

Инструменты разработки

Пакеты

Версия 8.0



Эта документация предоставляется с ограничениями на использование и защищена законами об интеллектуальной собственности. За исключением случаев, прямо разрешенных в вашем лицензионном соглашении или разрешенных законом, вы не можете использовать, копировать, воспроизводить, переводить, транслировать, изменять, лицензировать, передавать, распространять, демонстрировать, выполнять, публиковать или отображать любую часть в любой форме или посредством любые значения. Обратный инжиниринг, дизассемблирование или декомпиляция этой документации, если это не требуется по закону для взаимодействия, запрещены.

Информация, содержащаяся в данном документе, может быть изменена без предварительного уведомления и не может гарантировать отсутствие ошибок. Если вы обнаружите какие-либо ошибки, сообщите нам о них в письменной форме.

Содержание

Пакеты	4
Классификация пакетов	4
Структура пакета	4
Зависимости и иерархия пакетов	5
Добавление зависимостей пакета	6
Список зависимостей в метаданных	8
Иерархия пакетов приложения	8
Основные пакеты приложения	9
Пакет Custom	10
Создать пользовательский пакет	12
Шаг 1	12
Шаг 2	13
Шаг 3	14
Шаг 4	14
Шаг 5	15
Привязать данные к пакету	15
Исходный код	16
Алгоритм реализации примера	16

Пакеты

Основы

Любой продукт Creatio представляет собой определенный набор пакетов. С их помощью выполняются все конфигурационные изменения.

Классификация пакетов

Пакет Creatio — это совокупность конфигурационных элементов (схем, данных, SQL-скриптов, дополнительных библиотек), которые реализуют определенный блок функциональности. Физически пакет представляет собой каталог, содержащий определенный набор подкаталогов и файлов.

Любой продукт представляет собой конечный набор пакетов. Чтобы расширить или изменить функциональность продукта, нужно установить пакет, в котором реализованы все необходимые изменения.

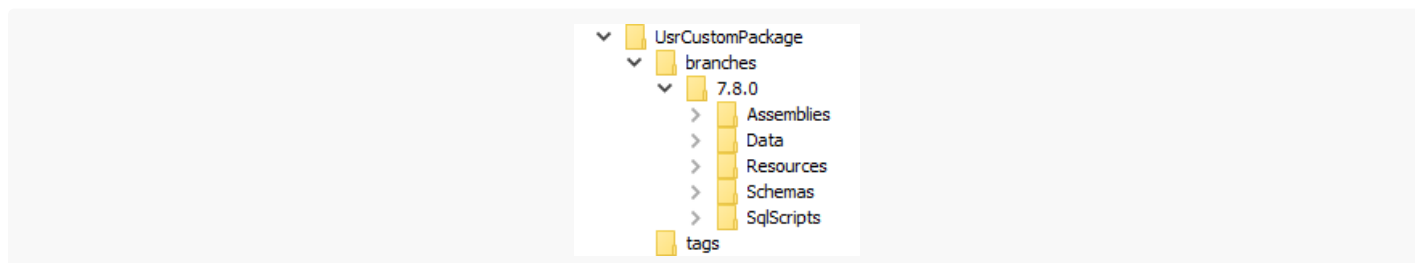
Пакеты Creatio условно можно разделить на два вида:

- **Предустановленные пакеты.** Поставляются вместе с системой и устанавливаются в рабочее пространство по умолчанию. К таким относятся пакеты с базовой функциональностью (например, `Base`, `NUI`), а также пакеты, созданные сторонними разработчиками. Такие пакеты устанавливаются из zip-архивов [как приложения marketplace](#) или с помощью [утилиты WorkspaceConsole](#).
- **Пользовательские пакеты** — пакеты, [создаваемые пользователями системы](#). Они могут быть привязаны к хранилищу SVN.

Конфигурационные элементы из предустановленных пакетов недоступны для изменения. Разработка дополнительной функциональности и модификация существующей выполняется исключительно в пользовательских пакетах.

Структура пакета

При фиксации пакета в систему контроля версий в хранилище пакета создается папка с именем пакета, а внутри нее — каталоги `branches` и `tags`.



В папке `branches` хранятся все версии данного пакета. Каждая версия хранится в отдельной вложенной папке, имя которой совпадает с номером версии пакета в системе, например, 7.8.0.

Важно. Структура, учитывающая версии пакета, осталась для совместимости с приложения версий ниже 7.9.

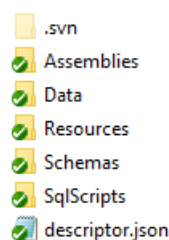
Каталог `tags` предназначен для хранения меток. Метки в системе управления версиями — это "снимок" проекта в определенный момент времени, статическая копия файлов, необходимая для фиксации определенного важного этапа разработки.

Рабочая копия пакета сохраняется локально в файловой системе. Путь для хранения пакетов задается в конфигурационном файле `ConnectionStrings.config` в атрибуте `connectionString` элемента `defPackagesWorkingCopyPath`

```
<add name="defPackagesWorkingCopyPath" connectionString="TEMP\APPLICATION\WORKSPACE\TerrasoftPac
```

По этому пути создается папка с именем пакета.

Структура папки пакета в файловой системе



В каталоге `Schemas` хранятся схемы пакета. Привязанные к пакету внешние сборки, данные и SQL-сценарии содержатся, соответственно, в каталогах `Assemblies`, `Data` и `SqlScripts`. Все текстовые ресурсы пакета, переведенные на разные языки, вынесены в отдельный каталог `Resources`.

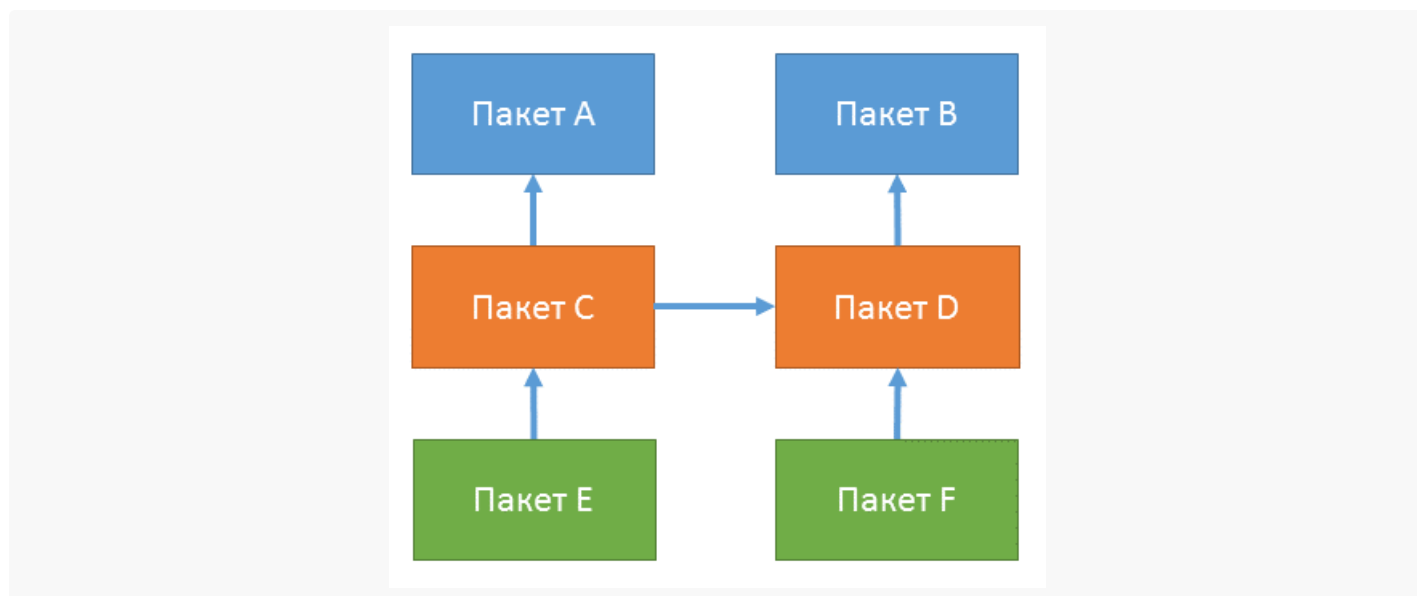
Важно. Начиная с версии 7.11.3 в структуру пакета добавлен каталог `Files`, содержащий файловый контент (см. ["Работа с файловым контентом"](#)).

