

Пакеты

Общие принципы работы с пакетами

Версия 7.17



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

Общие принципы работы с пакетами	4
Классификация пакетов	4
Структура пакета	6
Зависимости и иерархия пакетов	7
Привязка данных к пакету	13
Создать пользовательский пакет	13
1. Создать пакет	13
2. Заполнить свойства пакета	14
3. Определить зависимости пакета	15
4. Проверить зависимости пакета Custom	16
Привязать данные к пакету	16
1. Создать раздел	16
2. Добавить в раздел демонстрационные записи	17
3. Привязать к пакету данные	18
4. Проверить привязки данных	21

Общие принципы работы с пакетами


Основы

Любой продукт Creatio представляет собой определенный набор пакетов.



Пакет Creatio — это совокупность [конфигурационных элементов](#), которые реализуют блок функциональности. Физически пакет представляет собой папку, содержащую определенный набор вложенных папок и файлов.

Классификация пакетов

Типы пакетов:

-  — предустановленные пакеты. Являются частью приложения и по умолчанию устанавливаются в рабочее пространство. Недоступны для изменения.

Виды предустановленных пакетов:

- Пакеты с базовой функциональностью (например, [Base], [NUI]).
- Пакеты сторонних разработчиков.
Устанавливаются из *.zip-архивов с помощью [Creatio IDE](#) или с помощью [утилиты WorkspaceConsole](#).
-  — пользовательские пакеты. Созданы другими пользователями системы и заблокированы для изменения в системе контроля версий. Недоступны для изменения.
-  — пользовательские пакеты. Созданы текущим пользователем либо загружены из системы контроля версий. Доступны для изменения.

Для расширения или изменения функциональности необходимо установить пакет с требуемой функциональностью. Разработка дополнительной функциональности и модификация существующей выполняется исключительно в пользовательских пакетах.

Основные пакеты приложения

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

Основные пакеты приложения

Название пакета	Описание
[<i>Base</i>]	Базовые схемы основных объектов, разделов системы и связанных с ними схем объектов, страниц, процессов и т. д.
[<i>Platform</i>]	Модули и страницы мастера разделов, дизайнеров реестра и итогов и т. п.
[<i>Managers</i>]	Клиентские модули менеджеров схем.
[<i>NUI</i>]	Функциональность, связанная с пользовательским интерфейсом системы.
[<i>Ulv2</i>]	
[<i>DesignerTools</i>]	Схемы дизайнеров и их элементов.
[<i>ProcessDesigner</i>]	Схемы дизайнера процессов.

Пакет [*Custom*]

В процессе работы мастер разделов или мастер деталей создает схемы, которые необходимо сохранить в пользовательский пакет. В только что установленном приложении нет пакетов, доступных для изменения, а в предустановленные пакеты невозможно внести изменения. Для этого предназначен специальный предустановленный пакет [*Custom*]. Он позволяет добавлять схемы как вручную, так и с помощью мастеров.

Особенности пакета [*Custom*]:

- Пакет [*Custom*] невозможно добавить в систему контроля версий. Поэтому его схемы можно перенести на другую рабочую среду только при помощи функциональности [экспорта и импорта](#) пакетов.
- В отличие от других предустановленных пакетов, пакет [*Custom*] невозможно 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 невозможно добавить пакет [*Custom*].

Рекомендуемые варианты использования пакета [*Custom*]:

- Не предполагается перенос изменений в другую рабочую среду.
В процессе работы мастер разделов или мастер деталей не только создает различные схемы, но и привязывает данные к текущему пакету. Для пакета [*Custom*] не предусмотрено использование стандартного механизма импорта пакетов. Поэтому если текущим пакетом является пакет [*Custom*], то перенести привязанные данные в другой пользовательский пакет можно только с помощью

запросов к базе данных. Мы настоятельно не рекомендуем использовать этот способ, поскольку изменения могут повлиять на структуру базы данных, что приведет к неработоспособности приложения.

При значительной доработке пользовательской функциональности необходимо [создать пользовательский пакет](#) с использованием системы контроля версий.

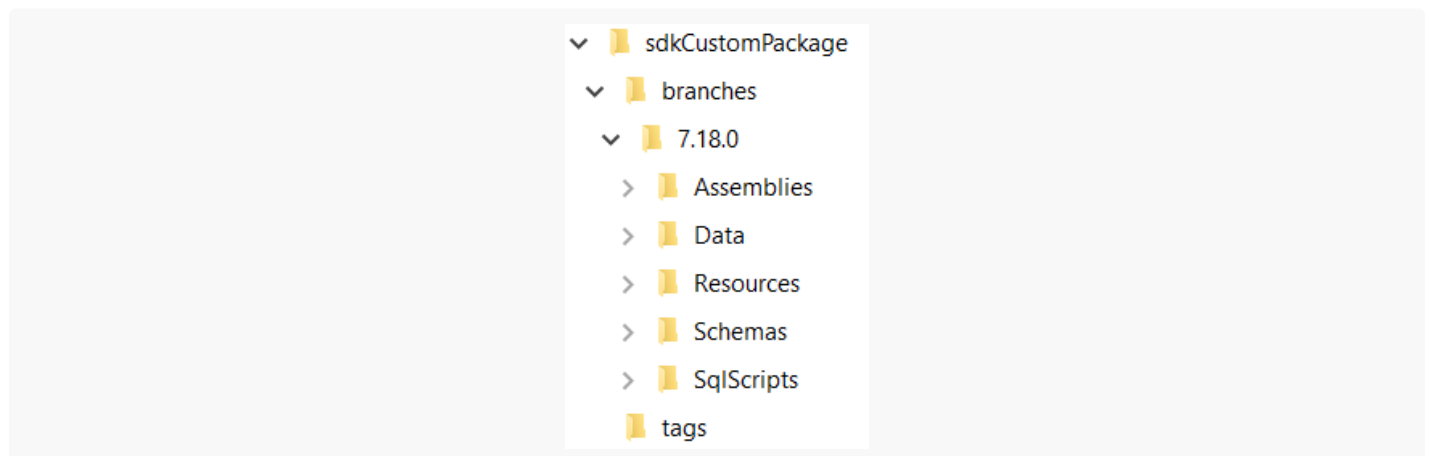
- Изменения выполняются при помощи мастеров или вручную, при этом объем изменений небольшой.
- Нет необходимости использовать систему контроля версий.

