Это неполная документация. Полная – в папке SSL23\_Docs\_ITS

2017-09-09

2.3.6

https://its.1c.ru/db/bsp23doc#content:528:hdoc

**Глава 1. Состав библиотеки**

Библиотека «1С:Библиотека стандартных подсистем» включает в себя набор универсальных функциональных подсистем, предназначенных для использования в прикладных решениях на платформе «1С:Предприятие 8.3». Библиотека не является законченным (с предметной точки зрения) прикладным решением. Подсистемы библиотеки могут использоваться в конфигурации-потребителе как все вместе, так и по отдельности.

Предназначена для разработки конфигураций, которые рассчитаны для работы на платформе «1С:Предприятие 8.3» версии 8.3.7 и выше. При этом значения свойств конфигурации должны быть следующими:

● **Основной режим запуска** должен быть установлен в **Управляемое приложение**.

● В список **Назначения использования** должно входить значение **Персональный компьютер**.

● **Вариант встроенного языка** должен быть установлен в **Русский**.

● **Режим управления блокировкой данных** должен быть установлен в **Управляемый**.

● **Режим совместимости** должен быть установлен в **Не использовать**.

● **Режим использования модальности** может быть установлен в **Не использовать**.

● **Режим совместимости интерфейса** может принимать значения **Версия 8.2**, **Версия 8.2. Разрешить Такси** или **Такси. Разрешить Версия 8.2**.

● **Режим использования синхронных вызовов расширений и внешних компонент** должен быть установлен в значение **Использовать**.

По способу внедрения все подсистемы библиотеки можно условно разделить на следующие категории:

1. Подсистемы, реализующие самостоятельную функциональность.

2. Интегрируемые подсистемы, реализующие функциональность, предназначенную для использования в тех или иных объектах конфигурации-потребителя (тесная интеграция).

Подсистемы первой категории внедряются простым переносом функциональности, как правило, выводятся в интерфейс конфигурации и не требуют существенных дополнительных настроек. Такие подсистемы могут содержать пользовательские интерфейсы и предоставлять программный интерфейс, который можно использовать в конфигурации-потребителе. Обновление этих подсистем при выходе очередной версии библиотеки зачастую сводится к простому обновлению соответствующих объектов конфигурации. Примеры подсистем: «Валюты», «Завершение работы пользователей».

При внедрении интегрируемых подсистем после переноса функциональности подсистемы требуется определить состав объектов конфигурации-потребителя, для которых нужно выполнить внедрение функциональности. После этого необходимо выполнить некоторое количество дополнительных настроек, внести изменения в код и формы выбранных объектов, т. е. выполнить тесную интеграцию подсистемы в объекты конфигурации-потребителя. Обновление таких подсистем при выходе очередной версии библиотеки также сводится к простому обновлению подсистемы и, как правило, не требует повторного внесения изменений в объекты конфигурации-потребителя.

Библиотека содержит не только подсистемы, но и предлагает некоторые методики разработки прикладных решений, например методику установки параметров сеанса.

Описание ключевых функций всех подсистем библиотеки приведено в таблице 1.1.

**Примечание**

Некоторые подсистемы могут иметь ограничения по использованию в прикладных решениях, рассчитанных на работу в локальном режиме или в модели сервиса.

Таблица 1.1. Назначение подсистем библиотеки

● Значками «Работа в модели сервиса» и «Работа в локальном режиме» отмечены подсистемы, поддерживающие и/или обеспечивающие работу конфигурации в этих режимах.

● «Внешние пользователи» – подсистема рассчитана на работу внешних пользователей (подробнее о внешних пользователях см. раздел «[Пользователи](https://its.1c.ru/db/content/bsp23doc/src/3.37.%20%D0%BF%D0%BE%D0%BB%D1%8C%D0%B7%D0%BE%D0%B2%D0%B0%D1%82%D0%B5%D0%BB%D0%B8.htm?_=1504708101#_пользователи)»).

● «Тесная интеграция» – подсистема предназначена для внедрения (интеграции) в объекты конфигурации-потребителя. Значение **Да/Нет** означает наличие различных вариантов внедрения подсистемы.

# Глава 2. Инструкция по внедрению библиотеки

Первое внедрение подсистем библиотеки в конфигурацию и обновление версии уже внедренной библиотеки заметно отличаются. В целом процесс внедрения состоит из следующих этапов:

1. Перенос объектов метаданных из файла поставки библиотеки в конфигурацию.

2. Настройка перенесенных объектов библиотеки.

3. Использование объектов библиотеки при разработке конфигурации.

Все функциональные подсистемы, входящие состав библиотеки, относятся к родительской подсистеме «Стандартные подсистемы» в дереве объектов метаданных конфигуратора. Подсистемы, предназначенные для конфигураций, рассчитанных на работу в модели сервиса, подчинены подсистеме «Работа в модели сервиса».

Для задачи первоначального внедрения библиотеки в прикладных решениях в дистрибутив входит внешняя обработка **ПервоеВнедрениеБСП.epf**. Она позволяет выбрать подсистемы для внедрения с учетом зависимостей, а также удалить код неиспользуемых подсистем. Для проверки корректности встраивания при первом внедрении БСП, а также для проверок при последующих обновлениях версии библиотеки, в составе дистрибутива поставляется внешний отчет **ПроверкаВнедренияБСП.erf**. Он позволяет выявлять различные проблемы внедрения библиотеки (проверка корректности заполнения состава типов, проверка наличия вставок кода и т. п.).

Для быстрого начала разработки новой конфигурации на основе библиотеки можно также воспользоваться [краткой инструкцией](https://its.1c.ru/db/content/bsp23doc/src/2.4.%20%D0%B1%D1%8B%D1%81%D1%82%D1%80%D0%BE%D0%B5%20%D0%BD%D0%B0%D1%87%D0%B0%D0%BB%D0%BE%20%D1%80%D0%B0%D0%B7%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%BA%D0%B8%20_%D1%81%20%D0%BD%D1%83%D0%BB%D1%8F_.htm?_=1504708101#_инструкция_по_быстрому), содержащей только минимально необходимый список действий для запуска конфигурации.

# 2.1. Перенос объектов метаданных из файла поставки библиотеки в конфигурацию

## Первое внедрение

Для выполнения переноса объектов библиотеки в конфигурацию при первом внедрении нужно выбрать в режиме **Конфигуратор** команду меню **Конфигурация – Сравнить, объединить с конфигурацией из файла...** В диалоговом окне указать файл поставки библиотеки и подтвердить запрос о постановке к ней на поддержку.

Шаг 1. В открывшемся окне сравнения и объединения конфигураций снять все флажки и отметить переносимые объекты с помощью команды **Действия – Отметить по подсистемам конфигурации поставщика**. Из группы подсистем **Стандартные подсистемы выбрать**:

* ● обязательные подсистемы согласно таблице 2.1;
* ● дополнительные подсистемы, обязательные к переносу в конфигурацию для работы в модели сервиса согласно таблице 2.2;
* ● подсистемы, которые требуется внедрить в вашу конфигурацию;
* ● подсистемы, от которых зависят переносимые вами подсистемы, согласно таблицам 2.1–2.3\*.

Далее нажать кнопку **Установить**.

**Важно!**

Необходимо убедиться, что в список подсистем для переноса включены все обязательные подсистемы из таблицы 2.1. Если предполагается работа конфигурации в модели сервиса, то обязательными также являются подсистемы из таблицы 2.2. В противном случае конфигурация окажется неработоспособной.

\*Для получения списка подсистем с учетом зависимостей можно воспользоваться обработкой **ПервоеВнедрениеБСП.epf**, входящей в состав дистрибутива.

Шаг 2. Дополнительно нужно установить или снять флажки для отдельных объектов библиотеки и их свойств согласно инструкциям в таблице 2.5.

Далее выбрать режим объединения **Взять из файла** (**Действия – Установить режим для всех…**) и нажать кнопку **Выполнить**. В окне со списком зависимых объектов нажать кнопку **Продолжить**.

Шаг 3. После завершения сравнения и объединения перейти ко второму этапу – [настройка объектов библиотеки](https://its.1c.ru/db/content/bsp23doc/src/2.1.%20%D0%BF%D0%B5%D1%80%D0%B5%D0%BD%D0%BE%D1%81%20%D0%BE%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BE%D0%B2%20%D0%BC%D0%B5%D1%82%D0%B0%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85%20%D0%B8%D0%B7%20%D1%84%D0%B0%D0%B9%D0%BB%D0%B0%20%D0%BF%D0%BE%D1%81%D1%82%D0%B0%D0%B2%D0%BA%D0%B8%20%D0%B1%D0%B8%D0%B1%D0%BB%D0%B8%D0%BE%D1%82%D0%B5%D0%BA%D0%B8%20%D0%B2%20%D0%BA%D0%BE%D0%BD%D1%84%D0%B8%D0%B3%D1%83%D1%80%D0%B0%D1%86%D0%B8%D1%8E.htm?_=1504708101#_настройка_объектов_библиотеки). В зависимости от числа внедряемых подсистем этот шаг может занять от одного до нескольких часов.

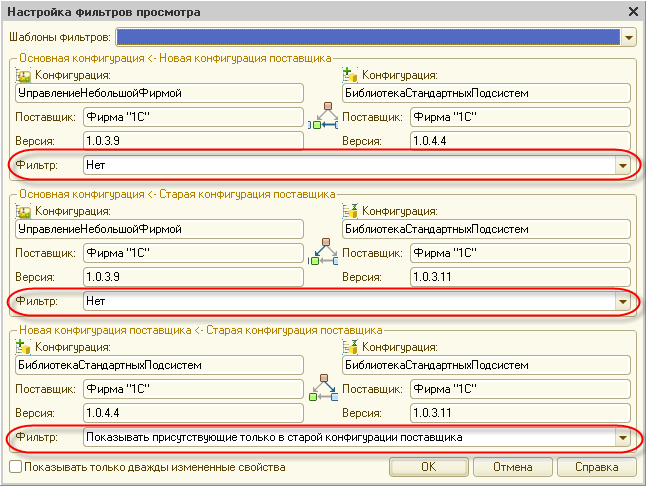
После завершения настройки объектов библиотеки рекомендуется проверить корректность внедрения при помощи внешнего отчета **ПроверкаВнедренияБСП.erf**, входящего в состав дистрибутива.

## Обновление версии библиотеки

В этом разделе описана общая схема обновления на новую версию библиотеки. Отдельные особенности обновления каждой конкретной версии библиотеки приведены в сопроводительном документе **updateSSL.htm**, входящем в поставку библиотеки. Как правило, при переходе между редакциями, подредакциями и релизами библиотеки (первая, вторая и третья цифры в номере версии) требуется выполнить дополнительные действия по обновлению, а при переходе между исправительными релизами библиотеки (четвертая цифра в номере версии) никаких дополнительных действий не требуется. В редких случаях, когда дополнительные действия все же необходимы при обновлении на исправительный релиз библиотеки, соответствующие инструкции содержатся не только в **updateSSL.htm**, но и в файле **readme.txt** в разделе **Важно**.

Для обновления версии библиотеки нужно выбрать команду меню **Конфигурация – Поддержка – Обновить конфигурацию**. В диалоговом окне указать вариант **Выбор файла обновления**, задать путь к файлу поставки библиотеки и подтвердить запрос об обновлении версии конфигурации поставщика.

Шаг 1. Предварительно необходимо подготовить к удалению из конфигурации устаревшие объекты метаданных библиотеки. Для этого в открывшемся окне сравнения и объединения конфигураций снять все флажки напротив объектов метаданных, затем установить фильтр сравнения и объединения в значение **Показывать присутствующие только в старой конфигурации поставщика** в группе **Новая конфигурация поставщика – Старая конфигурация поставщика**, как указано на рисунке.



Далее отметить все объекты метаданных флажками, выбрать режим объединения **Взять из новой конфигурации поставщика** (**Действия – Установить режим для всех…**) и нажать кнопку **Выполнить**.

При наличии ссылок на устаревшие объекты библиотеки из других объектов конфигурации или из других объектов самой библиотеки кнопка **Продолжить** недоступна. Следует вручную очистить все ссылки из других объектов конфигурации и перейти к шагу 2 (при этом ссылки из библиотечных объектов будут удалены автоматически на шаге 3).

Если же ссылок на устаревшие объекты библиотеки не было обнаружено и по кнопке **Продолжить** удалось поставить конфигурацию на поддержку, то следует отменить постановку на поддержку, вернувшись к исходной конфигурации БД (меню **Конфигурация – Конфигурация базы данных – Вернуться к конфигурации БД**).

Шаг 2. Повторно выполнить команду **Конфигурация – Поддержка – Обновить конфигурацию**, указав в варианте **Выбор файла обновления** путь к файлу поставки библиотеки.

В открывшемся окне сравнения и объединения конфигураций снять все флажки и отметить переносимые объекты с помощью команды **Действия – Отметить по подсистемам новой конфигурации поставщика**. Из группы подсистем «Стандартные подсистемы» выбрать:

* ● обязательные подсистемы согласно таблице 2.1;
* ● дополнительные подсистемы, обязательные к переносу в конфигурацию для работы в модели сервиса согласно таблице 2.2;
* ● все ранее внедренные подсистемы, а также подсистемы, которые дополнительно требуется внедрить в вашу конфигурацию;
* ● подсистемы, от которых зависят отмеченные выше подсистемы, согласно таблицам 2.1–2.3.

Далее нажать кнопку **Установить**.

**Важно!**

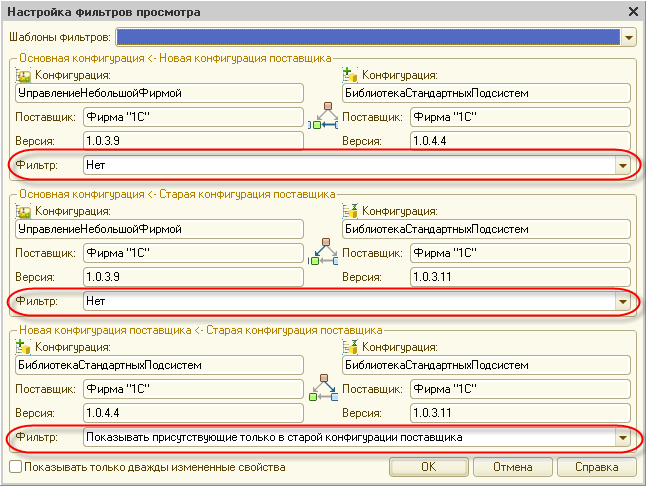
Необходимо убедиться, что в список отмеченных подсистем включены все обязательные подсистемы из таблицы 2.1 (вместе с относящимися к ним объектами метаданных). Если предполагается работа конфигурации в модели сервиса, то обязательными также являются подсистемы из таблицы 2.2. В противном случае конфигурация окажется неработоспособной.

Если подсистема **Администрирование** была внедрена ранее или планируется ее внедрение, то отметить ее флажком в окне сравнения и объединения конфигураций (объект метаданных «подсистема»).

Выбрать режим объединения **Взять из новой конфигурации поставщика** (**Действия – Установить режим для всех…**).

Дополнительно нужно установить или снять флажки для отдельных объектов библиотеки и их свойств согласно инструкциям в таблице 2.6. В зависимости от числа обновляемых и дополнительно устанавливаемых подсистем этот шаг может занять от 15 минут до одного часа.

Затем необходимо отметить к удалению из конфигурации все устаревшие объекты метаданных библиотеки. Для этого установить фильтр сравнения и объединения в значение **Показывать присутствующие только в старой конфигурации поставщика** в группе **Новая конфигурация поставщика – Старая конфигурация поставщика**, как указано на рисунке.



Далее отметить все объекты метаданных флажками и нажать кнопку **Выполнить**. В окне со списком зависимых объектов нажать кнопку **Продолжить**.

Шаг 3. После завершения сравнения и объединения перейти ко второму этапу – [настройка объектов библиотеки](https://its.1c.ru/db/content/bsp23doc/src/2.1.%20%D0%BF%D0%B5%D1%80%D0%B5%D0%BD%D0%BE%D1%81%20%D0%BE%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BE%D0%B2%20%D0%BC%D0%B5%D1%82%D0%B0%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85%20%D0%B8%D0%B7%20%D1%84%D0%B0%D0%B9%D0%BB%D0%B0%20%D0%BF%D0%BE%D1%81%D1%82%D0%B0%D0%B2%D0%BA%D0%B8%20%D0%B1%D0%B8%D0%B1%D0%BB%D0%B8%D0%BE%D1%82%D0%B5%D0%BA%D0%B8%20%D0%B2%20%D0%BA%D0%BE%D0%BD%D1%84%D0%B8%D0%B3%D1%83%D1%80%D0%B0%D1%86%D0%B8%D1%8E.htm?_=1504708101#_настройка_объектов_библиотеки).

После завершения настройки объектов библиотеки рекомендуется проверить корректность внедрения при помощи внешнего отчета **ПроверкаВнедренияБСП.erf**, входящего в состав дистрибутива.

## Зависимости между подсистемами библиотеки

Таблица 2.1. Подсистемы, обязательные к переносу в конфигурацию

|  |  |
| --- | --- |
| **Подсистема** | **От чего зависит** |
| Базовая функциональность |  |
| Обновление версии ИБ |  |
| Пользователи | Контактная информация (\*) |

\* При внедрении подсистемы «Пользователи» для использования в модели сервиса подсистема «Контактная информация» является обязательной. В противном случае подсистема может быть внедрена отдельно. См. инструкции по внедрению без подсистемы в разделе «[Особые случаи внедрения подсистемы](https://its.1c.ru/db/content/bsp23doc/src/3.59.%20%D0%BE%D1%81%D0%BE%D0%B1%D1%8B%D0%B5%20%D1%81%D0%BB%D1%83%D1%87%D0%B0%D0%B8%20%D0%B2%D0%BD%D0%B5%D0%B4%D1%80%D0%B5%D0%BD%D0%B8%D1%8F%20%D0%BF%D0%BE%D0%B4%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC.htm?_=1504708101#_особые_случаи_внедрения)» в соответствующем разделе главы 3.

Таблица 2.2. Дополнительные подсистемы, обязательные к переносу в конфигурацию для работы в модели сервиса

|  |  |
| --- | --- |
| **Подчиненная подсистема в ветке «Работа в модели сервиса»** | **От чего зависит** |
| Базовая функциональность в модели сервиса | Обмен данными  Регламентные задания |
| Обновление версии ИБ в модели сервиса | Завершение работы пользователей |
| Очередь заданий |  |
| Пользователи в модели сервиса |  |
| Удаленное администрирование |  |

Таблица 2.3. Необязательные подсистемы, допускающие выборочную установку

| **Подсистема** | **От чего зависит** |
| --- | --- |
| Адресный классификатор | Контактная информация (\*)  Получение файлов из Интернета  Библиотека интернет-поддержки (БИП) – Базовая функциональность (\*) |
| Анализ журнала регистрации | Варианты отчетов (\*)  Рассылка отчетов (\*) |
| Анкетирование | Настройка порядка элементов |
| Банки (\*\*\*) | Получение файлов из Интернета |
| Бизнес-процессы и задачи | Работа с почтовыми сообщениями (\*)  Управление доступом (\*) |
| Валюты (\*\*\*) | Получение файлов из Интернета |
| Варианты отчетов | Работа с почтовыми сообщениями  Дополнительные отчеты и обработки (\*)  Рассылка отчетов (\*) |
| Версионирование объектов |  |
| Взаимодействия | Контактная информация  Настройка порядка элементов  Отправка SMS  Полнотекстовый поиск  Присоединенные файлы  Работа с почтовыми сообщениями  Свойства  Файловые функции |
| Графики работы | Календарные графики |
| Групповое изменение объектов |  |
| Даты запрета изменения |  |
| Дополнительные отчеты и обработки (\*\*\*) | Печать  Варианты отчетов (\*)  Групповое изменение объектов (\*) |
| Завершение работы пользователей |  |
| Загрузка данных из файла | Дополнительные отчеты и обработки (\*)  Групповое изменение объектов (\*) |
| Заметки пользователя |  |
| Заполнение объектов |  |
| Запрет редактирования реквизитов объектов |  |
| Защита персональных данных | Печать |
| Интеграция с «1С-Коннект» |  |
| Информация при запуске |  |
| Календарные графики |  |
| Контактная информация | Адресный классификатор (\*)  Настройка порядка элементов |
| Напоминания пользователя |  |
| Настройка порядка элементов |  |
| Настройки программы | Все подсистемы (\*) |
| Обмен данными (\*\*\*, \*\*\*\*) | Работа с почтовыми сообщениями  Обновление конфигурации (\*)  Получение файлов из Интернета (\*)  Префиксация объектов (\*)  Регламентные задания (\*) |
| Обновление конфигурации | Завершение работы пользователей  Работа с почтовыми сообщениями (\*)  Проверка легальности получения обновлений (\*) |
| Отправка SMS | Получение файлов из Интернета |
| Оценка производительности |  |
| Печать | Работа с почтовыми сообщениями (\*)  Присоединенные файлы (\*)  Дополнительные отчеты и обработки (\*) |
| Поиск и удаление дублей |  |
| Полнотекстовый поиск |  |
| Получение файлов из Интернета |  |
| Префиксация объектов |  |
| Присоединенные файлы | Файловые функции |
| Проверка легальности получения обновлений |  |
| Работа с контрагентами | Адресный классификатор  Контактная информация  Библиотека интернет-поддержки (БИП) – Базовая функциональность (\*) |
| Работа с почтовыми сообщениями |  |
| Работа с файлами | Свойства  Файловые функции  Управление доступом (\*) |
| Рассылка отчетов | Варианты отчетов  Дополнительные отчеты и обработки (\*)  Контактная информация  Получение файлов из Интернета  Работа с почтовыми сообщениями  Работа с файлами (\*)  Управление доступом (\*)  Групповое изменение объектов (\*) |
| Регламентные задания |  |
| Резервное копирование ИБ | Завершение работы пользователей |
| Свойства | Запрет редактирования реквизитов объектов |
| Склонение представлений объектов |  |
| Структура подчиненности |  |
| Текущие дела |  |
| Удаление помеченных объектов |  |
| Управление доступом (\*\*\*) |  |
| Управление итогами и агрегатами |  |
| Файловые функции (\*\* , \*\*\*) |  |
| Центр мониторинга | Оценка производительности (\*) |
| Шаблоны сообщений | Взаимодействия (\*)  Отправка SMS (\*)  Печать (\*)  Работа с почтовыми сообщениями (\*) |
| Электронная подпись | Контактная информация (\*)  Адресный классификатор (\*)  Печать (\*)  Работа с контрагентами (\*)  Библиотека интернет-поддержки (БИП) – Базовая функциональность (\*) |

\* Нежесткая зависимость, т. е. подсистема может быть внедрена отдельно. В соответствующем разделе главы 3 могут содержаться дополнительные инструкции по отдельному внедрению подсистемы.

\*\* Подсистема «Файловые функции» не предназначена для самостоятельного внедрения. Только совместно с подсистемами «Работа с файлами» и/или «Присоединенные файлы».

\*\*\* При внедрении в конфигурацию, рассчитанную на работу в модели сервиса, следует также отметить соответствующую подсистему с постфиксом **ВМоделиСервиса** из родительской подсистемы «Работа в модели сервиса». Например, если к внедрению отмечена подсистема «Валюты», следует также отметить и подсистему «Валюты в модели сервиса».

\*\*\*\* Подсистема «Обмен данными» обязательна к внедрению при необходимости разработки обмена данными в распределенной информационной базе (РИБ).

Таблица 2.4. Необязательные подсистемы, допускающие выборочную установку для конфигураций, рассчитанных на работу в модели сервиса

| **Подчиненная подсистема в ветке «Работа в модели сервиса»** | **От чего зависит** |
| --- | --- |
| Банки в модели сервиса | Банки  Поставляемые данные |
| Валюты в модели сервиса | Валюты  Поставляемые данные |
| Дополнительные отчеты и обработки в модели сервиса | Дополнительные отчеты и обработки  Поставляемые данные  Удаленное администрирование |
| Поставляемые данные | Файловые функции |
| Управление доступом в модели сервиса | Управление доступом |
| Файловые функции в модели сервиса | Файловые функции |

## Объекты и свойства объектов, для которых необходимо дополнительно изменить отметку перед объединением конфигурации с библиотекой

Таблица 2.5. Объекты и свойства объектов, для которых необходимо дополнительно изменить отметку перед объединением конфигурации с библиотекой при первом внедрении подсистемы в конфигурацию

| **№** | **Название объекта** | **Инструкция** |
| --- | --- | --- |
| **Для подсистемы «Базовая функциональность»** | | |
| 1. | Подсистема **ПодключаемыеОтчетыИОбработки** | Установить флажок (только флажок самой подсистемы, состав подсистемы отмечать к объединению не нужно) |
| 2. | Язык **Русский** | Установить флажок\* |
| \* Если виден в дереве метаданных в окне сравнения и объединения конфигураций |
| 3. | Свойства корневого объекта конфигурации:  **● Основные роли**  **● Модуль управляемого приложения**  **● Модуль сеанса**  **● Модуль внешнего соединения**  **● Модуль обычного приложения**  **● Режим использования модальности**  **● Режим использования синхронных вызовов расширений платформы и внешних компонент**  **● Режим совместимости интерфейса**  ● **Режим совместимости** | Установить флажок |
| 4. | Остальные свойства корневого объекта конфигурации | Снять флажок |
| **Для подсистемы «Настройки программы»** | | |
| 5. | Подсистема **Администрирование** | Установить флажок (только флажок самой подсистемы, состав подсистемы отмечать к объединению не нужно) |
| 6. | Общие команды:  ●**ДополнительныеОбработкиАдминистрирование**,  **● ДополнительныеОтчетыАдминистрирование** | Снять флажок \* |
| \* Только при внедрении без подсистемы «Дополнительные отчеты и обработки» |
| 7. | Общая команда **ПанельОтчетовАдминистрирование** | Снять флажок \* |
| \* Только при внедрении без подсистемы «Варианты отчетов» |
| \* Только при внедрении без подсистемы «Префиксация объектов» |

Таблица 2.6. Объекты и свойства объектов, для которых необходимо дополнительно изменить отметку перед объединением конфигурации с библиотекой при обновлении подсистемы в конфигурации

**Свойства конфигурации**

| **№** | **Название объекта** | **Инструкция** |
| --- | --- | --- |
| **Подсистема «Базовая функциональность»** | | |
| 1. | **●**Свойства корневого объекта конфигурации | ● Снять флажок |

**Подсистемы**

| **№** | **Название объекта** | **Инструкция** |
| --- | --- | --- |
| **Подсистема «Настройки программы»** | | |
| 2. | **●ПодключаемыеОтчетыИОбработки** | Установить флажок (только флажок самой подсистемы, состав подсистемы отмечать к объединению не нужно).  Для свойства **Командный интерфейс** установить режим объединения **Объединить с приоритетом новой конфигурации поставщика**.  Для свойства **Состав** установить режим объединения **Объединить** |
| 3. | **● Администрирование** | Установить флажок (только флажок самой подсистемы, состав подсистемы отмечать к объединению не нужно).  Для свойства **Командный интерфейс** установить режим объединения **Объединить с приоритетом новой конфигурации поставщика**.  ● Для свойства **Состав** установить режим объединения **Объединить** |

**Общие модули**

Снять флажок для всех переопределяемых общих модулей:

| **№** | **Название объекта** |
| --- | --- |
| **Подсистема «Базовая функциональность»** | |
| 1. | ● **ОбщегоНазначенияКлиентПереопределяемый**.  ● **ОбщегоНазначенияПереопределяемый**.  ● **ПодсистемыКонфигурацииПереопределяемый**.  ● **РаботаВБезопасномРежимеПереопределяемый** |
| **Все подсистемы ветки «Работа в модели сервиса»** | |
| 2. | ● **ИнтерфейсыСообщенийВМоделиСервисаПереопределяемый**.  ● **ОбменДаннымиВМоделиСервисаПереопределяемый**.  ● **ОбменСообщениямиПереопределяемый**.  ● **ОчередьЗаданийПереопределяемый**.  ● **ПоставляемыеДанныеПереопределяемый**.  ● **РаботаВМоделиСервисаПереопределяемый**.  ● **СообщенияВМоделиСервисаПереопределяемый**.  ● **ТрансляцияXDTOПереопределяемый** |
| **Подсистема «Банки»** | |
| 3. | ● **РаботаСБанкамиПереопределяемый** |
| **Подсистема «Бизнес-процессы и задачи»** | |
| 4. | ● **БизнесПроцессыИЗадачиКлиентПереопределяемый**.  ● **БизнесПроцессыИЗадачиПереопределяемый** |
| **Подсистема «Варианты отчетов»** | |
| 5. | ● **ВариантыОтчетовПереопределяемый**.  ● **ОтчетыКлиентПереопределяемый**.  ● **ОтчетыПереопределяемый** |
| **Подсистема «Взаимодействия»** | |
| 6. | ● **ВзаимодействияКлиентПереопределяемый**.  ● **ВзаимодействияКлиентСерверПовтИспПереопределяемый**.  ● **ВзаимодействияПереопределяемый** |
| **Подсистема «Валюты»** | |
| 7. | ● **РаботаСКурсамиВалютПереопределяемый** |
| **Подсистема «Групповое изменение объектов»** | |
| 8. | ● **ГрупповоеИзменениеОбъектовПереопределяемый** |
| **Подсистема «Даты запрета изменения»** | |
| 9. | ● **ДатыЗапретаИзмененияПереопределяемый** |
| **Подсистема «Дополнительные отчеты и обработки»** | |
| 10. | **● ДополнительныеОтчетыИОбработкиПереопределяемый**.  ● **ДополнительныеОтчетыИОбработкиКлиентПереопределяемый** |
| **Подсистема «Заполнение объектов»** | |
| 11. | ● **ЗаполнениеОбъектовПереопределяемый** |
| **Подсистема «Запрет редактирования реквизитов объектов»** | |
| 12. | ● **ЗапретРедактированияРеквизитовОбъектовПереопределяемый** |
| **Подсистема «Защита персональных данных»** | |
| 13. | ● **ЗащитаПерсональныхДанныхПереопределяемый** |
| **Подсистема «Информация при запуске»** | |
| 14. | **● ИнформацияПриЗапускеПереопределяемый** |
| **Подсистема «Календарные графики»** | |
| 15. | ● **КалендарныеГрафикиПереопределяемый** |
| **Подсистема «Напоминания пользователя»** | |
| 16. | ● **НапоминанияПользователяКлиентСерверПереопределяемый** |
| **Подсистема «Обмен данными»** | |
| 17. | ● **ОбменДаннымиПереопределяемый** |
| **Подсистема «Обновление версии ИБ»** | |
| 18. | ● **ОбновлениеИнформационнойБазыКлиентПереопределяемый**.  ● **ОбновлениеИнформационнойБазыПереопределяемый** |
| **Подсистема «Отправка SMS»** | |
| 19. | ● **ОтправкаSMSПереопределяемый** |
| **Подсистема «Печать»** | |
| 20. | ● **УправлениеПечатьюПереопределяемый** |
| **Подсистема «Поиск и удаление дублей»** | |
| 21. | ● **ПоискИУдалениеДублейПереопределяемый** |
| **Подсистема «Полнотекстовый поиск»** | |
| 22. | ● **ПолнотекстовыйПоискСерверПереопределяемый** |
| **Подсистема «Пользователи»** | |
| 23. | ● **ПользователиПереопределяемый** |
| **Подсистема «Префиксация объектов»** | |
| 24. | ● **ПрефиксацияОбъектовКлиентСерверПереопределяемый**.  ● **ПрефиксацияОбъектовПереопределяемый** |
| **Подсистема «Присоединенные файлы»** | |
| 25. | **ПрисоединенныеФайлыПереопределяемый** |
| **Подсистема «Работа с почтовыми сообщениями»** | |
| 26. | **● РаботаСПочтовымиСообщениямиПереопределяемый**.  ● **РаботаСПочтовымиСообщениямиКлиентПереопределяемый** |
| **Подсистема «Работа с контрагентами»** | |
| 27. | ● **РаботаСКонтрагентамиКлиентПереопределяемый**.  ● **РаботаСКонтрагентамиКлиентСерверПереопределяемый**.  ● **РаботаСКонтрагентамиПереопределяемый** |
| **Подсистема «Работа с файлами»** | |
| 28. | ● **РаботаСФайламиКлиентПереопределяемый**.  ● **РаботаСФайламиПереопределяемый** |
| **Подсистема «Рассылка отчетов»** | |
| 29. | ● **РассылкаОтчетовПереопределяемый** |
| **Подсистема «Регламентные задания»** | |
| 30. | ● **РегламентныеЗаданияПереопределяемый** |
| **Подсистема «Свойства»** | |
| 31. | ● **УправлениеСвойствамиПереопределяемый** |
| **Подсистема «Структура подчиненности»** | |
| 32. | ● **СтруктураПодчиненностиПереопределяемый** |
| **Подсистема «Текущие дела»** | |
| 33. | ● **ТекущиеДелаПереопределяемый** |
| **Подсистема «Управление доступом»** | |
| 34. | ● **УправлениеДоступомПереопределяемый** |
| **Подсистема «Шаблоны сообщений»** | |
| 35. | ● **ШаблоныСообщенийПереопределяемый** |
| **Подсистема «Электронная подпись»** | |
| 36. | **● ЭлектроннаяПодписьКлиентПереопределяемый**.  **● ЭлектроннаяПодписьПереопределяемый** |