В файле `descriptor.json` хранятся метаданные пакета в формате JSON — его идентификатор, наименование, версия, зависимости и т.д.

Зависимости и иерархия пакетов

Разработка приложения Creatio базируется на основных принципах проектирования программного обеспечения, в частности, [принципа отсутствия повторений \(DRY\)](#). В архитектуре Creatio этот принцип был применен к механизму пакетов и реализован с помощью зависимостей пакетов друг от друга. Каждый пакет содержит определенную функциональность приложения, которая не должна повторяться в других пакетах. Для того чтобы такую функциональность можно было использовать в любом другом пакете, необходимо пакет, содержащий эту функциональность, добавить в зависимости пакета, в котором она будет использована.

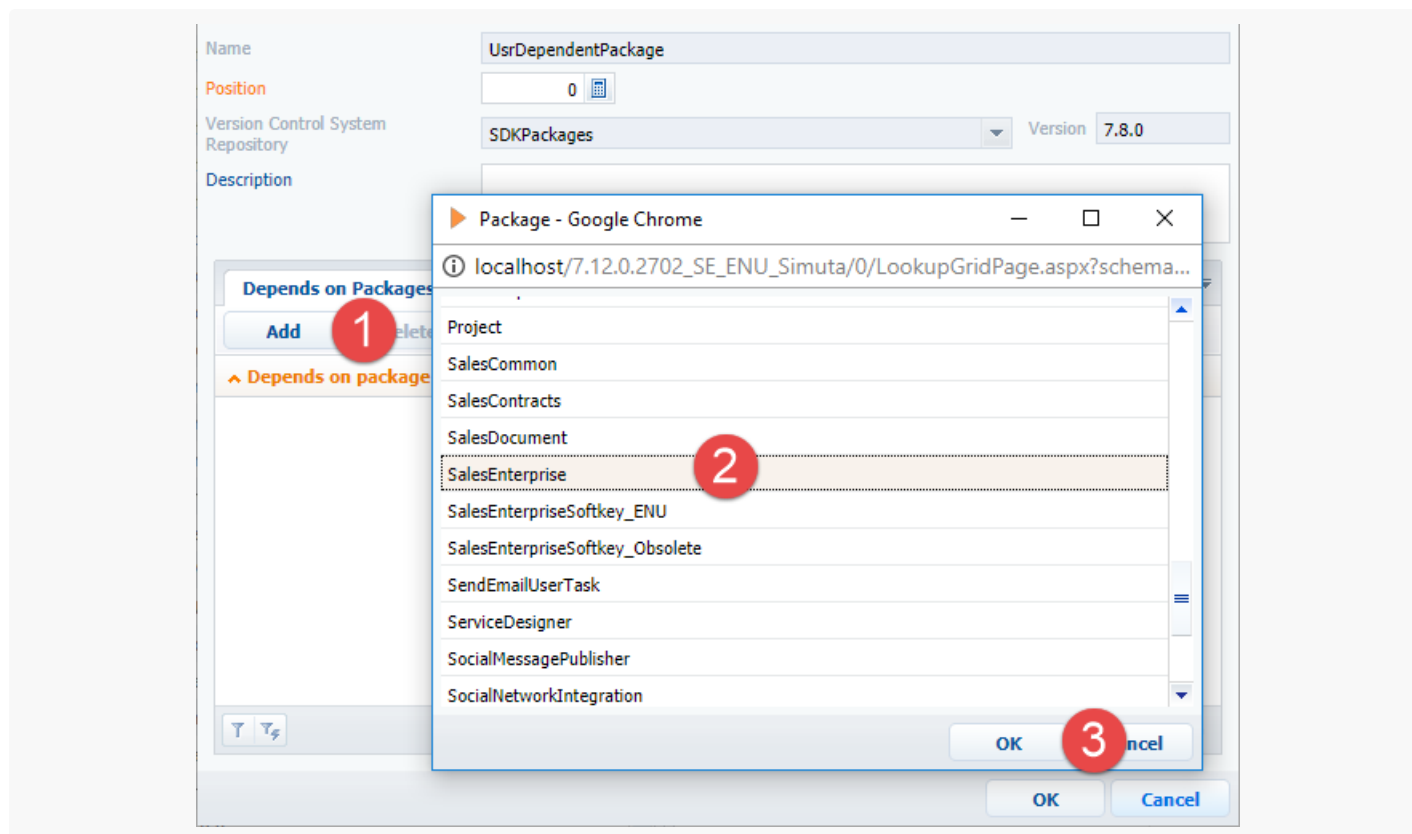
Пакет может иметь несколько зависимостей. Например, в пакете С установлены зависимости от пакетов А и D. Таким образом, вся функциональность пакетов А и D доступна в пакете С.



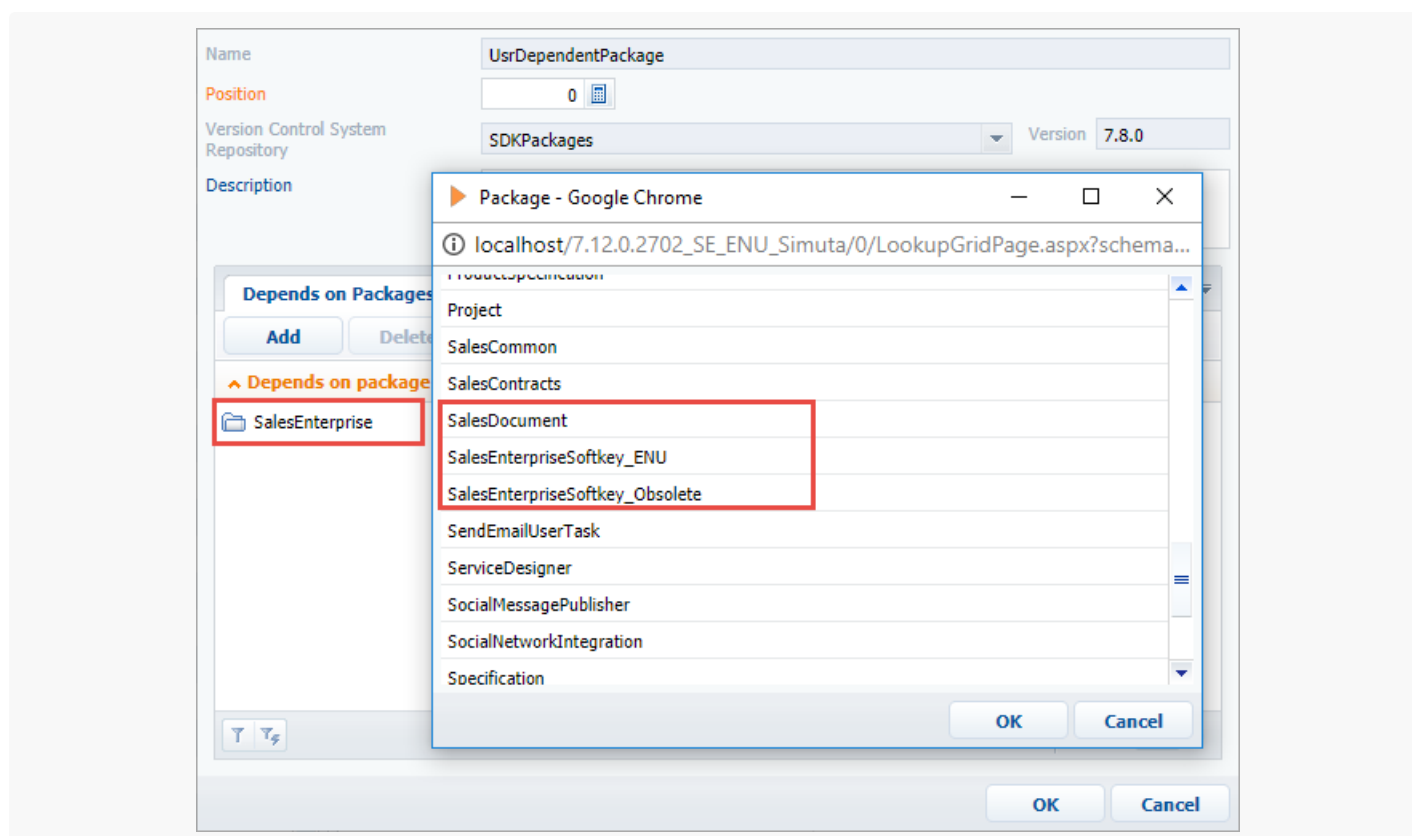
Зависимости пакетов формируют иерархические цепочки. Это означает, что в пакете доступна не только функциональность пакета, добавленного в зависимости текущего, но и функциональность всех пакетов, от которых зависит пакет, добавленный в зависимости текущего. Ближайшей аналогией иерархии пакетов является иерархия наследования классов в объектно-ориентированном программировании. Так, например, в пакете Е доступна функциональность не только пакета С, от которого он зависит, но и функциональность пакетов А, В и D. А в пакете F доступна функциональность пакетов В и D.