Пользовательский пакет

Чтобы выполнять разработку в пользовательском пакете, необходимо в системной настройке [*Текущий пакет*] (код [*CurrentPackageId*]) указать имя пользовательского пакета.

Структура пакета

При фиксации пакета в [системе контроля версий](#) в хранилище пакета создается папка с именем пакета.



Структура папки с именем пакета:

- Папка `branches`.
Назначение — хранение версий текущего пакета. Версия пакета — отдельная вложенная папка, имя которой совпадает с номером версии пакета в системе (например, 7.18.0).
- Папка `tags`.
Назначение — хранение меток. **Метки** в системе контроля версий — это "снимок" проекта в определенный момент времени, статическая копия файлов, необходимая для фиксации этапа разработки.

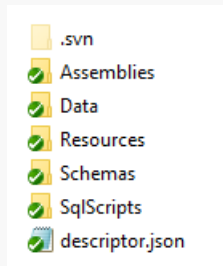
Рабочая копия пакета сохраняется локально в файловой системе. Путь для хранения пакетов задается в конфигурационном файле `ConnectionStrings.config` в атрибуте `connectionString` элемента `defPackagesWorkingCopyPath`.

`ConnectionStrings.config`

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

Структура папки пакета в файловой системе:

- Папка `Schemas` — содержит схемы пакета.
- Папка `Assemblies` — содержит внешние сборки, привязанные к пакету.
- Папка `Data` — содержит данные, привязанные к пакету.
- Папка `SqlScripts` — содержит SQL-сценарии, привязанные к пакету.
- Папка `Resources` — содержит локализованные ресурсы пакета.
- Папка `Files` — содержит [файловый контент](#) пакета.
- Файл `descriptor.json` — хранит метаданные пакета в формате JSON. К метаданным пакета относятся идентификатор, наименование, версия, зависимости и т. д.



Зависимости и иерархия пакетов

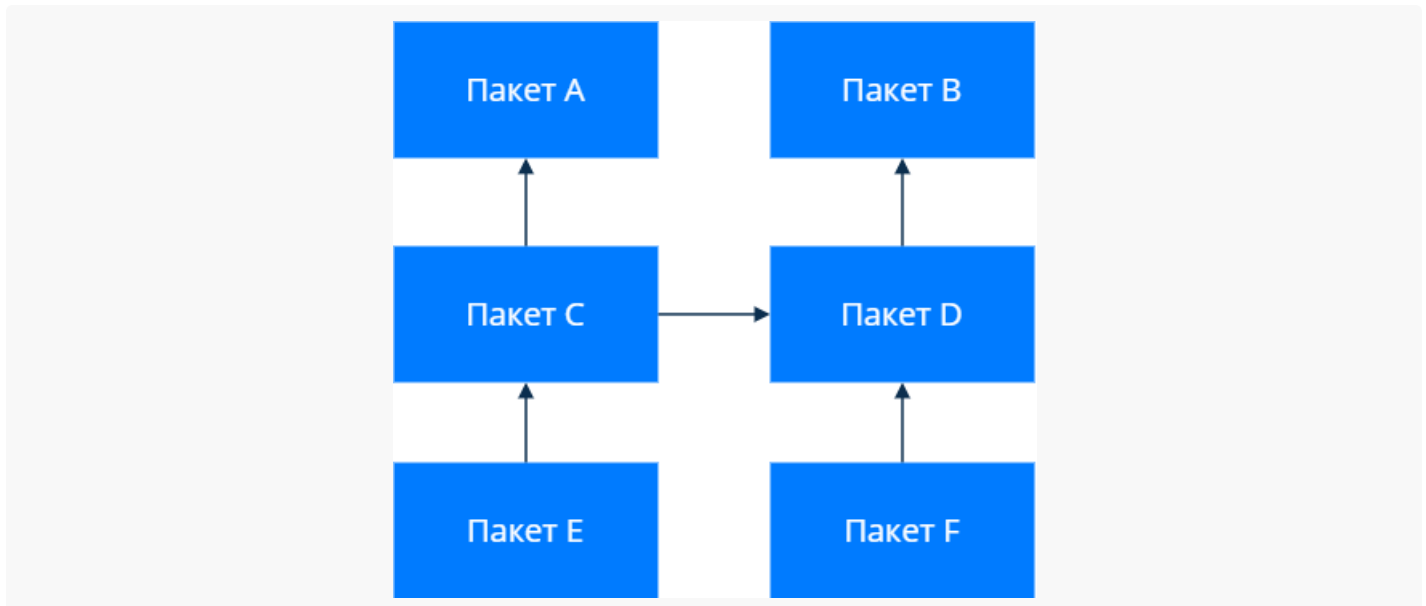
Разработка приложения Creatio базируется на основных принципах проектирования программного обеспечения, в частности, **принципа отсутствия повторений (DRY)**.

В архитектуре Creatio этот принцип реализован с помощью **зависимостей пакетов**. Каждый пакет содержит определенную функциональность приложения, которая не должна повторяться в других пакетах. Чтобы такую функциональность можно было использовать в другом пакете, необходимо пакет, содержащий эту функциональность, добавить в зависимости пакета, в котором она будет использована.

Виды зависимостей:

- Чтобы текущий пакет наследовал всю **функциональность приложения**, в качестве родительского пакета необходимо выбрать пакет, который в иерархии находится следующим после пакета [*Custom*].
- Чтобы текущий пакет наследовал **функциональность пакета**, в качестве родительского пакета необходимо выбрать пакет, функциональность которого необходимо наследовать.

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



Зависимости пакетов формируют **иерархические цепочки**. Это означает, что в пакете доступна не только функциональность дочернего пакета, но и функциональность всех пакетов, для которых дочерний пакет является родительским. Ближайшей аналогией иерархии пакетов является иерархия наследования классов в объектно-ориентированном программировании. Так, например, в пакете E доступна функциональность не только пакета C, от которого он зависит, но и функциональность пакетов A, B и D. А в пакете F доступна функциональность пакетов B и D.

Иерархия пакетов приложения

Иерархия и зависимости пакетов отображены на **диаграмме зависимостей пакетов**. Чтобы открыть диаграмму:

1. Перейдите в раздел [*Конфигурация*] ([*Configuration*]).
2. В выпадающем списке [*Действия*] ([*Actions*]) панели инструментов в группе [*Пакеты*] ([*Packages*]) выберите [*Диаграмма зависимостей пакетов*] ([*Package dependencies diagram*]).

