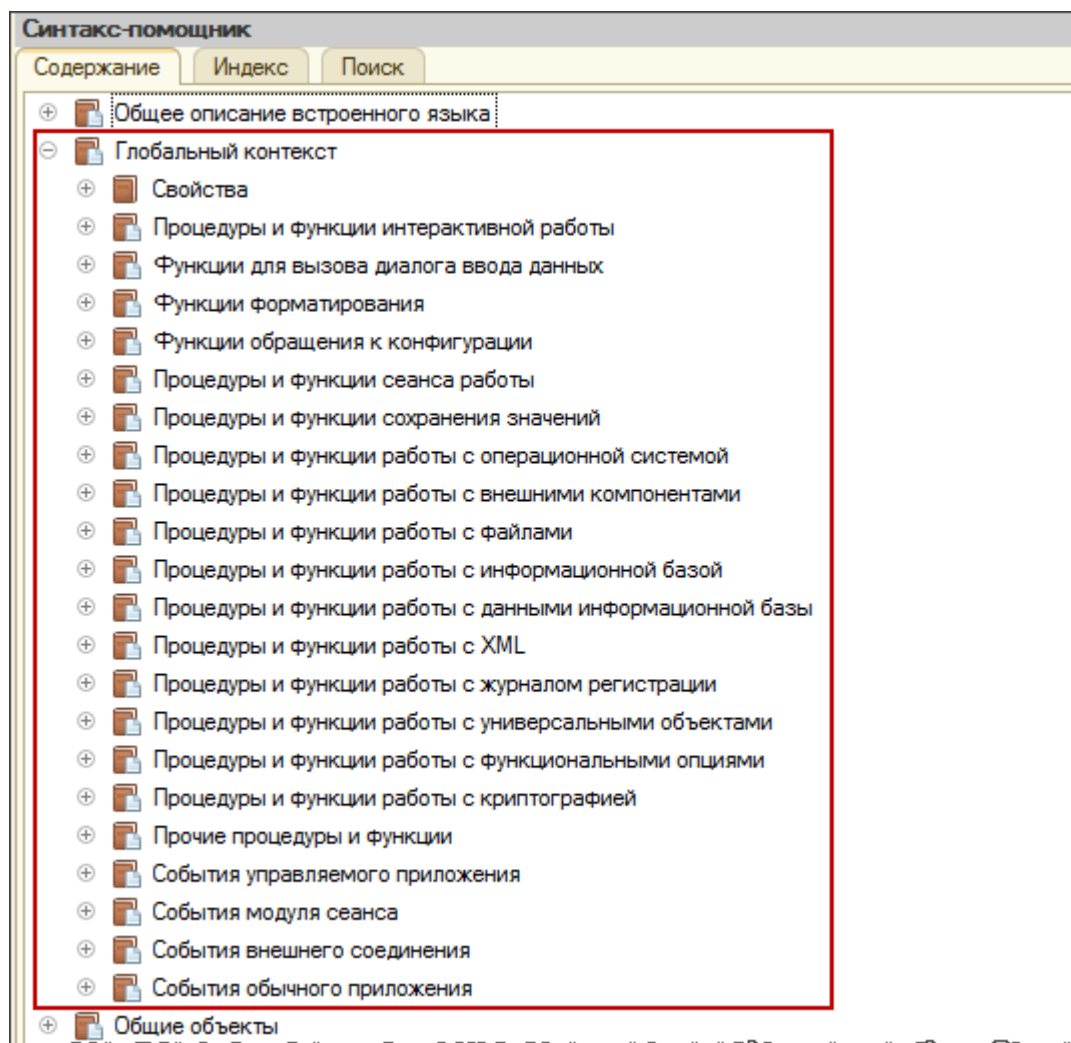
В другой точке контекст может измениться. Даже в одном модуле в разных процедурах может быть совершенно разный контекст.

Именно поэтому важно представлять контекст данной точки, где мы ведем разработку, чтобы понимать: какие свойства и методы мы можем использовать. Выделяют несколько видов контекста.

Глобальный контекст, который доступен в любой точке конфигурации (в принципе, нужно еще делить контекст на Сервер и на Клиент).