Добавление зависимостей пакета

Зависимости можно добавить только в пользовательский пакет и только после его создания. Для этого необходимо в карточке пакета на детали [*Зависит от пакетов*] ([*Depends on packages*]) нажать на кнопку [*Добавить*] ([*Add*]) (1). В появившемся диалоге справочника пакетов необходимо выбрать нужный пакет (2) и нажать на кнопку [*OK*] (3).



После этого выбранный пакет будет отображен в списке зависимостей текущего пакета, а при добавлении новой зависимости он будет скрыт из справочника пакетов.



После создания пакета он автоматически добавляется в зависимости предустановленного пакета [*Custom*].

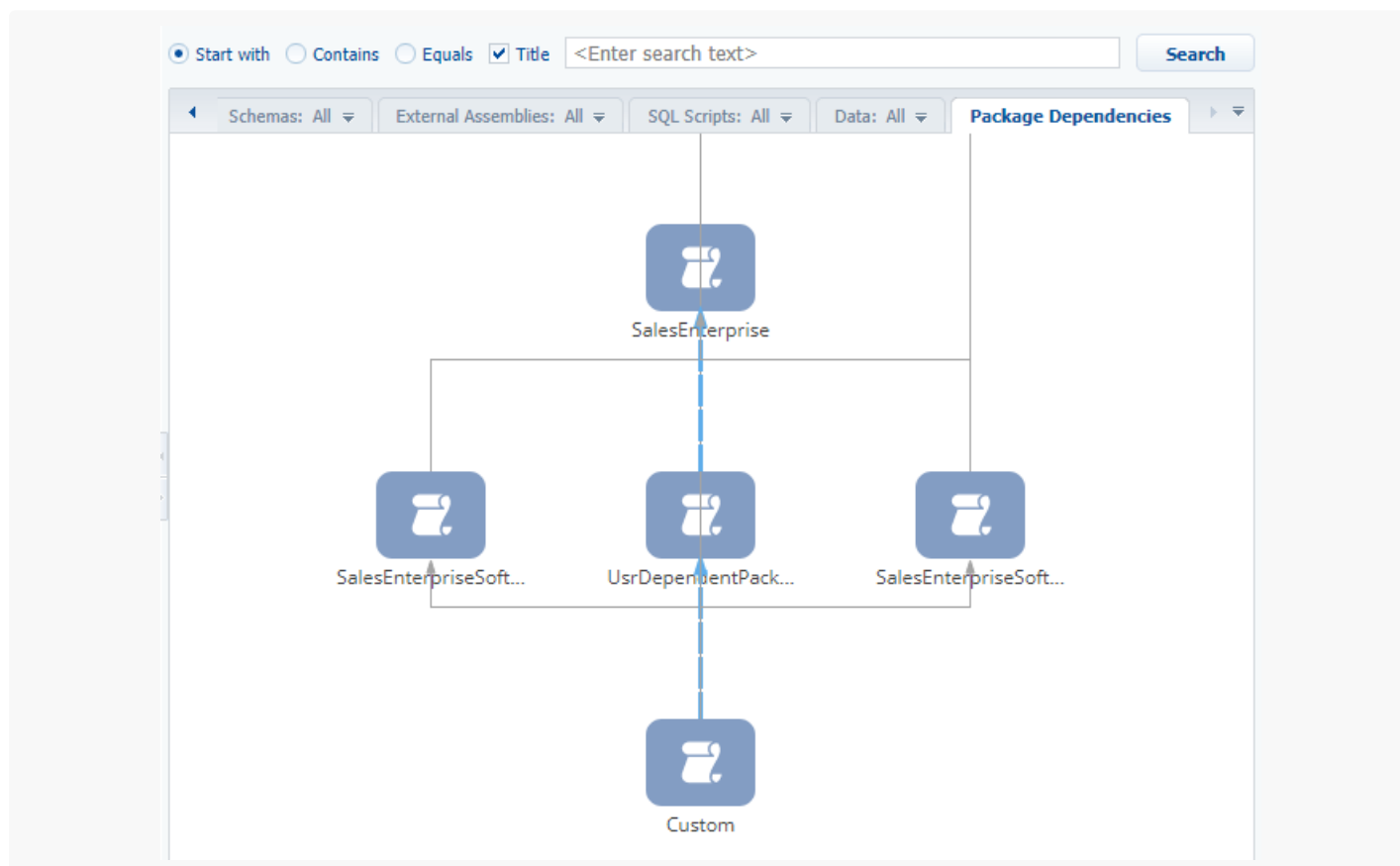
Список зависимостей в метаданных

Список зависимостей пакета хранится в его метаданных в свойстве `DependsOn` объекта, определенного в файле `descriptor.json`. Свойство `DependsOn` является массивом объектов, в которых указывается имя пакета, его версия и уникальный идентификатор, по которому можно определить пакет в базе данных приложения. Файл `descriptor.json` создается приложением для каждой версии пакета. Пример файла `descriptor.json`:

```
{
  "Descriptor": {
    "UId": "51b3ed42-678c-4da3-bd16-8596b95c0546",
    "PackageVersion": "7.8.0",
    "Name": "UsrDependentPackage",
    "ModifiedOnUtc": "\\Date(1522653150000)\\",
    "Maintainer": "Customer",
    "DependsOn": [
      {
        "UId": "e14dcfb1-e53c-4439-a876-af7f97083ed9",
        "PackageVersion": "7.8.0",
        "Name": "SalesEnterprise"
      }
    ]
  }
}
```

Иерархия пакетов приложения

Иерархию и зависимости пакетов приложения удобно исследовать, используя диаграмму зависимостей пакетов. Эта диаграмма расположена на вкладке [*Зависимости пакетов*] ([*Package Dependencies*]) раздела [*Конфигурация*].



Если кликнуть по узловому элементу диаграммы с именем пакета, то с помощью анимированных стрелок отобразятся связи с пакетами, которые зависят от выбранного, и от которых зависит выбранный пакет. Например, для продукта *SalesEnterprise*, пакет [*UsrDependentPackage*] зависит только от пакета [*SalesEnterprise*] и всех его зависимостей. Также от него зависит пакет [*Custom*].

Основные пакеты приложения

К основным пакетам приложения можно отнести пакеты, которые обязательно присутствуют во всех продуктах.

Основные пакеты приложения

Название пакета	Содержимое
Base	Базовые схемы основных объектов, разделов системы и связанных с ними схем объектов, страниц, процессов и др.
Platform	Модули и страницы мастера разделов, дизайнеров реестра и итогов и т. п.
Managers	Клиентские модули менеджеров схем.
NUI	Функциональность, связанная с пользовательским интерфейсом системы.
UIv2	Функциональность, связанная с пользовательским интерфейсом системы.
DesignerTools	Схемы дизайнеров и их элементов.
ProcessDesigner	Схемы дизайнера процессов.