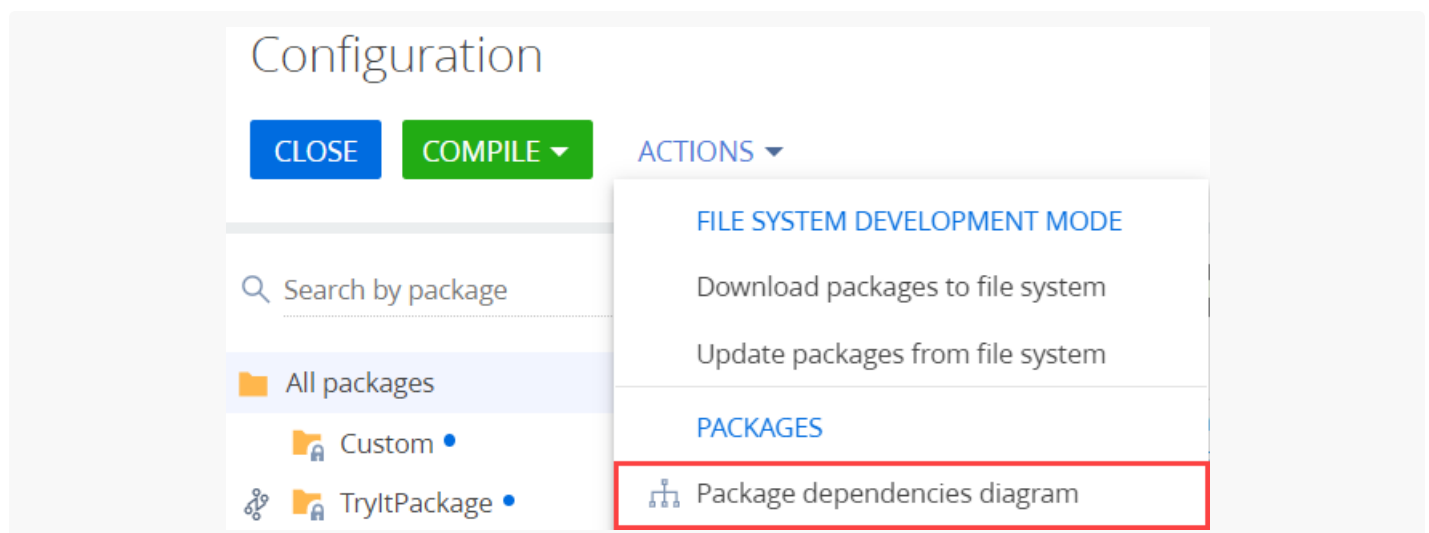
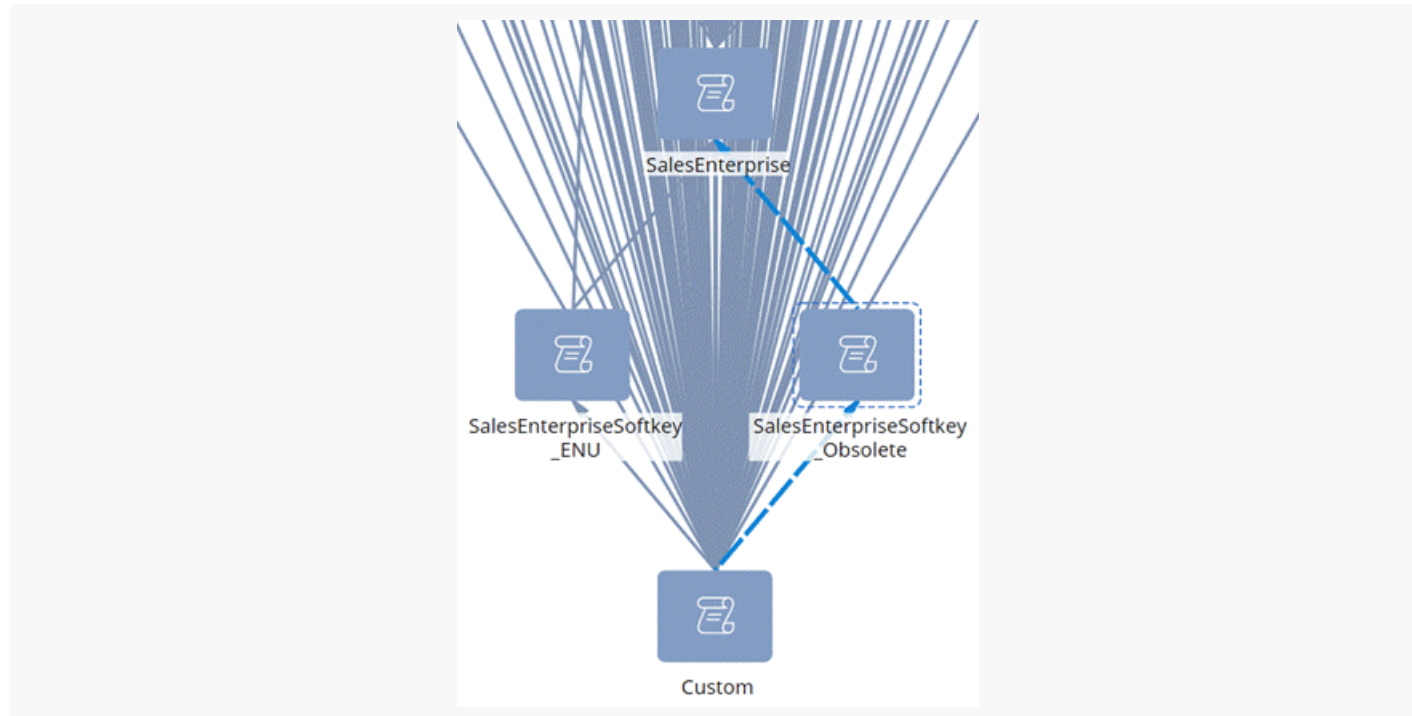


Диаграмма зависимостей будет открыта в новой вкладке.

Если кликнуть по узловому элементу диаграммы с именем пакета, то в виде анимированных стрелок отобразятся связи с другими пакетами. Например, в продукте SalesEnterprise пакет [*SalesEnterpriseSoftkey_Obsolete*] зависит только от пакета [*SalesEnterprise*] и всех его родительских пакетов. Также пакет [*SalesEnterpriseSoftkey_Obsolete*] является родительским для пакета [*Custom*].