**Роли**

| **№** | **Название объекта** | **Инструкция** |
| --- | --- | --- |
| **Подсистема «Базовая функциональность»** | | |
| 1. | **● ПолныеПрава**  ●**АдминистраторСистемы** | ● Для свойства **Права** установить **Режим объединения…** в **Объединить с приоритетом новой конфигурации поставщика** |

**Общие реквизиты**

| **№** | **Название объекта** | **Инструкция** |
| --- | --- | --- |
| **Все подсистемы ветки «Работа в модели сервиса»** | | |
| 1. | **● ОбластьДанныхОсновныеДанные**.  **● ОбластьДанныхВспомогательныеДанные** | Снять флажок |

**Критерии отбора**

| **№** | **Название объекта** | **Инструкция** |
| --- | --- | --- |
| **Подсистема «Структура подчиненности»** | | |
| 1. | **● СвязанныеДокументы**, свойство **Тип** и свойство **Состав** | ● Снять флажок |

**Определяемые типы**

| **№** | **Название объекта** | **Инструкция** |
| --- | --- | --- |
| **Подсистема «Базовая функциональность»** | | |
| 1. | **● Организация** | ● Снять флажок |
| **Подсистема «Анкетирование»** | | |
| 2. | **● Респондент** | ● Установить **Режим объединения…**в **Объединить с приоритетом новой конфигурации поставщика** |
| **Подсистема «Базовая функциональность»** | | |
| 3. | **● МестоХраненияФункциональныхОпций** | ● Установить **Режим объединения…**в **Объединить с приоритетом новой конфигурации поставщика** |
| **Подсистема «Бизнес-процессы и задачи»** | | |
| 4. | ● **БизнесПроцесс**.  ● **ОбъектАдресации**.  **● ПредметЗадачи** | ● Установить **Режим объединения…**в **Объединить с приоритетом новой конфигурации поставщика** |
| **Подсистема «Взаимодействия»** | | |
| 5. | ● **КонтактВзаимодействия**.  **● ПредметВзаимодействия** | ● Установить **Режим объединения…**в **Объединить с приоритетом новой конфигурации поставщика** |
| **Подсистема «Версионирование объектов»** | | |
| 6. | ● **ВерсионируемыеДанные**.  **● ВерсионируемыеДанныеОбъект** | ● Установить **Режим объединения…**в **Объединить с приоритетом новой конфигурации поставщика** |
| **Подсистема «Даты запрета изменения»** | | |
| 7. | **● АдресатЗапретаИзменения** | ● Установить **Режим объединения…**в **Объединить с приоритетом новой конфигурации поставщика** |
| **Подсистема «Дополнительные отчеты и обработки»** | | |
| 8. | **● ОбъектСДополнительнымиКомандами** | ● Установить **Режим объединения…**в **Объединить с приоритетом новой конфигурации поставщика** |
| **Подсистема «Заметки пользователя»** | | |
| 9. | ● **ПредметЗаметок**.  **● ПредметЗаметокОбъект** | ● Установить **Режим объединения…**в **Объединить с приоритетом новой конфигурации поставщика** |
| **Подсистема «Защита персональных данных»** | | |
| 10. | **● СубъектПерсональныхДанных** | ● Установить **Режим объединения…**в **Объединить с приоритетом новой конфигурации поставщика** |
| **Подсистема «Контактная информация»** | | |
| 11. | **● ВладелецКонтактнойИнформации** | ● Установить **Режим объединения…**в **Объединить с приоритетом новой конфигурации поставщика** |
| **Подсистема «Настройка порядка элементов»** | | |
| 12. | **● ОбъектСНастраиваемымПорядком** | ● Установить **Режим объединения…**в **Объединить с приоритетом новой конфигурации поставщика** |
| **Подсистема «Напоминания пользователя»** | | |
| 13. | ● **ПредметНапоминания**.  **● ПредметНапоминанияОбъект** | ● Установить **Режим объединения…**в **Объединить с приоритетом новой конфигурации поставщика** |
| **Подсистема «Пользователи»** | | |
| 14. | **● Подразделение**.  **● ФизическоеЛицо** | ● Снять флажок |
| 15. | **● Пользователь**.  ● **ВнешнийПользователь**.  **● ВнешнийПользовательОбъект** | ● Установить **Режим объединения…**в **Объединить с приоритетом новой конфигурации поставщика** |
| **Подсистема «Присоединенные файлы»** | | |
| 16. | ● **ПрисоединенныйФайл**.  ● **ПрисоединенныйФайлОбъект**.  ● **ВладелецПрисоединенныхФайлов**.  **● ВладелецПрисоединенныхФайловОбъект** | ● Установить **Режим объединения…**в **Объединить с приоритетом новой конфигурации поставщика** |
| **Подсистема «Работа с контрагентами»** | | |
| 17. | ● **Контрагент**.  **● КонтрагентОбъект** | ● Установить **Режим объединения…**в **Объединить с приоритетом новой конфигурации поставщика** |
| **Подсистема «Работа с файлами»** | | |
| 18. | ● **ВладелецФайлов**.  **● ВладелецФайловОбъект** | ● Установить **Режим объединения…**в **Объединить с приоритетом новой конфигурации поставщика** |
| **Подсистема «Рассылка отчетов»** | | |
| 19. | **● ПолучательРассылки** | ● Установить **Режим объединения…**в **Объединить с приоритетом новой конфигурации поставщика** |
| **Подсистема «Свойства»** | | |
| 20. | **● ВладелецДополнительныхСведений** | ● Установить **Режим объединения…**в **Объединить с приоритетом новой конфигурации поставщика** |
| **Подсистема «Склонение представлений объектов»** | | |
| 21. | **● ОбъектСклонения** | ● Установить **Режим объединения…**в **Объединить с приоритетом новой конфигурации поставщика** |
| **Подсистема «Управление доступом»** | | |
| 22. | ● **ВладелецНастроекПрав**.  ● **ВладелецНастроекПравОбъект**.  **● ЗначениеДоступа**.  ● **ЗначениеДоступаСГруппамиЗначенийДоступаОбъект**.  ● **ВладелецНаборовЗначенийДоступаОбъект**.  ●**ВладелецСОграничениемПоНаборамЗначенийДоступаОбъект**.  **●ВладелецВнешнихЗначенийВНаборахЗначенийДоступаОбъект** | ● Установить **Режим объединения…**в **Объединить с приоритетом новой конфигурации поставщика** |
| 23. | **Подсистема «Шаблоны сообщений»** |  |
| 24. | **● ПредметШаблонаСообщения** | **●**Установить **Режим объединения…**в **Объединить с приоритетом новой конфигурации поставщика** |

**Общие команды**

| **№** | **Название объекта** | **Инструкция** |
| --- | --- | --- |
| **Подсистема «Обмен данными»** | | |
| 1. | Свойство**Тип параметра команды**:  ● **ВыполнитьОбменДанными**.  ● **ВыполнитьОбменДаннымиИнтерактивно**.  ● **НастроитьПараметрыТранспортаСообщенийОбмена**.  ● **ОткрытьПравилаКонвертацииОбъектов**.  ● **ОткрытьПравилаРегистрацииОбъектов**.  ● **ОткрытьСценарииОбменовДанными**.  ● **ПерейтиВЖурналРегистрацииСобытийВыгрузкиДанных**.  ● **ПерейтиВЖурналРегистрацииСобытийЗагрузкиДанных**.  ● **СоставОтправляемыхДанных**.  **● УдалитьНастройкуСинхронизации** | ● Снять флажок |
| **Подсистема «Структура подчиненности»** | | |
| 2. | **● СтруктураПодчиненности**, свойство **Тип параметра команды** | ● Снять флажок |
| **Подсистема «Управление доступом»** | | |
| 3. | **● НастроитьПрава**, свойство **Тип параметра команды** | ● Снять флажок |

**Общие макеты**

| **№** | **Название объекта** | **Инструкция** |
| --- | --- | --- |
| **Подсистема «Работа с контрагентами»** | | |
| 4. | **● ИнструкцияПоПроверкеКонтрагентов** | Снять флажок |

**Языки**

| **№** | **Название объекта** | **Инструкция** |
| --- | --- | --- |
| **Подсистема «Базовая функциональность»** | | |
| 5. | **● Русский** | ● Установить флажок |

**Справочники**

| **№** | **Название объекта** | **Инструкция** |
| --- | --- | --- |
| **Подсистема «Бизнес-процессы и задачи»** | | |
| 6. | **● РолиИсполнителей**, свойство **Предопределенные** | ● Установить **Режим объединения…**в **Объединить с приоритетом новой конфигурации поставщика** |
| **Подсистема «Контактная информация»** | | |
| 7. | **● ВидыКонтактнойИнформации**, свойство **Предопределенные** | ● Установить **Режим объединения…**в **Объединить с приоритетом новой конфигурации поставщика** |
| **Подсистема «Пользователи»** | | |
| 8. | **ВнешниеПользователи**, команда**ВнешнийДоступ** свойство **Тип параметра команды** | ● Снять флажок |
| **Подсистема «Свойства»** | | |
| 9. | **●**Справочник**НаборыДополнительныхРеквизитовИСведений**, свойство **Предопределенные** | ● Установить **Режим объединения…**в **Объединить с приоритетом новой конфигурации поставщика** |
| **Подсистема «Управление доступом»** | | |
| 10. | **● ПрофилиГруппДоступа**, свойство **Предопределенные** | ● Установить **Режим объединения…**в **Объединить с приоритетом новой конфигурации поставщика** |

**Перечисления**

| **№** | **Название объекта** | **Инструкция** |
| --- | --- | --- |
| **Подсистема «Отправка SMS»** | | |
| 1. | **● ПровайдерыSMS** | ● Снять флажок |

**Планы видов характеристик**

| **№** | **Название объекта** | **Инструкция** |
| --- | --- | --- |
| **Подсистема «Анкетирование»** | | |
| 1. | **● ВопросыДляАнкетирования**, свойство **Тип** | ● Установить **Режим объединения…**в **Объединить с приоритетом новой конфигурации поставщика** |
| **Подсистема «Бизнес-процессы и задачи»** | | |
| 2. | **● ОбъектыАдресацииЗадач**, свойства **Предопределенные**и **Тип** | ● Установить **Режим объединения…**в **Объединить с приоритетом новой конфигурации поставщика** |
| **Подсистема «Даты запрета изменения»** | | |
| 3. | **● РазделыДатЗапретаИзменения**, свойства **Предопределенные** и**Тип** | ● Установить **Режим объединения…**в **Объединить с приоритетом новой конфигурации поставщика** |
| **Подсистема «Свойства»** | | |
| 4. | **● ДополнительныеРеквизитыИСведения**, свойства **Предопределенные** и**Тип** | ● Установить **Режим объединения…**в **Объединить с приоритетом новой конфигурации поставщика** |

**Обработки**

| **№** | **Название объекта** | **Инструкция** |
| --- | --- | --- |
| **Подсистема «Настройки программы»** | | |
| 1. | **● ПанельАдминистрированияБСП** | ● Если внесены изменения в формы, то снять флажок |

**Бизнес-процессы**

| **№** | **Название объекта** | **Инструкция** |
| --- | --- | --- |
| **Подсистема «Бизнес-процессы и задачи»** | | |
| 1. | **● Задание**, свойство **Вводится на основании** | ● Снять флажок |

# 2.2. Настройка объектов библиотеки

После завершения сравнения и объединения объекты метаданных библиотеки перенесены в конфигурацию, но еще не настроены. Для настройки перенесенных объектов библиотеки нужно выполнить инструкции согласно таблице 2.7.

В том случае, если действия при первом внедрении библиотеки и при обновлении ее версии различаются, инструкции в таблице 2.7 явно делятся на части: «При первом внедрении» и «При обновлении версии библиотеки». Особого внимания требуют модули корневого объекта конфигурации и переопределяемые общие модули, так как автоматическое обновление таких узких мест конфигурации невозможно. После таблицы 2.7 также приведены общие инструкции по первоначальной настройке и обновлению объектов библиотеки в конфигурации.

Таблица 2.7. Объекты библиотеки, которые требуют дополнительных действий после сравнения и объединения конфигурации с библиотекой

**Общие объекты «Библиотеки стандартных подсистем»**

| **№** | **Название объекта** | **Инструкция** |
| --- | --- | --- |
| 1. | ● Модуль сеанса | При первом внедрении:  ● Перенести целиком обработчик **УстановкаПараметровСеанса** |
| 2. | ● Модуль управляемого приложения.  ● Модуль обычного приложения | При первом внедрении в обработчики **ПередНачаломРаботыСистемы**, **ПриНачалеРаботыСистемы** и **ПередЗавершениемРаботыСистемы** перенести блоки кода, отмеченные комментарием:  // СтандартныеПодсистемы  …  // Конец СтандартныеПодсистемы  Также перенести фрагменты кода из области определения глобальных переменных.  При обновлении версии библиотеки перенести из файла поставки библиотеки блоки кода по внедряемым подсистемам в обработчиках **ПередНачаломРаботыСистемы**, **ПриНачалеРаботыСистемы**, **ПередЗавершениемРаботыСистемы** и области определения глобальных переменных:  // <ПутьПодсистемы>  …  // Конец <ПутьПодсистемы> |
| 3. | Свойство **Версия** | При первом внедрении установить в соответствии с шаблоном Р.П.З.С, где:  Р – номер редакции (минимум 1 цифра);  П – номер подредакции (минимум 1 цифра);  З – номер версии (минимум 1 цифра);  С – номер сборки (минимум 1 цифра).  Например, 1.0.0.1. Подробнее см. статью на ИТС [Нумерация редакций и версий](https://its.1c.ru/db/v8std/content/-2145783164/hdoc).  При каждом обновлении увеличивать версии. |

**Общие объекты «Библиотеки стандартных подсистем» для конфигураций, рассчитанных на работу в модели сервиса**

| **№** | **Название объекта** | **Инструкция** |
| --- | --- | --- |
| 1. | Общий реквизит **ОбластьДанныхОсновныеДанные** | При первом внедрении:  ● Исключить из состава общего реквизита все объекты конечной конфигурации, по которым предполагается общее наполнение для всех областей.  При каждом обновлении:  ● Открыть состав общего реквизита и для всех объектов библиотеки установить свойство **Использование** в значение, указанное в файле поставки библиотеки |
| 2. | Общий реквизит **ОбластьДанныхВспомогательныеДанные** | При первом внедрении:  ● Исключить из состава общего реквизита все объекты конечной конфигурации, к которым требуется доступ из сеансов с неустановленным использованием разделения.  При каждом обновлении:  ● Открыть состав общего реквизита и для всех объектов библиотеки установить свойство **Использование** в значение, указанное в файле поставки библиотеки |
| 3. | Общие модули:  ● **ОбменСообщениямиПереопределяемый**.  ● **ОчередьЗаданийПереопределяемый**.  ● **РаботаВМоделиСервисаПереопределяемый** | Рекомендуется придерживаться [общего подхода по настройке переопределяемых общих модулей](https://its.1c.ru/db/content/bsp23doc/src/2.2.%20%D0%BD%D0%B0%D1%81%D1%82%D1%80%D0%BE%D0%B9%D0%BA%D0%B0%20%D0%BE%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BE%D0%B2%20%D0%B1%D0%B8%D0%B1%D0%BB%D0%B8%D0%BE%D1%82%D0%B5%D0%BA%D0%B8.htm?_=1504708101#_настройка_и_обновление) |
| 4. | Функциональная опция **ИспользоватьРазделениеПоОбластямДанных** | При первом внедрении:  ● Определить состав объектов конфигурации, которые должны быть размещены в пользовательском интерфейсе только при работе в модели сервиса (и не должны при работе в локальном режиме).  ● Включить эти объекты в состав функциональной опции |
| 5. | Функциональная опция **НеИспользоватьРазделениеПоОбластямДанных** | При первом внедрении:  ● Определить состав объектов конфигурации, которые должны быть размещены в пользовательском интерфейсе только при работе в локальном режиме (и не должны при работе в модели сервиса).  ● Включить эти объекты в состав функциональной опции |
| 6. | Общие команды **ВыгрузитьДанныеВЛокальнуюВерсию**,**ВыгрузитьДанныеДляПереходаВСервис**, **ДобавитьПользователейСервиса**, **ЗагрузитьДанныеВОбласть**,**ЗагрузитьДанныеИзСервиса**, **СоздатьРезервнуюКопию** | Разместить в командном интерфейсе конфигурации |
| 7. | Общая форма **НастройкиРезервногоКопированияПриложений** | Разместить в командном интерфейсе конфигурации |

**Общие модули**

Рекомендуется придерживаться [общего подхода по настройке переопределяемых общих модулей](https://its.1c.ru/db/content/bsp23doc/src/2.2.%20%D0%BD%D0%B0%D1%81%D1%82%D1%80%D0%BE%D0%B9%D0%BA%D0%B0%20%D0%BE%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BE%D0%B2%20%D0%B1%D0%B8%D0%B1%D0%BB%D0%B8%D0%BE%D1%82%D0%B5%D0%BA%D0%B8.htm?_=1504708101#_настройка_и_обновление).

| **№** | **Название объекта** | **Инструкция** |
| --- | --- | --- |
| **Подсистема «Базовая функциональность»** | | |
| 1. | ● **ОбщегоНазначенияВызовСервераПереопределяемый**.  **● ОбщегоНазначенияПереопределяемый** | См. также дополнительные инструкции по настройке в разделе «[Базовая функциональность](https://its.1c.ru/db/content/bsp23doc/src/3.4.%20%D0%B1%D0%B0%D0%B7%D0%BE%D0%B2%D0%B0%D1%8F%20%D1%84%D1%83%D0%BD%D0%BA%D1%86%D0%B8%D0%BE%D0%BD%D0%B0%D0%BB%D1%8C%D0%BD%D0%BE%D1%81%D1%82%D1%8C.htm?_=1504708101#_базовая_функциональность)» |
| **Подсистема «Бизнес-процессы и задачи»** | | |
| 2. | **● БизнесПроцессыИЗадачиПереопределяемый**.  ● **БизнесПроцессыИЗадачиКлиентПереопределяемый** | |
| **Подсистема «Варианты отчетов»** | | |
| 3. | ● **ВариантыОтчетовПереопределяемый**.  ● **ОтчетыКлиентПереопределяемый**.  ● **ОтчетыПереопределяемый** | См. также дополнительные инструкции по настройке в разделе «[Варианты отчетов](https://its.1c.ru/db/content/bsp23doc/src/3.8.%20%D0%B2%D0%B0%D1%80%D0%B8%D0%B0%D0%BD%D1%82%D1%8B%20%D0%BE%D1%82%D1%87%D0%B5%D1%82%D0%BE%D0%B2.htm?_=1504708101#_варианты_отчетов)» |
| **Подсистема «Взаимодействия»** | | |
| 4. | ● **ВзаимодействияКлиентПереопределяемый**.  ● **ВзаимодействияКлиентСерверПовтИспПереопределяемый**.  ● **ВзаимодействияПереопределяемый** | См. также дополнительные инструкции по настройке в разделе «[Взаимодействия](https://its.1c.ru/db/content/bsp23doc/src/3.10.%20%D0%B2%D0%B7%D0%B0%D0%B8%D0%BC%D0%BE%D0%B4%D0%B5%D0%B9%D1%81%D1%82%D0%B2%D0%B8%D1%8F.htm?_=1504708101#_взаимодействия)» |
| **Подсистема «Групповое изменение объектов»** | | |
| 5. | ● **ГрупповоеИзменениеОбъектовПереопределяемый** | |
| **Подсистема «Даты запрета изменения»** | | |
| 6. | ● **ДатыЗапретаИзмененияПереопределяемый** | |
| **Подсистема «Дополнительные отчеты и обработки»** | | |
| 7. | ● **ДополнительныеОтчетыИОбработкиПереопределяемый** | См. также дополнительные инструкции по настройке в разделе «[Дополнительные отчеты и обработки](https://its.1c.ru/db/content/bsp23doc/src/3.14.%20%D0%B4%D0%BE%D0%BF%D0%BE%D0%BB%D0%BD%D0%B8%D1%82%D0%B5%D0%BB%D1%8C%D0%BD%D1%8B%D0%B5%20%D0%BE%D1%82%D1%87%D0%B5%D1%82%D1%8B%20%D0%B8%20%D0%BE%D0%B1%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%BA%D0%B8.htm?_=1504708101#_дополнительные_отчеты_и)» |
| **Подсистема «Заполнение объектов»** | | |
| 8. | ● **ЗаполнениеОбъектовПереопределяемый** | |
| **Подсистема «Запрет редактирования реквизитов объектов»** | | |
| 9. | ● **ЗапретРедактированияРеквизитовОбъектовПереопределяемый** | |
| **Подсистема «Защита персональных данных»** | | |
| 10. | ● **ЗащитаПерсональныхДанныхПереопределяемый** | Описать заполнение полей, необходимых для печатной формы согласия. Описать заполнение таблиц:  ● **Сведения о персональных данных**.  **● Области персональных данных** |
| **Подсистема «Информация при запуске»** | | |
| 11. | ● **ИнформацияПриЗапускеПереопределяемый** | |
| **Подсистема «Напоминания пользователя»** | | |
| 12. | ● **НапоминанияПользователяПереопределяемый** | |
| **Подсистема «Обмен данными»** | | |
| 13. | **● ОбменДаннымиПереопределяемый** | См. также дополнительные инструкции по настройке в разделе «[Обмен данными](https://its.1c.ru/db/content/bsp23doc/src/3.28.%20%D0%BE%D0%B1%D0%BC%D0%B5%D0%BD%20%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D0%BC%D0%B8.htm?_=1504708101#_обмен_данными)» |
| **Подсистема «Обновление версии ИБ»** | | |
| 14. | ●**ОбновлениеИнформационнойБазыКлиентПереопределяемый**.  **● ОбновлениеИнформационнойБазыПереопределяемый** | См. также дополнительные инструкции по настройке в разделе «[Обновление версии ИБ](https://its.1c.ru/db/content/bsp23doc/src/3.28.%20%D0%BE%D0%B1%D0%BC%D0%B5%D0%BD%20%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D0%BC%D0%B8.htm?_=1504708101#_обновление_версии_иб)» |
| **Подсистема «Отправка SMS»** | | |
| 15. 15. | ● **ОтправкаSMSПереопределяемый** | См. также дополнительные инструкции по настройке в разделе «[Обновление конфигурации](https://its.1c.ru/db/content/bsp23doc/src/3.29.%20%D0%BE%D0%B1%D0%BD%D0%BE%D0%B2%D0%BB%D0%B5%D0%BD%D0%B8%D0%B5%20%D0%B2%D0%B5%D1%80%D1%81%D0%B8%D0%B8%20%D0%B8%D0%B1.htm?_=1504708101#_обновление_конфигурации)» |
| **Подсистема «Печать»** | | |
| 16. | **● УправлениеПечатьюПереопределяемый** | См. также дополнительные инструкции по настройке в разделе «[Печать](https://its.1c.ru/db/content/bsp23doc/src/3.33.%20%D0%BF%D0%B5%D1%87%D0%B0%D1%82%D1%8C.htm?_=1504708101#_печать)» |
| **Подсистема «Поиск и удаление дублей»** | | |
| 17. | ● **ПоискИУдалениеДублейПереопределяемый** | |
| **Подсистема «Пользователи»** | | |
| 18. | **● ПользователиПереопределяемый** | См. также дополнительные инструкции по настройке в разделе «[Пользователи](https://its.1c.ru/db/content/bsp23doc/src/3.37.%20%D0%BF%D0%BE%D0%BB%D1%8C%D0%B7%D0%BE%D0%B2%D0%B0%D1%82%D0%B5%D0%BB%D0%B8.htm?_=1504708101#_пользователи)» |
| **Подсистема «Префиксация объектов»** | | |
| 19. | **● ПрефиксацияОбъектовКлиентСерверПереопределяемый** | См. также дополнительные инструкции по настройке в разделе «[Префиксация объектов](https://its.1c.ru/db/content/bsp23doc/src/3.37.%20%D0%BF%D0%BE%D0%BB%D1%8C%D0%B7%D0%BE%D0%B2%D0%B0%D1%82%D0%B5%D0%BB%D0%B8.htm?_=1504708101#_префиксация_объектов)» |
| **Подсистема «Работа с контрагентами»** | | |
| 20. | ● **РаботаСКонтрагентамиКлиентПереопределяемый**.  ● **РаботаСКонтрагентамиКлиентСерверПереопределяемый**.  **● РаботаСКонтрагентамиПереопределяемый** | См. также дополнительные инструкции по настройке в разделе «[Работа с контрагентами](https://its.1c.ru/db/content/bsp23doc/src/3.42.%20%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%B0%20%D1%81%20%D0%BA%D0%BE%D0%BD%D1%82%D1%80%D0%B0%D0%B3%D0%B5%D0%BD%D1%82%D0%B0%D0%BC%D0%B8.htm?_=1504708101#_работа_с_контрагентами)» |
| **Подсистема «Работа с почтовыми сообщениями»** | | |
| 21. | **● ЭлектроннаяПочтаКлиентПереопределяемый** | См. также дополнительные инструкции по настройке в разделе «[Работа с почтовыми сообщениями](https://its.1c.ru/db/content/bsp23doc/src/3.43.%20%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%B0%20%D1%81%20%D0%BF%D0%BE%D1%87%D1%82%D0%BE%D0%B2%D1%8B%D0%BC%D0%B8%20%D1%81%D0%BE%D0%BE%D0%B1%D1%89%D0%B5%D0%BD%D0%B8%D1%8F%D0%BC%D0%B8.htm?_=1504708101#_работа_с_почтовыми)» |
| **Подсистема «Рассылка отчетов»** | | |
| 22. | ● **РассылкаОтчетовПереопределяемый** | См. также дополнительные инструкции по настройке в разделе «[Рассылка отчетов](https://its.1c.ru/db/content/bsp23doc/src/3.45.%20%D1%80%D0%B0%D1%81%D1%81%D1%8B%D0%BB%D0%BA%D0%B0%20%D0%BE%D1%82%D1%87%D0%B5%D1%82%D0%BE%D0%B2.htm?_=1504708101#_рассылка_отчетов)» |
| **Подсистема «Свойства»** | | |
| 23. | ● **УправлениеСвойствамиПереопределяемый** | См. также дополнительные инструкции по настройке в разделе «[Свойства](https://its.1c.ru/db/content/bsp23doc/src/3.48.%20%D1%81%D0%B2%D0%BE%D0%B9%D1%81%D1%82%D0%B2%D0%B0.htm?_=1504708101#_свойства)» |
| **Подсистема «Структура подчиненности»** | | |
| 24. | ● **СтруктураПодчиненностиПереопределяемый** | См. также дополнительные инструкции по настройке в разделе «[Структура подчиненности](https://its.1c.ru/db/content/bsp23doc/src/3.49.%20%D1%81%D0%BA%D0%BB%D0%BE%D0%BD%D0%B5%D0%BD%D0%B8%D0%B5%20%D0%BF%D1%80%D0%B5%D0%B4%D1%81%D1%82%D0%B0%D0%B2%D0%BB%D0%B5%D0%BD%D0%B8%D0%B9%20%D0%BE%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BE%D0%B2.htm?_=1504708101#_структура_подчиненности)» |
| **Подсистема «Текущие дела»** | | |
| 25. | ● **ТекущиеДелаПереопределяемый** | См. также дополнительные инструкции по настройке в разделе «[Текущие дела](https://its.1c.ru/db/content/bsp23doc/src/3.51.%20%D1%82%D0%B5%D0%BA%D1%83%D1%89%D0%B8%D0%B5%20%D0%B4%D0%B5%D0%BB%D0%B0.htm?_=1504708101#_текущие_дела)» |
| **Подсистема «Управление доступом»** | | |
| 26. | ● **УправлениеДоступомПереопределяемый** | См. также дополнительные инструкции по настройке в разделе «[Управление доступом](https://its.1c.ru/db/content/bsp23doc/src/3.53.%20%D1%83%D0%BF%D1%80%D0%B0%D0%B2%D0%BB%D0%B5%D0%BD%D0%B8%D0%B5%20%D0%B4%D0%BE%D1%81%D1%82%D1%83%D0%BF%D0%BE%D0%BC.htm?_=1504708101#_управление_доступом)» |
| **Подсистема «Файловые функции»** | | |
| 27. | ● **ФайловыеФункцииПереопределяемый** | См. также дополнительные инструкции по настройке в разделе «[Файловые функции](https://its.1c.ru/db/content/bsp23doc/src/3.55.%20%D1%84%D0%B0%D0%B9%D0%BB%D0%BE%D0%B2%D1%8B%D0%B5%20%D1%84%D1%83%D0%BD%D0%BA%D1%86%D0%B8%D0%B8.htm?_=1504708101#_файловые_функции)» |
| **Подсистема «Центр мониторинга»** | | |
| 28. | ● **ЦентрМониторингаПереопределяемый** | |
| **Подсистема «Файловые функции»** | | |
| 29. | ● **ШаблоныСообщенийПереопределяемый** | См. также дополнительные инструкции по настройке в разделе «Шаблоны сообщений» |