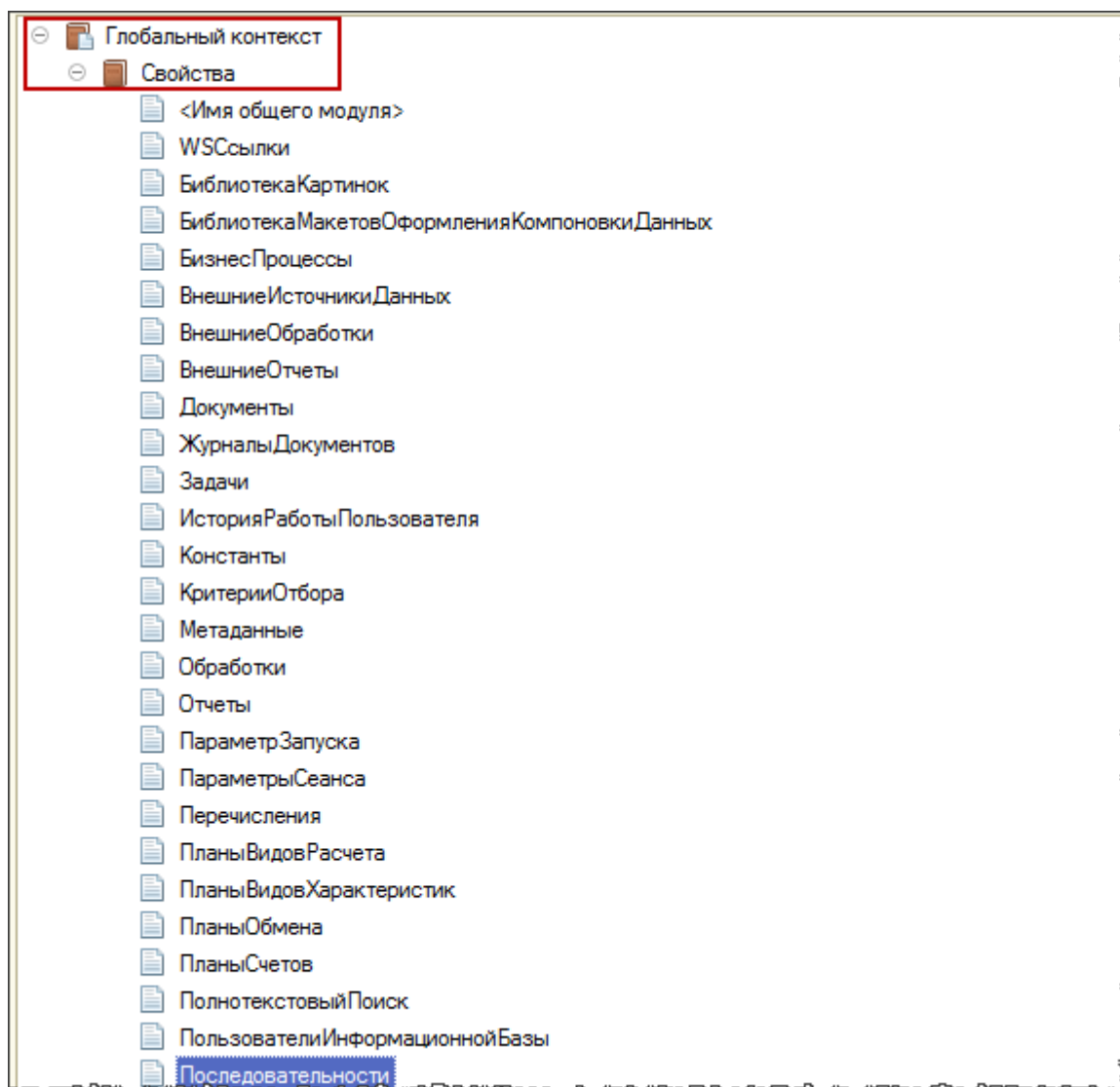
Глобальный контекст доступен во всех местах конфигурации. Глобальный контекст образуется прежде всего процедурами и функциями глобального контекста Платформы.

В синтакс-помощнике есть отдельная ветка «Глобальный контекст», в которой видно, что существуют различные функции, которые будут доступны при вызове.