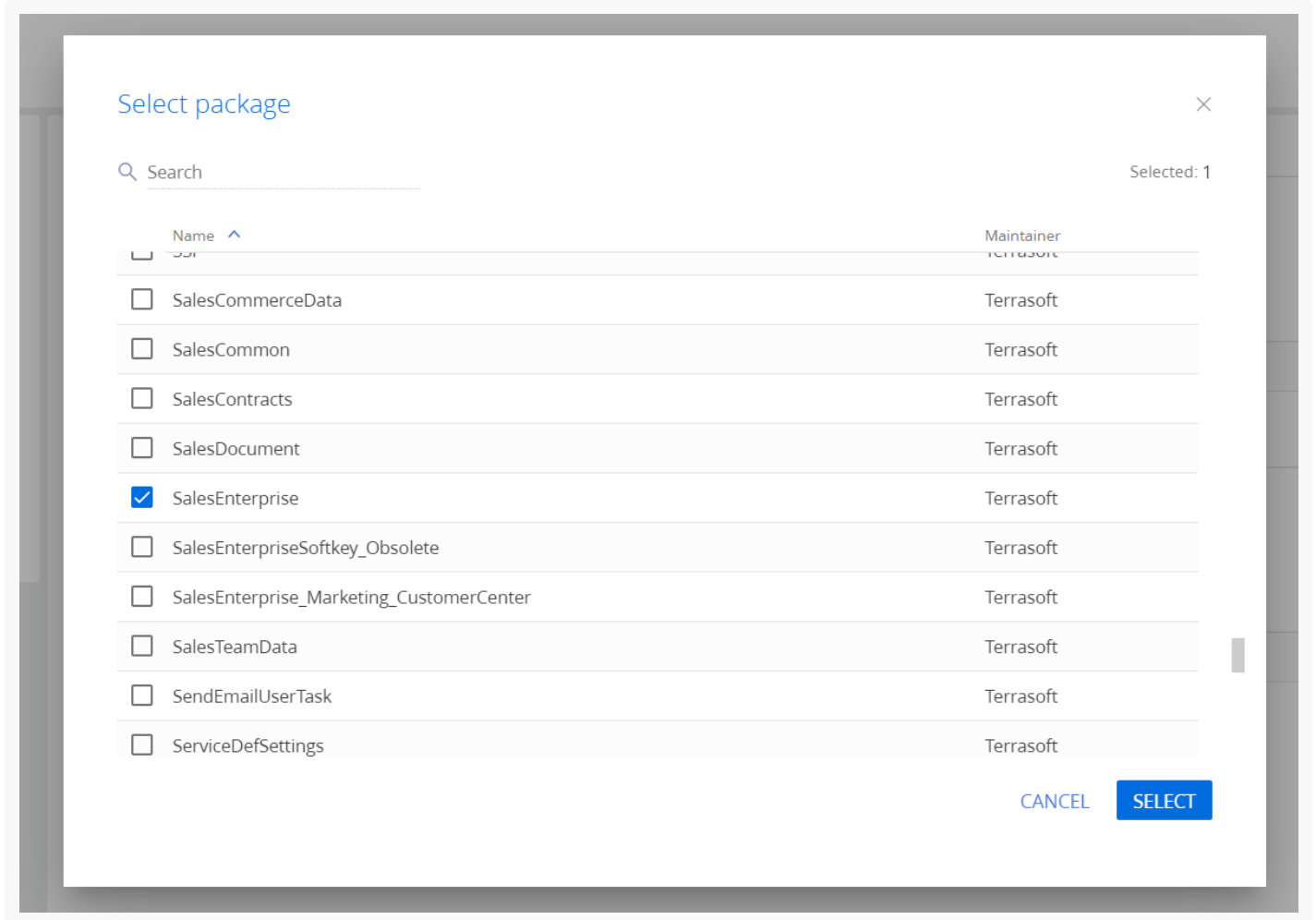


Добавление зависимостей пакета

Зависимости можно добавить в пользовательский пакет при создании пакета или уже после него.

Чтобы **добавить зависимости**:

1. Перейдите на страницу пакета.
2. На вкладке [*Зависимости*] ([*Dependencies*]) на детали [*Зависит от пакетов*] ([*Depends on packages*]) нажмите кнопку [*Добавить*] ([*Add*]).
3. В появившемся окне справочника пакетов выберите необходимый пакет и нажмите кнопку [*Выбрать*] ([*Select*]).



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

Package properties

CLOSE ACTIONS ▾

Name TestPackage1	DEPENDENCIES SYSTEM INFORMATION
Repository address http://tscore-svn:8050/svn/tsfmdoc/SDKP...	Depends on Packages ⓘ
Repository version SDKPackages	🔍 Search by package
Package version 1.0.0	Name ^
Maintainer Customer ⓘ	📁 SalesEnterpriseSoftkey_ENU
Description	+ Add
	Dependent Packages ⓘ
	🔍 Search by package
	Name ^
	📁 Custom

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

Package properties

[CLOSE](#) [ACTIONS](#) ▾

Name
TestPackage1

Repository address
http://tscore-svn:8050/svn/tsfmdoc/SDKP...

Repository version
SDKPackages

Package version
1.0.0

Maintainer
Customer ⓘ

Description

DEPENDENCIES **SYSTEM INFORMATION**

Depends on Packages ⓘ

🔍 Search by package

Name ^

📁 SalesEnterpriseSoftkey_ENU

+ Add

Dependent Packages ⓘ

🔍 Search by package

Name ^

📁 Custom

Список зависимостей в метаданных пакета

Список зависимостей хранится в **метаданных пакета**, которые можно посмотреть в свойстве `DependsOn` объекта, определенного в файле `descriptor.json`.