Пакет [Custom]

В процессе своей работы мастер разделов или мастер деталей создает различные схемы, которые необходимо сохранить в пакет. Однако в только что установленном приложении доступных для изменения пользовательских пакетов нет. А в предустановленные пакеты изменения внести нельзя.

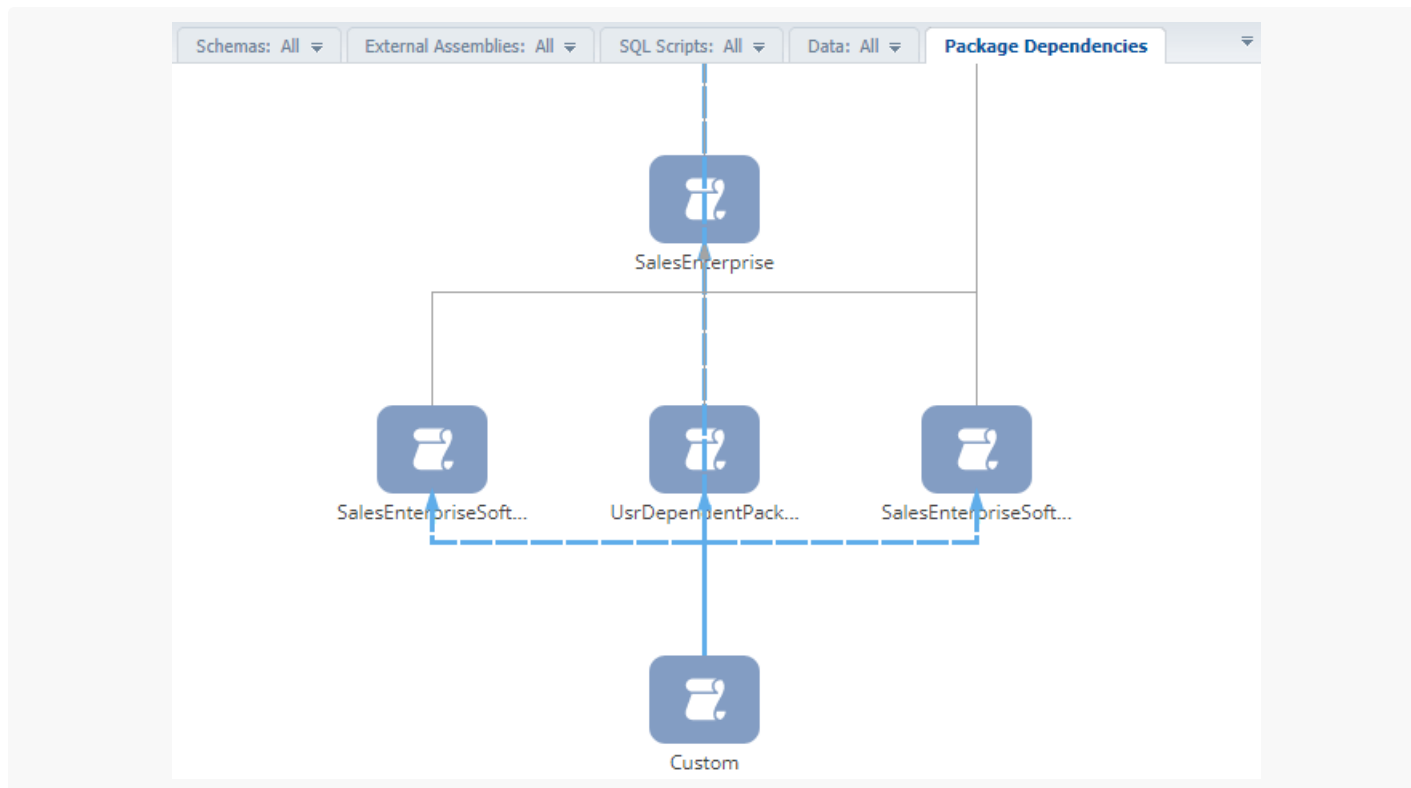
Для разрешения подобных конфликтов предназначен специальный предустановленный пакет [Custom]. Он позволяет добавлять схемы как вручную, так и с помощью мастеров.

Особенности пакета [Custom]

Как и все предустановленные пакеты, пакет [Custom] нельзя добавить в хранилище учета версий (SVN). Поэтому его схемы можно перенести в другое приложение только при помощи [экспорта и импорта](#).

В отличие от других предустановленных пакетов, пакет [Custom] нельзя выгрузить в файловую систему при помощи утилиты [WorkspaceConsole](#).

В пакете [Custom] установлены зависимости от всех предустановленных пакетов приложения. При создании или установке пользовательского пакета в пакет [Custom] автоматически добавляется зависимость от пользовательского пакета. Таким образом пакет [Custom] всегда должен быть последним в иерархии пакетов. В зависимости пользовательских пакетов пакет [Custom] добавить нельзя.



На заметку. Технически пользовательский пакет можно сделать последним в иерархии при помощи системной настройки [*Идентификатор пользовательского пакета*] (`CustomPackageUid`). Однако добавить в его зависимости предустановленные пакеты (в том числе и пакет [*Custom*]) можно только в том случае, если разработка ведется без использования SVN.

Устанавливать вместо пакета [*Custom*] любой другой пакет в качестве последнего в иерархии крайне не рекомендуется!

Пакет [*Custom*] и Мастер разделов

В процессе своей работы мастер разделов или мастер деталей не только создает различные схемы, но и привязывает данные к текущему пакету. Однако, если текущим пакетом является пакет [*Custom*], то перенести привязанные данные в другой пользовательский пакет практически невозможно. Поэтому рекомендуется в качестве текущего пакета использовать любой пользовательский пакет, но не [*Custom*].

Для того чтобы поменять текущий пакет, необходимо использовать системную настройку [*Текущий пакет*] (`CurrentPackageId`).

Рекомендации

Пакет [*Custom*] рекомендуется использовать в следующих случаях:

- Когда не предполагается перенос изменений в другое приложение.
- Если изменения выполняются при помощи мастеров или вручную, при этом объем изменений небольшой.

- Если нет необходимости использовать SVN.

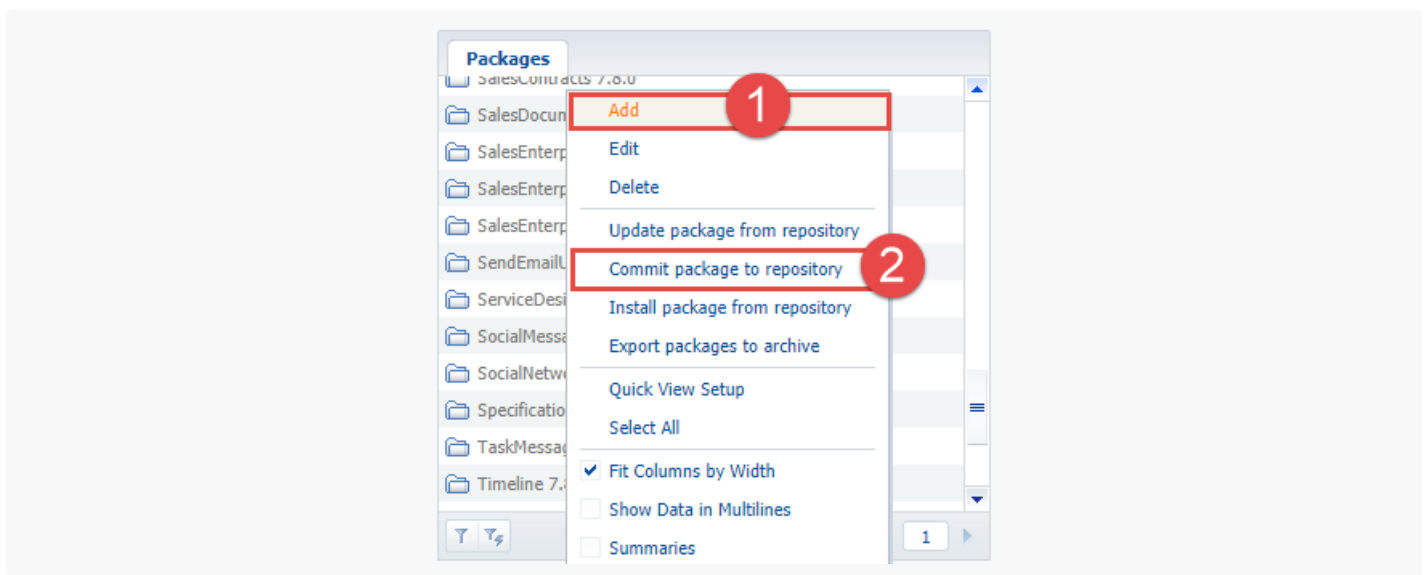
При необходимости разработать значительный объем новой функциональности более целесообразным будет создание нового пользовательского пакета с использованием SVN.