**Роли**

| **№** | **Название объекта** | **Инструкция** |
| --- | --- | --- |
| **Подсистема «Базовая функциональность»** | | |
| 1. | ● **БазовыеПрава**.  ●**БазовыеПраваВнешнегоПользователя** | При внедрении совместно с подсистемой «Свойства», но без подсистемы «Управление доступом» выполнить инструкции раздела «[Внедрение подсистем "Свойства" и "Управление доступом"](https://its.1c.ru/db/content/bsp23doc/src/3.59.%20%D0%BE%D1%81%D0%BE%D0%B1%D1%8B%D0%B5%20%D1%81%D0%BB%D1%83%D1%87%D0%B0%D0%B8%20%D0%B2%D0%BD%D0%B5%D0%B4%D1%80%D0%B5%D0%BD%D0%B8%D1%8F%20%D0%BF%D0%BE%D0%B4%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC.htm?_=1504708101#_внедрение_подсистем_)» |
| 2. 3. | ● **ПолныеПрава** | Со всех объектов метаданных снять права:  ● **Право интерактивного удаления**,  ● **Интерактивное удаление предопределенных данных**,  ● **Интерактивная пометка удаления предопределенных данных**,  ● **Интерактивное снятие пометки удаления предопределенных данных**,  ● **Интерактивное удаление помеченных предопределенных данных**.  При внедрении совместно с подсистемой «Работа в модели сервиса» включить в роли права ко всем объектам, входящим в состав общего реквизита-разделителя **ОбластьДанныхОсновныеДанные**.  При внедрении без подсистемы «Работа в модели сервиса» включить в роль все права, кроме перечисленных выше, на все объекты метаданных конфигурации |
| **Подсистема «Свойства»** | | |
| 3. | ●**ИзменениеДополнительныхСведений**.  **● ЧтениеДополнительныхСведений** | При внедрении в конфигурацию без подсистемы «Управление доступом» выполнить инструкции раздела «[Внедрение подсистем "Свойства" и "Управление доступом"](https://its.1c.ru/db/content/bsp23doc/src/3.59.%20%D0%BE%D1%81%D0%BE%D0%B1%D1%8B%D0%B5%20%D1%81%D0%BB%D1%83%D1%87%D0%B0%D0%B8%20%D0%B2%D0%BD%D0%B5%D0%B4%D1%80%D0%B5%D0%BD%D0%B8%D1%8F%20%D0%BF%D0%BE%D0%B4%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC.htm?_=1504708101#_внедрение_подсистем_)» |
| **Подсистема «Защита персональных данных»** | | |
| 4. | **● Создать роли** | Реализовать роли, предоставляющие доступ к объектам:  **●**документ **СогласиеНаОбработкуПерсональныхДанных**,  **●**документ **ОтзывСогласияНаОбработкуПерсональныхДанных**,  ● регистр сведений **СогласияНаОбработкуПерсональныхДанных**.  См. пример роли **\_ДемоДобавлениеИзменениеСогласийНаОбработкуПерсональныхДанных**и **\_ДемоЧтениеСогласийНаОбработкуПерсональныхДанных** |

**Критерии отбора**

| **№** | **Название объекта** | **Инструкция** |
| --- | --- | --- |
| **Подсистема «Структура подчиненности»** | | |
| 1. | ●**СвязанныеДокументы** | При первом внедрении, определив свойство **Состав**, установить возможные типы родительских документов, справочников и ПВХ в свойстве **Тип** и определить состав реквизитов дочерних документов, справочников и ПВХ, в которых будет происходить поиск родительских объектов |

**Регламентные задания**

| **№** | **Название объекта** | **Инструкция** |
| --- | --- | --- |
| **Подсистема «Управление итогами и обработками»** | | |
| 2. | ● **ОбновлениеАгрегатов**.  ● **ПерестроениеАгрегатов**.  **●УстановкаПериодаРассчитанныхИтогов**.  Свойство **Расписание** | При первом внедрении определить расписание регламентных заданий. При каждом обновлении не переносить расписание из поставки |

**Определяемые типы**

| **№** | **Название объекта** | **Инструкция** |
| --- | --- | --- |
| **Подсистема «Анкетирование»** | | |
| 1. | **● Респондент** | При первом внедрении:  ● Принять решение по поводу состава объектов конфигурации, которые должны быть сопоставлены респондентам анкет. Например, справочники **Респонденты**, **Сотрудники**, **Партнеры**.  ● Указать допустимые типы респондентов в составе типов свойства **Тип** |
| **Подсистема «Базовая функциональность»** | | |
| 2. | **● Организация** | Если в конфигурации имеется справочник **Организации**, при первом внедрении указать в свойстве **Тип**ссылку на него |
| **Подсистема «Бизнес-процессы и задачи»** | | |
| 3. | ● **БизнесПроцесс** | При первом внедрении указать в свойстве **Тип**ссылки на все бизнес-процессы в конфигурации |
| 4. | ● **ПредметЗадачи** | При первом внедрении:  ● Принять решение по поводу состава объектов конфигурации, которые должны быть основанием для бизнес-процесса **Задание** или других бизнес-процессов (определить список предметов задач).  ● Указать в свойстве **Тип**ссылки на предметы задач |
| 5. | ● **ОбъектАдресации** | При первом внедрении:  ● На основании полученных ролей исполнителей составить список объектов, которые должны иметь ролевую адресацию (т. е. определить «объекты адресации» – см. раздел «[Бизнес-процессы и задачи](https://its.1c.ru/db/content/bsp23doc/src/3.6.%20%D0%B1%D0%B8%D0%B7%D0%BD%D0%B5%D1%81-%D0%BF%D1%80%D0%BE%D1%86%D0%B5%D1%81%D1%81%D1%8B%20%D0%B8%20%D0%B7%D0%B0%D0%B4%D0%B0%D1%87%D0%B8.htm?_=1504708101#_бизнес-процессы_и_задачи)»).  ● Указать в свойстве **Тип**ссылки на «объекты адресации» |
| 6. | ● **ОтложенныеБизнесПроцессы** | При первом внедрении:  ● Принять решение по поводу состава бизнес-процессов, которые могут быть запущены отложенно.  ● Указать в свойстве **Тип** ссылки на данные бизнес-процессы |
| **Подсистема «Версионирование объектов»** | | |
| 7. | **● ВерсионируемыеДанные**.  ● **ВерсионируемыеДанныеОбъект** | При первом внедрении провести настройку состава типа согласно инструкции в разделе «[Версионирование объектов](https://its.1c.ru/db/content/bsp23doc/src/3.8.%20%D0%B2%D0%B0%D1%80%D0%B8%D0%B0%D0%BD%D1%82%D1%8B%20%D0%BE%D1%82%D1%87%D0%B5%D1%82%D0%BE%D0%B2.htm?_=1504708101#_версионирование_объектов)» |
| **Подсистема «Взаимодействия»** | | |
| 8. | ● **КонтактВзаимодействия** | При первом и последующих внедрениях:  ● Принять решение по поводу состава объектов конфигурации, которые должны указываться в качестве контактов взаимодействий.  ● Добавить допустимые типы контактов в состав типов свойства **Тип** |
| 9. | ● **ПредметВзаимодействия** | При первом и последующих внедрениях:  ● Принять решение по поводу состава объектов конфигурации, которые должны указываться в качестве предметов взаимодействий.  ● Добавить допустимые типы предметов в состав типов свойства **Тип** |
| **Подсистема «Даты запрета изменения»** | | |
| 10. | ● **АдресатЗапретаИзменения** | При первом внедрении:  ● Принять решение по поводу состава планов обмена, для которых администратор будет настраивать даты запрета загрузки, и добавить их типы в состав типов реквизита.  См. также дополнительные инструкции по настройке в разделе «[Даты запрета изменения](https://its.1c.ru/db/content/bsp23doc/src/3.13.%20%D0%B4%D0%B0%D1%82%D1%8B%20%D0%B7%D0%B0%D0%BF%D1%80%D0%B5%D1%82%D0%B0%20%D0%B8%D0%B7%D0%BC%D0%B5%D0%BD%D0%B5%D0%BD%D0%B8%D1%8F.htm?_=1504708101#_даты_запрета_изменения)» |
| **Подсистема «Дополнительные отчеты и обработки»** | | |
| 11. | ● **ОбъектСДополнительнымиКомандами** | При первом внедрении:  ● Принять решение по поводу состава объектов конфигурации, с которыми необходимо использовать дополнительные отчеты и обработки различных видов.  ● Перечислить типы этих объектов в свойстве **Тип**.  Подробную инструкцию по настройке см. в разделе «[Дополнительные отчеты и обработки](https://its.1c.ru/db/content/bsp23doc/src/3.14.%20%D0%B4%D0%BE%D0%BF%D0%BE%D0%BB%D0%BD%D0%B8%D1%82%D0%B5%D0%BB%D1%8C%D0%BD%D1%8B%D0%B5%20%D0%BE%D1%82%D1%87%D0%B5%D1%82%D1%8B%20%D0%B8%20%D0%BE%D0%B1%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%BA%D0%B8.htm?_=1504708101#_дополнительные_отчеты_и)» |
| **Подсистема «Заметки пользователя»** | | |
| 12. | **● ПредметЗаметок**.  ● **ПредметЗаметокОбъект** | ● Определить состав объектов конфигурации, по поводу которых пользователи должны иметь возможность вводить заметки.  ● Отметить выбранные типы объектов в свойстве **Тип**.  Подробную инструкцию по настройке см. в разделе «[Заметки пользователя](https://its.1c.ru/db/content/bsp23doc/src/3.17.%20%D0%B7%D0%B0%D0%BC%D0%B5%D1%82%D0%BA%D0%B8%20%D0%BF%D0%BE%D0%BB%D1%8C%D0%B7%D0%BE%D0%B2%D0%B0%D1%82%D0%B5%D0%BB%D1%8F.htm?_=1504708101#_заметки_пользователя)» |
| **Подсистема «Защита персональных данных»** | | |
| 13. | **● СубъектПерсональныхДанных** | При первом внедрении заполнить состав типов определяемого типа **СубъектПерсональныхДанных**. В качестве типа выбрать сущность, являющуюся субъектом персональных данных в конфигурации |
| **Подсистема «Контактная информация»** | | |
| 14. | **● ВладелецКонтактнойИнформации** | При первом внедрении:  ● Принять решение по поводу состава объектов конфигурации, которые должны содержать контактную информацию.  ● Перечислить типы этих объектов в свойстве **Тип**.  Инструкцию по настройке см. в разделе «[Контактная информация](https://its.1c.ru/db/content/bsp23doc/src/3.24.%20%D0%BA%D0%BE%D0%BD%D1%82%D0%B0%D0%BA%D1%82%D0%BD%D0%B0%D1%8F%20%D0%B8%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%86%D0%B8%D1%8F.htm?_=1504708101#_контактная_информация)» |
| **Подсистема «Напоминания пользователя»** | | |
| 15. | **● ПредметНапоминания**.  **● ПредметНапоминанияОбъект** | ● Определить состав объектов ссылочного типа, для которых требуется ввод пользовательских напоминаний.  ● Отметить выбранные типы объектов в свойстве **Тип.**  Подробную инструкцию по настройке см. в разделе «[Напоминания пользователя](https://its.1c.ru/db/content/bsp23doc/src/3.24.%20%D0%BA%D0%BE%D0%BD%D1%82%D0%B0%D0%BA%D1%82%D0%BD%D0%B0%D1%8F%20%D0%B8%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%86%D0%B8%D1%8F.htm?_=1504708101#_напоминания_пользователя)» |
| **Подсистема «Настройка порядка элементов»** | | |
| 16. | **● ОбъектСНастраиваемымПорядком** | При первом внедрении:  ● Принять решение по поводу состава объектов конфигурации, для которых необходимо настраивать порядок элементов.  ● Перечислить типы этих объектов в свойстве **Тип**.  ● Подробную инструкцию по настройке см. в разделе «[Настройка](https://its.1c.ru/db/content/bsp23doc/src/3.26.%20%D0%BD%D0%B0%D1%81%D1%82%D1%80%D0%BE%D0%B9%D0%BA%D0%B0%20%D0%BF%D0%BE%D1%80%D1%8F%D0%B4%D0%BA%D0%B0%20%D1%8D%D0%BB%D0%B5%D0%BC%D0%B5%D0%BD%D1%82%D0%BE%D0%B2.htm?_=1504708101#_настройка_порядка_элементов)порядка элементов» |
| **Подсистема «Пользователи»** | | |
| 17. | **● Подразделение**.  **● ФизическоеЛицо** | При первом внедрении:  ● Принять решение о необходимости использования реквизитов **Подразделение** и **ФизическоеЛицо**справочника **Пользователи**.  ● Указать ссылки на соответствующие справочники в определяемых типах |
| 18. | **● ВнешнийПользователь**.  **● ВнешнийПользовательОбъект**.  **● Пользователь** | При первом внедрении:  ● Принять решение по поводу состава объектов конфигурации, которые должны быть сопоставлены внешним пользователям системы. Например, справочники **Физические лица**, **Сотрудники**, **Партнеры**.  ● Перечислить допустимые типы внешних пользователей в составе типов (в состав типа **Пользователь**также входит тип **СправочникСсылка.Пользователи**) |
| **Подсистема «Присоединенные файлы»** | | |
| 19. | ● **ПрисоединенныйФайл**.  ● **ПрисоединенныйФайлОбъект**  ● **ВладелецПрисоединенныхФайлов**.  **● ВладелецПрисоединенныхФайловОбъект** | При первом внедрении:  ● Принять решение по поводу состава объектов конфигурации, которые должны содержать присоединенные файлы («объекты с файлами»).  ● Перечислить типы этих объектов, а также типы справочников файлов в свойстве **Тип**.  Инструкцию по настройке см. в разделе «[Присоединенные файлы](https://its.1c.ru/db/content/bsp23doc/src/3.39.%20%D0%BF%D1%80%D0%B8%D1%81%D0%BE%D0%B5%D0%B4%D0%B8%D0%BD%D0%B5%D0%BD%D0%BD%D1%8B%D0%B5%20%D1%84%D0%B0%D0%B9%D0%BB%D1%8B.htm?_=1504708101#_присоединенные_файлы)» |
| **Подсистема «Работа с контрагентами»** | | |
| 20. | ● **Контрагент**.  ● **КонтрагентОбъект** | При первом внедрении:  ● Указать тип справочника контрагентов в свойстве **Тип** |
| **Подсистема «Работа с файлами»** | | |
| 21. | ● **ВладелецФайлов**.  **● ВладелецФайлов Объект** | При первом внедрении:  ● Принять решение по поводу состава объектов конфигурации, которые должны содержать присоединенные файлы («объекты с файлами»).  ● Перечислить типы этих объектов в свойстве **Тип**.  Инструкцию по настройке см. в разделе «[Работа с файлами](https://its.1c.ru/db/content/bsp23doc/src/3.44.%20%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%B0%20%D1%81%20%D1%84%D0%B0%D0%B9%D0%BB%D0%B0%D0%BC%D0%B8.htm?_=1504708101#_работа_с_файлами)» |
| **Подсистема «Рассылка отчетов»** | | |
| 22. | **● ПолучательРассылки** | При первом внедрении:  ● Определить возможных получателей рассылки. Для этих объектов должна вестись контактная информация о еmail-адресах.  ● Перечислить типы получателей рассылки в свойстве **Тип** |
| **Подсистема «Свойства»** | | |
| 23. | **● ВладелецДополнительныхСведений** | При первом внедрении:  ● Принять решение по поводу состава объектов, в которых должна использоваться подсистема (составить список «объектов с дополнительными сведениями»).  ● Указать в свойстве **Тип**ссылки на объекты, в которых будет использоваться подсистема свойств («объекты с дополнительными сведениями»).  Инструкцию по настройке см. в разделе «[Свойства](https://its.1c.ru/db/content/bsp23doc/src/3.48.%20%D1%81%D0%B2%D0%BE%D0%B9%D1%81%D1%82%D0%B2%D0%B0.htm?_=1504708101#_свойства)» |
| **Подсистема «Управление доступом»** | | |
| 24. | ● **ВладелецНастроекПрав**.  ● **ВладелецНастроекПравОбъект**.  **● ЗначениеДоступа**.  ● **ЗначениеДоступаСГруппамиЗначенийДоступаОбъект**.  ● **ВладелецНаборовЗначенийДоступаОбъект**.  ●**ВладелецСОграничениемПоНаборамЗначенийДоступаОбъект**.  ●**ВладелецВнешнихЗначенийВНаборахЗначенийДоступаОбъект** | При первом внедрении выполнить настройку в соответствии с описанием в разделе «[Управление доступом](https://its.1c.ru/db/content/bsp23doc/src/3.53.%20%D1%83%D0%BF%D1%80%D0%B0%D0%B2%D0%BB%D0%B5%D0%BD%D0%B8%D0%B5%20%D0%B4%D0%BE%D1%81%D1%82%D1%83%D0%BF%D0%BE%D0%BC.htm?_=1504708101#_управление_доступом)».  При каждом обновлении убедиться, что состав типов включает в себя все поставляемые типы. |
| **Подсистема «Шаблона сообщений»** | | |
| 25. | **● ПредметШаблонаСообщения** | При первом и последующих внедрениях:  ● Принять решение по поводу состава объектов конфигурации, которые должны указываться в качестве контактов взаимодействий.  Указать в свойстве **Тип**ссылки на предметы шаблонов сообщений. |

**Общие формы**

| **№** | **Название объекта** | **Инструкция** |
| --- | --- | --- |
| **Подсистема «Отправка SMS»** | | |
| 1. | **● Общая форма НастройкаОтправкиSMS** | При первом внедрении поместить в командный интерфейс |

**Общие команды**

| **№** | **Название объекта** | **Инструкция** |
| --- | --- | --- |
| **Подсистема «Интеграция с "1С-Коннект" ("Бухфон")»** | | |
| 1. | **● Запустить1СБухфон** | При первом внедрении:  ● Принять решение по поводу места размещения команды на рабочем столе и поместить в выбранную форму.  ● Увеличить размеры команды на форме до ширины 16, высоты 3.  ● В процедурах модуля формы прописать вызов процедур подсистемы согласно описанию в главе 3 в разделе «[Интеграция с "1С-Коннект" ("Бухфон")](https://its.1c.ru/db/content/bsp23doc/src/3.21.%20%D0%B8%D0%BD%D1%82%D0%B5%D0%B3%D1%80%D0%B0%D1%86%D0%B8%D1%8F%20%D1%81%20_1%D1%81-%D0%BA%D0%BE%D0%BD%D0%BD%D0%B5%D0%BA%D1%82%20(%D0%B1%D1%83%D1%85%D1%84%D0%BE%D0%BD)_.htm?_=1504708101#_интеграция_с_)» |
| 2. | ● **МастерЗапуска1СБухфон** | При первом внедрении:  ● Разместить на форме настроек пользователя |
| **Подсистема «Обмен данными»** | | |
| 3. | ● **ВыполнитьОбменДанными**.  ● **ВыполнитьОбменДаннымиИнтерактивно**.  ● **НастроитьПараметрыТранспортаСообщенийОбмена**.  ● **ОткрытьПравилаКонвертацииОбъектов**.  ● **ОткрытьПравилаРегистрацииОбъектов**.  ● **ОткрытьСценарииОбменовДанными**.  ●**ПерейтиВЖурналРегистрацииСобытийВыгрузкиДанных**.  ●**ПерейтиВЖурналРегистрацииСобытийЗагрузкиДанных**.  ● **СоставОтправляемыхДанных**.  **● УдалитьНастройкуСинхронизации** | При первом внедрении настроить согласно инструкции по настройке в разделе «[Обмен данными](https://its.1c.ru/db/content/bsp23doc/src/3.28.%20%D0%BE%D0%B1%D0%BC%D0%B5%D0%BD%20%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D0%BC%D0%B8.htm?_=1504708101#_обмен_данными)» |
| **Подсистема «Структура подчиненности»** | | |
| 4. | ● **СтруктураПодчиненности** | При первом внедрении установить свойство **Тип параметра команды** для тех документов, справочников и ПВХ, которые предполагается выводить в отчет **Структура подчиненности** |
| **Подсистема «Управление доступом»** | | |
| 5. | ● **НастроитьПрава**, свойство **Тип параметра команды** | При первом внедрении выполнить настройку в соответствии с описанием в разделе «[Управление доступом](https://its.1c.ru/db/content/bsp23doc/src/3.53.%20%D1%83%D0%BF%D1%80%D0%B0%D0%B2%D0%BB%D0%B5%D0%BD%D0%B8%D0%B5%20%D0%B4%D0%BE%D1%81%D1%82%D1%83%D0%BF%D0%BE%D0%BC.htm?_=1504708101#_управление_доступом)».  При каждом обновлении убедиться, что состав типов включает состав поставляемых типов |

**Общие макеты**

| **№** | **Название объекта** | **Инструкция** |
| --- | --- | --- |
| **Подсистема «Обновление версии ИБ»** | | |
| 1. | ● **ОписаниеИзмененийСистемы** | При первом внедрении создать макет согласно разделу «[Обновление версии ИБ](https://its.1c.ru/db/content/bsp23doc/src/3.28.%20%D0%BE%D0%B1%D0%BC%D0%B5%D0%BD%20%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D0%BC%D0%B8.htm?_=1504708101#_обновление_версии_иб)» и в дальнейшем при разработке конфигурации сопровождать его |
| **Подсистема «Работа с контрагентами»** | | |
| 2. 2. | ●**ИнструкцияПоПроверкеКонтрагентов** | Доработать инструкцию в соответствии с особенностями реализации проверки в конкретном прикладном решении |

**Справочники**

| **№** | **Название объекта** | **Инструкция** |
| --- | --- | --- |
| **Подсистема «Бизнес-процессы и задачи»** | | |
| 1. | ● **РолиИсполнителей** | При первом внедрении создать необходимый набор предопределенных элементов ролей исполнителей бизнес-процессов, предусмотренных в конфигурации. Инструкцию по настройке см. в разделе «[Бизнес-процессы и задачи](https://its.1c.ru/db/content/bsp23doc/src/3.6.%20%D0%B1%D0%B8%D0%B7%D0%BD%D0%B5%D1%81-%D0%BF%D1%80%D0%BE%D1%86%D0%B5%D1%81%D1%81%D1%8B%20%D0%B8%20%D0%B7%D0%B0%D0%B4%D0%B0%D1%87%D0%B8.htm?_=1504708101#_бизнес-процессы_и_задачи)» |
| **Подсистема «Валюты»** | | |
| 2. | ● **Валюты** | При первом внедрении поместить справочник **Валюты** в командный интерфейс. Подробнее см. в разделе «[Валюты](https://its.1c.ru/db/content/bsp23doc/src/3.7.%20%D0%B2%D0%B0%D0%BB%D1%8E%D1%82%D1%8B.htm?_=1504708101#_валюты)» |
| **Подсистема «Контактная информация»** | | |
| 3. | ● **ВидыКонтактнойИнформации** | При первом внедрении, если требуется, создать необходимый набор предопределенных элементов справочника исходя из потребностей конфигурации. Инструкцию по настройке см. в разделе «[Контактная информация](https://its.1c.ru/db/content/bsp23doc/src/3.24.%20%D0%BA%D0%BE%D0%BD%D1%82%D0%B0%D0%BA%D1%82%D0%BD%D0%B0%D1%8F%20%D0%B8%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%86%D0%B8%D1%8F.htm?_=1504708101#_контактная_информация)» |
| **Подсистема «Оценка производительности»** | | |
| 4. | **● КлючевыеОперации** | Если принято решение использовать предопределенные элементы, то при первом внедрении создать необходимый набор предопределенных элементов и произвести первоначальное заполнение реквизитов из формы списка. См. также дополнительные инструкции в разделе «[Оценка производительности](https://its.1c.ru/db/content/bsp23doc/src/3.32.%20%D0%BE%D1%86%D0%B5%D0%BD%D0%BA%D0%B0%20%D0%BF%D1%80%D0%BE%D0%B8%D0%B7%D0%B2%D0%BE%D0%B4%D0%B8%D1%82%D0%B5%D0%BB%D1%8C%D0%BD%D0%BE%D1%81%D1%82%D0%B8.htm?_=1504708101#_оценка_производительности)» |
| **Подсистема «Пользователи»** | | |
| 5. | **● ВнешниеПользователи**, команда **ВнешнийДоступ**, свойство **Тип параметра команды** | При первом внедрении:  ● Принять решение по поводу состава объектов конфигурации, которые должны быть сопоставлены внешним пользователям системы. Например, справочники **Физические лица**, **Сотрудники**, **Партнеры**.  ● Перечислить допустимые типы внешних пользователей в составе типов (в состав типа **Пользователь** также входит тип **СправочникСсылка.Пользователи**) |
| **Подсистема «Свойства»** | | |
| 6. | ●**НаборыДополнительныхРеквизитовИСведений** | При первом внедрении создать необходимый набор предопределенных элементов для каждого «объекта со свойствами» в формате **Тип\_Имя**. Например, **Справочник\_ФизическиеЛица** для справочника **ФизическиеЛица**.  Инструкцию по настройке см. в разделе «[Свойства](https://its.1c.ru/db/content/bsp23doc/src/3.48.%20%D1%81%D0%B2%D0%BE%D0%B9%D1%81%D1%82%D0%B2%D0%B0.htm?_=1504708101#_свойства)» |

**Журналы документов**

| **№** | **Название объекта** | **Инструкция** |
| --- | --- | --- |
| **Подсистема «Защита персональных данных»** | | |
| 1. | ●**СогласияНаОбработкуПерсональныхДанных** | ● Если не используется подсистема **Настройки программы**, то при первом внедрении разместить в командном интерфейсе |

**Отчеты**

| **№** | **Название объекта** | **Инструкция** |
| --- | --- | --- |
| **Подсистема «Защита персональных данных»** | | |
| 1. | **●СогласияНаОбработкуПерсональныхДанныхДействующие**.  ● **СогласияНаОбработкуПерсональныхДанныхИстекающие** | ● Если не используется подсистема **Настройки программы**, то при первом внедрении разместить в командном интерфейсе |

**Обработки**

| **№** | **Название объекта** | **Инструкция** |
| --- | --- | --- |
| **Подсистема «Информация при запуске»** | | |
| 1. | **● ИнформацияПриЗапуске** | Для макетов обработки при первом внедрении:  ● Принять решение по поводу состава страниц с информацией, которые должны выводиться при запуске программы.  ● Загрузить страницы в макеты обработки согласно инструкции к подсистеме |
| **Подсистема «Настройки программы»** | | |
| 2. 2. | **●ПанельАдминистрированияБСП** | При каждом обновлении:  ● Если в форму панели были добавлены свои объекты или внесены изменения, то выполнять сравнение/объединение всех форм обработки для выяснения состава дополнений и изменений в поставке библиотеки |
| **Подсистема «Оценка производительности»** | | |
| 3. | ● **ОценкаПроизводительности** | Если не используется подсистема **Настройки программы**, то при первом внедрении разместить в командном интерфейсе |

**Планы видов характеристик**

| **№** | **Название объекта** | **Инструкция** |
| --- | --- | --- |
| **Подсистема «Анкетирование»** | | |
| 1. | ● **ВопросыДляАнкетирования** | При первом внедрении:  ● Принять решение по поводу состава объектов конфигурации, которые должны указываться в качестве ответов на вопросы анкеты (определить состав «типов ответов»).  ● Добавить допустимые типы ответов в состав типов свойства **Тип значения** |
| **Подсистема «Бизнес-процессы и задачи»** | | |
| 2. | ● **ОбъектыАдресацииЗадач** | При первом внедрении:  ● Указать в свойстве **Тип** плана видов характеристик, а также в свойстве **Тип** предопределенного элемента **ВсеОбъектыАдресации** этого плана видов характеристик все типы «объектов адресации».  ● Создать необходимый набор предопределенных элементов для объектов адресации (инструкцию по настройке см. в разделе «[Бизнес-процессы и задачи](https://its.1c.ru/db/content/bsp23doc/src/3.6.%20%D0%B1%D0%B8%D0%B7%D0%BD%D0%B5%D1%81-%D0%BF%D1%80%D0%BE%D1%86%D0%B5%D1%81%D1%81%D1%8B%20%D0%B8%20%D0%B7%D0%B0%D0%B4%D0%B0%D1%87%D0%B8.htm?_=1504708101#_бизнес-процессы_и_задачи)») |
| **Подсистема «Свойства»** | | |
| 3. | ●**ДополнительныеРеквизитыИСведения** | При первом внедрении указать возможные типы значений свойств. Состав типов данных может быть расширен любыми типами объектов метаданных конфигурации.  Инструкцию по настройке см. в разделе «[Свойства](https://its.1c.ru/db/content/bsp23doc/src/3.48.%20%D1%81%D0%B2%D0%BE%D0%B9%D1%81%D1%82%D0%B2%D0%B0.htm?_=1504708101#_свойства)» |

**Бизнес-процессы**

| **№** | **Название объекта** | **Инструкция** |
| --- | --- | --- |
| **Подсистема «Бизнес-процессы и задачи»** | | |
| 1. | ●**Задание** | При первом внедрении в свойстве **Вводится на основании** указать ссылки на объекты метаданных, по которым необходимо создавать задания (предметы задач). Инструкцию по настройке см. в разделе «[Бизнес-процессы и задачи](https://its.1c.ru/db/content/bsp23doc/src/3.6.%20%D0%B1%D0%B8%D0%B7%D0%BD%D0%B5%D1%81-%D0%BF%D1%80%D0%BE%D1%86%D0%B5%D1%81%D1%81%D1%8B%20%D0%B8%20%D0%B7%D0%B0%D0%B4%D0%B0%D1%87%D0%B8.htm?_=1504708101#_бизнес-процессы_и_задачи)» |

## Настройка и обновление переопределяемых общих модулей библиотеки

Для настройки в конфигурации переопределяемых общих модулей рекомендуется придерживаться общего подхода:

1. **При первой настройке** переопределяемого общего модуля следует ознакомиться с документацией к его экспортным процедурам и функциям, приведенной в комментариях к ним. И при необходимости вписать реализацию в экспортные процедуры и функции модуля.