Свойство `DependsOn` — массив 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": {
    "UIId": "51b3ed42-678c-4da3-bd16-8596b95c0546",
    "PackageVersion": "7.18.0",
    "Name": "UsrDependentPackage",
    "ModifiedOnUtc": "\\Date(1522653150000)\\",
    "Maintainer": "Customer",
    "DependsOn": [
```

```

    {
      "Uuid": "e14dcfb1-e53c-4439-a876-af7f97083ed9",
      "PackageVersion": "7.18.0",
      "Name": "SalesEnterprise"
    }
  ]
}
}

```

Привязка данных к пакету

При переносе изменений между [рабочими средами](#) часто возникает необходимость вместе с разработанной функциональностью предоставлять некоторые данные. Это может быть, например, наполнение справочников, новые системные настройки, демонстрационные записи раздела и т. д.

При создании раздела с помощью мастера к пакету автоматически привязываются данные, необходимые для регистрации и корректной работы раздела.

Name	Title	Status	Type	Object	Modified on	Package
SysModule_SectionManager_8a3879e5f91c49cf81a9d7dd5b3a47c4			Data	SysModule	4/23/2018, 1:12:06 PM	sdkBookExample
SysModuleInWorkplace_SectionInWorkplaceManager_a69d5bf641024326adeca22d02f6dde6			Data	SysModuleInWorkplace	4/23/2018, 1:12:07 PM	sdkBookExample
SysModuleEntity_SysModuleEntityManager_8654e87e4fd34e0a91d903ad427373d9			Data	SysModuleEntity	4/23/2018, 1:12:04 PM	sdkBookExample
SysModuleEdit_SysModuleEditManager_e3a3124a5cd346919574d2d5f5f750c4			Data	SysModuleEdit	4/23/2018, 1:12:05 PM	sdkBookExample
SysImage_f8d3f0121ed2433a996610d4b00cf09a			Data	SysImage	4/23/2018, 1:12:07 PM	sdkBookExample
SysDetail_DetailManager_b78e48663abc45cb916779059502af6b			Data	SysDetail	4/23/2018, 1:35:29 PM	sdkBookExample

Привязать необходимые данные к пакету, содержащему разработанную функциональность, можно в разделе [*Конфигурация*] ([*Configuration*]).

Создать пользовательский пакет

Легкий

1. Создать пакет

1. Перейдите в дизайнер системы по кнопке
2. В блоке [*Конфигурирование разработчиком*] ([*Admin area*]) перейдите по ссылке [*Управление конфигурацией*] ([*Advanced settings*]).
3. В области работы с пакетами нажмите кнопку .

2. Заполнить свойства пакета

При нажатии на кнопку будет отображена карточка пакета, в которой необходимо заполнить свойства пакета.

Свойства пакета:

- [*Название*] ([*Name*]) — название пакета (обязательное свойство). Не может совпадать с названием существующих пакетов.
- [*Описание*] ([*Description*]) — описание пакета, например, расширенная информация о функциональности, которая будет реализована в пакете.
- [*Хранилище системы контроля версий*] ([*Version control system repository*]) — название хранилища системы контроля версий, в котором будут фиксироваться изменения пакета (обязательное свойство). Хранилища, которые находятся в перечне хранилищ конфигурации, но не помечены как активные, не попадут в выпадающий список доступных хранилищ.

Важно. Поле [*Хранилище системы контроля версий*] ([*Version control system repository*]) заполняется при создании нового пакета и в дальнейшем недоступно для редактирования.

- [*Версия*] ([*Version*]) — версия пакета (обязательное свойство). 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. Указываемая версия пакета не обязательно должна совпадать с фактической версией приложения.

Package

Name*
sdkTestPackage

Description

Version control system repository
SDKPackages

Version*
7.18.0

CANCEL CREATE AND ADD DEPENDENCIES SAVE

Содержимое свойств пакета будет сохранено в метаданных пакета.

Метаданные свойств пакета