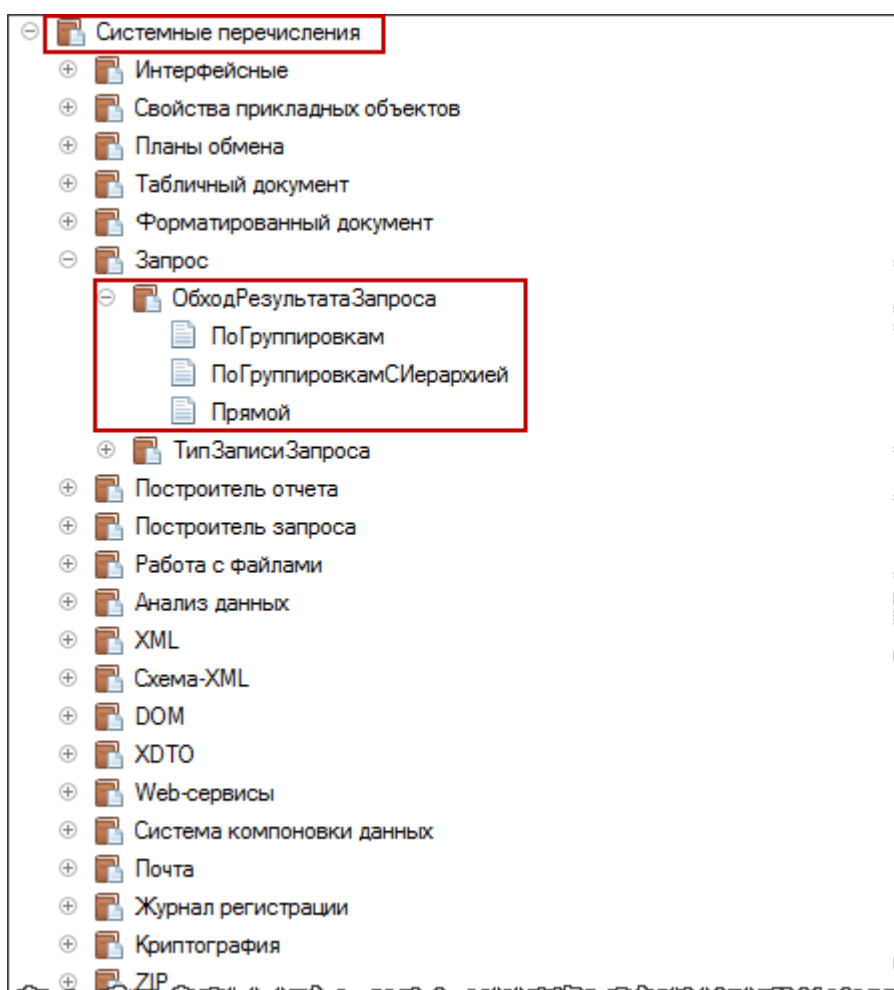


Конечно же, в глобальный контекст также входят и встроенные функции работы со строкой, числом, датой и т.д. Все это тоже образует глобальный контекст.

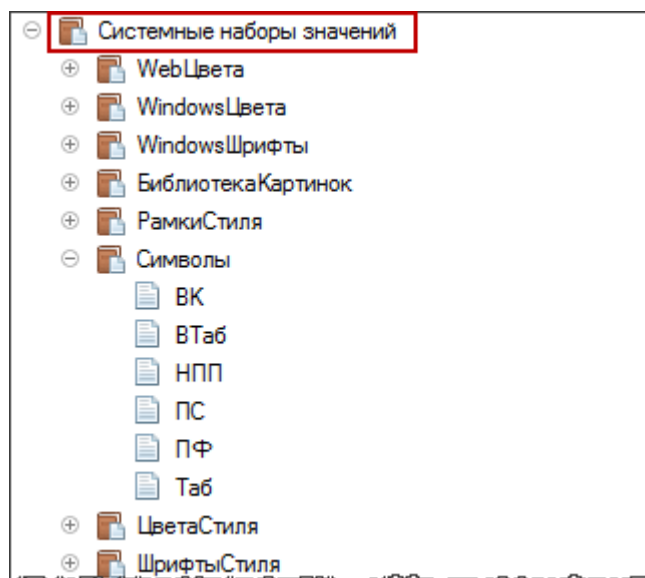
Обратите внимание, что существуют не только методы глобального контекста, но и свойства. К ним можно обращаться.



Также в глобальный контекст входят системные перечисления и системные значения. Например, перечисление *ОбходРезультатаЗапроса* имеет три фиксированных значения. Мы можем обращаться к этому перечислению в любой точке конфигурации.



Системные наборы имеют похожее назначение с перечислениями. Указывается набор значений и выбор одного из значений.



Кроме того, в глобальный контекст также входят экспортные методы общих модулей. Существуют модули для описания общих алгоритмов, и некоторые их методы можно делать экспортными, тогда они могут вызываться из других программных модулей.

Помимо Глобального контекста существует локальный контекст модуля. Это переменные, процедуры и функции конкретного модуля. Если спуститься еще на уровень ниже, существует локальный контекст процедуры.

В процедуре можно объявлять свои собственные переменные. В модуле, например, могут существовать две переменные с одинаковым названием: одна объявлена на уровне контекста модуля, а вторая на уровне самой процедуры. Это две разные переменные.

Контекст – это первое, на что нужно обращать внимание при написании программного кода. На контекст влияет то, где процедура скомпилирована. Контекст выполнения программного кода в платформе 8.3 может делиться на Клиентский и Серверный.