2. **При каждом последующем обновлении** переопределяемого общего модуля требуется перенести новые экспортные процедуры и функции, удалить устаревшие и убедиться, что комментарии, количество и имена параметров у всех функций совпадают с их библиотечными эквивалентами. При необходимости вписать реализацию в новые экспортные процедуры и функции модуля, а также актуализировать реализацию уже существующих функций, если в новой версии библиотеки было изменено их назначение или состав параметров.

## Удаление лишних связей с неиспользуемыми подсистемами

Некоторые объекты библиотеки могут поставляться по умолчанию уже настроенными для работы с другими подсистемами библиотеки. В этом случае необходимо оставить в них только те блоки кода, которые относятся к внедряемым подсистемам. Блок кода, относящийся к той или иной подсистеме, отмечается начальным и конечным комментариями:

// <ПутьКПодсистеме>

…

// Конец <ПутьКПодсистеме>

Например, если в конфигурацию не была внедрена подсистема «Адресный классификатор», то необходимо удалить все фрагменты кода, отмеченные комментариями:

// СтандартныеПодсистемы.АдресныйКлассификатор

…

// Конец СтандартныеПодсистемы.АдресныйКлассификатор

Или, например, если в конфигурацию не была внедрена подсистема «Обмен сообщениями», то необходимо удалить все фрагменты кода, отмеченные комментариями:

// СтандартныеПодсистемы.РаботаВМоделиСервиса.ОбменСообщениями

…

// Конец СтандартныеПодсистемы.РаботаВМоделиСервиса.ОбменСообщениями

Для массового удаления фрагментов кода подсистем, которые не используются в конфигурации, можно воспользоваться обработкой **ПервоеВнедрениеБСП.epf**. Перед ее запуском может также потребоваться:

* ● закомментировать код модулей управляемого и обычного приложений, а также модуля сеанса – для возможности запуска обработки;
* ● включить возможность изменения в настройках поддержки для библиотечных объектов.

## Особенности обновления на конкретную версию библиотеки

Особенности обновления каждой конкретной версии библиотеки описаны в сопроводительном документе **updateSSL.htm**, входящем в поставку библиотеки. При обновлении через несколько версий необходимо последовательно выполнить все шаги, описанные в этом документе, начиная с самой младшей версии.

## Увеличение номера версии конфигурации

После завершения внедрения (обновления) библиотеки в конфигурацию следует увеличить номер версии конфигурации. Это необходимо для того, чтобы смогли выполниться все процедуры обновления данных информационной базы, предусмотренные в новой версии библиотеки. См. также раздел «[Обновление вспомогательных данных во время разработки](https://its.1c.ru/db/content/bsp23doc/src/3.4.%20%D0%B1%D0%B0%D0%B7%D0%BE%D0%B2%D0%B0%D1%8F%20%D1%84%D1%83%D0%BD%D0%BA%D1%86%D0%B8%D0%BE%D0%BD%D0%B0%D0%BB%D1%8C%D0%BD%D0%BE%D1%81%D1%82%D1%8C.htm?_=1504708101#_обновление_вспомогательных_данных)».

## Контроль корректности внедрения и обновления

Рекомендуется выполнить следующие контрольные процедуры:

1. Проверить конфигурацию с установленным флажком **Расширенная проверка** (конфигуратор – меню **Конфигурация – Проверка конфигурации…**).

2. Запустить конфигурацию в режиме обновления с предыдущей версии, а также в режиме первоначального заполнения (на «пустой» информационной базе).

3. Воспользоваться специальным инструментом – внешней обработкой **ПроверкаВнедренияБСП.erf** из дистрибутива библиотеки.

С помощью этих процедур можно быстро выявить проблемы несовместимости, связанные с объектами библиотеки, которые были переименованы или удалены в новой версии библиотеки, а также различные ошибки из-за человеческого фактора, связанные с ручными операциями при внедрении (обновлении). Кроме того, помощник внедрения позволяет выявить несоответствия между запланированным и фактическим объемом внедрения библиотеки в конфигурацию.

## Внедрение подсистем библиотеки в объекты конфигурации и разработка ролей

Некоторые подсистемы предназначены для внедрения непосредственно в объекты конфигурации-потребителя и поэтому предполагают тесную интеграцию с объектами конфигурации. Для этого в соответствующих разделах главы 3 «[Настройка и использование подсистем при разработке конфигурации» предусмотрены отдельные инструкции](https://its.1c.ru/db/content/bsp23doc/src/%D0%B3%D0%BB%D0%B0%D0%B2%D0%B0%203.%20%D0%BD%D0%B0%D1%81%D1%82%D1%80%D0%BE%D0%B9%D0%BA%D0%B0%20%D0%B8%20%D0%B8%D1%81%D0%BF%D0%BE%D0%BB%D1%8C%D0%B7%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5%20%D0%BF%D0%BE%D0%B4%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%20%D0%BF%D1%80%D0%B8%20%D1%80%D0%B0%D0%B7%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%BA%D0%B5%20%D0%BA%D0%BE%D0%BD%D1%84%D0%B8%D0%B3%D1%83%D1%80%D0%B0%D1%86%D0%B8%D0%B8.htm?_=1504708101#_настройка_и_использование)».

Там же даны рекомендации по настройке прав доступа пользователей. В большинстве случаев подсистемы предоставляют готовый набор ролей к своим объектам. Однако могут быть случаи, когда разработка системы ролей, в том числе настройка ролей конфигурации для объектов, поставляемых библиотекой, остается за разработчиком конфигурации.

## Разработка справочной информации к объектам библиотеки

Большинство объектов библиотеки не содержат справочной информации, поскольку ее содержимое сильно зависит от специфики конфигурации-потребителя. Справочную информацию к объектам библиотеки следует разрабатывать непосредственно в конфигурации.

# 2.3. Использование в конфигурации

Для использования большинства подсистем библиотеки достаточно выполнить инструкции по внедрению и настройке, вывести в командный интерфейс объекты подсистем и выполнить тесную интеграцию с объектами конфигурации.

Также в подсистемах предусмотрен программный интерфейс, который может быть опционально задействован в конфигурации. Описание программного интерфейса содержится в исходных текстах самой библиотеки. Описание дополнительных возможностей подсистем для разработчиков также можно найти в подразделах «[Использование при разработке конфигурации](https://its.1c.ru/db/content/bsp23doc/src/3.2.%20%D0%B0%D0%BD%D0%B0%D0%BB%D0%B8%D0%B7%20%D0%B6%D1%83%D1%80%D0%BD%D0%B0%D0%BB%D0%B0%20%D1%80%D0%B5%D0%B3%D0%B8%D1%81%D1%82%D1%80%D0%B0%D1%86%D0%B8%D0%B8.htm?_=1504708101#_использование_при_разработке)» главы 3.

# 2.4. Быстрое начало разработки «с нуля»

Короткая инструкция для быстрого начала разработки собственной конфигурации на базе «Библиотеки стандартных подсистем» (БСП). В ней описаны только обязательные действия, которые необходимы для первого запуска разрабатываемого решения. Полный список действий см. в [главе 2](https://its.1c.ru/db/content/bsp23doc/src/%D0%B3%D0%BB%D0%B0%D0%B2%D0%B0%202.%20%D0%B8%D0%BD%D1%81%D1%82%D1%80%D1%83%D0%BA%D1%86%D0%B8%D1%8F%20%D0%BF%D0%BE%20%D0%B2%D0%BD%D0%B5%D0%B4%D1%80%D0%B5%D0%BD%D0%B8%D1%8E%20%D0%B1%D0%B8%D0%B1%D0%BB%D0%B8%D0%BE%D1%82%D0%B5%D0%BA%D0%B8.htm?_=1504708101#_инструкция_по_внедрению).

## Подготовка

1. Определиться со списком внедряемых подсистем. В самом простом случае внедряются все подсистемы БСП. При частичном внедрении можно воспользоваться обработкой **ПервоеВнедрениеБСП** из комплекта поставки, которая позволяет выбрать подсистемы для внедрения с учетом их зависимостей друг от друга.

2. Создать новую пустую базу.

## Сравнение, объединение

1. Вызвать меню **Конфигурация** – **Сравнить, объединить с конфигурацией из файла**.

2. Выбрать файл **1Cv8.cf** из поставки БСП.

3. На предложение выполнить полную загрузку конфигурации ответить **Нет**.

4. На предложение постановки на поддержку ответить **Да**.

5. Снять все флажки (снять флажок с корневого элемента).

6. Выполнить **Действия** – **Отметить по подсистемам файла**.

7. Отметить подсистемы, выбранные на шаге 1 раздела **Подготовка** и нажать **Установить**.

8. Отметить к объединению свойства корневого элемента: **Основные роли**, **Модуль управляемого приложения**, **Модуль сеанса**, **Модуль внешнего соединения**, **Модуль обычного приложения**, **Режим использования модальности**, **Режим использования синхронных вызовов расширений платформы и внешних компонент**, **Режим совместимости интерфейса**, **Режим совместимости**. Нажать **Выполнить**.

9. В окне **Настройка правил поддержки** нажать **Ок**.

## Действия после сравнения, объединения

1. Установить в свойствах конфигурации имя конфигурации, например, «МояКонфигурация».

2. Установить в свойствах конфигурации номер версии разрабатываемой конфигурации. Например, при начале разработки конфигурации можно установить номер 1.0.0.1.

3. Скопировать общий модуль ОбновлениеИнформационнойБазы.

4. Заменить в названии скопированного модуля на имя или сокращение имени конфигурации (например, **ОбновлениеИнформационнойБазыМК**).

5. Заменить текст модуля на:

Процедура ПриДобавленииПодсистемы(Описание) Экспорт

Описание.Имя = "МояКонфигурация";

Описание.Версия = "1.0.0.1";

// Требуется библиотека стандартных подсистем.

Описание.ТребуемыеПодсистемы.Добавить("СтандартныеПодсистемы");

КонецПроцедуры

Процедура ПриДобавленииОбработчиковОбновления(Обработчики) Экспорт

КонецПроцедуры

Процедура ПередОбновлениемИнформационнойБазы() Экспорт

КонецПроцедуры

Процедура ПослеОбновленияИнформационнойБазы(Знач ПредыдущаяВерсия, Знач ТекущаяВерсия,

Знач ВыполненныеОбработчики, ВыводитьОписаниеОбновлений, МонопольныйРежим) Экспорт

КонецПроцедуры

Процедура ПриПодготовкеМакетаОписанияОбновлений(Знач Макет) Экспорт

КонецПроцедуры

Процедура ПриДобавленииОбработчиковПереходаСДругойПрограммы(Обработчики) Экспорт

КонецПроцедуры

Процедура ПриОпределенииРежимаОбновленияДанных(РежимОбновленияДанных, СтандартнаяОбработка) Экспорт

КонецПроцедуры

Процедура ПриЗавершенииПереходаСДругойПрограммы(Знач ПредыдущееИмяКонфигурации, Знач ПредыдущаяВерсияКонфигурации, Параметры) Экспорт

КонецПроцедуры

заменив в процедуре **ПриДобавленииПодсистемы** имя конфигурации и номер версии на установленные на шагах 1 и 2 соответственно.

6. Включить возможность внесения изменений в модуль **ПодсистемыКонфигурацииПереопределяемый**. Добавить в него в процедуру **ПриДобавленииПодсистем** строчку **МодулиПодсистем.Добавить("ОбновлениеИнформационнойБазыМК")**; заменив имя модуля на имя модуля, созданного на шаге 4.

7. Выполнить первый запуск. Убедиться в отсутствии ошибок при начальном заполнении.

**Глава 3. Настройка и использование подсистем при разработке конфигурации**

[3.1. Адресный классификатор](https://its.1c.ru/db/content/bsp23doc/src/3.1.%20%D0%B0%D0%B4%D1%80%D0%B5%D1%81%D0%BD%D1%8B%D0%B9%20%D0%BA%D0%BB%D0%B0%D1%81%D1%81%D0%B8%D1%84%D0%B8%D0%BA%D0%B0%D1%82%D0%BE%D1%80.htm?_=1504708101)  
[3.2. Анализ журнала регистрации](https://its.1c.ru/db/content/bsp23doc/src/3.2.%20%D0%B0%D0%BD%D0%B0%D0%BB%D0%B8%D0%B7%20%D0%B6%D1%83%D1%80%D0%BD%D0%B0%D0%BB%D0%B0%20%D1%80%D0%B5%D0%B3%D0%B8%D1%81%D1%82%D1%80%D0%B0%D1%86%D0%B8%D0%B8.htm?_=1504708101)  
[3.3. Анкетирование](https://its.1c.ru/db/content/bsp23doc/src/3.3.%20%D0%B0%D0%BD%D0%BA%D0%B5%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5.htm?_=1504708101)  
[3.4. Базовая функциональность](https://its.1c.ru/db/content/bsp23doc/src/3.4.%20%D0%B1%D0%B0%D0%B7%D0%BE%D0%B2%D0%B0%D1%8F%20%D1%84%D1%83%D0%BD%D0%BA%D1%86%D0%B8%D0%BE%D0%BD%D0%B0%D0%BB%D1%8C%D0%BD%D0%BE%D1%81%D1%82%D1%8C.htm?_=1504708101)  
[3.5. Банки](https://its.1c.ru/db/content/bsp23doc/src/3.5.%20%D0%B1%D0%B0%D0%BD%D0%BA%D0%B8.htm?_=1504708101)  
[3.6. Бизнес-процессы и задачи](https://its.1c.ru/db/content/bsp23doc/src/3.6.%20%D0%B1%D0%B8%D0%B7%D0%BD%D0%B5%D1%81-%D0%BF%D1%80%D0%BE%D1%86%D0%B5%D1%81%D1%81%D1%8B%20%D0%B8%20%D0%B7%D0%B0%D0%B4%D0%B0%D1%87%D0%B8.htm?_=1504708101)  
[3.7. Валюты](https://its.1c.ru/db/content/bsp23doc/src/3.7.%20%D0%B2%D0%B0%D0%BB%D1%8E%D1%82%D1%8B.htm?_=1504708101)  
[3.8. Варианты отчетов](https://its.1c.ru/db/content/bsp23doc/src/3.8.%20%D0%B2%D0%B0%D1%80%D0%B8%D0%B0%D0%BD%D1%82%D1%8B%20%D0%BE%D1%82%D1%87%D0%B5%D1%82%D0%BE%D0%B2.htm?_=1504708101)  
[3.9. Версионирование объектов](https://its.1c.ru/db/content/bsp23doc/src/3.9.%20%D0%B2%D0%B5%D1%80%D1%81%D0%B8%D0%BE%D0%BD%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5%20%D0%BE%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BE%D0%B2.htm?_=1504708101)  
[3.10. Взаимодействия](https://its.1c.ru/db/content/bsp23doc/src/3.10.%20%D0%B2%D0%B7%D0%B0%D0%B8%D0%BC%D0%BE%D0%B4%D0%B5%D0%B9%D1%81%D1%82%D0%B2%D0%B8%D1%8F.htm?_=1504708101)  
[3.11. Графики работы](https://its.1c.ru/db/content/bsp23doc/src/3.11.%20%D0%B3%D1%80%D0%B0%D1%84%D0%B8%D0%BA%D0%B8%20%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D1%8B.htm?_=1504708101)  
[3.12. Групповое изменение объектов](https://its.1c.ru/db/content/bsp23doc/src/3.12.%20%D0%B3%D1%80%D1%83%D0%BF%D0%BF%D0%BE%D0%B2%D0%BE%D0%B5%20%D0%B8%D0%B7%D0%BC%D0%B5%D0%BD%D0%B5%D0%BD%D0%B8%D0%B5%20%D0%BE%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BE%D0%B2.htm?_=1504708101)  
[3.13. Даты запрета изменения](https://its.1c.ru/db/content/bsp23doc/src/3.13.%20%D0%B4%D0%B0%D1%82%D1%8B%20%D0%B7%D0%B0%D0%BF%D1%80%D0%B5%D1%82%D0%B0%20%D0%B8%D0%B7%D0%BC%D0%B5%D0%BD%D0%B5%D0%BD%D0%B8%D1%8F.htm?_=1504708101)  
[3.14. Дополнительные отчеты и обработки](https://its.1c.ru/db/content/bsp23doc/src/3.14.%20%D0%B4%D0%BE%D0%BF%D0%BE%D0%BB%D0%BD%D0%B8%D1%82%D0%B5%D0%BB%D1%8C%D0%BD%D1%8B%D0%B5%20%D0%BE%D1%82%D1%87%D0%B5%D1%82%D1%8B%20%D0%B8%20%D0%BE%D0%B1%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%BA%D0%B8.htm?_=1504708101)  
[3.15. Завершение работы пользователей](https://its.1c.ru/db/content/bsp23doc/src/3.15.%20%D0%B7%D0%B0%D0%B2%D0%B5%D1%80%D1%88%D0%B5%D0%BD%D0%B8%D0%B5%20%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D1%8B%20%D0%BF%D0%BE%D0%BB%D1%8C%D0%B7%D0%BE%D0%B2%D0%B0%D1%82%D0%B5%D0%BB%D0%B5%D0%B9.htm?_=1504708101)  
[3.16. Загрузка данных из файла](https://its.1c.ru/db/content/bsp23doc/src/3.16.%20%D0%B7%D0%B0%D0%B3%D1%80%D1%83%D0%B7%D0%BA%D0%B0%20%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85%20%D0%B8%D0%B7%20%D1%84%D0%B0%D0%B9%D0%BB%D0%B0.htm?_=1504708101)  
[3.17. Заметки пользователя](https://its.1c.ru/db/content/bsp23doc/src/3.17.%20%D0%B7%D0%B0%D0%BC%D0%B5%D1%82%D0%BA%D0%B8%20%D0%BF%D0%BE%D0%BB%D1%8C%D0%B7%D0%BE%D0%B2%D0%B0%D1%82%D0%B5%D0%BB%D1%8F.htm?_=1504708101)  
[3.18. Заполнение объектов](https://its.1c.ru/db/content/bsp23doc/src/3.18.%20%D0%B7%D0%B0%D0%BF%D0%BE%D0%BB%D0%BD%D0%B5%D0%BD%D0%B8%D0%B5%20%D0%BE%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BE%D0%B2.htm?_=1504708101)  
[3.19. Запрет редактирования реквизитов объектов](https://its.1c.ru/db/content/bsp23doc/src/3.19.%20%D0%B7%D0%B0%D0%BF%D1%80%D0%B5%D1%82%20%D1%80%D0%B5%D0%B4%D0%B0%D0%BA%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F%20%D1%80%D0%B5%D0%BA%D0%B2%D0%B8%D0%B7%D0%B8%D1%82%D0%BE%D0%B2%20%D0%BE%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BE%D0%B2.htm?_=1504708101)  
[3.20. Защита персональных данных](https://its.1c.ru/db/content/bsp23doc/src/3.20.%20%D0%B7%D0%B0%D1%89%D0%B8%D1%82%D0%B0%20%D0%BF%D0%B5%D1%80%D1%81%D0%BE%D0%BD%D0%B0%D0%BB%D1%8C%D0%BD%D1%8B%D1%85%20%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85.htm?_=1504708101)  
[3.21. Интеграция с «1С-Коннект (Бухфон)»](https://its.1c.ru/db/content/bsp23doc/src/3.21.%20%D0%B8%D0%BD%D1%82%D0%B5%D0%B3%D1%80%D0%B0%D1%86%D0%B8%D1%8F%20%D1%81%20_1%D1%81-%D0%BA%D0%BE%D0%BD%D0%BD%D0%B5%D0%BA%D1%82%20(%D0%B1%D1%83%D1%85%D1%84%D0%BE%D0%BD)_.htm?_=1504708101)  
[3.22. Информация при запуске](https://its.1c.ru/db/content/bsp23doc/src/3.22.%20%D0%B8%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%86%D0%B8%D1%8F%20%D0%BF%D1%80%D0%B8%20%D0%B7%D0%B0%D0%BF%D1%83%D1%81%D0%BA%D0%B5.htm?_=1504708101)  
[3.23. Календарные графики](https://its.1c.ru/db/content/bsp23doc/src/3.23.%20%D0%BA%D0%B0%D0%BB%D0%B5%D0%BD%D0%B4%D0%B0%D1%80%D0%BD%D1%8B%D0%B5%20%D0%B3%D1%80%D0%B0%D1%84%D0%B8%D0%BA%D0%B8.htm?_=1504708101)  
[3.24. Контактная информация](https://its.1c.ru/db/content/bsp23doc/src/3.24.%20%D0%BA%D0%BE%D0%BD%D1%82%D0%B0%D0%BA%D1%82%D0%BD%D0%B0%D1%8F%20%D0%B8%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%86%D0%B8%D1%8F.htm?_=1504708101)  
[3.25. Напоминания пользователя](https://its.1c.ru/db/content/bsp23doc/src/3.25.%20%D0%BD%D0%B0%D0%BF%D0%BE%D0%BC%D0%B8%D0%BD%D0%B0%D0%BD%D0%B8%D1%8F%20%D0%BF%D0%BE%D0%BB%D1%8C%D0%B7%D0%BE%D0%B2%D0%B0%D1%82%D0%B5%D0%BB%D1%8F.htm?_=1504708101)  
[3.26. Настройка порядка элементов](https://its.1c.ru/db/content/bsp23doc/src/3.26.%20%D0%BD%D0%B0%D1%81%D1%82%D1%80%D0%BE%D0%B9%D0%BA%D0%B0%20%D0%BF%D0%BE%D1%80%D1%8F%D0%B4%D0%BA%D0%B0%20%D1%8D%D0%BB%D0%B5%D0%BC%D0%B5%D0%BD%D1%82%D0%BE%D0%B2.htm?_=1504708101)  
[3.27. Настройки программы](https://its.1c.ru/db/content/bsp23doc/src/3.27.%20%D0%BD%D0%B0%D1%81%D1%82%D1%80%D0%BE%D0%B9%D0%BA%D0%B8%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D1%8B.htm?_=1504708101)  
[3.28. Обмен данными](https://its.1c.ru/db/content/bsp23doc/src/3.28.%20%D0%BE%D0%B1%D0%BC%D0%B5%D0%BD%20%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D0%BC%D0%B8.htm?_=1504708101)  
[3.29. Обновление версии ИБ](https://its.1c.ru/db/content/bsp23doc/src/3.29.%20%D0%BE%D0%B1%D0%BD%D0%BE%D0%B2%D0%BB%D0%B5%D0%BD%D0%B8%D0%B5%20%D0%B2%D0%B5%D1%80%D1%81%D0%B8%D0%B8%20%D0%B8%D0%B1.htm?_=1504708101)  
[3.30. Обновление конфигурации](https://its.1c.ru/db/content/bsp23doc/src/3.30.%20%D0%BE%D0%B1%D0%BD%D0%BE%D0%B2%D0%BB%D0%B5%D0%BD%D0%B8%D0%B5%20%D0%BA%D0%BE%D0%BD%D1%84%D0%B8%D0%B3%D1%83%D1%80%D0%B0%D1%86%D0%B8%D0%B8.htm?_=1504708101)  
[3.31. Отправка SMS](https://its.1c.ru/db/content/bsp23doc/src/3.31.%20%D0%BE%D1%82%D0%BF%D1%80%D0%B0%D0%B2%D0%BA%D0%B0%20sms.htm?_=1504708101)  
[3.32. Оценка производительности](https://its.1c.ru/db/content/bsp23doc/src/3.32.%20%D0%BE%D1%86%D0%B5%D0%BD%D0%BA%D0%B0%20%D0%BF%D1%80%D0%BE%D0%B8%D0%B7%D0%B2%D0%BE%D0%B4%D0%B8%D1%82%D0%B5%D0%BB%D1%8C%D0%BD%D0%BE%D1%81%D1%82%D0%B8.htm?_=1504708101)  
[3.33. Печать](https://its.1c.ru/db/content/bsp23doc/src/3.33.%20%D0%BF%D0%B5%D1%87%D0%B0%D1%82%D1%8C.htm?_=1504708101)  
[3.34. Поиск и удаление дублей](https://its.1c.ru/db/content/bsp23doc/src/3.34.%20%D0%BF%D0%BE%D0%B8%D1%81%D0%BA%20%D0%B8%20%D1%83%D0%B4%D0%B0%D0%BB%D0%B5%D0%BD%D0%B8%D0%B5%20%D0%B4%D1%83%D0%B1%D0%BB%D0%B5%D0%B9.htm?_=1504708101)  
[3.35. Полнотекстовый поиск](https://its.1c.ru/db/content/bsp23doc/src/3.35.%20%D0%BF%D0%BE%D0%BB%D0%BD%D0%BE%D1%82%D0%B5%D0%BA%D1%81%D1%82%D0%BE%D0%B2%D1%8B%D0%B9%20%D0%BF%D0%BE%D0%B8%D1%81%D0%BA.htm?_=1504708101)  
[3.36. Получение файлов из Интернета](https://its.1c.ru/db/content/bsp23doc/src/3.36.%20%D0%BF%D0%BE%D0%BB%D1%83%D1%87%D0%B5%D0%BD%D0%B8%D0%B5%20%D1%84%D0%B0%D0%B9%D0%BB%D0%BE%D0%B2%20%D0%B8%D0%B7%20%D0%B8%D0%BD%D1%82%D0%B5%D1%80%D0%BD%D0%B5%D1%82%D0%B0.htm?_=1504708101)  
[3.37. Пользователи](https://its.1c.ru/db/content/bsp23doc/src/3.37.%20%D0%BF%D0%BE%D0%BB%D1%8C%D0%B7%D0%BE%D0%B2%D0%B0%D1%82%D0%B5%D0%BB%D0%B8.htm?_=1504708101)  
[3.38. Префиксация объектов](https://its.1c.ru/db/content/bsp23doc/src/3.38.%20%D0%BF%D1%80%D0%B5%D1%84%D0%B8%D0%BA%D1%81%D0%B0%D1%86%D0%B8%D1%8F%20%D0%BE%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BE%D0%B2.htm?_=1504708101)  
[3.39. Присоединенные файлы](https://its.1c.ru/db/content/bsp23doc/src/3.39.%20%D0%BF%D1%80%D0%B8%D1%81%D0%BE%D0%B5%D0%B4%D0%B8%D0%BD%D0%B5%D0%BD%D0%BD%D1%8B%D0%B5%20%D1%84%D0%B0%D0%B9%D0%BB%D1%8B.htm?_=1504708101)  
[3.40. Проверка легальности получения обновлений](https://its.1c.ru/db/content/bsp23doc/src/3.40.%20%D0%BF%D1%80%D0%BE%D0%B2%D0%B5%D1%80%D0%BA%D0%B0%20%D0%BB%D0%B5%D0%B3%D0%B0%D0%BB%D1%8C%D0%BD%D0%BE%D1%81%D1%82%D0%B8%20%D0%BF%D0%BE%D0%BB%D1%83%D1%87%D0%B5%D0%BD%D0%B8%D1%8F%20%D0%BE%D0%B1%D0%BD%D0%BE%D0%B2%D0%BB%D0%B5%D0%BD%D0%B8%D0%B9.htm?_=1504708101)  
[3.41. Работа в модели сервиса](https://its.1c.ru/db/content/bsp23doc/src/3.41.%20%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%B0%20%D0%B2%20%D0%BC%D0%BE%D0%B4%D0%B5%D0%BB%D0%B8%20%D1%81%D0%B5%D1%80%D0%B2%D0%B8%D1%81%D0%B0.htm?_=1504708101)  
[3.42. Работа с контрагентами](https://its.1c.ru/db/content/bsp23doc/src/3.42.%20%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%B0%20%D1%81%20%D0%BA%D0%BE%D0%BD%D1%82%D1%80%D0%B0%D0%B3%D0%B5%D0%BD%D1%82%D0%B0%D0%BC%D0%B8.htm?_=1504708101)  
[3.43. Работа с почтовыми сообщениями](https://its.1c.ru/db/content/bsp23doc/src/3.43.%20%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%B0%20%D1%81%20%D0%BF%D0%BE%D1%87%D1%82%D0%BE%D0%B2%D1%8B%D0%BC%D0%B8%20%D1%81%D0%BE%D0%BE%D0%B1%D1%89%D0%B5%D0%BD%D0%B8%D1%8F%D0%BC%D0%B8.htm?_=1504708101)  
[3.44. Работа с файлами](https://its.1c.ru/db/content/bsp23doc/src/3.44.%20%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%B0%20%D1%81%20%D1%84%D0%B0%D0%B9%D0%BB%D0%B0%D0%BC%D0%B8.htm?_=1504708101)  
[3.45. Рассылка отчетов](https://its.1c.ru/db/content/bsp23doc/src/3.45.%20%D1%80%D0%B0%D1%81%D1%81%D1%8B%D0%BB%D0%BA%D0%B0%20%D0%BE%D1%82%D1%87%D0%B5%D1%82%D0%BE%D0%B2.htm?_=1504708101)  
[3.46. Регламентные задания](https://its.1c.ru/db/content/bsp23doc/src/3.46.%20%D1%80%D0%B5%D0%B3%D0%BB%D0%B0%D0%BC%D0%B5%D0%BD%D1%82%D0%BD%D1%8B%D0%B5%20%D0%B7%D0%B0%D0%B4%D0%B0%D0%BD%D0%B8%D1%8F.htm?_=1504708101)  
[3.47. Резервное копирование ИБ](https://its.1c.ru/db/content/bsp23doc/src/3.47.%20%D1%80%D0%B5%D0%B7%D0%B5%D1%80%D0%B2%D0%BD%D0%BE%D0%B5%20%D0%BA%D0%BE%D0%BF%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5%20%D0%B8%D0%B1.htm?_=1504708101)  
[3.48. Свойства](https://its.1c.ru/db/content/bsp23doc/src/3.48.%20%D1%81%D0%B2%D0%BE%D0%B9%D1%81%D1%82%D0%B2%D0%B0.htm?_=1504708101)  
[3.49. Склонение представлений объектов](https://its.1c.ru/db/content/bsp23doc/src/3.49.%20%D1%81%D0%BA%D0%BB%D0%BE%D0%BD%D0%B5%D0%BD%D0%B8%D0%B5%20%D0%BF%D1%80%D0%B5%D0%B4%D1%81%D1%82%D0%B0%D0%B2%D0%BB%D0%B5%D0%BD%D0%B8%D0%B9%20%D0%BE%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BE%D0%B2.htm?_=1504708101)  
[3.50. Структура подчиненности](https://its.1c.ru/db/content/bsp23doc/src/3.50.%20%D1%81%D1%82%D1%80%D1%83%D0%BA%D1%82%D1%83%D1%80%D0%B0%20%D0%BF%D0%BE%D0%B4%D1%87%D0%B8%D0%BD%D0%B5%D0%BD%D0%BD%D0%BE%D1%81%D1%82%D0%B8.htm?_=1504708101)  
[3.51. Текущие дела](https://its.1c.ru/db/content/bsp23doc/src/3.51.%20%D1%82%D0%B5%D0%BA%D1%83%D1%89%D0%B8%D0%B5%20%D0%B4%D0%B5%D0%BB%D0%B0.htm?_=1504708101)  
[3.52. Удаление помеченных объектов](https://its.1c.ru/db/content/bsp23doc/src/3.52.%20%D1%83%D0%B4%D0%B0%D0%BB%D0%B5%D0%BD%D0%B8%D0%B5%20%D0%BF%D0%BE%D0%BC%D0%B5%D1%87%D0%B5%D0%BD%D0%BD%D1%8B%D1%85%20%D0%BE%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BE%D0%B2.htm?_=1504708101)  
[3.53. Управление доступом](https://its.1c.ru/db/content/bsp23doc/src/3.53.%20%D1%83%D0%BF%D1%80%D0%B0%D0%B2%D0%BB%D0%B5%D0%BD%D0%B8%D0%B5%20%D0%B4%D0%BE%D1%81%D1%82%D1%83%D0%BF%D0%BE%D0%BC.htm?_=1504708101)  
[3.54. Управление итогами и агрегатами](https://its.1c.ru/db/content/bsp23doc/src/3.54.%20%D1%83%D0%BF%D1%80%D0%B0%D0%B2%D0%BB%D0%B5%D0%BD%D0%B8%D0%B5%20%D0%B8%D1%82%D0%BE%D0%B3%D0%B0%D0%BC%D0%B8%20%D0%B8%20%D0%B0%D0%B3%D1%80%D0%B5%D0%B3%D0%B0%D1%82%D0%B0%D0%BC%D0%B8.htm?_=1504708101)  
[3.55. Файловые функции](https://its.1c.ru/db/content/bsp23doc/src/3.55.%20%D1%84%D0%B0%D0%B9%D0%BB%D0%BE%D0%B2%D1%8B%D0%B5%20%D1%84%D1%83%D0%BD%D0%BA%D1%86%D0%B8%D0%B8.htm?_=1504708101)  
[3.56. Центр мониторинга](https://its.1c.ru/db/content/bsp23doc/src/3.56.%20%D1%86%D0%B5%D0%BD%D1%82%D1%80%20%D0%BC%D0%BE%D0%BD%D0%B8%D1%82%D0%BE%D1%80%D0%B8%D0%BD%D0%B3%D0%B0.htm?_=1504708101)  
[3.57. Шаблоны сообщений](https://its.1c.ru/db/content/bsp23doc/src/3.57.%20%D1%88%D0%B0%D0%B1%D0%BB%D0%BE%D0%BD%D1%8B%20%D1%81%D0%BE%D0%BE%D0%B1%D1%89%D0%B5%D0%BD%D0%B8%D0%B9.htm?_=1504708101)  
[3.58. Электронная подпись](https://its.1c.ru/db/content/bsp23doc/src/3.58.%20%D1%8D%D0%BB%D0%B5%D0%BA%D1%82%D1%80%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F%20%D0%BF%D0%BE%D0%B4%D0%BF%D0%B8%D1%81%D1%8C.htm?_=1504708101)  
[3.59. Особые случаи внедрения подсистем](https://its.1c.ru/db/content/bsp23doc/src/3.59.%20%D0%BE%D1%81%D0%BE%D0%B1%D1%8B%D0%B5%20%D1%81%D0%BB%D1%83%D1%87%D0%B0%D0%B8%20%D0%B2%D0%BD%D0%B5%D0%B4%D1%80%D0%B5%D0%BD%D0%B8%D1%8F%20%D0%BF%D0%BE%D0%B4%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC.htm?_=1504708101)