Создать пользовательский пакет

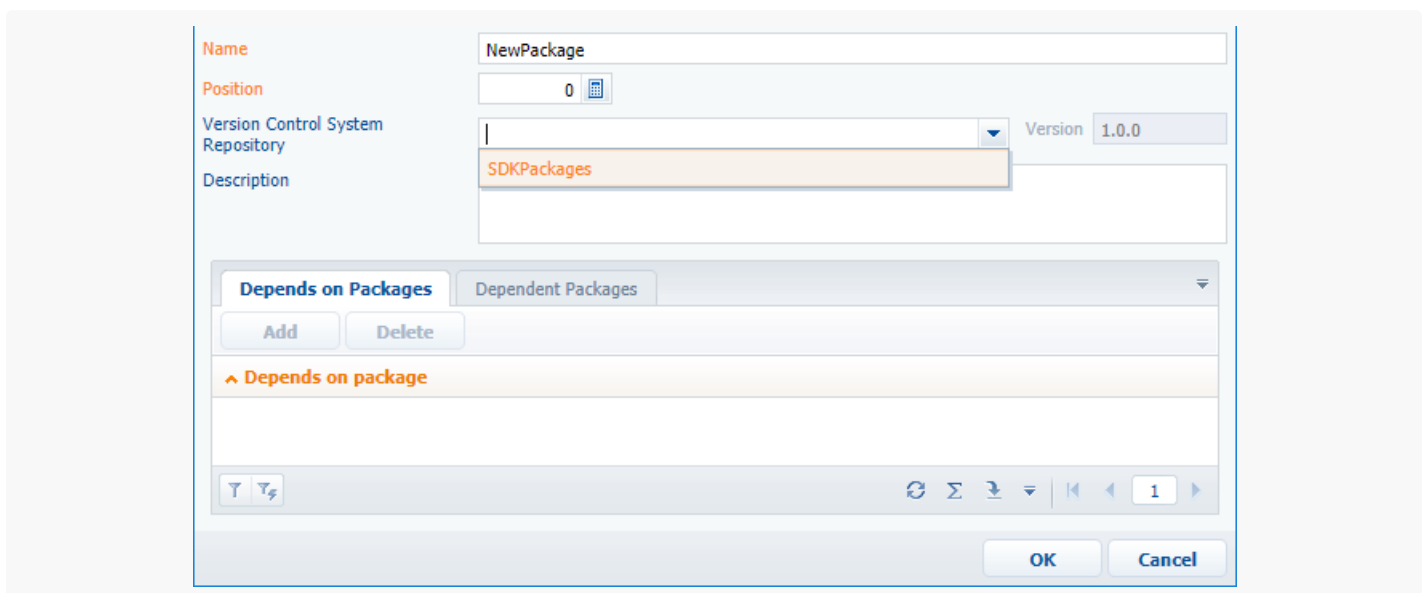
 Легкий

Шаг 1

Чтобы создать новый пользовательский пакет в контекстном меню на вкладке [*Пакеты*] ([*Packages*]) раздела [*Конфигурация*] ([*Configuration*]) выберите действие [*Добавить*] ([*Add*]) (1).



В результате откроется карточка пакета.



Шаг 2

Заполните поля карточки пакета.

Основные поля карточки пакета:

- [*Название*] ([*Name*]) — название пакета. Обязательное для заполнения поле. Не может совпадать с названием уже существующих пакетов.
- [*Позиция*] ([*Position*]) — позиция пакета в [иерархии](#). Обязательное для заполнения поле. По умолчанию имеет значение 0.
- [*Хранилище системы контроля версий*] ([*Version Control System Repository*]) — название хранилища системы контроля версий, в котором будут фиксироваться изменения пакета. Список доступных хранилищ формируется из списка хранилищ системы контроля версий. Хранилища, которые находятся в списке хранилищ конфигурации, но не помечены как активные, не попадут в выпадающий список доступных хранилищ. Поле является обязательным для заполнения.

Важно. Поле [*Хранилище системы контроля версий*] заполняется при создании нового пакета и в дальнейшем недоступно для редактирования. Если система контроля версий не используется, то это поле не отображается.

- [*Версия*] — версия пакета. Обязательное для заполнения поле. Версия пакета может содержать цифры, символы латинского алфавита и знаки "." и "_". Добавляемое значение должно начинаться с цифры или буквы. Все элементы пакета имеют ту же версию, что и сам пакет. Версия пакета не обязательно должна совпадать с версией приложения.
- [*Описание*] — описание пакета, например, расширенная информация о функциональности, которая реализуется в пакете. Не обязательное поле.

На заметку. При создании нового пакета сразу же указать его зависимости нельзя. Добавление зависимости выполняется при редактировании уже созданного пакета.

Содержимое основных полей карточки пакета будет сохранено в его метаданных.

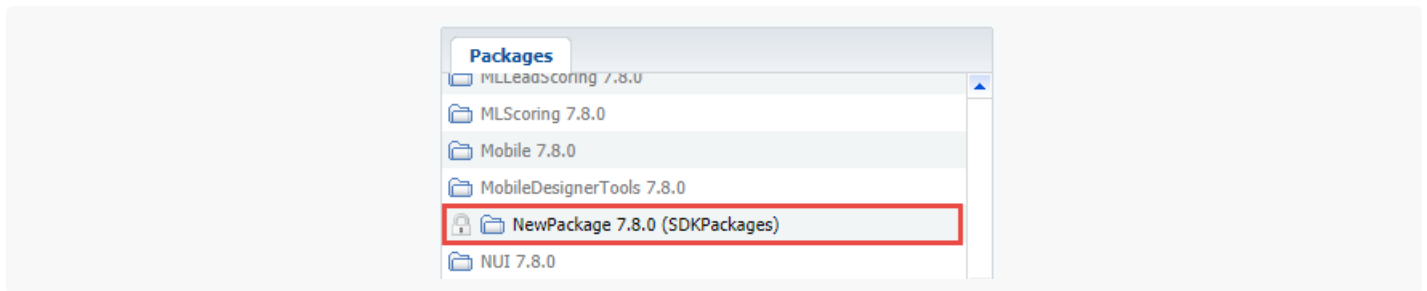
Метаданные основных полей карточки пакета

```
{
  "Descriptor": {
    "Uid": "1c1443d7-87df-4b48-bfb8-cc647755c4c1",
    "PackageVersion": "7.8.0",
    "Name": "NewPackage",
    "ModifiedOnUtc": "\/Date(1522657977000)\/",
    "Maintainer": "Customer",
    "DependsOn": []
  }
}
```

Кроме этих свойств метаданные пакета содержат информацию о зависимостях (свойство `DependsOn`) и информацию о разработчике (`Maintainer`). Значение свойства `Maintainer` устанавливается с помощью системной настройки [*Издатель*].

Шаг 3

После заполнения всех полей карточки нажмите на кнопку [*OK*] и пакет будет создан и отобразится на вкладке [*Пакеты*] ([*Packages*]).