```
{
  "Descriptor": {
    "UIId": "1c1443d7-87df-4b48-bfb8-cc647755c4c1",
    "PackageVersion": "7.18.0",
    "Name": "NewPackage",
    "ModifiedOnUtc": "\\Date(1522657977000)\\",
    "Maintainer": "Customer",
    "DependsOn": []
  }
}
```

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

3. Определить зависимости пакета

Чтобы текущий пакет наследовал функциональность приложения, необходимо определить **зависимости пакета**.

Чтобы **добавить зависимости** пакета:

1. В карточке пакета нажмите кнопку [*Создать и добавить зависимости*] ([*Create and add dependencies*]).
2. На вкладке [*Зависимости*] ([*Dependencies*]) в детали [*Зависит от пакетов*] ([*Depends on packages*]) установите необходимые зависимости. Чтобы текущий пакет наследовал всю **функциональность приложения**, в качестве родительского пакета необходимо выбрать пакет, который в иерархии находится следующим после пакета [*Custom*].

4. Проверить зависимости пакета [*Custom*]

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

 Средний

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

Демонстрационные записи:

- Книга David Flanagan "JavaScript: The Definitive Guide: Activate Your Web Pages", ISBN 978-0596805524, издательство "Apress", стоимость \$33.89.
- Книга Andrew Troelsen "Pro C# 7: With .NET and .NET Core", ISBN 978-1484230176, издательство "Apress", стоимость \$56.99.

1. Создать раздел

В нашем примере в [мастере разделов](#) предварительно был создан раздел [*Книги*] ([*Books*]). Поля раздела представлены в таблице.

Свойства колонок страницы записей раздела

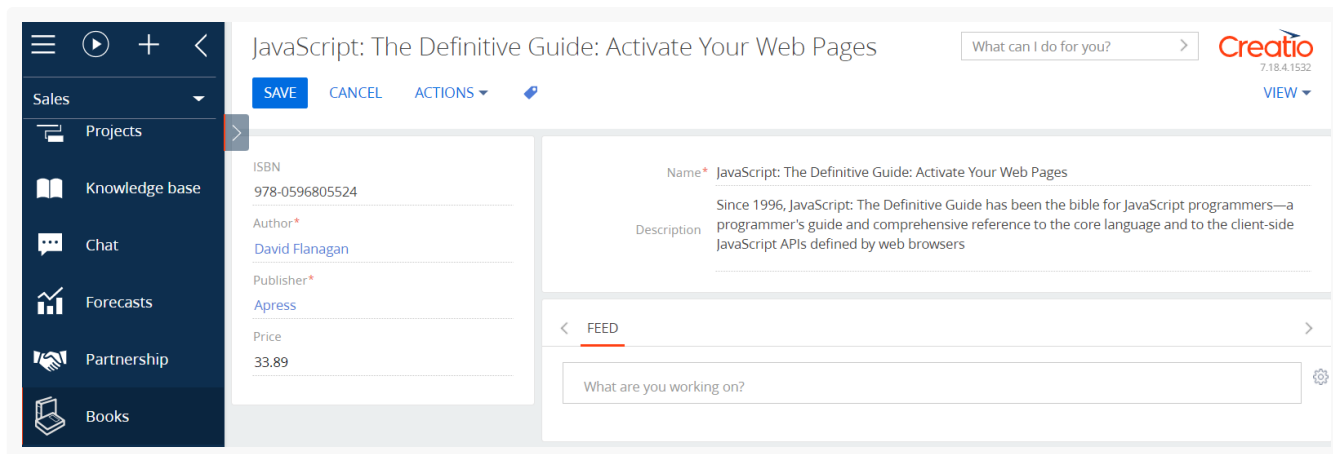
[Заголовок] ([Title])	[Код] ([Code])	Тип данных	Обязательность поля
[Название] ([Name])	<input type="text" value="UserName"/>	Строка (String)	Обязательное поле
[ISBN]	<input type="text" value="UsrISBN"/>	Строка (String)	
[Автор] ([Author])	<input type="text" value="UsrAuthor"/>	Справочник (Lookup) [Контакт] ([Contact])	Обязательное поле
[Издатель] ([Publisher])	<input type="text" value="UsrPublisher"/>	Справочник (Lookup) [Контрагент] ([Account])	Обязательное поле
[Стоимость] ([Price])	<input type="text" value="UsrPrice"/>	Дробное число (Decimal)	

Создание раздела подробно рассмотрено в статье [Создать новый раздел](#).

2. Добавить в раздел демонстрационные записи

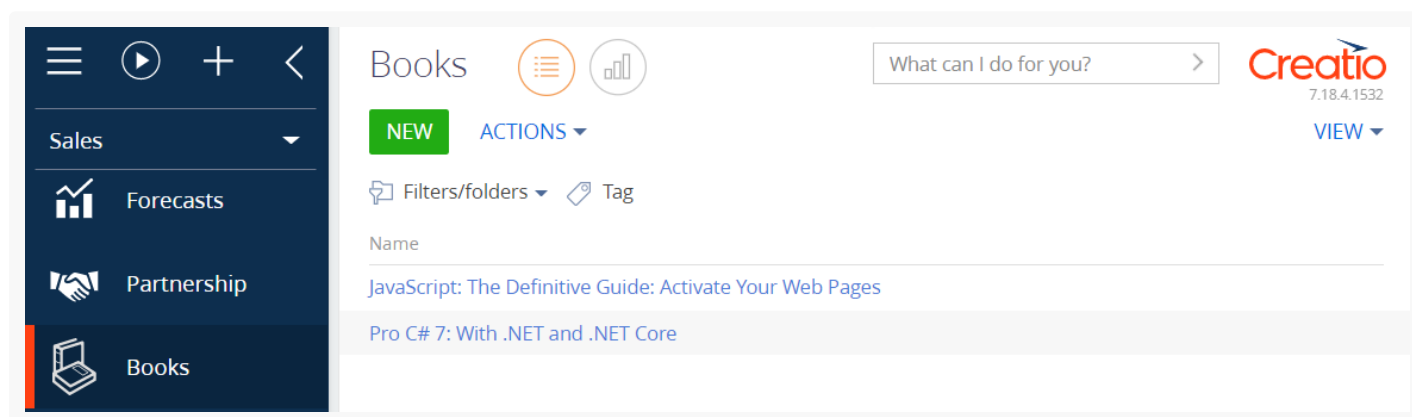
Чтобы **добавить записи** в реестр раздела [Книги] ([Books]):

- В разделе [Контакты] ([Contacts]) добавьте запись и заполните поле [ФИО] ([Full name]) значением "David Flanagan".
- В разделе [Контакты] ([Contacts]) добавьте запись и заполните поле [ФИО] ([Full name]) значением "Andrew Troelsen".
- В разделе [Контрагенты] ([Accounts]) добавьте запись и заполните поле [Название] ([Name]) значением "Apress".
- Добавьте книгу** `JavaScript: The Definitive Guide: Activate Your Web Pages`:
 - Перейдите в раздел [Книги] ([Books]).
 - Нажмите [Добавить] ([New]).