# 3.1. Адресный классификатор

Подсистема «Адресный классификатор» предназначена для загрузки, хранения и предоставления другим подсистемам конфигурации адресных сведений Федеральной информационной адресной системы (ФИАС). Подсистема позволяет работать в одном из двух вариантов:

1. При наличии постоянного подключения к Интернету адресные сведения не загружаются в программу, а запрашиваются онлайн с помощью веб-сервиса фирмы «1С».

2. Для автономной работы без (постоянного) подключения к Интернету адресные сведения могут быть загружены в программу с сайта фирмы «1С» или из указанной папки на диске.

## Настройка

Если в конфигурации не используется подсистема «Настройки программы», то в командном интерфейсе ответственного за ведение нормативно-справочной информации необходимо разместить:

● константу **ИсточникДанныхАдресногоКлассификатора**. См. пример в форме **ПоддержкаИОбслуживание обработки ПанельАдминистрированияБСП**;

● команды **ЗагрузитьАдресныйКлассификатор**, **ОчиститьАдресныйКлассификатор**, **ПроверитьОбновлениеАдресногоКлассификатора** регистра сведений **АдресныеОбъекты** для работы в варианте с загрузкой данных в программу по требованию.

## Настройка прав доступа пользователей

Для настройки прав доступа пользователей к данным подсистемы «Адресный классификатор» следует использовать роли, приведенные ниже.

Таблица 3.1.

| **№** | **Роли и их назначение** |
| --- | --- |
| 1. | **БазовыеПрава** (из подсистемы «Базовая функциональность»)  Просмотр и использование адресных сведений |
| 2. | **ДобавлениеИзменениеАдресныхСведений**  Добавление и изменение адресных сведений |
| 3. | **АдминистраторСистемы** (из подсистемы «Базовая функциональность»)  Удаление помеченных на удаление объектов подсистемы |

Пример настройки прав доступа пользователей приведен ниже.

Таблица 3.2.

|  |  |  |
| --- | --- | --- |
| **№** | **Группа пользователей и ее функции** | **Состав ролей** |
| 1. | Администратор | **АдминистраторСистемы** (из подсистемы «Базовая функциональность») |
| 2. | Ответственный за нормативно-справочную информацию | **БазовыеПрава** (из подсистемы «Базовая функциональность»)  **ЗапускТонкогоКлиента** (из подсистемы «Базовая функциональность»)  **ДобавлениеИзменениеАдресныхСведений** |

## Использование при разработке конфигурации

Программный интерфейс подсистемы представлен общим модулем **АдресныйКлассификатор**. Доступ к данным адресного классификатора рекомендуется производить только через его программный интерфейс.

### Настройка обмена данными

Все объекты метаданных подсистемы не должны участвовать в обмене. Данные адресного классификатора должны заполняться и поддерживаться в актуальном состоянии отдельно в каждой информационной базе, между которыми настроена синхронизация данных.

# 3.2. Анализ журнала регистрации

Подсистема «Анализ журнала регистрации» предназначена для диагностики работы программы на основе данных журнала регистрации. Включает в себя следующие варианты отчетов:

● **Анализ активности пользователей (пользователя)**,

● **Контроль журнала регистрации**,

● **Продолжительность работы регламентных заданий**.

## Настройка

Разместить в командном интерфейсе администратора отчет **АнализЖурналаРегистрации**.

### Настройка прав доступа пользователей

Для настройки прав доступа пользователей к данным подсистемы «Анализ журнала регистрации» следует использовать роль, указанную ниже.

Таблица 3.3.

| **№** | **Роли и их назначение** |
| --- | --- |
| 1. | **ПолныеПрава** (из подсистемы «Базовая функциональность»)  Просмотр отчетов по журналу регистрации |

## Использование при разработке конфигурации

Программный интерфейс подсистемы представлен модулем менеджера отчета **АнализЖурналаРегистрации**.

### Настройка обмена данными

Для настройки обмена данными никаких действий не требуется, так как подсистема не предоставляет собственных данных.

# 3.3. Анкетирование

Подсистема «Анкетирование» предназначена для составления анкет, проведения опросов и анализа результатов опросов. С помощью веб-клиента можно проводить опросы через Интернет.

Для аутентификации респондентов в системе используются возможности подсистемы «Пользователи». Авторизовавшийся респондент получает доступ к предназначенным для него опросам, а также к архиву ранее заполненных им анкет.

Подсистема «Анкетирование» предоставляет ряд отчетов, которые позволяют анализировать в различных разрезах результаты одного выбранного опроса и сравнивать результаты, полученные при проведении нескольких опросов.

## Настройка

Для того чтобы респонденты имели возможность указывать в качестве ответов на вопросы анкеты объекты информационной базы, необходимо принять решение по поводу состава типов таких объектов (определить типы ответов). Например, справочники **Номенклатура**, **Партнеры**, **Подразделения**. Список допустимых типов ответов надо добавить в состав типов свойства **Тип** **значения** ПВХ **ВопросыДляАнкетирования**.

Затем необходимо принять решение по поводу состава объектов конфигурации, которые могут быть сопоставлены с респондентами анкеты. Например, справочники **Контрагенты**, **Партнеры**, **Физические лица**. Список допустимых типов респондентов надо указать в составе типов определяемого типа **Респондент**.

Допустимые типы респондентов должны быть также включены в список внешних пользователей системы. Порядок действий по настройке подсистемы «Пользователи» описан в разделе «[Пользователи](https://its.1c.ru/db/content/bsp23doc/src/3.37.%20%D0%BF%D0%BE%D0%BB%D1%8C%D0%B7%D0%BE%D0%B2%D0%B0%D1%82%D0%B5%D0%BB%D0%B8.htm?_=1504708101#_пользователи)».

### Настройка пользовательского интерфейса

Для использования подсистемы в конфигурации необходимо настроить командный интерфейс исходя из функциональных обязанностей различных пользователей данной подсистемы.

Если в конфигурации не используется подсистема «Настройки программы», то в рабочем месте администратора программы необходимо разместить константу **ИспользоватьАнкетирование**. См. пример в форме **Органайзер**обработки **ПанельАдминистрированияБСП**.

В командный интерфейс ответственного за составление анкет следует включить:

● ПВХ **ВопросыДляАнкетирования**,

● справочник **ШаблоныАнкет**.

В командном интерфейсе ответственного за проведение опросов необходимо разместить два документа:

● **НазначениеОпросов**,

● **Анкета**.

В командный интерфейс ответственного за анализ результатов анкетирования необходимо включить ряд отчетов:

● **АнализОпроса**,

● **АнкетированиеАналитическийОтчет**.

Для респондента в рабочую область рабочего стола следует поместить основную форму обработки **СписокРеспондента**.

### Настройка прав доступа пользователей

Для настройки прав доступа пользователей к данным подсистемы «Анкетирование» следует использовать роли, приведенные ниже.

Таблица 3.4.

| **№** | **Роли и их назначение** |
| --- | --- |
| 1. | **ЧтениеОтветовНаВопросыАнкет**  Чтение анкет, опросов, ответов на вопросы анкет и анализ ответов в отчетах |
| 2. | **ДобавлениеИзменениеОтветовНаВопросыАнкет**  Участие в опросах в качестве респондента |
| 3. | **ДобавлениеИзменениеШаблоновАнкет**  Подготовка шаблонов анкет. Добавление и изменение вопросов для анкетирования |
| 4. | **ДобавлениеИзменениеОпросов**  Добавление и изменение опросов |
| 5. | **ПолныеПрава** (из подсистемы «Базовая функциональность»).  Включение и отключение использования подсистемы «Анкетирование». Удаление помеченных на удаление объектов подсистемы |

Дополнительно следует создать вспомогательные роли или использовать подходящие роли, существующие в конфигурации, для обеспечения доступа к данным, которые не относятся к подсистеме «Анкетирование», но требуются для работы с ней.

Таблица 3.5.

| **№** | **Вспомогательные роли и их назначение** |
| --- | --- |
| 1. | **<ЧтениеДанныхДляОтветовНаВопросыАнкет>**  Роль для чтения объектов метаданных, значения которых могут выступать в качестве ответов на вопросы |
| 2. | **<ЧтениеДанныхРеспондентов>**  Чтение объектов метаданных, значения которых могут выступать в качестве респондентов |

Примеры настройки прав доступа пользователей приведены ниже.

Таблица 3.6.

| **№** | **Группа пользователей и ее функции** | **Состав ролей** |
| --- | --- | --- |
| 1. | Администратор | **ПолныеПрава** (из подсистемы «Базовая функциональность») |
| 2. | Ответственный за анализ результатов анкетирования:  ● просмотр заполненных анкет,  ● использование аналитических отчетов подсистемы | **БазовыеПрава** (из подсистемы «Базовая функциональность»):  ● **ЗапускТонкогоКлиента** (из подсистемы «Базовая функциональность»),  ● **ЧтениеВнешнихПользователей** (из подсистемы «Пользователи»),  ● **ЧтениеОтветовНаВопросыАнкет**,  ● **<ЧтениеДанныхРеспондентов>**,  **● <ЧтениеДанныхДляОтветовНаВопросыАнкет>** |
| 3. | Составитель шаблонов анкет:  ● создание и редактирование шаблонов анкет и вопросов для анкетирования | **БазовыеПрава** (из подсистемы «Базовая функциональность»):  ● **ЗапускТонкогоКлиента** (из подсистемы «Базовая функциональность»),  ● **ДобавлениеИзменениеШаблоновАнкет**,  **● <ЧтениеДанныхДляОтветовНаВопросыАнкет>** |
| 4. | Ответственный за проведение опросов:  ● назначение опросов | **БазовыеПрава** (из подсистемы «Базовая функциональность»):  ● **ЗапускТонкогоКлиента** (из подсистемы «Базовая функциональность»),  ● **ЧтениеВнешнихПользователей** (из подсистемы «Пользователи»),  ● **ДобавлениеИзменениеОпросов**,  **● <ЧтениеДанныхРеспондентов>** |
| 5. | Респондент:  ● участие в опросах в качестве респондента | **БазовыеПраваВнешнегоПользователя** (из подсистемы «Базовая функциональность»):  ● **ЗапускВебКлиента** (из подсистемы «Базовая функциональность»),  **● <ЧтениеДанныхДляОтветовНаВопросыАнкет>** |

## Использование при разработке конфигурации

### Настройка обмена данными

Все объекты метаданных подсистемы, содержащие данные, рекомендуется включать в планы обмена распределенной ИБ (РИБ).

# 3.4. Базовая функциональность

Подсистема «Базовая функциональность» содержит базовый функционал, обязательный для всех прикладных решений, использующих библиотеку. К базовому функционалу относятся процедуры и функции общего назначения, ряд универсальных обработок, а также стандартные роли: **ПолныеПрава**, **БазовыеПрава** и другие.

## Настройка

### Базовые типы

Если в конфигурации имеется справочник **Организации**, то необходимо:

* ● указать ссылку на него в свойстве **Тип**определяемого типа **Организация**;
* ● реализовать в модуле менеджера справочника **Организации** в области **ПрограммныйИнтерфейс** две экспортные функции:

// Возвращает организацию по умолчанию.

// Если в ИБ есть только одна организация, которая не помечена на удаление и не является предопределенной,

// то будет возвращена ссылка на нее, иначе будет возвращена пустая ссылка.

//

// Возвращаемое значение:

//     СправочникСсылка.Организации - ссылка на организацию.

//

Функция ОрганизацияПоУмолчанию() Экспорт

КонецФункции

// Возвращает количество элементов справочника Организации.

// Не учитывает предопределенные и помеченные на удаление элементы.

//

// Возвращаемое значение:

//     Число - количество организаций.

//

Функция КоличествоОрганизаций() Экспорт

КонецФункции

### Инициализация параметров сеанса

Для инициализации параметров сеанса требуется вписать имя параметра сеанса и путь к его обработчику в процедуру **ОбработчикиИнициализацииПараметровСеанса** общего модуля **ОбщегоНазначенияПереопределяемый**. При этом обработчик инициализации должен принимать два параметра:

* ● **ИмяПараметра** – строка – имя инициализируемого параметра;
* ● **УстановленныеПараметры** – массив – имена параметров, которые были инициализированы.

### Проверка минимальной версии платформы

Для проверки того, что версия платформы подходит для запуска конфигурации, необходимо вызвать функцию **СтандартныеПодсистемыКлиент.ПроверитьВерсиюПлатформы** в обработчике **ПриНачалеРаботыСистемы**модуля управляемого и обычного приложения. Пример реализации можно найти в модуле управляемого и обычного приложения демонстрационной конфигурации.

### Подключить формы объектов программы

Для подключения форм объектов программы, в которых требуется выводить подключаемые команды, надо:

* ● В процедуре **ПриСозданииНаСервере** (обработчик события формы) вставить вызов по шаблону:

// СтандартныеПодсистемы.ПодключаемыеКоманды

ПодключаемыеКоманды.ПриСозданииНаСервере(ЭтотОбъект, , Истина);

// Конец СтандартныеПодсистемы.ПодключаемыеКоманды

* ● В модуле формы вставить процедуры (обработчики команд):

// СтандартныеПодсистемы.ПодключаемыеКоманды

&НаКлиенте

Процедура Подключаемый\_ВыполнитьКоманду(Команда)

ПодключаемыеКомандыКлиент.ВыполнитьКоманду(ЭтотОбъект, Команда, <ОбъектИлиТаблицаФормы>, Истина);

КонецПроцедуры

&НаСервере

Процедура Подключаемый\_ВыполнитьКомандуНаСервере(Контекст, Результат)

ПодключаемыеКоманды.ВыполнитьКоманду(ЭтотОбъект, Контекст, <ОбъектИлиТаблицаФормы>, Результат);

КонецПроцедуры

&НаКлиенте

Процедура Подключаемый\_ОбновитьКоманды()

ПодключаемыеКомандыКлиентСервер.ОбновитьКоманды(ЭтотОбъект, <ОбъектИлиТаблицаФормы>);

КонецПроцедуры

// Конец СтандартныеПодсистемы.ПодключаемыеКоманды

* ● Если форма является формой объекта, тогда:
* ● В параметре <**ОбъектИлиТаблицаФормы**> следует передавать реквизит формы типа **ДанныеФормыСтруктура**. Например:

ПодключаемыеКомандыКлиент.ВыполнитьКоманду(ЭтотОбъект, Команда, Объект);

* ● В обработчике **ПриЧтенииНаСервере** формы следует вставить вызов по шаблону:

&НаСервере

Процедура ПриЧтенииНаСервере(ТекущийОбъект)

// СтандартныеПодсистемы.ПодключаемыеКоманды

ПодключаемыеКомандыКлиентСервер.ОбновитьКоманды(ЭтотОбъект, <ОбъектИлиТаблицаФормы>);

// Конец СтандартныеПодсистемы.ПодключаемыеКоманды

КонецПроцедуры

* ● В обработчике **ПриОткрытии** формы следует вставить вызов по шаблону:

&НаКлиенте

Процедура ПриОткрытии(Отказ)

// СтандартныеПодсистемы.ПодключаемыеКоманды

ПодключаемыеКомандыКлиент.НачатьОбновлениеКоманд(ЭтотОбъект);

// Конец СтандартныеПодсистемы.ПодключаемыеКоманды

КонецПроцедуры

* ● Также рекомендуется вызывать процедуры обновления видимости команд после изменения ключевых реквизитов (значения которых могут быть использованы в условиях видимости команд):
* ● **ПодключаемыеКомандыКлиент**.**НачатьОбновлениеКоманд** - для обновления видимости команд на клиенте (подключает обработчик ожидания **Подключаемый\_ОбновитьКоманды**).
* ● **ПодключаемыеКомандыКлиентСервер.ОбновитьКоманды** - для обновления видимости команд на сервере (используется в случае, если в процессе вызова уже делается серверный вызов).
* ● Если форма является формой списка, тогда:
* ● В параметре <**ОбъектИлиТаблицаФормы**> следует передавать элемент формы типа **ТаблицаФормы**, связанный с динамическим списком. Например:

ПодключаемыеКомандыКлиент.ВыполнитьКоманду(ЭтотОбъект, Команда, Элементы.Список);

* ● В обработчике **ПриАктивизацииСтроки** таблицы формы следует вставить вызов по шаблону:

&НаКлиенте

Процедура <ИмяТаблицы>ПриАктивизацииСтроки(Элемент)

// СтандартныеПодсистемы.ПодключаемыеКоманды

ПодключаемыеКомандыКлиент.НачатьОбновлениеКоманд(ЭтотОбъект);

// Конец СтандартныеПодсистемы.ПодключаемыеКоманды

КонецПроцедуры

Внимание. Поле **Ссылка** динамического списка формы должно быть доступно в обработчике команды. Для этого необходимо в свойствах поля **Ссылка**, вложенного в реквизит формы типа **ДинамическийСписок**, включить флажок **Использовать всегда**.

* ● Опционально. Добавить реквизит формы **ПараметрыПодключаемыхКоманд** произвольного типа. Это позволит не создавать реквизит формы динамически, что ускорит время открытия формы.

### Настройка исключений поиска ссылок на объекты

В тех случаях, когда ссылки между объектами информационной базы не являются значимыми для таких операций, как удаление объекта или поиск ссылок на объект, необходимо настроить список исключений поиска ссылок в функции **ПриДобавленииИсключенийПоискаСсылок**в переопределяемом модуле **ОбщегоНазначенияПереопределяемый**.

Список исключений поиска ссылок может также использоваться другими подсистемами или в функциональности конфигурации. См. также раздел «[Удаление помеченных объектов](https://its.1c.ru/db/content/bsp23doc/src/3.52.%20%D1%83%D0%B4%D0%B0%D0%BB%D0%B5%D0%BD%D0%B8%D0%B5%20%D0%BF%D0%BE%D0%BC%D0%B5%D1%87%D0%B5%D0%BD%D0%BD%D1%8B%D1%85%20%D0%BE%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BE%D0%B2.htm?_=1504708101#_очистка_мест_использования)».

В частности, при анализе мест использования с помощью функции **ОбщегоНазначения.МестаИспользования** в результатах поиска ссылающихся объектов не учитываются ссылки из этого списка исключений.

### Настройка пользовательского интерфейса

При запуске система автоматически устанавливает заголовок окна программы по значению функции **Пользователи.АвторизованныйПользователь** и значению константы **ЗаголовокСистемы**.

Пример заголовка системы:

**Торговый дом «Ромашка» / Романова / Бухгалтерия предприятия, редакция 2.0 / (1С:Предприятие)**.

Если в конфигурации не используется подсистема «Настройки программы», то в рабочем месте администратора программы необходимо разместить константу **ЗаголовокСистемы**. Для того чтобы изменения вступили в силу, требуется вызвать процедуру **УстановитьПроизвольныйЗаголовокПриложения** общего модуля **ОбщегоНазначенияКлиент**. См. пример в форме **ОбщиеНастройки** обработки **ПанельАдминистрированияБСП**.

Если в конфигурации не используется подсистема «Настройки программы», то в рабочем месте администратора программы необходимо дополнительно разместить:

* ● общую команду **УстановитьРасширениеРаботыСФайлами**,
* ● обработку **Журнал регистрации**,
* ● команду для вызова формы **ПараметрыАдминистрированияСервернойИБ** в форме административных настроек системы.

См. пример размещения в формах обработки **ПанельАдминистрированияБСП**.

В форме персональных настроек следует разместить поле для изменения настройки, управляющей подтверждением при закрытии программы. См. пример реализации в демонстрационной конфигурации на закладке **Общие формы \_ДемоМоиНастройки**.

## Использование при разработке конфигурации

### Программный интерфейс

Программный интерфейс подсистемы представлен экспортными функциями общих модулей:

* ● **ОбщегоНазначения**, **ОбщегоНазначенияКлиент**, **ОбщегоНазначенияКлиентСервер**,**ОбщегоНазначенияПереопределяемый**,
* ● **СтандартныеПодсистемыКлиентПовтИсп**,
* ● **СтроковыеФункцииКлиентСервер** и др.

### Выполнение кода при запуске системы

Действия, выполняемые перед началом работы системы, при начале работы системы и перед завершением работы системы следует располагать в процедурах общего модуля **ОбщегоНазначенияКлиентПереопределяемый**:**ПередНачаломРаботыСистемы**, **ПриНачалеРаботыСистемы** и **ПередЗавершениемРаботыСистемы**соответственно. При этом следует учитывать, что при работе в модели сервиса данные процедуры могут быть вызваны не только при осуществлении фактического входа (выхода) пользователя из системы, но и при интерактивном входе (выходе) администратора информационной базы в область данных.

С целью минимизации количества серверных вызовов при старте системы не рекомендуется напрямую вызывать серверные процедуры и функции из кода модуля приложения и модуля управляемого приложения. Для передачи клиенту параметров, необходимых для выполнения клиентского кода, следует использовать функцию **ПараметрыРаботыКлиентаПриЗапуске** общего модуля **СтандартныеПодсистемыКлиентПовтИсп**. При первом вызове этой функции происходит одно обращение к серверу, после чего полученное значение кешируется на клиенте для всех последующих вызовов этой функции.

При необходимости расширить количество передаваемых параметров следует добавлять новые параметры в функцию **ПараметрыРаботыКлиентаПриЗапуске** переопределяемого общего модуля **ОбщегоНазначенияПереопределяемый**. Новые параметры рекомендуется помещать после параметров библиотечных подсистем в произвольном порядке относительно друг друга, например:

**Параметры.Вставить("ИнформационнаяБазаФайловая", ОбщегоНазначения.ИнформационнаяБазаФайловая());**

Аналогичный подход рекомендуется применять и для взаимодействия с сервером при дальнейшей работе пользователя в программе. См. функцию **ПараметрыРаботыКлиента** переопределяемого общего модуля **ОбщегоНазначенияПереопределяемый**.

### Обновление вспомогательных данных во время разработки

В ряде случаев при разработке конфигурации может потребоваться обновление вспомогательных данных, которые влияют на работу программы: кеши некоторых свойств метаданных, служебные регистры сведений и т. п.

* ● Для полного обновления этих данных предназначена внешняя обработка **Инструменты разработчика: Обновление вспомогательных данных**, которая входит в состав дистрибутива и в демонстрационную конфигурацию.
* ● Выполнить оптимальное обновление вспомогательных данных можно следующим способом: указать параметр запуска **ЗапуститьОбновлениеИнформационнойБазы** в конфигураторе или через параметр командной строки **/С**.
* ● При закладке изменений в хранилище, которые могут привести к необходимости обновления вспомогательных данных, можно увеличивать номер версии конфигурации, тогда у других участников коллективной разработки автоматически запустятся обязательные обработчики обновления.

### Использование идентификаторов объектов метаданных

При необходимости хранить в базе данных ссылку на объект метаданных (например, ссылка на объект метаданных **Справочник.Организации**) рекомендуется вместо строкового реквизита с полным именем объекта метаданных использовать ссылку на соответствующий элемент справочника **ИдентификаторыОбъектовМетаданных** или **ИдентификаторыОбъектовРасширений**. Такая потребность возникает, например, в различных подсистемах БСП: для хранения настроек версионирования данных, списка выбранных ролей в профилях групп доступа и т. п.

Такой подход позволяет:

* ● повысить производительность запросов, которые обращаются к реквизитам данного типа;
* ● снизить размер таблиц в базе данных, в которых используются реквизиты данного типа;
* ● выводить в пользовательском интерфейсе представление ссылки вместо строкового имени объекта метаданных;
* ● избавиться от разработки обработчиков обновления и первоначального заполнения ИБ для актуализации строковых реквизитов с полными именами объектов метаданных при изменениях метаданных конфигурации.

Программно ссылку на объект метаданных можно получить с помощью функции **ИдентификаторОбъектаМетаданных** общего модуля **ОбщегоНазначения**.

Справочник **ИдентификаторыОбъектовМетаданных** заполняется автоматически при первом запуске (и каждом обновлении) конфигурации согласно метаданным конфигурации, а также автоматически актуализируется при каждом обновлении версии конфигурации (учитываются переименованные, добавленные и удаленные объекты метаданных). Полный список объектов метаданных (справочники, документы, регистры и т. п.), обновление идентификаторов которых выполняется автоматически, см. в комментарии к функции **ИдентификаторОбъектаМетаданных** общего модуля **ОбщегоНазначения**. Справочник **ИдентификаторыОбъектовРасширений** заполняется еще и при добавлении расширений через общую форму **УстановленныеРасширения**, а также по требованию (при первом обращении из кода).

Исключение составляют подсистемы и роли, обновление идентификаторов которых требуется выполнять вручную. Необходимость обновления идентификаторов подсистем и ролей возникает при их переименовании (а для подсистем – также и в случае их перемещения из одной родительской подсистемы в другую). В противном случае возможно рассогласование идентификаторов объектов метаданных с объектами метаданных конфигурации.

Для обновления идентификаторов необходимо воспользоваться процедурой **ПриДобавленииПереименованийОбъектовМетаданных** общего модуля **ОбщегоНазначенияПереопределяемый** и увеличить номер версии конфигурации. Например, следующий фрагмент кода описывает, что в версии конфигурации 2.0.1.2 подсистема **\_ДемоПоставляемыеДанные** была перенесена из подсистемы **\_ДемоРаботаВМоделиСервиса** в **\_ДемоАдминистрирование**:

Процедура ПриДобавленииПереименованийОбъектовМетаданных (Итог) Экспорт

ОбщегоНазначения.ДобавитьПереименование(Итог, "2.0.1.2",

"Подсистема.\_ДемоРаботаВМоделиСервиса.Подсистема.\_ДемоПоставляемыеДанные",

"Подсистема.\_ДемоАдминистрирование.Подсистема.\_ДемоПоставляемыеДанные");

КонецПроцедуры

Обновление идентификаторов выполняется последовательно по версиям конфигурации, а в пределах одной версии – в порядке следования строк со сведениями о переименованиях. Обновление идентификаторов для переименованных подсистем выполняется также и для всех их дочерних подсистем (если они есть).