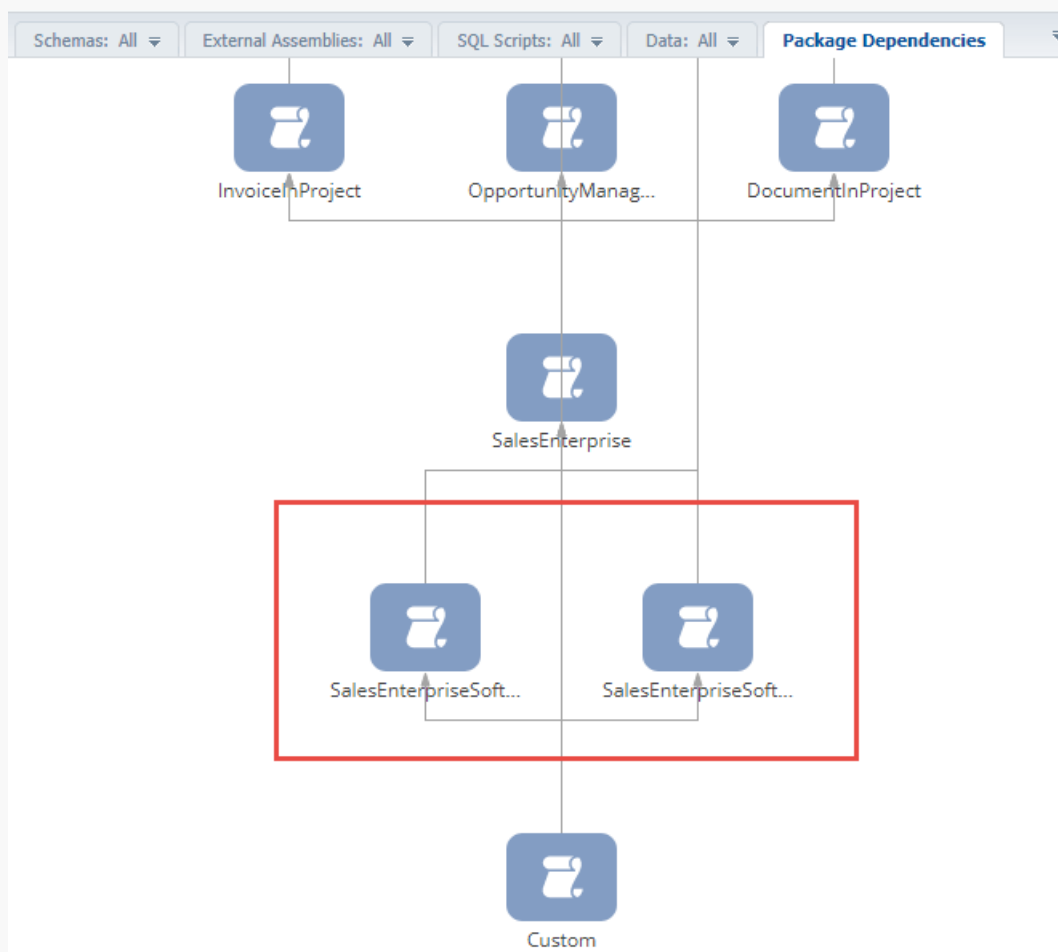


Шаг 4

Чтобы созданный пакет имел всю функциональность, которая заложена в систему, определите для него зависимости.

При этом достаточно указать самый последний пакет в иерархии предустановленных пакетов. Чтобы определить какой из пакетов в иерархии предустановленных пакетов является последним, перейдите на вкладку [*Зависимости пакетов*] раздела [*Конфигурация*]. Далее найдите предустановленные пакеты, которые находятся выше пакета [*Custom*]. Например, на рисунке последними в иерархии предустановленных пакетов будут пакеты [*SalesEnterpriseSoftkey_ENU*] и [*SalesEnterpriseSoftkey_Obsolete*].

На заметку. Пакет [*Custom*] добавить в зависимости нового пакета нельзя.



Шаг 5

В пакете [*Custom*] должны быть установлены зависимости от всех пакетов приложения. Поэтому необходимо удостовериться в том, что в нем установлена зависимость от созданного пакета.

Привязать данные к пакету



Сложный

При поставке пользователям пакетов часто возникает потребность предоставлять вместе с разработанной функциональностью также и некоторые данные. Это может быть, например, наполнение справочников, новые системные настройки, демонстрационные записи раздела и т. п.

Привязать необходимые данные к пакету, содержащему разработанную функциональность, можно на вкладке [*Данные*] раздела [*Конфигурация*].

Пример. Для пользовательского раздела [*Книги*] ([*Books*]) необходимо привязать две демонстрационные записи и связанные с ними записи других разделов.

Важно. При создании раздела с помощью мастера к пакету автоматически привязываются данные,

необходимые для регистрации и корректной работы раздела.

☒ Start with
 ☐ Contains
 ☐ Equals
 ☒ Title

Schemas: All
 External Assemblies: All
 SQL Scripts: All
 Data: All
 Packag

Add Edit Delete

Name	Package	Schema
SysDetail_DetailManager_b78e...	sdkBookExample	SysDetail
SysImage_f8d3f0121ed2433a9...	sdkBookExample	SysImage
SysModule_SectionManager_8...	sdkBookExample	SysModule
SysModuleEdit_SysModuleEdit...	sdkBookExample	SysModuleEdit
SysModuleEntity_SysModuleEn...	sdkBookExample	SysModuleEntity
SysModuleInWorkplace_Sectio...	sdkBookExample	SysModuleInWorkplace

1

Исходный код

Пакет с реализацией примера можно скачать по [ссылке](#).

Алгоритм реализации примера

1. Создать новый раздел [Книги]

Важно. Функциональность нового раздела следует создавать в [отдельном пакете разработки](#). Чтобы мастер раздела создавал схемы в пакете разработки, необходимо выбрать этот пакет в колонке [*Значение по умолчанию*] ([*Default value*]) системной настройки [*Текущий пакет*] ([*Current package*]). После завершения работы мастера в качестве текущего можно установить пакет [*Custom*].

Для создания нового раздела [Книги] воспользуйтесь [мастером разделов](#).

Свойства раздела [Книги] ([*Books*])

Books: General section properties

SAVE

CANCEL



SECTION

PAGE

BUSINESS RULES

CASES

BUSINESS PROCESSES



Select basic properties for section:

Title* Books

Code* UsrBook

Menu icon



Page settings:



One page for all records



Multiple pages

Свойства страницы редактирования записей

Books: Page

SAVE

CANCEL



SECTION

PAGE

BUSINESS RULES

CASES

BUSINESS PROCESSES



Page elements



Books



New column

☒ Boolean

Date

0.5 Decimal

123 Integer

Lookup

String

ISBN

Author *

Publisher *

0.5 Price

Name *

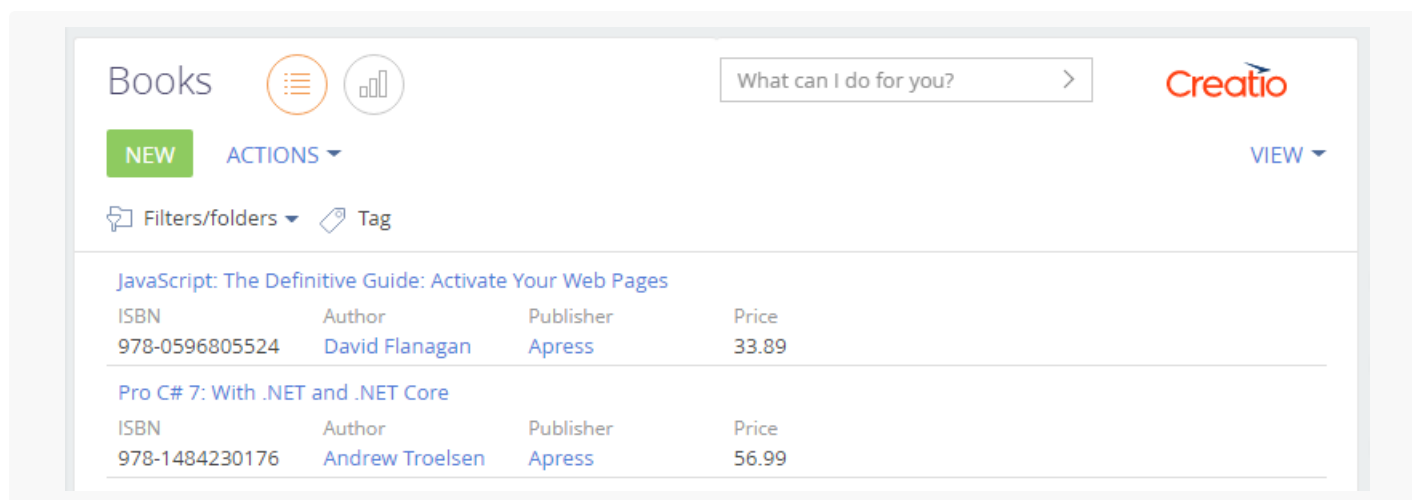
Description

Свойства колонок страницы редактирования записей раздела

Заголовок	Название (Код в БД)	Тип данных
Название (Name)	UserName	Строка (String).
Описание (Description)	UsrDescription	Строка (String). Многострочный текст (Multiline text).
ISBN	UsrISBN	Строка (String).
Автор (Author)	UsrAuthor	Справочник [<i>Контакт</i>] ([<i>Contact</i>]). Значение колонки будет привязано к одной из записей раздела [<i>Контакты</i>].
Издатель (Publisher)	UsrPublisher	Справочник [<i>Контрагент</i>] ([<i>Account</i>]). Значение колонки будет привязано к одной из записей раздела [<i>Контрагенты</i>].
Стоимость (Price)	UsrPrice	Дробное число (Decimal).

2. Добавить в раздел необходимые записи

Добавьте в раздел две демонстрационные записи. При необходимости также создайте записи в связанных разделах [*Контакты*] и [*Контрагенты*].



3. Привязать к пакету данные о контактах

Поскольку записи раздела [*Книги*] связаны с записями раздела [*Контакты*] по колонке UsrAuthor, то сначала необходимо привязать к пакету сведения об авторах. Для этого выполните действие [*Добавить*] ([*Add*]) на вкладке [*Данные*] ([*Data*]) раздела [*Конфигурация*] и установите следующие свойства страницы привязки данных:

1. [*Название*] ([*Name*]) — "ContactsInBooks".
2. [*Объект*] ([*Object*]) — "Контакт" ("Contact").
3. [*Тип установки*] ([*Installation type*]) — "Установка" ("Installation").
4. [*Колонки*] ([*Columns*]) — выберите только заполненные колонки. Колонка [*Id*] должна быть выбрана обязательно.
5. Фильтр данных — отфильтруйте требуемые данные, например, по имени контакта.

Properties | Bound Data

Name: ContactsInBooks

Object: Contact

Installation type: Installation

Columns: Add, Edit, Delete

Name	Forced up...	Key
1 Data entry compliance	<input type="checkbox"/>	<input type="checkbox"/>
Id	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Aa Full name	<input type="checkbox"/>	<input type="checkbox"/>
Owner	<input type="checkbox"/>	<input type="checkbox"/>

Filter: ☒ OR ☒ Full name that is equal to Andrew Troelsen ☒ Full name that is equal to David Flanagan

Display data | Records total: 2

Full name	Owner	Data entry compliance
Andrew Troelsen	Supervisor	10
David Flanagan	Supervisor	10

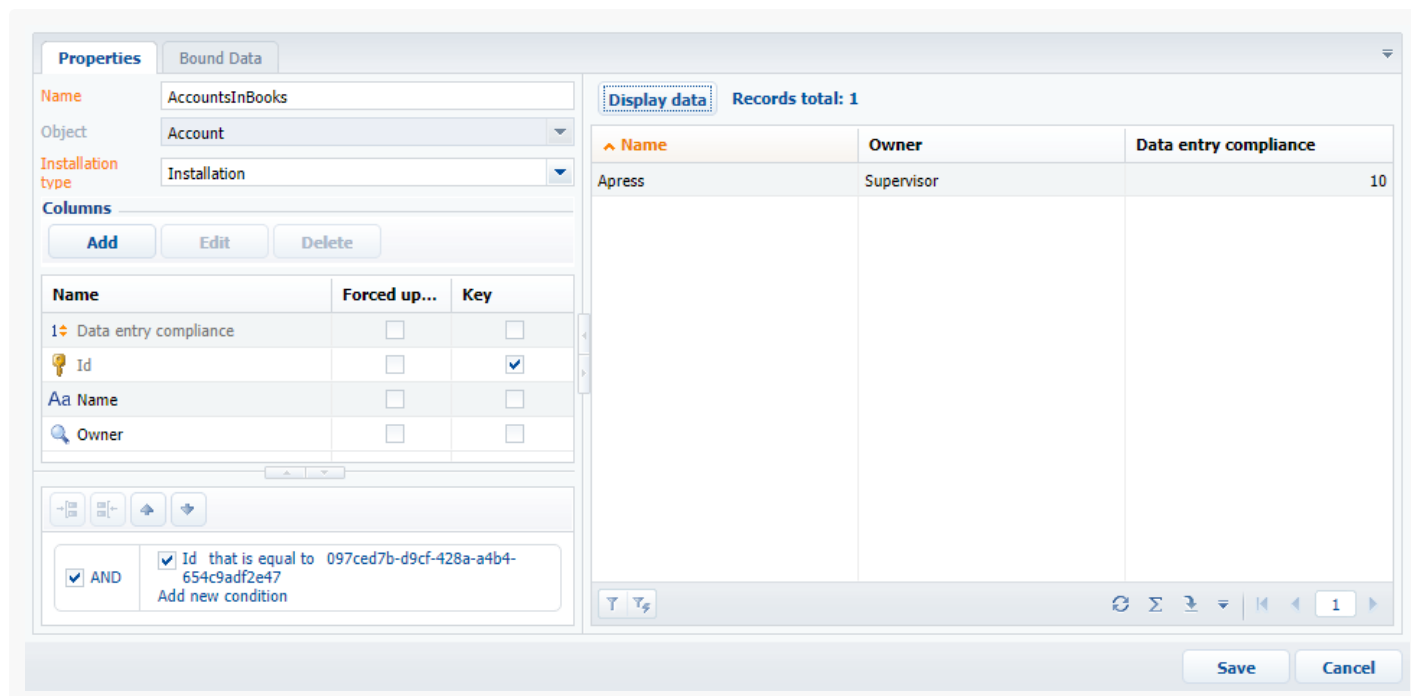
Save Cancel

На заметку. Рекомендуется использовать фильтрацию по колонке **Id** (см. следующий шаг), т. к. полное имя контакта может быть изменено.

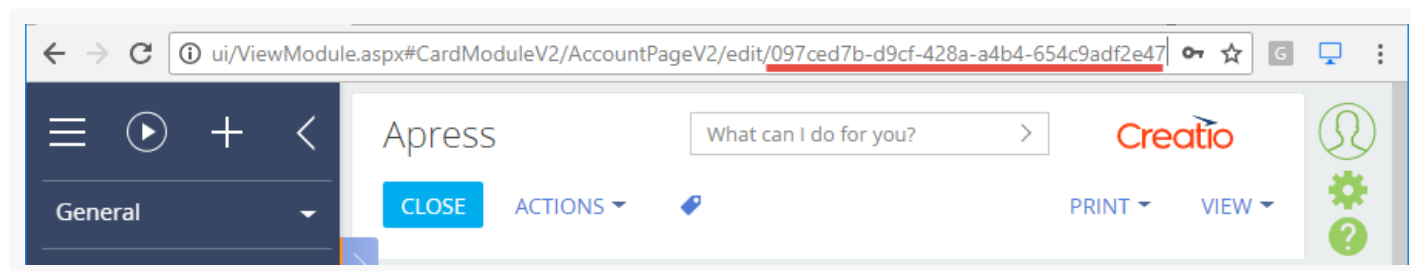
4. Привязать к пакету данные о контрагентах

Выполните действие [*Добавить*] ([*Add*]) на вкладке [*Данные*] ([*Data*]) раздела [*Конфигурация*] и установите следующие свойства страницы привязки данных:

1. [*Название*] ([*Name*]) — "AccountsInBooks".
2. [*Объект*] ([*Object*]) — "Контрагент" ("Account").
3. [*Тип установки*] ([*Installation type*]) — "Установка" ("Installation").
4. [*Колонки*] ([*Columns*]) — выберите только заполненные колонки. Колонка [*Id*] должна быть выбрана обязательно.
5. Фильтр данных — отфильтруйте требуемые данные, например, по идентификатору контрагента. Узнать идентификатор можно из адресной строки браузера, открыв страницу редактирования нужной записи.