По сути это означает, что программный код выполняется либо на стороне клиентской машины, либо на стороне Сервера приложений. Всякий раз, когда мы пишем программный код на 8.3, нам нужно думать, где этот код должен выполняться: на стороне Клиента или на стороне Сервера.

На Клиенте существуют свои поддерживаемые типы данных, а на Сервере – свои. Существуют типы данных, которые поддерживаются и на Клиенте, и на Сервере (т.е. их можно спокойно использовать в любом программном коде).

Важно понимать, что программный код, который скомпилирован на Клиенте, является

независимым от программного кода, который скомпилирован на Сервере. Если речь идет о Клиенте, то в основном это какие-то легкие действия.

Например, показ какого-то диалогового сообщения, задание вопроса пользователю. На клиенте можно использовать лишь ограниченное количество типов данных.

Клиент – это обычно слабая по мощности машина, поэтому нагружать его не стоит. На Сервере же должны выполняться обработки различных данных.

Важно понимать, какое возможно взаимодействие между Клиентом и Сервером. Существует возможность из процедур клиентских обращаться к процедурам серверным.

Система, после обработки соответствующего серверного кода вернет управление назад на точку вызова. Следует обратить внимание, что обратный вызов невозможен, т.е. со стороны Сервера клиентский метод мы вызвать не можем ни при каких условиях.

Клиентские процедуры могут вызывать как другие клиентские процедуры, так и серверные процедуры. Таким образом, при разработке в Платформе 1С:Предприятие 8.3 важно понимать, где будет скомпилирована процедура: на Клиенте или на Сервере, поскольку от этого будет зависеть текущий контекст модуля.

Если в модуле могут присутствовать как те, так и другие процедуры (например, модуль управляемой формы), то при создании процедуры необходимо указывать, где она будет скомпилирована: на Сервере или на Клиенте.

Для этого есть определенные директивы, которые четко указывают, где будет использоваться данная процедура. Исходя из этого в процедуре будут доступны соответствующие типы данных. Ниже представлены примеры клиентской и серверной процедуры модуля формы.

```
Обработка ПроверкаКонтрагентов: Форма
  &НаКлиенте
  Процедура ВыполнитьКоманду(Команда)
    ВыполнитьНаСервере();
  КонецПроцедуры
  &НаСервереБезКонтекста
  Процедура ВыполнитьНаСервере()
    Запрос = Новый Запрос;
    Запрос.Текст =
      "ВЫБРАТЬ
      | Контрагенты.Ссылка
      | ИЗ
      | Справочник.Контрагенты КАК Контрагенты
      | ГДЕ
      | (НЕ Контрагенты.Ссылка В
      |   (ВЫБРАТЬ РАЗЛИЧНЫЕ
      |     КонтактныеЛицаКонтрагентов.Владелец
      |     ИЗ
      |     Справочник.КонтактныеЛицаКонтрагентов КАК КонтактныеЛицаКонтрагентов));";
    Результат = Запрос.Выполнить();
    ВыборкаДетальныеЗаписи = Результат.Выбрать();
    Пока ВыборкаДетальныеЗаписи.Следующий() Цикл
      Сообщить(ВыборкаДетальныеЗаписи.Ссылка);
    КонецЦикла;
  КонецПроцедуры
```

Когда мы описываем некую переменную, то мы также должны указывать (как для процедуры или функции место компиляции), где эта переменная будет находиться: на Клиенте или на Сервере. Т.е., если указать, что переменная будет доступна на Сервере, то ее можно будет использовать в серверных процедурах.

В целях оптимизации кода следует в большинстве случаев с Клиента обращаться к контексту *&НаСервереБезКонтекста*, а не к контексту *&НаСервере*. Дело в том, что при втором варианте на Сервер передаются еще все данные формы.

Обратите внимание, что для каждого объекта в синтакс-помощнике указывается доступность данного свойства или метода (сервер, тонкий клиент, толстый клиент, внешнее соединение).

Артемов Артем,

г. Москва

Дополнительные материалы

Все статьи проекта Курсы-по-1С.рф: <http://курсы-по-1с.рф/blog/articles/>

Курсы по программированию в 1С v.8

Базовый и Продвинутой курсы по Программированию на Платформе 1С 8
<http://www.Spec8.ru/>



Базовый курс по программированию в 1С v.8

Курс про **готовые приемы и решения**
90% задач по программированию в 1С



Продвинутой курс по программированию в 1С v.8

Больше, чем Вы можете себе представить
Детальнее требований на **1С:Специалист**

«Курс по подготовке к Аттестации по Платформе 1С 8.2 / 8.3»

<http://курсы-по-1с.рф/dev-attestation/>



Подготовка к Аттестации по Платформе 1С v.8

Аттестация по Платформе – **с первого раза**
Экономия 100 - 150 часов подготовки