Отдельным случаем, когда обновление идентификаторов происходит автоматически, является переименование объектов метаданных с целью изменения структуры данных с помощью создания новой копии объекта метаданных. Такая необходимость возникает, когда реструктуризация объекта метаданных невозможна. Например, при сокращении длины кода справочника **Подразделения** с 50 до 11 может потребоваться создать новый справочник, а старый переименовать в **УдалитьПодразделения**. В этом случае идентификатор справочника будет автоматически назначен новому справочнику с тем же именем.

Однако если при этом новый справочник будет назван иначе, например **СтруктурныеЕдиницы**, то потребуется отразить переименование справочника **Подразделения** в справочник **СтруктурныеЕдиницы**, как это описано выше.

При активной разработке конфигурации данные справочников **ИдентификаторыОбъектовМетаданных** можно также обновлять вручную, без увеличения номера версии конфигурации, с помощью обработки **ИнструментыРазработчикаОбновлениеВспомогательныхДанных.epf**, которая входит в состав дистрибутива и в демонстрационную конфигурацию. Однако это допустимо только в случае, когда не производилось переименование ролей или подсистем. В противном случае необходимо увеличить версию конфигурации и вписать переименование в процедуру **ПриДобавленииПереименованийОбъектовМетаданных**, как написано выше. Если этого не сделать, то элемент справочника, который соответствует роли или подсистеме, будет помечен на удаление и будет создан новый, что приведет к рассогласованию идентификаторов объектов метаданных с объектами метаданных конфигурации. Рассогласования можно исправить вручную. Для этого нужно открыть форму элемента справочника **ИдентификаторыОбъектовМетаданных**, выполнить команду **Включить возможность редактирования** в меню **Все действия** и задать соответствующее полное имя.

### Версионирование программных интерфейсов

При разработке нескольких конфигураций, которые должны взаимодействовать друг с другом (например, с помощью веб-сервисов или обмена сообщениями), как правило, на стороне каждой из конфигураций реализуется программный интерфейс, к которому обращается другая конфигурация. По мере развития каждой из конфигураций часто возникает необходимость внесения изменений в программные интерфейсы, предоставляемые этими конфигурациями своим корреспондентам.

В общем случае при изменении реализации программного интерфейса (далее – просто интерфейса), например набора методов или структуры их параметров, код вызывающей стороны становится неработающим, поскольку поведение вызывающих методов «жестко» завязано на конкретную реализацию интерфейса.

Для исключения коллизий с некорректным использованием измененного интерфейса следует использовать механизм версионирования интерфейсов.

Определения:

* ● Семейство интерфейсов – набор интерфейсов, решающих одну задачу и представленных в развитии реализации именами объектов метаданных с постфиксами номеров версий.
* ● Базовое имя интерфейсов семейства (или просто «базовое имя») – имя его представителя без постфиксного номера версии.
* ● Номер версии состоит из четырех групп, разделенных точками. Каждая из этих групп может содержать исключительно числовое значение. Например, **2.0.1.6**.
* ● Приращение номеров версий осуществляется стандартным для версионирования ПО образом. Например, **1.0.2.1, 1.0.2.2, … 1.0.3.1…** и т. д.
* ● Интерфейс, носящий базовое имя, считается интерфейсом версии **1.0.1.1**.

Действия на стороне, предоставляющей интерфейс (например, веб-сервис **MessageExchange**):

* ● Создать интерфейс семейства с новой реализацией, наименовав его по правилу: <Базовое имя> + «\_» + <Номер версии с подчеркиваниями вместо точек>. Например, **MessageExchange\_2\_0\_1\_6**.
* ● Присвоить семейству интерфейсов имя. Например, **ОбменСообщениями**. Имя семейству интерфейсов присваивается произвольно, исходя из его назначения.
* ● В теле процедуры **ПриОпределенииПоддерживаемыхВерсийПрограммныхИнтерфейсов** общего модуля **ОбщегоНазначенияПереопределяемый** к секции подсистемы интерфейса добавить вставку нового номера версии, как показано в примере описания процедуры. Если секция отсутствует, необходимо ее создать.

Действия на стороне, использующей интерфейс:

* ● Механизм версионирования обслуживается веб-сервисом **InterfaceVersion**, доступ к которому предоставляет роль **УдаленныйДоступБазоваяФункциональность**. Таким образом, функционирование механизма возможно только для пользователя, обладающего этой ролью.
* ● Перед вызовом методов интерфейса публикующей стороны нужно запросить у нее список номеров поддерживаемых версий через вызов функции **ПолучитьВерсииИнтерфейса** общего модуля **ОбщегоНазначения**, как показано в примере ее описания. При этом в качестве параметра **ИмяИнтерфейса**нужно задать имя семейства интерфейсов.
* ● В зависимости от полученного набора номеров версий в логических ветках осуществлять вызовы тех или иных методов с актуальной структурой параметров, обращаясь к конкретным интерфейсам семейства.

Примеры использования можно найти в демонстрационной конфигурации (см. вызовы функции **ПолучитьВерсииИнтерфейса**).

### Безопасное хранилище паролей

Для хранения паролей и другой конфиденциальной информации следует использовать безопасное хранилище. Для помещения данных в безопасное хранилище предназначена процедура **ЗаписатьДанныеВБезопасноеХранилище**, для получения ранее сохраненных данных – функция**ПрочитатьДанныеИзБезопасногоХранилища**, а для удаления данных – функция**УдалитьДанныеИзБезопасногоХранилища**общего модуля **ОбщегоНазначения**.

Если в форме необходимо разместить поле для ввода пароля, то необходимо:

* ● Создать реквизит формы с типом **Строка**, включить свойство **Сохраняемые данные**, а также вспомогательный реквизит с типом **Булево** для фиксации факта изменения пароля. Например, **Пароль** и **ПарольИзменен**.
* ● Создать на форме поле со свойством **РежимПароля = Да** и связать его с ранее созданным реквизитом формы.
* ● Для сведения к минимуму возможности получения пароля злоумышленниками не следует при открытии формы присваивать его реквизиту формы и передавать с сервера на клиент. Вместо этого в событие формы **ПриСозданиеНаСервере** на сервере надо разместить следующий код:

УстановитьПривилегированныйРежим(Истина);

Пароли = ОбщегоНазначения.ПрочитатьДанныеИзБезопасногоХранилища(Объект.Ссылка, "Пароль, ПарольSMTP");

УстановитьПривилегированныйРежим(Ложь);

Пароль = ?(ЗначениеЗаполнено(Пароли.Пароль), ЭтотОбъект.УникальныйИдентификатор, "");

ПарольSMTP = ?(ЗначениеЗаполнено(Пароли.ПарольSMTP), ЭтотОбъект.УникальныйИдентификатор, "");

* ● В событие **ПриЗаписиНаСервере**:

Если ПарольИзменен Тогда

УстановитьПривилегированныйРежим(Истина);

ОбщегоНазначения.ЗаписатьДанныеВБезопасноеХранилище(ТекущийОбъект.Ссылка, Пароль);

УстановитьПривилегированныйРежим(Ложь);

КонецЕсли;

Если ПарольSMTPИзменен Тогда

УстановитьПривилегированныйРежим(Истина);

ОбщегоНазначения.ЗаписатьДанныеВБезопасноеХранилище(ТекущийОбъект.Ссылка, ПарольSMTP, "ПарольSMTP");

УстановитьПривилегированныйРежим(Ложь);

КонецЕсли;

* ● В модуле объекта разместить следующий код в событие**ПередУдалением**:

УстановитьПривилегированныйРежим(Истина);

ОбщегоНазначения.УдалитьДанныеИзБезопасногоХранилища(Ссылка);

УстановитьПривилегированныйРежим(Ложь);

См. пример в форме **ФормаЭлемента** справочника **УчетныеЗаписиЭлектроннойПочты**.

### Стандартные роли и дополнительные права

Библиотека не навязывает ту или иную методику разработки ролей в конфигурации. В общем виде при разработке системы ролей могут использоваться два подхода, которые различаются степенью детализации (укрупненности) ролей:

1. При первом подходе проектируются **прикладные роли**, предоставляющие доступ ко всему множеству объектов метаданных, которые требуются для работы определенной категории пользователей системы. Например, «Бухгалтер», «Кассир» и т. д. Такие роли самостоятельно назначают пользователям (группам пользователей) и, как правило, расширяют дополнительными правами, например: «Действия главного бухгалтера», «Печать непроведенных документов», «Запуск тонкого клиента» и т. п.

2. Во втором случае проектируются **роли-функции**, предоставляющие «атомарный» доступ к определенному подмножеству объектов метаданных, с которым различные пользователи могут работать как с одной функцией системы. Такие роли не назначаются пользователям поодиночке, а объединяются в профили групп доступа и назначаются пользователям (группам пользователей) в совокупности. При этом профили групп доступа выступают аналогами прикладных ролей, например: «Бухгалтер», «Кассир» и т. д.

В состав библиотеки входят роли, которые можно использовать для настройки доступа пользователей к объектам информационной базы в обоих перечисленных случаях:

1. Обязательные роли.

2. Роли, общие для нескольких подсистем.

3. Роли, поставляемые в составе отдельных подсистем (приведены в разделах по настройке конкретных подсистем).

Обязательные роли должны всегда присутствовать в конфигурации, в то время как остальные роли из состава библиотеки являются опциональными: в определенных случаях они могут быть удалены из конфигурации.

#### Обязательные роли

Таблица 3.7.

| **№** | **Роли и их назначение** |
| --- | --- |
| 1. | **ПолныеПрава**  Обязательная роль, которая предоставляет неограниченный доступ ко всем прикладным данным, но не дает прав доступа для администрирования информационной базы в целом (обновление конфигурации, работа в конфигураторе и т. п.). Включает все права доступа, кроме права интерактивного удаления.  При работе в модели сервиса назначается администраторам абонентов (областей данных) и предоставляет неограниченный доступ ко всем данным текущей области, а также позволяет выполнять администрирование пользователей, настройку программы, удаление помеченных объектов и другие административные действия с областью данных.  При работе в локальном режиме назначается совместно с ролью **АдминистраторСистемы** для администрирования.  В базовых версиях конфигурации роль **ПолныеПрава** предоставляет неограниченный доступ ко всем данным и конфигурации информационной базы |
| 2. | **АдминистраторСистемы**  Обязательная роль, предоставляющая права администрирования информационной базы в целом (обновление конфигурации, работа в конфигураторе и т. п.).  Назначается пользователям только совместно с ролью **ПолныеПрава**.  При работе в модели сервиса используется для администраторов сервиса. Включает неограниченный доступ ко всем неразделенным данным, кроме права интерактивного удаления.  В базовых версиях конфигурации роль **АдминистраторСистемы** не используется |
| 3. | **Администрирование**  Предоставляет права «Административные функции» и «Активные пользователи» |
| 4. | **ЗапускВебКлиента**  Предоставляет право «Веб-клиент» |
| 5. | **ЗапускТонкогоКлиента**  Предоставляет право «Тонкий клиент» |
| 6. | **ЗапускТолстогоКлиента**  Предоставляет право «Толстый клиент» |
| 7. | **ЗапускВнешнегоСоединения**  Предоставляет право «Внешнее соединение» |
| 8. | **ЗапускAutomation**  Предоставляет право «Automation» |
| 9. | **ИнтерактивноеОткрытиеВнешнихОтчетовИОбработок**  Предоставляет права «Интерактивное открытие внешних отчетов» и «Интерактивное открытие внешних обработок» |
| 10. | **ОбновлениеКонфигурацииБазыДанных**  Предоставляет право «Обновление конфигурации базы данных» |
| 11. | **ВыводНаПринтерФайлБуферОбмена**  Предоставляет право «Вывод» |
| 12. | **ПросмотрЖурналаРегистрации**  Предоставляет право «Журнал регистрации» |
| 13. | **РежимВсеФункции**  Предоставляет право «Режим "Все функции"» |
| 14. | **СохранениеДанныхПользователя**  Предоставляет право «Сохранение данных пользователя» |

#### Роли, общие для нескольких подсистем

В отличие от ролей, поставляемых в составе отдельных подсистем, данная группа ролей предоставляет доступ к объектам сразу нескольких подсистем.

Таблица 3.8.

| **№** | **Роли и их назначение** |
| --- | --- |
| 1. | **БазовыеПрава**  Роль, предоставляющая доступ к тем объектам информационной базы, которые должны быть всегда доступны всем пользователям.  Назначается всем пользователям |
| 2. | **БазовыеПраваВнешнегоПользователя**  Роль, предоставляющая доступ к тем объектам информационной базы, которые должны быть всегда доступны всем внешним пользователям.  Назначается всем внешним пользователям |
| 3. | **УдаленныйДоступБазоваяФункциональность**  Предоставляет доступ к вызовам операций веб-сервисов подсистемы «Базовая функциональность» |

Далее рассмотрим подробнее, каким образом библиотечные роли следует применять при использовании обоих подходов к разработке системы ролей в конфигурации.

### Подход № 1. Прикладные роли и дополнительные права

В простейшем случае роль соответствует функциональным обязанностям определенной категории пользователей системы, например: «Бухгалтер», «Кассир». Такие прикладные роли применимы в конфигурациях, в которых функциональные обязанности пользователей заранее известны и не предполагают изменений на внедрениях.

Совместно с прикладными ролями пользователям могут назначаться одна или несколько ролей, предоставляющих дополнительные права. Название роли, предоставляющей дополнительные права на определенные действия, должно описывать эти действия, например: «Действия главного бухгалтера», «Печать непроведенных документов», «Запуск тонкого клиента», «Удаление помеченных объектов» и т. п. Не следует применять в именах ролей слово «право» (исключения составляют обязательные роли **ПолныеПрава**, **БазовыеПрава** и **БазовыеПраваВнешнегоПользователя**).

Роли, предоставляющие дополнительные права, не предназначены для самостоятельного использования, они применяются в комбинации с прикладной ролью и служат для расширения доступа к данным той или иной прикладной области. Например:

* ● роль «Действия генерального директора» может назначаться пользователям только совместно с «прикладной» ролью «Генеральный директор»;
* ● роль «Действия главного бухгалтера» – только вместе с ролью «Бухгалтер»;
* ● роль «Печать непроведенных документов» может использоваться совместно с произвольной «прикладной» ролью.

Перед началом работы конфигурации следует проверить состав ролей текущего пользователя. Если пользователю не назначена хотя бы одна прикладная роль, предполагающая самостоятельное использование, работа пользователя не может быть продолжена.

**Если в конфигурации не используется подсистема «Управление доступом»**, то для упрощения администрирования прав пользователей рекомендуется оставить в конфигурации только обязательные роли, удалив все остальные библиотечные роли. Следует также удалить обязательную роль **Базовые права**, а ее состав прав включить в одну или несколько прикладных ролей.

**При использовании подсистемы «Управление доступом»**допускается оставить в конфигурации все библиотечные роли как есть и рекомендуется подготовить типовые профили групп доступа пользователей. Подробнее о настройке профилей групп доступа с помощью возможностей подсистемы «Управление доступом» см. в разделе «[Управление доступом](https://its.1c.ru/db/content/bsp23doc/src/3.53.%20%D1%83%D0%BF%D1%80%D0%B0%D0%B2%D0%BB%D0%B5%D0%BD%D0%B8%D0%B5%20%D0%B4%D0%BE%D1%81%D1%82%D1%83%D0%BF%D0%BE%D0%BC.htm?_=1504708101#_управление_доступом)».

### Подход № 2. Роли-функции

Если функциональные обязанности пользователей конфигурации не известны на этапе разработки, а определяются на конкретном внедрении, прикладные роли следует разделять на более специализированные роли, декомпозируя их до уровня отдельных ролей-функций.

Роли-функции не предназначены для самостоятельного использования, а назначаются пользователям (группам пользователей) всегда в комбинации с другими ролями-функциями. Совокупность ролей-функций, предоставляющих определенной категории пользователей необходимый набор прав для выполнения своих обязанностей, образует профиль группы доступа. Например, профиль «Менеджер по продажам» может включать такие роли-функции, как «Добавление и изменение заказов покупателей» и «Чтение нормативно-справочной информации», а «Начальник отдела продаж» – «Добавление и изменение заказов покупателей», «Добавление и изменение нормативно-справочной информации» и «Печать непроведенных документов».

Настройка прав пользователей с помощью отдельных ролей-функций не требует внесения изменений в конфигурацию как на этапе внедрения, так и на этапе эксплуатации (администрирования) информационной системы.

При проектировании роли-функции нужно учитывать, что она участвует в формировании пользовательского интерфейса. Поэтому для достижения гибкости управления пользовательским интерфейсом могут потребоваться «атомарные» роли, которые предоставляют доступ к одному объекту метаданных.

Так в случае с объектами метаданных, которые используются в качестве разрезов ограничения доступа (например, организаций, складов, контрагентов, касс и др.), для чтения лучше создавать «атомарные роли», т. к. они будут использоваться в сочетании с ролями – например, для доступа к документам. При таком подходе, например, в интерфейс кладовщика не попадет доступ к списку касс, который ему не требуется, что соответствует принципу управляемого интерфейса.

**Для реализации подхода № 2 рекомендуется внедрение в конфигурацию подсистемы «Управление доступом»**. Подробнее о настройке профилей групп доступа с помощью возможностей подсистемы «Управление доступом» см. в разделе «[Управление доступом](https://its.1c.ru/db/content/bsp23doc/src/3.53.%20%D1%83%D0%BF%D1%80%D0%B0%D0%B2%D0%BB%D0%B5%D0%BD%D0%B8%D0%B5%20%D0%B4%D0%BE%D1%81%D1%82%D1%83%D0%BF%D0%BE%D0%BC.htm?_=1504708101#_управление_доступом)».

#### Требования к наименованию ролей-функций

Название роли-функции рекомендуется начинать с описания действия, которое она позволяет выполнить пользователю в информационной системе:

* ● **Чтение** – для права чтения и просмотра, например: **ЧтениеЗаказов**, **ЧтениеНормативноСправочнойИнформации** и т. п.;
* ● **ДобавлениеИзменение** – для права добавления, которое должно сочетаться с правом изменения добавленного, правами чтения и просмотра, например, **ДобавлениеИзменениеЗаказов**;
* ● **Изменение** – для права изменения, например **ИзменениеЗадач**;
* ● **Редактирование** – для права интерактивного редактирования, которое должно сочетаться с правом изменения того, что редактируется, например **РедактированиеЗадач**;
* ● **Просмотр** – для права интерактивного просмотра, которое должно сочетаться с правом чтения того, что просматривается, например **ПросмотрПартнеров**;
* ● **ПометкаУдаления** – для права интерактивной установки пометки удаления, которое должно сочетаться с правом изменения того, что помечается на удаление, например **ПометкаУдаленияБизнесПроцессов**.

Если роль дает права сразу на несколько действий, то в синониме роли их следует перечислять через запятую и соединять союзом «и», например: «Чтение заказов покупателей», «Добавление и изменение нормативно-справочной информации», «Добавление и изменение заказов покупателей».

#### Управление командным интерфейсом с помощью ролей-функций

Видимость элементов пользовательского интерфейса в зависимости от прав пользователей рекомендуется задавать в соответствующих ролях-функциях. Например, в роль «Чтение заказов покупателей» следует включать общие команды, которые открывают формы заказов покупателей.

Однако в некоторых случаях все же необходимо заводить дополнительные роли для управления командным интерфейсом. Как правило, такие интерфейсные роли не содержат ограничений доступа к данным, а связываются непосредственно с реквизитами объектов, элементами форм и командами. Их наличие у текущего пользователя может также программно проверяться из кода на встроенном языке.

Название интерфейсной роли должно описывать разрешенное действие, например «Просмотр контактной информации». Исключение составляют роли, которые предоставляют доступ к разделам командного интерфейса (просмотр подсистем). Такие роли начинаются со слова «Раздел», например: **РазделЗапасыИЗакупки**, **РазделПродажи**, **РазделПродажиИПроведениеСделок**, **РазделПродажиИВозвраты**, **РазделПродажиРозничныеПродажи**.

Примеры интерфейсных ролей приведены ниже.

Таблица 3.9.

| **№** | **Роли и их назначение** |
| --- | --- |
| 1. | **<Раздел нормативно-справочная информация>**  Роль управляет видимостью раздела командного интерфейса, в котором размещены команды открытия форм объектов метаданных, относящихся к нормативно-справочной информации, например: форма списка справочника **Валюты**, адресного классификатора и т. п. |
| 2. | **<Раздел совместная работа>**  Роль управляет видимостью раздела командного интерфейса, используемого для работы с задачами и бизнес-процессами |

Примеры настройки прав доступа пользователей приведены ниже.

Таблица 3.10.

| **№** | **Группа пользователей и ее функции** | **Состав ролей** |
| --- | --- | --- |
| 1. | Администратор:  ● имеет полный доступ к системе,  **●**выполняет настройку прав (состав ролей) и ограничений доступа | ● **АдминистраторСистемы**,  **● ПолныеПрава** |
| 2. | Ответственный за нормативно-справочную информацию | ● **ЗапускТонкогоКлиента**,  ● **БазовыеПрава**,  ● **ДобавлениеИзменениеАдресныхСведений**,  ● **ДобавлениеИзменениеБанков**,  ● **ДобавлениеИзменениеКалендарныхГрафиков**,  ●**ДобавлениеИзменениеДополнительныхРеквизитовИСведений**,  ● **ДобавлениеИзменениеКурсовВалют**,  ● **ДобавлениеИзменениеГрафиковРаботы**,  ● **ДобавлениеИзменениеВидовКонтактнойИнформации**,  **● <Раздел нормативно-справочная информация>** |

### Контроль замены ссылок при записи объектов и выполнение связанных действий

При замене ссылок в документах ([поиске и удалении дублей](https://its.1c.ru/db/content/bsp23doc/src/3.34.%20%D0%BF%D0%BE%D0%B8%D1%81%D0%BA%20%D0%B8%20%D1%83%D0%B4%D0%B0%D0%BB%D0%B5%D0%BD%D0%B8%D0%B5%20%D0%B4%D1%83%D0%B1%D0%BB%D0%B5%D0%B9.htm?_=1504708101#_поиск_и_удаление) или программном вызове **ОбщегоНазанчения.ЗаменитьСсылки**) документы записываются без перепроведения, чтобы не нарушать последовательности. Как следствие, не срабатывает событие **ОбработкаПроведения** и вся прикладная логика проведения по регистрам. Однако в некоторых случаях при замене ссылок может потребоваться выполнять часть этой логики. Для этого в события записи объектов (**ПередЗаписью** и **ПриЗаписи**) через **ДополнительныеСвойства** передаются параметры:

* ● **ЗаменаСсылок – Булево – Истина**. Позволяет определить, что отработал механизм замены ссылок.
* ● **ВыполненныеЗамены – Массив**. Описание выполненных замен. Элементы типа **Структура** с полями:
* ● **СсылкаДубля –** **Произвольный**. Элемент, который был заменен.
* ● **СсылкаОригинала – Произвольный**. Элемент, на который выполнена замена.
* ● **ВидРеквизита – Строка**. Вид реквизита, в котором выполнена замена. Может принимать следующие значения: **Реквизиты**, **СтандартныеРеквизиты**, **ТабличныеЧасти**, **СтандартныеТабличныеЧасти**, **Движения**, **Последовательности**, **НаборЗаписей**.
* ● **ИмяРеквизита – Строка**. Имя реквизита или имя табличной части.
* ● **Индекс – Число**, **Неопределено**. Индекс строки, в которой выполнена замена дубля на оригинал. Заполняется в случае, когда замена выполнена в табличной части.
* ● **ИмяКолонки – Строка**, **Неопределено**. Имя колонки, в которой выполнена замена дубля на оригинал. Заполняется в случае, когда замена выполнена в табличной части.

Эти параметры можно использовать для того, чтобы определить, «где» и «что именно было изменено» и при необходимости выполнить связанные действия. Например, создать ключи аналитики и выполнить их замену в регистрах.

### Подключение отчетов и обработок к другим объектам метаданных

В отчетах и обработках конфигурации и расширений можно подключать дополнительные команды к тем объектам, в которые внедрены «[Печать](https://its.1c.ru/db/content/bsp23doc/src/3.33.%20%D0%BF%D0%B5%D1%87%D0%B0%D1%82%D1%8C.htm?_=1504708101#_печать)» или «[Заполнение объектов](https://its.1c.ru/db/content/bsp23doc/src/3.4.%20%D0%B1%D0%B0%D0%B7%D0%BE%D0%B2%D0%B0%D1%8F%20%D1%84%D1%83%D0%BD%D0%BA%D1%86%D0%B8%D0%BE%D0%BD%D0%B0%D0%BB%D1%8C%D0%BD%D0%BE%D1%81%D1%82%D1%8C.htm?_=1504708101#_заполнение_объектов)». Для этого:

* ● Отчет или обработка должна входить в состав подсистемы **ПодключаемыеОтчетыИОбработки**.
* ● В ее модуле менеджера должна быть определена процедура **ПриОпределенииНастроек** по шаблону:

#Область ПрограммныйИнтерфейс

// Определяет состав программного интерфейса для интеграции с конфигурацией.

//

// Параметры:

//   Настройки - Структура - Настройки интеграции этого объекта.

//       См. возвращаемое значение функции ПодключаемыеКоманды.НастройкиПодключаемыхОтчетовИОбработок().

//

Процедура ПриОпределенииНастроек(Настройки) Экспорт

// Пример:

//Настройки.Размещение.Добавить(Метаданные.Документы.ИмяДокумента);

//Настройки.ДобавитьКомандыПечати = Истина;

КонецПроцедуры

#КонецОбласти

* ● Подробное описание процедур **ДобавитьКомандыПечати** и **ДобавитьКомандыЗаполнения** и их параметров см. в документации к подсистемам «[Печать](https://its.1c.ru/db/content/bsp23doc/src/3.33.%20%D0%BF%D0%B5%D1%87%D0%B0%D1%82%D1%8C.htm?_=1504708101#_печать)» и «[Заполнение объектов](https://its.1c.ru/db/content/bsp23doc/src/3.4.%20%D0%B1%D0%B0%D0%B7%D0%BE%D0%B2%D0%B0%D1%8F%20%D1%84%D1%83%D0%BD%D0%BA%D1%86%D0%B8%D0%BE%D0%BD%D0%B0%D0%BB%D1%8C%D0%BD%D0%BE%D1%81%D1%82%D1%8C.htm?_=1504708101#_заполнение_объектов)».

### Настройка обмена данными

Для настройки обмена данными следует руководствоваться общими правилами.

В планы обмена распределенной информационной базы (РИБ) рекомендуется включать все объекты метаданных подсистемы, кроме перечисленных:

* ● константа **АвтоматическиНастраиватьРазрешенияВПрофиляхБезопасности**,
* ● константа **АдресПубликацииИнформационнойБазыВИнтернете**,
* ● константа **АдресПубликацииИнформационнойБазыВЛокальнойСети**,
* ● константа **ГлавныйУзел**,
* ● константа **ЗаголовокСистемы**,
* ● константа **ИдентификаторИнформационнойБазы**,
* ● константа **ИспользоватьРазделениеПоОбластямДанных**,
* ● константа **ИспользуютсяПрофилиБезопасности**,
* ● константа **НеИспользоватьРазделениеПоОбластямДанных**,
* ● константа **ПрофильБезопасностиИнформационнойБазы**,
* ● константа **ЭтоАвтономноеРабочееМесто**,
* ● справочник **ВерсииРасширений**,
* ● справочник **ИдентификаторыОбъектовРасширений**,
* ● регистр сведений **БезопасноеХранилищеДанных**,
* ● регистр сведений **БезопасноеХранилищеДанныхОбластейДанных**,
* ● регистр сведений **ЗапросыРазрешенийНаИспользованиеВнешнихРесурсов**,
* ● регистр сведений **ИдентификаторыОбъектовВерсийРасширений**,
* ● регистр сведений **ПараметрыРаботыВерсийРасширений**,
* ● регистр сведений **РазрешенияНаИспользованиеВнешнихРесурсов**,
* ● регистр сведений **РежимыПодключенияВнешнихМодулей**,
* ● регистр сведений **СеансыВерсийРасширений**.
* ● регистр сведений **КэшПрограммныхИнтерфейсов**,
* ● регистр сведений **НеразделенныеПользователи**.

В планах обмена распределенной информационной базы (РИБ) рекомендуется отключать регистрацию изменений для следующих объектов метаданных подсистемы (см. также раздел «[Особенности создания начального образа подчиненного узла распределенной ИБ](https://its.1c.ru/db/content/bsp23doc/src/3.28.%20%D0%BE%D0%B1%D0%BC%D0%B5%D0%BD%20%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D0%BC%D0%B8.htm?_=1504708101#_особенности_создания_начального)»):

* ● константа **ПараметрыСлужебныхСобытий**.

Из остальных планов обмена (не РИБ) рекомендуется исключать следующие объекты метаданных:

* ● константа **ПараметрыПодключаемыхКоманд**,
* ● константа **ПараметрыСлужебныхСобытий**,
* ● справочник **ВерсииРасширений**,
* ● справочник **ИдентификаторыОбъектовМетаданных**,
* ● справочник **ИдентификаторыОбъектовРасширений**,
* ● регистр сведений **ИдентификаторыОбъектовВерсийРасширений**,
* ● регистр сведений **ПараметрыРаботыВерсийРасширений**,
* ● регистр сведений **СеансыВерсийРасширений**.

### Настройка рабочей даты пользователя

Довольно часто на практике информация в учетные системы вводится не сразу в момент совершения операции, а позже по времени (задним числом). Например, фактическое поступление на склад товаров от поставщика может произойти раньше, чем будут получены соответствующие финансовые документы. Задним числом чаще всего работают бухгалтеры, отражающие уже свершившиеся ранее факты хозяйственной жизни.

Поэтому для удобства пользователей разработчик конфигурации может предусмотреть возможность настройки даты, которую необходимо подставлять в новые документы, – так называемой «рабочей даты».

В общем модуле **ОбщегоНазначения** расположены процедура и функции для программного управления такой настройкой:

* ● процедура **УстановитьРабочуюДатуПользователя**,
* ● функция **РабочаяДатаПользователя**,
* ● функция **ТекущаяДатаПользователя**.