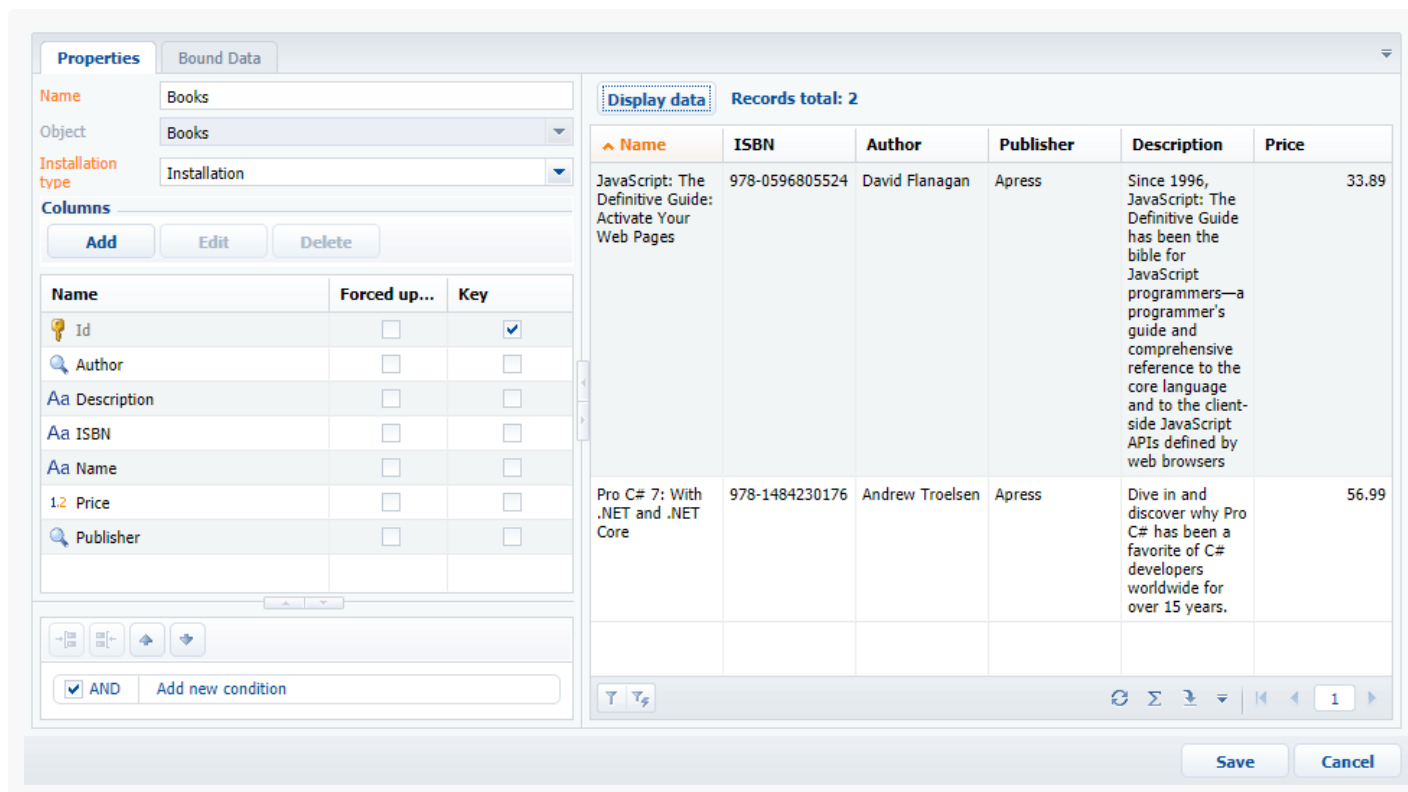
Определение идентификатора контрагента



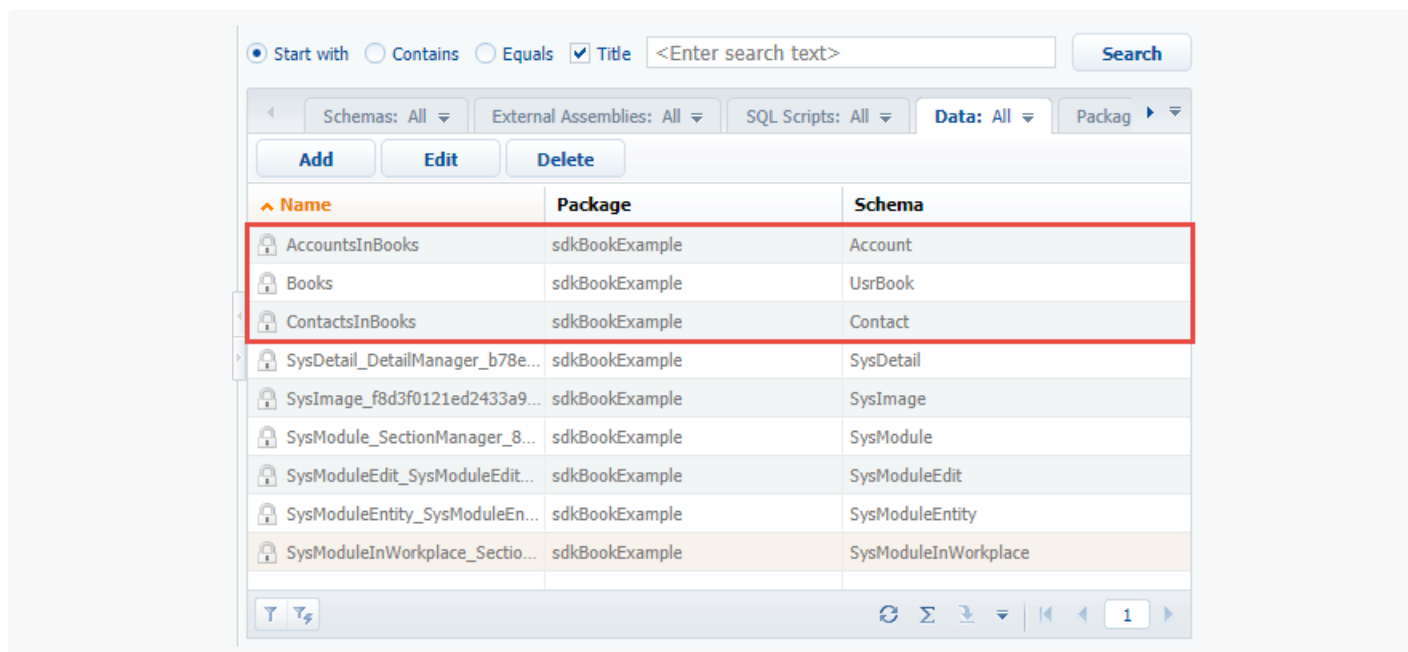
5. Привязать к пакету данные пользовательского раздела

Выполните действие [*Добавить*] ([*Add*]) на вкладке [*Данные*] ([*Data*]) раздела [*Конфигурация*] и установите следующие свойства страницы привязки данных:

1. [*Название*] ([*Name*]) — "Books".
2. [*Объект*] ([*Object*]) — "Книги" ("Books").
3. [*Тип установки*] ([*Installation type*]) — "Установка" ("Installation").
4. [*Колонки*] ([*Columns*]) — выберите только заполненные колонки. Колонка [*Id*] должна быть выбрана обязательно.
5. Фильтр данных — отфильтруйте требуемые данные. Если раздел содержит только две записи, то фильтр можно не использовать.



В результате выполнения примера к пакету будут привязаны три дополнительных набора данных для трех разделов.



Пакет можно выгрузить в архив, используя [функциональность экспорта](#). После установки пакета в другое приложение все привязанные записи отобразятся в соответствующих разделах.