 - Заполните **поля** карточки книги:
 - [Название] ([Name]) — "JavaScript: The Definitive Guide: Activate Your Web Pages".
 - [ISBN] — "978-0596805524".
 - [Автор] ([Author]) — выберите "David Flanagan".
 - [Издатель] ([Publisher]) — выберите "Apress".
 - [Стоимость] ([Price]) — "33.89".



5. Аналогичным образом добавьте книгу `Pro C# 7: With .NET and .NET Core`.

Реестр раздела [Книги] ([Books]) представлен на рисунке ниже.



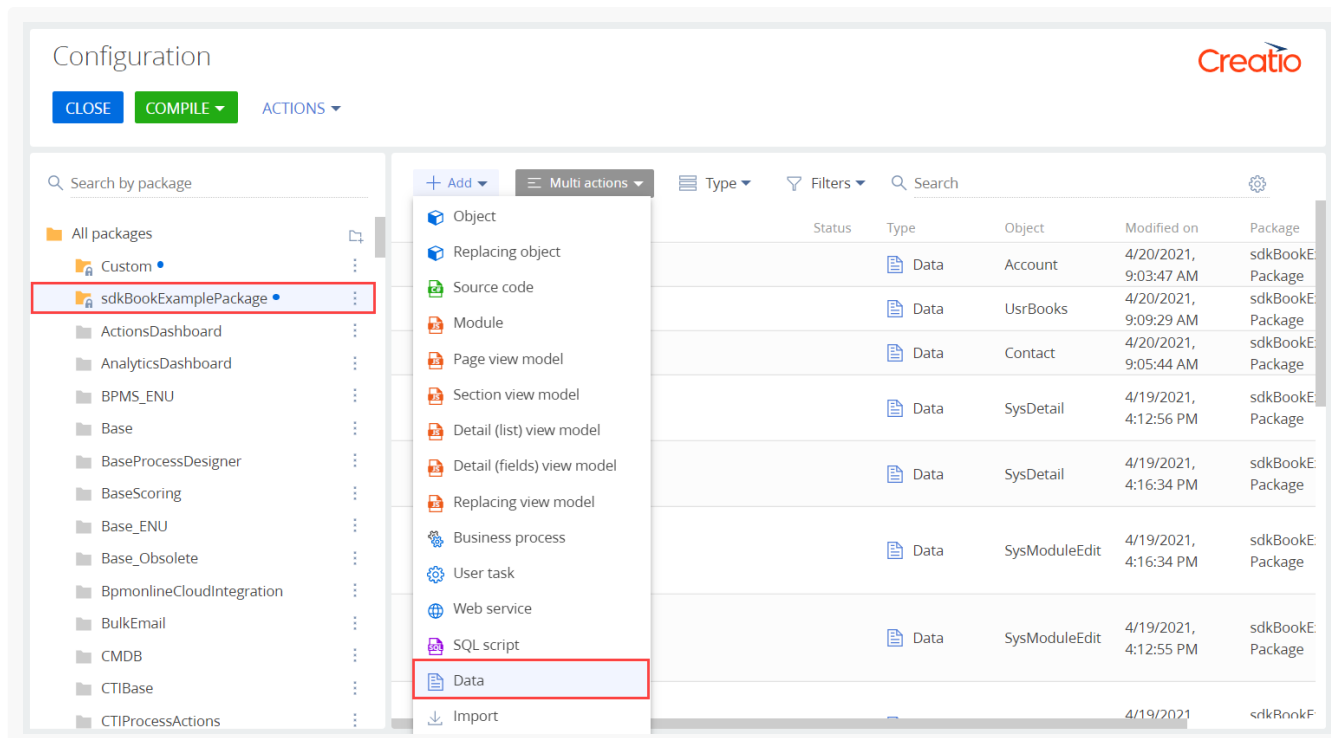
3. Привязать к пакету данные

Поскольку записи раздела [Книги] ([Books]) связаны с записями раздела [Контакты] ([Contacts]) по колонке [*UsrAuthor*], то сначала необходимо привязать к пакету сведения об авторах.

Чтобы **выполнить привязку** данных к пакету:

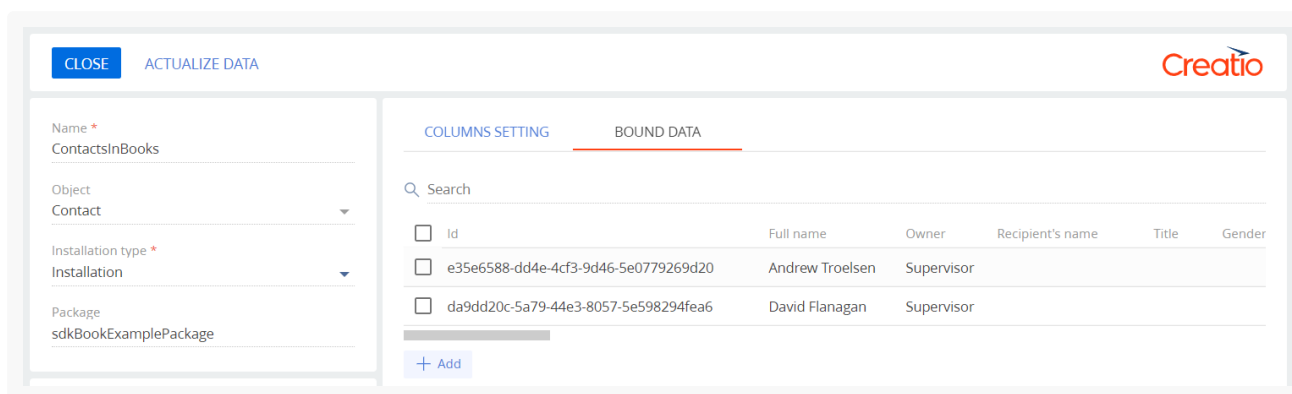
1. Выполните **привязку контактов**:

- a. Перейдите в раздел [Конфигурация] ([Configuration]) и выберите пользовательский пакет.
- b. На панели инструментов рабочей области нажмите кнопку [Добавить] ([Add]) и выберите в списке вид конфигурационного элемента [Данные] ([Data]).



c. Заполните **свойства** страницы привязки данных:

- [*Название*] ([*Name*]) — "ContactsInBooks".
- [*Объект*] ([*Object*]) — "Контакт" ("Contact").
- [*Тип установки*] ([*Installation type*]) — "Установка" ("Installation").
- На вкладке [*Прикрепленные данные*] ([*Bound data*]) выберите записи, которые в колонке [*ФИО*] ([*Full name*]) содержат значения "David Flanagan" и "Andrew Troelsen".

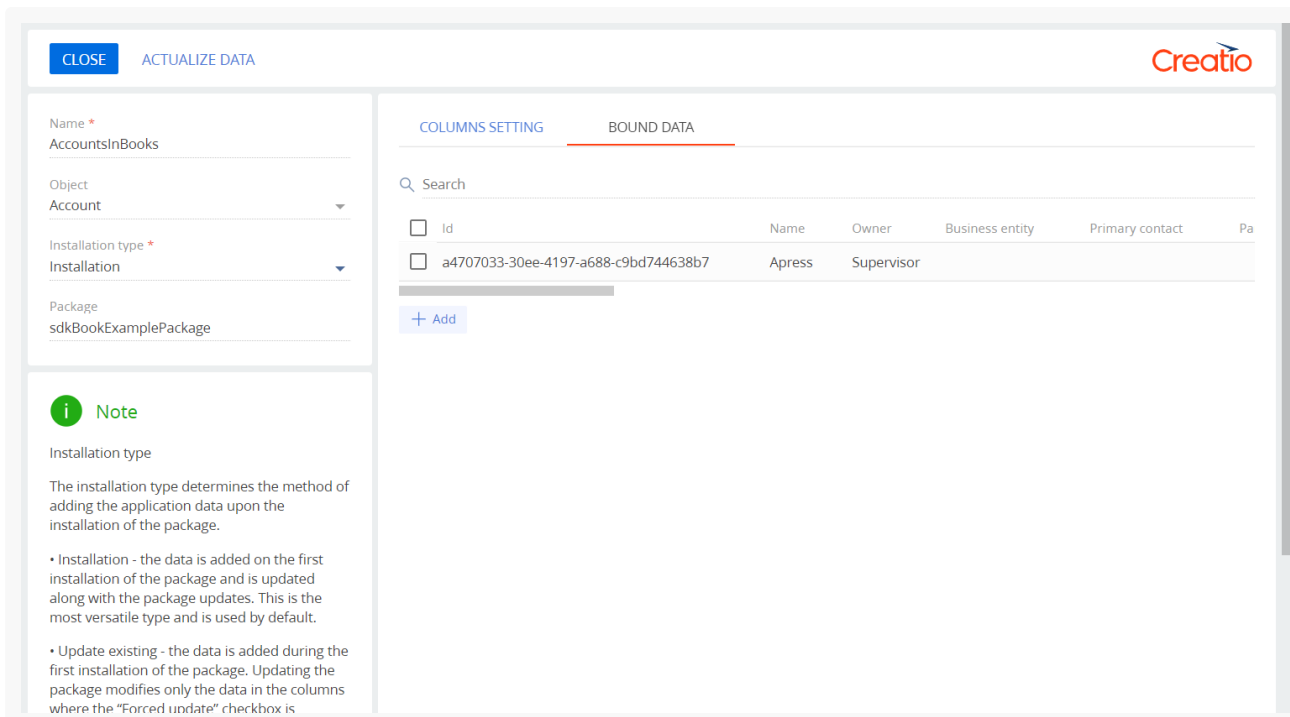


e. Сохраните данные.

2. Выполните **привязку контрагента**:

- Перейдите в раздел [*Конфигурация*] ([*Configuration*]) и выберите пользовательский пакет.
- На панели инструментов рабочей области нажмите кнопку [*Добавить*] ([*Add*]) и выберите в списке вид конфигурационного элемента [*Данные*] ([*Data*]).
- Заполните **свойства** страницы привязки данных:

- a. [*Название*] ([*Name*]) — "AccountsInBooks".
- b. [*Объект*] ([*Object*]) — "Контрагент" ("Account").
- c. [*Тип установки*] ([*Installation type*]) — "Установка" ("Installation").
- d. На вкладке [*Прикрепленные данные*] ([*Bound data*]) выберите запись, которая в колонке [*Название*] ([*Name*]) содержит значение "Apress".



- e. Сохраните данные.

3. Выполните **привязку книг**:

- a. Перейдите в раздел [*Конфигурация*] ([*Configuration*]) и выберите пользовательский пакет.
- b. На панели инструментов рабочей области нажмите кнопку [*Добавить*] ([*Add*]) и выберите в списке вид конфигурационного элемента [*Данные*] ([*Data*]).
- c. Заполните **свойства** страницы привязки данных:
 - a. [*Название*] ([*Name*]) — "Books".
 - b. [*Объект*] ([*Object*]) — "UsrBooks".
 - c. [*Тип установки*] ([*Installation type*]) — "Установка" ("Installation").
 - d. На вкладке [*Прикрепленные данные*] ([*Bound data*]) выберите записи, которые в колонке [*Название*] ([*Name*]) содержат значения "JavaScript: The Definitive Guide: Activate Your Web Pages" и "Pro C# 7: With .NET and .NET Core".