Пример их использования приведен в демонстрационной конфигурации в модуле общей формы **\_ДемоМоиНастройки**. С помощью переключателя на форме пользователь может выбрать в качестве даты создания документа текущую дату компьютера либо иную указанную дату.

Сведения о рабочей дате сохраняются в хранилище общих настроек информационной базы, для использования которого необходимо, чтобы пользователю была назначена роль **СохранениеДанныхПользователя**.

При создании новых документов для заполнения их даты из настройки пользователя можно использовать подписки на события **ОбработкаЗаполнения и ПриКопировании** (см. в демонстрационной конфигурации подписки на события **\_ДемоЗаполнитьДатуДокументаПоРабочейДате** и **\_ДемоЗаполнитьДатуДокументаПоРабочейДатеПриКопировании**).

При использовании подписок на события необходимо учитывать, что их обработчики вызываются после того, как будет выполнен обработчик события в модуле объекта. Поэтому если алгоритм автоматического заполнения данных нового документа использует его дату, то установку даты документа из настройки рабочей даты следует выполнить в самом начале процедуры – обработчика события.

### Расширение видов подключаемых команд

Для этого в модуле **ПодключаемыеКомандыПереопределяемый**:

* ● В процедуре **ПриОпределенииВидовПодключаемыхКоманд** добавить вид подключаемых команд и описать умолчания для вывода подменю с командами этого вида. Например:

Процедура ПриОпределенииВидовПодключаемыхКоманд(ВидыПодключаемыхКоманд) Экспорт

Вид = ВидыПодключаемыхКоманд.Добавить();

Вид.Имя = "<ИмяВидаКоманд>";

Вид.ИмяПодменю = "<ИмяПодменю>";

Вид.Заголовок = НСтр("ru = '<Заголовок подменю>'");

Вид.Картинка = БиблиотекаКартинок.<ИмяКартинкиПодменю>;

КонецПроцедуры

* ● Если нужна возможность разрабатывать и подключать команды данного вида при помощи расширений, тогда в процедуре **ПриОпределенииСоставаНастроекПодключаемыхОбъектов** следует описать простую настройку (например - флажок), которая будет служить маркером того, что объект расширения поставляет команды данного вида. Например:

Процедура ПриОпределенииСоставаНастроекПодключаемыхОбъектов(НастройкиПрограммногоИнтерфейса) Экспорт

Настройка = НастройкиПрограммногоИнтерфейса.Добавить();

Настройка.Ключ = "<ИмяФлажка>";

Настройка.ОписаниеТипов = Новый ОписаниеТипов("Булево");

Настройка.ВидыПодключаемыхОбъектов = "Обработка";

КонецПроцедуры

* ● В процедуре **ПриОпределенииКомандПодключенныхКОбъекту** добавить команды данного вида, которые соответствуют переданному контексту. Например:

Процедура ПриОпределенииКомандПодключенныхКОбъекту(НастройкиФормы, Источники, ПодключенныеОтчетыИОбработки, Команды) Экспорт

Если Источники.Строки.Найти(<ОбъектМетаданных>, "Метаданные") <> Неопределено Тогда

Команда = Команды.Добавить();

Команда.Вид = "<ИмяВидаКоманд>";

Команда.Представление = НСтр("ru = '<Заголовок команды>'");

Команда.РежимЗаписи = "НеЗаписывать";

Команда.ВидимостьВФормах = "<ИмяФормыОбъекта>";

Команда.Обработчик = "<МодульПодсистемы>.<ИмяПроцедуры>";

КонецЕсли;

КонецПроцедуры

# 3.5. Банки

Подсистема «Банки» предназначена для загрузки, хранения и предоставления другим подсистемам конфигурации классификатора банков. Загрузка классификатора может производиться как автоматически (при помощи регламентного задания с сайта 1С), так и вручную (с сайта 1С, с диска ИТС).

## Настройка

Для использования подсистемы в конфигурации необходимо разместить в пользовательском интерфейсе ответственного за НСИ ссылку на справочник **КлассификаторБанковРФ**.

### Настройка прав доступа пользователей

Для настройки прав доступа пользователей к данным подсистемы «Банки» следует использовать роли, приведенные ниже.

Таблица 3.11.

|  |  |
| --- | --- |
| **№** | **Роли и их назначение** |
| 1. | **БазовыеПрава**(из подсистемы «Базовая функциональность»)  Чтение записей классификатора банков РФ |
| 2. | **ДобавлениеИзменениеБанков**  Загрузка классификатора банков РФ |
| 3. | **ПолныеПрава**, **АдминистрированиеСистемы**(из подсистемы «Базовая функциональность»)  Редактирование, удаление помеченных на удаление записей классификатора банков РФ |

## Использование при разработке конфигурации

Данные классификатора банков РФ хранятся в справочнике **КлассификаторБанковРФ**. Данные о банках, измененные пользователем вручную, хранятся в справочниках банковских счетов. См. справочник **\_ДемоБанковскиеСчета** в демонстрационной конфигурации.

Для обращения к данным банковского счета при формировании печатной формы нужно применять следующий подход, проиллюстрированный на примере документа **\_ДемоСчетНаОплатуПокупателю**. Фрагмент текста запроса для выбора реквизитов банка:

ВЫБОР

КОГДА БанковскийСчетКонтрагента.РучноеИзменениеРеквизитовБанка

ТОГДА БанковскийСчетКонтрагента.БИКБанка

ИНАЧЕ КлассификаторБанковРФ.Код

КОНЕЦ КАК БИКБанк,

ВЫБОР

КОГДА БанковскийСчетКонтрагента.РучноеИзменениеРеквизитовБанка

ТОГДА БанковскийСчетКонтрагента.НаименованиеБанка

ИНАЧЕ КлассификаторБанковРФ.Наименование

КОНЕЦ КАК НаименованиеБанка,

ВЫБОР

КОГДА БанковскийСчетКонтрагента.РучноеИзменениеРеквизитовБанка

ТОГДА БанковскийСчетКонтрагента.КоррСчетБанка

ИНАЧЕ КлассификаторБанковРФ.КоррСчет

КОНЕЦ КАК КоррСчетБанка,

ВЫБОР

КОГДА БанковскийСчетКонтрагента.РучноеИзменениеРеквизитовБанка

ТОГДА БанковскийСчетКонтрагента.ГородБанка

ИНАЧЕ КлассификаторБанковРФ.Город

КОНЕЦ КАК ГородБанка,

ВЫБОР

КОГДА БанковскийСчетКонтрагента.РучноеИзменениеРеквизитовБанкаДляРасчетов

ТОГДА БанковскийСчетКонтрагента.БИКБанкаДляРасчетов

ИНАЧЕ КлассификаторБанковКорреспондентовРФ.Код

КОНЕЦ КАК БИКБанкаДляРасчетов,

ВЫБОР

КОГДА БанковскийСчетКонтрагента.РучноеИзменениеРеквизитовБанкаДляРасчетов

ТОГДА БанковскийСчетКонтрагента.НаименованиеБанкаДляРасчетов

ИНАЧЕ КлассификаторБанковКорреспондентовРФ.Наименование

КОНЕЦ КАК НаименованиеБанкаДляРасчетов,

ВЫБОР

КОГДА БанковскийСчетКонтрагента.РучноеИзменениеРеквизитовБанкаДляРасчетов

ТОГДА БанковскийСчетКонтрагента.КоррСчетБанкаДляРасчетов

ИНАЧЕ КлассификаторБанковКорреспондентовРФ.КоррСчет

КОНЕЦ КАК КоррСчетБанкаДляРасчетов,

ВЫБОР

КОГДА БанковскийСчетКонтрагента.РучноеИзменениеРеквизитовБанкаДляРасчетов

ТОГДА БанковскийСчетКонтрагента.ГородБанкаДляРасчетов

ИНАЧЕ КлассификаторБанковКорреспондентовРФ.Город

КОНЕЦ КАК ГородБанкаДляРасчетов,

СчетНаОплатуПокупателю.Контрагент.НаименованиеПолное КАК ПолучательНаименованиеПолное,

БанковскийСчетКонтрагента.НомерСчета КАК ПолучательНомерСчета

ИЗ

Документ.\_ДемоСчетНаОплатуПокупателю КАК СчетНаОплатуПокупателю

ЛЕВОЕ СОЕДИНЕНИЕ Справочник.КлассификаторБанковРФ КАК КлассификаторБанковРФ

ПО СчетНаОплатуПокупателю.БанковскийСчет.Банк = КлассификаторБанковРФ.Ссылка

ЛЕВОЕ СОЕДИНЕНИЕ Справочник.КлассификаторБанковРФ КАК КлассификаторБанковКорреспондентовРФ

ПО СчетНаОплатуПокупателю.БанковскийСчет.БанкДляРасчетов = КлассификаторБанковКорреспондентовРФ.Ссылка

ЛЕВОЕ СОЕДИНЕНИЕ Справочник.\_ДемоБанковскиеСчета КАК БанковскийСчетКонтрагента

ПО СчетНаОплатуПокупателю.БанковскийСчет = БанковскийСчетКонтрагента

Пример кода по выводу реквизитов банка в табличный документ:

ДанныеПечати = Новый Структура;

//...

Если ПустаяСтрока(Шапка.БИКБанкаДляРасчетов) Тогда

ДанныеПечати.Вставить("БанкПолучателяПредставление", СокрЛП(Шапка.НаименованиеБанка) + " " + СокрЛП(Шапка.ГородБанка));

ДанныеПечати.Вставить("ПредставлениеПоставщика", СокрЛП(Шапка.ПолучательНаименованиеПолное));

ДанныеПечати.Вставить("БИКБанкаПолучателя", СокрЛП(Шапка.БИКБанк));

ДанныеПечати.Вставить("СчетБанкаПолучателяПредставление", СокрЛП(Шапка.КоррСчетБанка));

ДанныеПечати.Вставить("СчетПолучателяПредставление", СокрЛП(Шапка.ПолучательНомерСчета));

Иначе

ДанныеПечати.Вставить("БанкПолучателяПредставление", СокрЛП(Шапка.НаименованиеБанкаДляРасчетов) + " " + СокрЛП(Шапка.ГородБанкаДляРасчетов));

ДанныеПечати.Вставить("ПредставлениеПоставщика", СокрЛП(Шапка.ПолучательНаименованиеПолное) + " р/с " + СокрЛП(Шапка.ПолучательНомерСчета) + " в " + СокрЛП(Шапка.НаименованиеБанка)+ " " + СокрЛП(Шапка.ГородБанка));

ДанныеПечати.Вставить("БИКБанкаПолучателя", СокрЛП(Шапка.БИКБанкаДляРасчетов));

ДанныеПечати.Вставить("СчетБанкаПолучателяПредставление", СокрЛП(Шапка.КоррСчетБанкаДляРасчетов));

ДанныеПечати.Вставить("СчетПолучателяПредставление", СокрЛП(Шапка.КоррСчетБанка));

КонецЕсли;

//...

Для Каждого ИмяОбласти Из МассивОбластейМакета Цикл

ОбластьМакета = Макет.ПолучитьОбласть(ИмяОбласти);

//...

ЗаполнитьЗначенияСвойств(ОбластьМакета.Параметры, ДанныеПечати);

ТабличныйДокумент.Вывести(ОбластьМакета);

//...

КонецЦикла;

Пример реализации печатной формы см. в демонстрационной конфигурации в документе **\_ДемоСчетНаОплатуПокупателю**.

### Настройка обмена данными

Все объекты метаданных подсистемы, содержащие данные, кроме константы **ИспользоватьАльтернативныйСерверДляЗагрузкиКлассификатораБанков** рекомендуется включать в планы обмена распределенной ИБ (РИБ).

# 3.6. Бизнес-процессы и задачи

Подсистема «Бизнес-процессы и задачи» предназначена для управления жизненным циклом задач, возникающих в системе как результат выполнения бизнес-процессов или интерактивного ввода пользователями системы. Задачи могут быть адресованы исполнителю или группе исполнителей как персонально (персональная адресация), так и с использованием ролей исполнителей (ролевая адресация). В подсистему входят пять функциональных блоков: настройка ролевой адресации, создание, исполнение, контроль и автоматический мониторинг задач.

Подсистема также предоставляет базовую функциональность, облегчающую разработку произвольных бизнес-процессов в конфигурации.

## Настройка

### Определение состава ролей исполнителей и объектов адресации в конфигурации

Необходимость в ролевой адресации обусловлена, как правило, применением в конфигурации бизнес-процессов. Подсистема «Бизнес-процессы и задачи» предоставляет разработчикам бизнес-процессов возможности по созданию ролей исполнителей, участвующих в бизнес-процессах, а также предъявляет определенные требования к разработке самих бизнес-процессов в конфигурации.

Справочник **РолиИсполнителей** содержит «плоский» список ролей исполнителей, участвующих в бизнес-процессах. Например, **Руководитель компании**, **Менеджер продаж**, **Разработчик** и т. п. Для каждой роли в этом справочнике необходимо создать предопределенный элемент, указав значения реквизитов **Имя** и **Наименование**. Значения остальных реквизитов элементов справочника должны устанавливаться в процедуре обновления информационной базы.

С подсистемой поставляется одна предопределенная роль **Координатор выполнения задач**, которая используется для получения адресатов уведомления в функциональном блоке автоматического мониторинга задач. Например, код процедуры обновления для установки реквизитов роли **Координатор выполнения задач**выглядит так:

ВсеОбъектыАдресации = ПланыВидовХарактеристик.ОбъектыАдресацииЗадач.ВсеОбъектыАдресации;

РольОбъект = Справочники.РолиИсполнителей.ОтветственныйЗаКонтрольИсполнения.ПолучитьОбъект();

РольОбъект.ИспользуетсяБезОбъектовАдресации = Истина;

РольОбъект.ИспользуетсяСОбъектамиАдресации = Истина;

РольОбъект.ТипыОсновногоОбъектаАдресации = ВсеОбъектыАдресации;

РольОбъект.Записать();

В определенных случаях указания одной только роли исполнителя бывает недостаточно, чтобы определить список исполнителей задачи. Тогда вместе с ролью нужно еще указывать основной и/или дополнительный объект адресации. Объектом адресации может быть любой справочник конфигурации, например **Проекты** или **Подразделения**. Список возможных типов объектов адресации для ролей, определенных в справочнике**РолиИсполнителей**, содержится в плане видов характеристик (ПВХ) **ОбъектыАдресацииЗадач**. Для каждого объекта адресации в этом ПВХ необходимо создать один предопределенный элемент, указав значения реквизитов **Имя**, **Наименование** и **Тип**. Например, если необходимо, чтобы пользователь вместе с ролью **Руководитель проекта** всегда указывал проект (элемент справочника **Проекты**), то нужно сделать следующее:

● создать предопределенный элемент ПВХ **ОбъектыАдресацииЗадач**, у которого в составе типов указать ссылку на справочник **Проекты**;

● в реквизите **ТипыОсновногоОбъектаАдресации** роли **Руководитель проекта** указать ссылку на этот элемент ПВХ;

● реквизит **ИспользуетсяБезОбъектовАдресации** роли **Руководитель проекта** установить в значение **Ложь**, а **ИспользуетсяСОбъектамиАдресации** – в значение **Истина**.

Не рекомендуется создавать элементы ПВХ **ОбъектыАдресацииЗадач** с составным типом данных. Вместо этого необходимо создавать несколько предметно-ориентированных ролей, каждая из которых соответствует объекту адресации одного типа. Исключение составляет предопределенный элемент **ВсеОбъектыАдресации**, который поставляется вместе с подсистемой. Состав типов предопределенного элемента «Все объекты адресации» должен всегда совпадать с составом типов ПВХ **ОбъектыАдресацииЗадач**.

Не следует создавать предопределенный элемент ПВХ **ОбъектыАдресацииЗадач**, когда используется только один объект адресации. Вместо этого необходимо у элемента **ВсеОбъектыАдресации** изменить наименование на представление этого объекта адресации (например, **Организация**) и указать его тип.

Добавление новых элементов ПВХ **ОбъектыАдресацииЗадач** или редактирование существующих в режиме **1С:Предприятие** недоступно.

При разработке бизнес-процессов, позволяющих организовывать взаимодействие с внешними пользователями (партнерами, респондентами и др. – подробнее см. раздел «[Пользователи](https://its.1c.ru/db/content/bsp23doc/src/3.37.%20%D0%BF%D0%BE%D0%BB%D1%8C%D0%B7%D0%BE%D0%B2%D0%B0%D1%82%D0%B5%D0%BB%D0%B8.htm?_=1504708101#_пользователи)»), необходимо принять решение по поводу состава ролей, доступных для выбора внешним пользователям. Например, это могут быть: **Служба техподдержки**, **Менеджер по продажам**, **Служба доставки** и т. п. Как правило, внешним пользователям недоступны имена конкретных сотрудников компании, поэтому в качестве исполнителя они могут указывать только предназначенные для них роли исполнителей. Например, код процедуры обновления для ограничения назначения роли **МенеджерПоПродажам**выглядит так:

РольОбъект = Справочники.РолиИсполнителей.МенеджерПоПродажам.ПолучитьОбъект();

РольОбъект.ИспользуетсяБезОбъектовАдресации = Истина;

  НазначениеРоли = РольОбъект.Назначение.Добавить();

  НазначениеРоли.ТипПользователей = Справочники.Партнеры.ПустаяСсылка();

  НазначениеРоли = РольОбъект.Назначение.Добавить();

  НазначениеРоли.ТипПользователей = Справочники.КонтактныеЛицаПартнеров.ПустаяСсылка();

РольОбъект.Записать();

### Настройка отложенного старта для бизнес-процессов

Подсистема также позволяет создавать бизнес-процессы, старт которых может быть отложен на определенный срок. Все бизнес-процессы, которые могут быть добавлены для отложенного старта, необходимо включить в состав определяемого типа **ОтложенныеБизнесПроцессы**. В состав данного типа по умолчанию уже включен поставляемый бизнес-процесс **Задание**.

Для настройки отложенного старта бизнес-процесса необходимо разместить в форме бизнес-процесса соответствующую команду, вызывающую метод программного интерфейса **БизнесПроцессыИЗадачиКлиент.НастроитьОтложенныйСтарт**, которая и вызывает форму настройки.

Бизнес-процессы, которым настроен отложенный старт, периодически анализируются на необходимость их старта регламентным заданием **СтартОтложенныхПроцессов**.

### Размещение в командном интерфейсе

Если в конфигурации не используется подсистема «Настройки программы», то на рабочем месте администратора программы необходимо разместить константы:

● **ИспользоватьБизнесПроцессыИЗадачи**,

● **ИспользоватьПодчиненныеБизнесПроцессы**,

● **ИзменятьЗаданияЗаднимЧислом**,

● **ИспользоватьДатуИВремяВСрокахЗадач**,

● **ИспользоватьДатуНачалаЗадач**.

См. пример в форме **Органайзер** обработки **ПанельАдминистрированияБСП**.

Для использования в конфигурации функционального блока настройки ролевой адресации необходимо:

● разместить в командном интерфейсе рабочего места администратора системы команду **РолиИИсполнителиЗадач**регистра сведений **ИсполнителиЗадач**;

● перечислить в составе типов определяемого типа **ОбъектАдресации** ссылки на справочники конфигурации, которые являются объектами адресации (состав типов команды должен совпадать с составом типов ПВХ **ОбъектыАдресацииЗадач**).

Для использования в конфигурации функционального блока создания задач необходимо:

● принять решение по поводу допустимых типов объектов метаданных, на основании которых в конфигурации будет возможно создавать задачи, буквально находясь в форме объекта. Такие объекты называются предметами задач. В качестве предметов могут выступать любые справочники, документы, планы видов характеристик и задачи;

● указать состав типов предметов задач:

* ● в реквизите **Вводится на основании** бизнес-процесса **Задание**;
* ● в свойстве **Тип**определяемого типа **ПредметЗадачи**;

● указать все типы бизнес-процессов в свойстве **Тип**определяемого типа **БизнесПроцесс**.

Для использования в конфигурации функционального блока исполнения задач необходимо:

● разместить в командном интерфейсе исполнителя задач команду **МоиЗадачи** задачи **ЗадачаИсполнителя** и отчеты **Задачи**, **ЗадачиИстекающиеНаДату**;

● для размещения списка своих задач на рабочем столе предназначена форма **МоиЗадачиДляРабочегоСтола**задачи **ЗадачаИсполнителя**.

Для использования в конфигурации функционального блока контроля исполнения необходимо разместить в командном интерфейсе:

● команду **МоиБизнесПроцессы** регистра сведений **СписокБизнесПроцессов**;

● команду **ВсеЗадачи** задачи **ЗадачаИсполнителя**;

● отчеты **ЗадачиИстекающиеНаДату**, **ПросроченныеЗадачи**, **Задачи**, **ЗависшиеЗадачи** и **БизнесПроцессы**.

### Особые случаи внедрения подсистемы

● [Внедрение подсистем «Бизнес-процессы и задачи» и «Управление доступом»](https://its.1c.ru/db/content/bsp23doc/src/3.59.%20%D0%BE%D1%81%D0%BE%D0%B1%D1%8B%D0%B5%20%D1%81%D0%BB%D1%83%D1%87%D0%B0%D0%B8%20%D0%B2%D0%BD%D0%B5%D0%B4%D1%80%D0%B5%D0%BD%D0%B8%D1%8F%20%D0%BF%D0%BE%D0%B4%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC.htm?_=1504708101#_внедрение_подсистем_).

### Настройка прав доступа пользователей

Для настройки прав доступа пользователей к данным подсистемы «Бизнес-процессы и задачи» следует использовать роли, приведенные ниже.

Таблица 3.12.

|  |  |
| --- | --- |
| **№** | **Роли и их назначение** |
| 1. | **ДобавлениеИзменениеРолейИсполнителей**  Добавление и изменение ролей исполнителей, объектов адресации. Изменение исполнителей ролей по объектам адресации. Просмотр отчетов подсистемы |
| 2. | **ПолныеПрава** (из подсистемы «Базовая функциональность»)  Включение и отключение использования бизнес-процессов и задач. Удаление помеченных на удаление объектов подсистемы |
| 3. | **ДобавлениеИзменениеЗаданий**  Добавление и изменение бизнес-процессов **Задание**. Просмотр отчетов: **Список истекающих задач**, **Список просроченных задач**, **Задачи**.  Роль содержит тексты ограничений доступа, которые используют возможности подсистемы «Управление доступом» |
| 4. | **ИзменениеВыполнениеЗадач**  Изменение и выполнение задач. Просмотр отчетов: **Список истекающих задач**, **Задачи**.  Роль содержит тексты ограничений доступа, которые используют возможности подсистемы «Управление доступом» |
| 5. | **ЧтениеЗаданий**  Чтение бизнес-процессов **Задание**. Чтение задач. Просмотр отчетов: **Список истекающих задач**, **Список просроченных задач**, **Задачи**.  Роль содержит тексты ограничений доступа, которые используют возможности подсистемы «Управление доступом» |

Дополнительно следует создать ряд ролей с текстами ограничения доступа, а также ряд вспомогательных ролей (или использовать подходящие роли, существующие в конфигурации) для обеспечения доступа к данным, которые не относятся к подсистеме «Бизнес-процессы и задачи», но требуются для работы с ней.

Таблица 3.13.

|  |  |
| --- | --- |
| **№** | **Вспомогательные роли и их назначение** |
| 1. | **<ЧтениеБизнесПроцессов>**  Роль или группа ролей для чтения других бизнес-процессов конфигурации |
| 2. | **<ДобавлениеИзменениеБизнесПроцессов>**  Роль или группа ролей для добавления и изменения других бизнес-процессов конфигурации |
| 3. | **<ДобавлениеИзменениеИсполнителейРолейПоОбъектамАдресации>**  Изменение исполнителей ролей по разрешенным объектам адресации. Просмотр отчетов подсистемы. Пример настройки прав к объектам метаданных см. в демонстрационной конфигурации: роль **\_ДемоДобавлениеИзменениеИсполнителейРолейПоОбъектамАдресации** |

Примеры настройки прав доступа пользователей приведены ниже.

Таблица 3.14.

| **№** | **Группа пользователей и ее функции** | **Состав ролей** |
| --- | --- | --- |
| 1. | Администратор | ● **ПолныеПрава**(из подсистемы «Базовая функциональность») |
| 2. | Ответственный за ведение списка ролей и назначение на них исполнителей | ● **БазовыеПрава** (из подсистемы «Базовая функциональность»),  ● **ЗапускТонкогоКлиента** (из подсистемы «Базовая функциональность»),  **● ДобавлениеИзменениеРолейИсполнителей** |
| 3. | Руководитель подразделения:  ● назначение исполнителей на роли в своем подразделении,  ● выдача заданий,  ● контроль выполнения задач,  ● просмотр отчетов по задачам подразделения | ● **БазовыеПрава**(из подсистемы «Базовая функциональность»),  ● **ЗапускТонкогоКлиента** (из подсистемы «Базовая функциональность»),  ●**<ДобавлениеИзменениеИсполнителейРолейПоОбъектамАдресации>**,  ● **ДобавлениеИзменениеЗаданий**,  ● **ИзменениеВыполнениеЗадач**,  **● <ДобавлениеИзменениеБизнесПроцессов>** |
| 4. | Сотрудник подразделения:  ● исполнение задач,  ● просмотр отчетов по своим задачам | ● **БазовыеПрава** (из подсистемы «Базовая функциональность»),  ● **ЗапускТонкогоКлиента** (из подсистемы «Базовая функциональность»),  ● **ЧтениеЗаданий**,  ● **ИзменениеВыполнениеЗадач**,  **● <ЧтениеБизнесПроцессов>** |

## Использование при разработке конфигурации

Подсистема предоставляет базовую функциональность для разработки произвольных бизнес-процессов в конфигурации.

### Требования к метаданным бизнес-процессов

Для унификации бизнес-процессов по общим реквизитам необходимо соблюдать следующие требования к метаданным бизнес-процессов:

● реквизит **Номер** бизнес-процесса должен быть типа **Строка**, длина – 11;

● у бизнес-процесса должны быть определены обязательные реквизиты:

* ● **Автор** (**СправочникСсылка.Пользователи**) – ответственный за данный экземпляр бизнес-процесса. Например, для бизнес-процесса **Задание** автором по умолчанию является пользователь, инициировавший бизнес-процесс. Если бизнес-процесс позволяет организовать взаимодействие с внешними пользователями (партнерами, респондентами и др.), то реквизит **Автор** может быть типа **СправочникСсылка, ВнешниеПользователи**или составного типа:**СправочникСсылка.Пользователи** и**СправочникСсылка.ВнешниеПользователи**;
* ● **ДатаЗавершения** (**Дата**) – дата фактического завершения бизнес-процесса;
* ● **Наименование** (**Строка**, **250**, переменная длина) – текстовое описание экземпляра бизнес-процесса. Например, в бизнес-процессе **Задание** в качестве наименования выступает текст задачи;

● в свойстве **Задачи** бизнес-процесса следует указывать задачу **ЗадачаИсполнителя**;

● в свойство **Поля блокировки данных** необходимо включить стандартный реквизит **ГлавнаяЗадача**;

● если предполагается использовать механизмы остановки и продолжения бизнес-процессов, то в бизнес-процессе следует определить реквизит **Состояние** (**ПеречислениеСсылка.СостоянияБизнесПроцессов**) и разместить на форме бизнес-процесса команды остановки и продолжения по аналогии с бизнес-процессом **Задание**;

● если предполагается использовать иерархию бизнес-процессов, то в бизнес-процессе следует определить реквизит **ГлавнаяЗадача** (**ЗадачаСсылка.ЗадачаИсполнителя**) и вывести его на форме бизнес-процесса по аналогии с бизнес-процессом **Задание**. Реквизит **ГлавнаяЗадача** необходимо включить в свойство **Поля блокировки данных**.

В модуле менеджера бизнес-процесса должны быть определены три экспортные процедуры и функции:

● **ФормаВыполненияЗадачи**,

● **ПриПеренаправленииЗадачи**,

● **ОбработкаВыполненияПоУмолчанию**.

Назначение, описание и пример их использования см. в модуле менеджера бизнес-процесса **Задание**.

Список всех бизнес-процессов, в модулях менеджеров которых определены эти процедуры и функции, необходимо перечислить в процедуре **ПриОпределенииБизнесПроцессов** общего модуля **БизнесПроцессыИЗадачиПереопределяемый**.

Если бизнес-процесс связан с некоторым объектом-основанием, в разрезе которого должен ограничиваться доступ к экземплярам бизнес-процесса, то такой бизнес-процесс должен иметь реквизит со ссылкой на объект-основание. Например, это может быть реквизит **Предмет** составного типа, как у бизнес-процесса **Задание**.

Для вывода понятных представлений бизнес-процессов в виде «Согласовать техпроект по Такси от 01.04.2014 (Задание)» следует также создать две подписки на события менеджеров бизнес-процессов **ОбработкаПолученияПолейПредставления** и **ОбработкаПолученияПредставления**. В качестве обработчиков подписок указать процедуры общего модуля **БизнесПроцессыИЗадачиКлиентСервер**:

● **ОбработкаПолученияПолейПредставленияБизнесПроцесса**,

● **ОбработкаПолученияПредставленияБизнесПроцесса**.

В качестве примера см. подписки на события **ПолучитьПоляПредставленияБизнесПроцесса** и **ПолучитьПредставлениеБизнесПроцесса**.

### Разработка прикладных форм выполнения задач

Бизнес-процессы могут подменять форму выполнения задач по умолчанию, которая реализована в задаче **ЗадачаИсполнителя**, и использовать свою прикладную форму. Для создания удобного пользовательского интерфейса исполнителя задачи рекомендуется использовать прикладные формы, сгенерированные в точках действия карты маршрута бизнес-процесса.