ACTUALIZE DATA

Columns Setting | **BOUND DATA**

Search

<input type="checkbox"/>	Id	Created on	Created by	Modified on	Modified by	Active processes	Name
<input type="checkbox"/>	c44dc8a5-4934-46f4-a3d9-0a0b9ca63409	4/19/2021, 5:05:47 PM	Supervisor	4/19/2021, 5:05:47 PM	Supervisor	0	JavaSc Web P
<input type="checkbox"/>	ab83e661-cd07-47b8-a646-ef76746a10a4	4/19/2021, 5:11:10 PM	Supervisor	4/19/2021, 5:11:10 PM	Supervisor	0	Pro C#

+ Add

Note

Installation type

The installation type determines the method of adding the application data upon the installation of the package.

- Installation - the data is added on the first installation of the package and is updated along with the package updates. This is the most versatile type and is used by default.
- Update existing - the data is added during the first installation of the package. Updating the package modifies only the data in the columns where the "Forced update" checkbox is

e. Сохраните данные.

4. Проверить привязки данных

В результате выполнения примера к пользовательскому пакету будут привязаны данные разделов "[Книги]" ("[Books]"), "[Контакты]" ("[Contacts]"), "[Контрагенты]" ("[Accounts]").

Configuration

CLOSE | **COMPILE** | **ACTIONS**

Search by package

All packages

- Custom
- sdkBookExamplePackage**
 - ActionsDashboard
 - AnalyticsDashboard
 - BPMS_ENU
 - Base
 - BaseProcessDesigner
 - BaseScoring
 - Base_ENU
 - Base_Obsolete
 - BpmonlineCloudIntegration
 - BulkEmail
 - CMDB
 - CTIBase
 - CTIProcessActions

+ Add | Multi actions | Type | Filters | Search

<input type="checkbox"/>	Name	Title	Status	Type	Object	Modified on	Package
<input type="checkbox"/>	AccountsInBooks *			Data	Account	4/20/2021, 9:03:47 AM	sdBookE Package
<input type="checkbox"/>	Books *			Data	UsrBooks	4/20/2021, 9:09:29 AM	sdBookE Package
<input type="checkbox"/>	ContactsInBooks *			Data	Contact	4/20/2021, 9:05:44 AM	sdBookE Package
<input type="checkbox"/>	SysDetail_DetailManager_987f2dae9a1e4eb79ee8861dc9428a1e *			Data	SysDetail	4/19/2021, 4:12:56 PM	sdBookE Package
<input type="checkbox"/>	SysDetail_DetailManager_a61ff163c0f146f1be7240bb57345693 *			Data	SysDetail	4/19/2021, 4:16:34 PM	sdBookE Package
<input type="checkbox"/>	SysModuleEdit_SysModuleEditManager_bdf3143752da45e4ab98f2ef7767d57b *			Data	SysModuleEdit	4/19/2021, 4:16:34 PM	sdBookE Package
<input type="checkbox"/>	SysModuleEdit_SysModuleEditManager_c3f22fb9ee54c148c352bf835ed4e2c *			Data	SysModuleEdit	4/19/2021, 4:12:55 PM	sdBookE Package
<input type="checkbox"/>	SysModuleEntity_SysModuleEntityManager_62ac...					4/19/2021	sdBookE Package

Теперь пакет полностью готов для переноса между [рабочими средами](#) с помощью механизма [экспорта и импорта пакетов](#) Creatio IDE. После установки пакета в другую рабочую среду все привязанные записи отобразятся в соответствующих разделах.