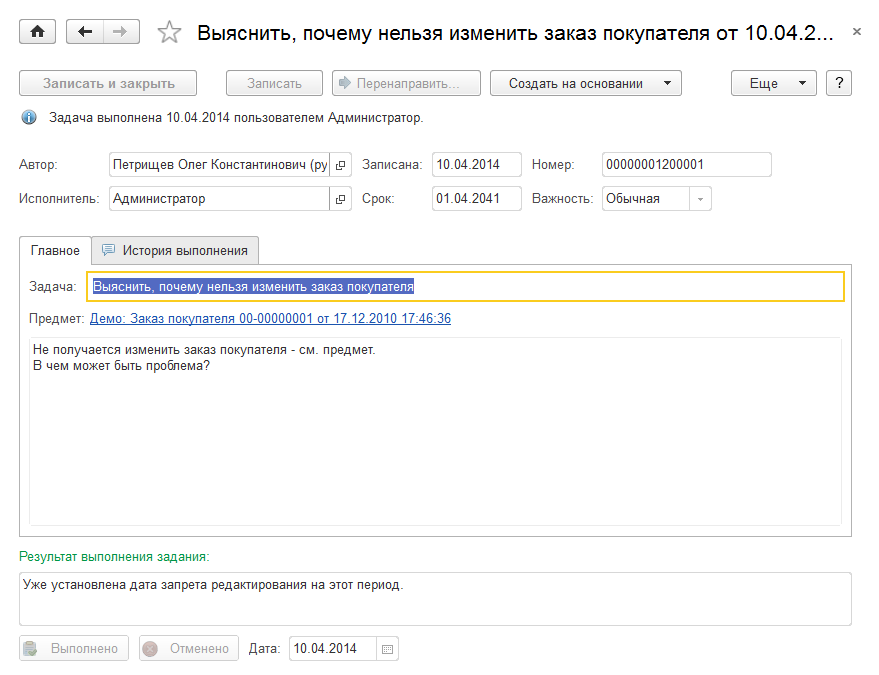
Внешний вид прикладной формы выполнения задачи рекомендуется выстраивать сверху вниз по порядку:

1. Опциональное информационное поле с информацией о дате и результате выполнения задачи. Поле спрятано, если задача еще выполняется.

2. Информация об отправителе задачи (как правило, это поле с автором бизнес-процесса), общие параметры выполнения задачи, такие как **Крайний срок**, **Важность**, **Дата начала**.

3. Содержание задачи, которое может включать текстовое описание постановки задачи, ссылку на предмет задачи, а также другие реквизиты бизнес-процесса (группа **Содержание**).

4. Результат выполнения задачи, который обязательно включает поля с исполнителем и датой фактического выполнения задачи. Результат может быть многовариантным; представляться как текстовым описанием, так и специализированными элементами управления формы (флажки, переключатели, списки выбора и т. п.). В этой же группе в правом нижнем углу формы также следует размещать в виде кнопок команды для выбора варианта выполнения задачи. Например: **Выполнено**, **Отменено**, **Утверждено**, **Отработано** и т. п. Выполнение такой команды приводит к закрытию формы задачи и переходу бизнес-процесса к следующей точке маршрута.



Примеры прикладных форм можно посмотреть в базовом бизнес-процессе **Задание**.

Для подключения прикладных форм выполнения задач в модуле менеджера бизнес-процесса необходимо определить экспортную функцию-обработчик **ФормаВыполненияЗадачи**. Функция **ФормаВыполненияЗадачи**вызывается каждый раз перед открытием формы задачи. Возвращаемое значение функции – структура с ключами **ИмяФормы** и **ПараметрыФормы**, значения которых используются для открытия формы с помощью метода контекста **ОткрытьФорму**.

Пример реализации функции **ФормаВыполненияЗадачи**:

Функция ФормаВыполненияЗадачи(ЗадачаСсылка, ТочкаМаршрутаСсылка) Экспорт

Результат = Новый Структура;

Если ТочкаМаршрутаСсылка = БизнесПроцессы.Задание.ТочкиМаршрута.Выполнить Тогда

Результат.Вставить("ПараметрыФормы", Новый Структура("Ключ", ЗадачаСсылка));

Результат.Вставить("ИмяФормы", "БизнесПроцесс.Поручение.Форма.Выполнить");

КонецЕсли;

Возврат Результат;

КонецФункции

Дополнительные рекомендации по разработке метаданных и прикладных форм выполнения задач:

● Рекомендуется давать имена формам выполнения задач по определенному шаблону: **Действие<ИмяТочкиМаршрута>**. Например, **ДействиеВыполнить**.

● Прикладная форма выполнения задачи обязательно должна содержать основной реквизит **Объект** типа **ЗадачаОбъект.ЗадачаИсполнителя**, а также может содержать произвольный набор дополнительных реквизитов, среди которых, как правило, есть реквизиты бизнес-процесса и объектов, связанных с процессом выполнения задачи.

● Если в форме присутствуют реквизиты бизнес-процесса (или других связанных объектов), то чтение и запись этих реквизитов рекомендуется выполнять в привилегированном режиме. Это позволит работать с формой задачи тем пользователям, которые при своей повседневной работе с системой не имеют прав на чтение или запись реквизитов бизнес-процесса, но, тем не менее, косвенно должны иметь к ним доступ из формы задачи.

Пример:

&НаСервере

Процедура ПрочитатьРеквизитыБизнесПроцесса()

ЗадачаОбъект = РеквизитФормыВЗначение("Объект");

УстановитьПривилегированныйРежим(Истина);

ЗаданиеОбъект = ЗадачаОбъект.БизнесПроцесс.ПолучитьОбъект();

ЗаданиеВыполнено = ЗаданиеОбъект.Выполнено;

ЗаданиеРезультатВыполнения = ЗаданиеОбъект.РезультатВыполнения;

ЗаданиеСодержание = ЗаданиеОбъект.Содержание;

КонецПроцедуры

&НаСервере

Процедура ЗаписатьРеквизитыБизнесПроцесса(ЗадачаОбъект)

УстановитьПривилегированныйРежим(Истина);

ЗаданиеОбъект = ЗадачаОбъект.БизнесПроцесс.ПолучитьОбъект();

ЗаблокироватьДанныеДляРедактирования(ЗаданиеОбъект.Ссылка);

ЗаданиеОбъект.Выполнено = ЗаданиеВыполнено;

ЗаданиеОбъект.Записать();

КонецПроцедуры

### Настройка обмена данными

Все объекты метаданных подсистемы, содержащие данные, рекомендуется включать в планы обмена распределенной ИБ (РИБ). Исключение из этого правила:

● Константу **ДатаУведомленияОНовыхЗадачах** не следует включать в планы обмена РИБ, если требуется независимо уведомлять исполнителей о новых задачах в узлах РИБ.

# 3.7. Валюты

Подсистема «Валюты» предназначена для загрузки, хранения и предоставления другим подсистемам конфигурации списка валют и их курсов. Загрузка курсов валют может производиться как автоматически (при помощи регламентного задания), так и вручную (через пользовательский интерфейс справочника **Валюты**).

## Настройка

Для использования подсистемы в конфигурации необходимо разместить в командном интерфейсе методиста-настройщика ссылку на справочник **Валюты**.

Право редактирования всех объектов подсистемы рекомендуется предоставить тем ролям конфигурации, которые отвечают за заполнение справочников нормативно-справочной информации.

Для автоматической загрузки курсов валют с сайта 1С предназначено регламентное задание **ЗагрузкаКурсовВалют**.

### Настройка прав доступа пользователей

Для настройки прав доступа пользователей к данным подсистемы «Валюты» следует использовать роли, приведенные ниже.

Таблица 3.15.

| **№** | **Роли и их назначение** |
| --- | --- |
| 1. | **ЧтениеКурсовВалют**  Чтение валют и курсов |
| 2. | **ДобавлениеИзменениеКурсовВалют**  Добавление и изменение валют и курсов. Загрузка курсов валют |
| 3. | **ПолныеПрава** (из подсистемы «Базовая функциональность»)  Удаление помеченных на удаление объектов подсистемы |

### Использование при разработке конфигурации

Список валют и их курсы хранятся в справочнике **Валюты** и регистре сведений **Курсы валют** соответственно.

### Настройка обмена данными

Все объекты метаданных подсистемы, содержащие данные, кроме константы **ИспользоватьАльтернативныйСерверДляЗагрузкиКурсовВалют**, рекомендуется включить в планы обмена распределенной ИБ (РИБ).

# 3.8. Варианты отчетов

Подсистема «Варианты отчетов» отображает все отчеты раздела на панели отчетов, предоставляет совместный доступ к пользовательским вариантам отчетов и предоставляет универсальную форму для отчетов конфигурации.

Варианты отчетов, созданные разработчиком (через конфигуратор), называются предопределенными, а сохраненные пользователем (в режиме **1С:Предприятие**) – пользовательскими.

Предопределенные варианты отчетов могут быть контекстными или неконтекстными:

● контекстные варианты отчетов могут использоваться только в контексте работы с каким-либо объектом программы (например, отчет по сотруднику, подразделению и т. п.). Как правило, такие варианты отчетов открываются из формы объекта;

● неконтекстные варианты отчетов могут одинаково открываться из любого места программы (например, из панели отчетов), не требуя передачи контекста. Как правило, такие варианты отчетов открываются из командного интерфейса.

Начальные настройки предопределенных вариантов отчетов зачитываются из метаданных, и их можно изменить на этапе внедрения подсистемы. При размещении отчетов в подсистемах конфигурации рекомендуется придерживаться рекомендаций стандарта «[Группировка команд](https://its.1c.ru/db/v8std/content/2149184223/hdoc)» (на сайте ИТС).

Размещение варианта отчета на панели отчетов можно изменить при сохранении или из его карточки. Настройки размещения выводятся в виде дерева подсистем с флажком использования и важностью.

На панели отчетов раздела каждый пользователь может изменить настройки видимости и быстрого доступа к вариантам отчетов под себя, переключившись в режим настройки.

Также на панели отчетов предусмотрен поиск по свойствам отчета: наименованию, описанию, наименованиям подсистем (разделов и групп командного интерфейса), полей, отборов и пользовательских настроек.

## Настройка

После завершения переноса объектов подсистемы необходимо выполнить ее настройку:

1. Подключить хранилище вариантов отчетов.

2. Подключить форму отчета.

3. Подключить подсистему к разделам командного интерфейса.

4. Изменить настройки вариантов отчетов.

5. Подключить обработчики полного обновления.

### Подключить хранилище вариантов отчетов

Необходимо принять решение по поводу варианта внедрения хранилища подсистемы:

1. Если конфигурация ранее не выпускалась (т. е. пользователи не сохраняли в своих базах никаких вариантов отчетов), то следует выбрать один из двух способов внедрения:

* ● **Полный**, если к подсистеме необходимо подключить все отчеты конфигурации, а также все внешние (дополнительные) отчеты. Для этого необходимо установить хранилище настроек **ХранилищеВариантовОтчетов** в свойстве конфигурации «Хранилище вариантов отчетов»;
* ● **Частичный**, если необходимо подключить подсистему выборочно только для некоторой части отчетов. Для этого необходимо установить хранилище настроек **ХранилищеВариантовОтчетов** в свойстве **Хранилище вариантов отчетов** каждого отчета, который нужно подключить к хранилищу подсистемы.

2. Если конфигурация уже выпускалась, т. е. существует вероятность того, что пользователи сохраняли в своих базах варианты отчетов, то следует использовать только **Частичный способ внедрения**. В этом случае ранее сохраненные пользователями варианты отчетов будут перенесены в хранилище подсистемы автоматически.

В противном случае, если в такой ситуации был выбран сценарий внедрения **Полный**, подсистема не сможет автоматически перенести ранее сохраненные пользователями варианты отчетов. В этом случае необходимо:

* ● включить в сопроводительную документацию к конфигурации инструкцию (или донести эту информацию другими способами) для администратора (пользователя, который выполняет обновление конфигурации), что он должен запустить обработку перехода перед обновлением;
* ● разработать и включить в состав дистрибутива обработку перехода. Пример подобной обработки входит в состав демонстрационной конфигурации – **ИнструментыРазработчикаВариантыОтчетовПереносВариантовОтчетов.epf**.

### Подключить форму отчета

Необходимо принять решение по поводу варианта внедрения формы отчета:

● **Полный**, если к подсистеме необходимо подключить все отчеты без основной формы. Для этого необходимо указать в свойствах конфигурации общие формы, поставляемые подсистемой:

* ● основная форма отчета – **ФормаОтчета**;
* ● основная форма настроек отчета – **ФормаНастроекОтчета**;
* ● основная форма варианта отчета – **ФормаВариантаОтчета**;

● **Частичный**, если необходимо подключить подсистему выборочно только для части отчетов без основной формы. Для этого необходимо указать в свойствах каждого отчета общие формы, поставляемые подсистемой:

* ● основная форма – **ФормаОтчета**;
* ● основная форма настроек – **ФормаНастроекОтчета**;
* ● основная форма варианта – **ФормаВариантаОтчета**.

Внимание: не поддерживается ситуация, когда отчет подключен только к некоторым из указанных форм. В частности, в случае «Полного» варианта внедрения, если для отчета на СКД определена собственная основная форма, то для него также следует определить собственную форму настроек (либо в свойствах этого отчета указать общую форму **ВспомогательнаяФормаНастроекОтчета**). Это требование также распространяется на дополнительные и внешние отчеты, использующиеся в конфигурации с «Полным» вариантом внедрения формы отчета.

В формах отчетов, которые не подключены к общей форме **ФормаОтчета**, необходимо определить обработчик события **ПриСохраненииПользовательскихНастроекНаСервере** и добавить в него код по шаблону:

&НаСервере

Процедура ПриСохраненииПользовательскихНастроекНаСервере(Настройки)

ВариантыОтчетов.ПриСохраненииПользовательскихНастроекНаСервере(ЭтотОбъект, Настройки);

КонецПроцедуры

В противном случае названия настроек этих отчетов не будут находиться системой поиска вариантов отчетов.

### Подключить подсистему к разделам командного интерфейса

Необходимо определить разделы командного интерфейса, в которых должны быть размещены панели отчетов. Рекомендуется выбрать все разделы, для которых предусмотрены отчеты.

Для каждого раздела необходимо создать отдельную общую команду для вызова панели отчетов:

● Имя команды рекомендуется задавать по шаблону **ПанельОтчетовИмяРаздела**. Синоним команды рекомендуется заполнить по принципу: «Отчеты по <ПредставлениеРаздела>». Например: «Отчеты по работе с файлами».

● В обработчике команды необходимо вставить вызов по шаблону:

&НаКлиенте

Процедура ОбработкаКоманды(ПараметрКоманды, ПараметрыВыполнения)

ВариантыОтчетовКлиент.ПоказатьПанельОтчетов("ИмяРаздела", ПараметрыВыполнения);

КонецПроцедуры

Выбранные разделы необходимо перечислить в процедуре **ОпределитьРазделыСВариантамиОтчетов** общего модуля **ВариантыОтчетовПереопределяемый**.

Предусмотрена возможность выводить панель отчетов в панели функций начальной страницы (рабочего стола):

● Создать общую команду, например **\_ДемоПанельОтчетовНачальнаяСтраница**, и размесить ее в командном интерфейсе основного раздела. В обработчике команды необходимо вставить вызов по шаблону:

ВариантыОтчетовКлиент.ПоказатьПанельОтчетов(ВариантыОтчетовКлиентСервер.ИдентификаторНачальнойСтраницы(), ПараметрыВыполнения);

● А в модуле менеджера отчета разместить процедуру по шаблону:

Процедура НастроитьВариантыОтчета(Настройки, НастройкиОтчета) Экспорт

НастройкиОтчета.Размещение.Вставить(ВариантыОтчетовКлиентСервер.ИдентификаторНачальнойСтраницы(), "Важный");

НастройкиВарианта = ВариантыОтчетов.ОписаниеВарианта(Настройки, НастройкиОтчета, "СчетаНаОплату");

НастройкиВарианта.Описание = НСтр("ru = 'Список счетов на оплату.'");

КонецПроцедуры

В заключение рекомендуется скрыть отчеты, подключенные к подсистеме, из панели действий выбранных разделов командного интерфейса.

### Изменить настройки вариантов отчетов

Настройки вариантов отчетов задаются в процедуре **НастроитьВариантыОтчетов** общего модуля **ВариантыОтчетовПереопределяемый**. Настройки можно менять непосредственно в переопределяемом модуле (удобно в небольших конфигурациях) или в модуле менеджера отчета (удобно при совместной разработке и больше соответствует библиотечному подходу). Для этого в процедуре **НастроитьВариантыОтчетов** общего модуля **ВариантыОтчетовПереопределяемый** следует разместить вызов модуля менеджера отчета по шаблону:

ВариантыОтчетов.НастроитьОтчетВМодулеМенеджера(Настройки, Метаданные.Отчеты.ИмяОтчета);

А в модуле менеджера отчета разместить процедуру по шаблону:

#Если Сервер Или ТолстыйКлиентОбычноеПриложение Или ВнешнееСоединение Тогда

#Область СлужебныйПрограммныйИнтерфейс

// Настройки размещения в панели отчетов.

//

// Параметры:

//   Настройки – Коллекция – Используется для описания настроек отчетов и вариантов

//       см. описание к ВариантыОтчетов.ДеревоНастроекВариантовОтчетовКонфигурации().

//   НастройкиОтчета – СтрокаДереваЗначений – Настройки размещения всех вариантов отчета.

//       См. "Реквизиты для изменения" функции ВариантыОтчетов.ДеревоНастроекВариантовОтчетовКонфигурации().

//

// Описание:

//   См. ВариантыОтчетовПереопределяемый.НастроитьВариантыОтчетов().

//

// Вспомогательные методы:

//   НастройкиВарианта = ВариантыОтчетов.ОписаниеВарианта(Настройки, НастройкиОтчета, "<ИмяВарианта>");

//   ВариантыОтчетов.УстановитьРежимВыводаВПанеляхОтчетов(Настройки, НастройкиОтчета, Истина/Ложь);

// Отчет

//   поддерживает только этот режим.

//

Процедура НастроитьВариантыОтчета(Настройки, НастройкиОтчета) Экспорт

НастройкиВарианта = ВариантыОтчетов.ОписаниеВарианта(Настройки, НастройкиОтчета, "ИмяВарианта");

КонецПроцедуры

#КонецОбласти

#КонецЕсли

Объекты **НастройкиОтчета** и **НастройкиВарианта** содержат следующие свойства, которые можно менять:

● **Включен** – если установить в **Ложь**, то вариант отчета не регистрируется в справочнике и как следствие не выводится в панелях отчетов и других формах подсистемы (всегда скрыт от пользователей). При этом вариант отчета остается в конфигурации, и его по-прежнему можно открывать при программном открытии формы отчета. Например, может потребоваться отключение контекстных вариантов отчетов.

● **Описание** – служит подсказкой на панели отчетов и содержит более подробное описание варианта отчета. **Описание** можно не заполнять для контекстных вариантов отчетов (которые не выводятся в панелях и списках отчетов). В противном случае **Описание** обязательно для заполнения.

● **ВидимостьПоУмолчанию** – если установить в **Ложь**, то вариант отчета по умолчанию будет скрыт на панелях отчетов. Скрытый вариант отчета можно вывести на панель отчетов администратору – сразу для всех пользователей (через форму элемента), пользователю – на свою панель отчетов (через режим настройки). Например, может потребоваться скрывать менее частотные варианты отчетов.

● **Описание** – служит подсказкой на панели отчетов и содержит более подробное описание варианта отчета.

● **Размещение** – служит для привязки вариантов отчетов к подсистемам конфигурации. Подсистемы первого уровня считаются разделами, второго и выше – группами. Если вариант включен в раздел, то на панели отчетов он будет выведен без группы. При необходимости для каждой подсистемы можно указать **Важность**:

* ● **Важный** – вариант отчета будет выделен жирным шрифтом и размещен в начале группы.
* ● **СмТакже** – вариант отчета будет выведен внизу панели отчетов, в группе **См. также**.

● **ФункциональныеОпции** – служит для скрытия вариантов отчетов по функциональным опциям. Содержит массив строк из имен функциональных опций конфигурации. Вариант отчета считается включенным в том случае, если включена одна из функциональных опций отчета (если они указаны) и одна из функциональных опций варианта отчета (если они указаны).

● **НастройкиДляПоиска** – дополнительные сведения для поиска этого варианта отчета. Является структурой со следующими свойствами:

* ● **НаименованияПолей** – представления полей варианта отчета;
* ● **НаименованияПараметровИОтборов** – представления настроек варианта отчета;
* ● **КлючевыеСлова**– дополнительная терминология (в т.ч. специализированная или устаревшая).

Разделитель терминов: **Символы.ПС**.

● **ОпределитьНастройкиФормы** – включает переопределение настроек формы отчета. Внимание: свойство доступно только для объекта **НастройкиОтчета**. Если установить в значение **Истина**, то в модуле объекта отчета следует определить процедуру **ОпределитьНастройкиФормы** по шаблону:

#Если Сервер Или ТолстыйКлиентОбычноеПриложение Или ВнешнееСоединение Тогда

#Область ПрограммныйИнтерфейс

// Настройки общей формы отчета подсистемы "Варианты отчетов".

//

// Параметры:

//   Форма - УправляемаяФорма, Неопределено - Форма отчета или форма настроек отчета.

//       Неопределено когда вызов без контекста.

//   КлючВарианта - Строка, Неопределено - Имя предопределенного

//       или уникальный идентификатор пользовательского варианта отчета.

//       Неопределено когда вызов без контекста.

//   Настройки - Структура - см. возвращаемое значение

//       ОтчетыКлиентСервер.ПолучитьНастройкиОтчетаПоУмолчанию().

//

Процедура ОпределитьНастройкиФормы(Форма, КлючВарианта, Настройки) Экспорт

КонецПроцедуры

#КонецОбласти

#КонецЕсли

Подробнее про изменение настроек формы отчета см. раздел «[Изменить настройки формы отчета](https://its.1c.ru/db/content/bsp23doc/src/3.8.%20%D0%B2%D0%B0%D1%80%D0%B8%D0%B0%D0%BD%D1%82%D1%8B%20%D0%BE%D1%82%D1%87%D0%B5%D1%82%D0%BE%D0%B2.htm?_=1504708101#_изменить_настройки_формы)».

### Изменить настройки формы отчета

Настройки формы отчета задаются в процедуре **ОпределитьНастройкиФормы** модуля объекта отчета. Как определить эту процедуру, уже описано выше. Настройки для изменения передаются трем параметрам этой процедуры и содержат следующие свойства:

● **ФормироватьСразу** – если установить в значение **Истина**, то отчет будет формироваться после открытия, после выбора пользовательских настроек и после выбора другого варианта отчета. По умолчанию – **Ложь**.

● **ВыводитьСуммуВыделенныхЯчеек** – если установить в значение **Ложь**, то в отчете не будет выводиться поле с автоматически рассчитываемой суммой выделенных ячеек. По умолчанию – **Истина**.

● **Печать – Структура** – параметры печати табличного документа (могут переопределяться пользователем):

* ● **ПолеСверху – Число** – отступ сверху при печати (в миллиметрах);
* ● **ПолеСлева – Число** – отступ слева при печати (в миллиметрах);
* ● **ПолеСнизу – Число** – отступ снизу при печати (в миллиметрах);
* ● **ПолеСправа – Число** – отступ справа при печати (в миллиметрах);
* ● **ОриентацияСтраницы – ОриентацияСтраницы** – «Портрет» или «Ландшафт»;
* ● **АвтоМасштаб – Булево** – автоматически подгонять масштаб под размер страницы;
* ● **МасштабПечати – Число** – масштаб изображения (в процентах).

● **События – Структура** – события, для которых определены обработчики в модуле объекта отчета (обработчики событий рекомендуется определять в области **ПрограммныйИнтерфейс**, после процедуры **ОпределитьНастройкиФормы**):

* ● **ПриСозданииНаСервере** – если установить в значение **Истина**, то в модуле объекта отчета следует определить процедуру **ПриСозданииНаСервере** по шаблону:

// Вызывается в обработчике одноименного события формы отчета после выполнения кода формы.

//

// Параметры:

//   Форма - УправляемаяФорма - Форма отчета.

//   Отказ - Передается из параметров обработчика "как есть".

//   СтандартнаяОбработка - Передается из параметров обработчика "как есть".

//

// См. также:

//   "УправляемаяФорма.ПриСозданииНаСервере" в синтакс-помощнике.

//

// Пример 1 - Добавление команды с обработчиком в "ОтчетыКлиентПереопределяемый.ОбработчикКоманды":

// Команда = Форма.Команды.Добавить("МояОсобеннаяКоманда");

// Команда.Действие = "Подключаемый\_Команда";

// Команда.Заголовок = НСтр("ru = 'Моя команда...'");

//

// Кнопка = Форма.Элементы.Добавить(Команда.Имя, Тип("КнопкаФормы"), Форма.Элементы.<ИмяПодменю>);

// Кнопка.ИмяКоманды = Команда.Имя;

//

// Форма.ПостоянныеКоманды.Добавить(КомандаСоздать.Имя);

//

Процедура ПриСозданииНаСервере(Форма, Отказ, СтандартнаяОбработка) Экспорт

// Обработка события.

КонецПроцедуры

* ● **ПередЗагрузкойНастроекВКомпоновщик**– если установить в значение **Истина**, то в модуле объекта отчета следует определить процедуру **ПередЗагрузкойНастроекВКомпоновщик**по шаблону:

// Вызывается перед загрузкой новых настроек. Используется для изменения схемы компоновки.

//   Например, если схема отчета зависит от ключа варианта или параметров отчета.

//   Чтобы изменения схемы вступили в силу следует вызывать метод ОтчетыСервер.ПодключитьСхему().

//

// Параметры:

//   Контекст - Произвольный -

//       Параметры контекста, в котором используется отчет.

//       Используется для передачи в параметрах метода ОтчетыСервер.ПодключитьСхему().

//   КлючСхемы - Строка -

//       Идентификатор текущей схемы компоновщика настроек.

//       По умолчанию не заполнен (это означает что компоновщик инициализирован на основании основной схемы).

//       Используется для оптимизации, чтобы переинициализировать компоновщик как можно реже).

//       Может не использоваться, если переинициализация выполняется безусловно.

//   КлючВарианта - Строка, Неопределено -

//       Имя предопределенного или уникальный идентификатор пользовательского варианта отчета.

//       Неопределено, когда вызов для варианта расшифровки или без контекста.

//   НовыеНастройкиКД - НастройкиКомпоновкиДанных, Неопределено -

//       Настройки варианта отчета, которые будут загружены в компоновщик настроек после его инициализации.

//       Неопределено, когда настройки варианта не надо загружать (уже загружены ранее).

//   НовыеПользовательскиеНастройкиКД - ПользовательскиеНастройкиКомпоновкиДанных, Неопределено -

//       Пользовательские настройки, которые будут загружены в компоновщик настроек после его инициализации.

//       Неопределено, когда пользовательские настройки не надо загружать (уже загружены ранее).

//

// Примеры:

// 1. Компоновщик отчета инициализируется на основании схемы из общих макетов:

// Если КлючСхемы <> "1" Тогда

// КлючСхемы = "1";

// СхемаКД = ПолучитьОбщийМакет("МояОбщаяСхемаКомпоновки");

// ОтчетыСервер.ПодключитьСхему(ЭтотОбъект, Контекст, СхемаКД, КлючСхемы);

// КонецЕсли;

//

// 2. Схема зависит от значения параметра, выведенного в пользовательские настройки отчета:

// Если ТипЗнч(НовыеПользовательскиеНастройкиКД) = Тип("ПользовательскиеНастройкиКомпоновкиДанных") Тогда

// ПолноеИмяОбъектаМетаданных = "";

// Для Каждого ЭлементКД Из НовыеПользовательскиеНастройкиКД.Элементы Цикл

// Если ТипЗнч(ЭлементКД) = Тип("ЗначениеПараметраНастроекКомпоновкиДанных") Тогда

// ИмяПараметра = Строка(ЭлементКД.Параметр);

// Если ИмяПараметра = "ОбъектМетаданных" Тогда

// ПолноеИмяОбъектаМетаданных = ЭлементКД.Значение;

// КонецЕсли;

// КонецЕсли;

// КонецЦикла;

// Если КлючСхемы <> ПолноеИмяОбъектаМетаданных Тогда

// КлючСхемы = ПолноеИмяОбъектаМетаданных;

// СхемаКД = Новый СхемаКомпоновкиДанных;

// // Наполнение схемы...

// ОтчетыСервер.ПодключитьСхему(ЭтотОбъект, Контекст, СхемаКД, КлючСхемы);

// КонецЕсли;

// КонецЕсли;

//

Процедура ПередЗагрузкойНастроекВКомпоновщик(Контекст, КлючСхемы, КлючВарианта, НовыеНастройкиКД, НовыеПользовательскиеНастройкиКД) Экспорт

// Обработка события.

КонецПроцедуры

* ● **ПередЗагрузкойВариантаНаСервере** – если установить в значение **Истина**, то в модуле объекта отчета следует определить процедуру **ПередЗагрузкойВариантаНаСервере** по шаблону:

// Вызывается в обработчике одноименного события формы отчета после выполнения кода формы.

//

// Параметры:

//   Форма - УправляемаяФорма - Форма отчета.

//   НовыеНастройкиКД - НастройкиКомпоновкиДанных - Настройки для загрузки в компоновщик настроек.

//

// См. также:

//   "Расширение управляемой формы для отчета.ПриЗагрузкеВариантаНаСервере" в синтакс-помощнике.

//

Процедура ПередЗагрузкойВариантаНаСервере(Форма, НовыеНастройкиКД) Экспорт

// Обработка события.

КонецПроцедуры

* ● **ПриЗагрузкеВариантаНаСервере** – если установить в значение **Истина**, то в модуле объекта отчета следует определить процедуру **ПриЗагрузкеВариантаНаСервере** по шаблону:

// Вызывается в обработчике одноименного события формы отчета после выполнения кода формы.

//

// Параметры:

//   Форма - УправляемаяФорма - Форма отчета.

//   НовыеНастройкиКД - НастройкиКомпоновкиДанных - Настройки для загрузки в компоновщик настроек.

//

// См. также:

//   "Расширение управляемой формы для отчета.ПриЗагрузкеВариантаНаСервере" в синтакс-помощнике.

//

Процедура ПриЗагрузкеВариантаНаСервере(Форма, НовыеНастройкиКД) Экспорт

// Обработка события.

КонецПроцедуры

* ● **ПриЗагрузкеПользовательскихНастроекНаСервере** – если установить в значение **Истина**, то в модуле объекта отчета следует определить процедуру **ПриЗагрузкеПользовательскихНастроекНаСервере** по шаблону:

// Вызывается в обработчике одноименного события формы отчета после выполнения кода формы.

//

// Параметры:

//   Форма - УправляемаяФорма - Форма отчета.

//   НовыеПользовательскиеНастройкиКД - ПользовательскиеНастройкиКомпоновкиДанных -

//       Пользовательские настройки для загрузки в компоновщик настроек.

//

// См. также:

//   "Расширение управляемой формы для отчета.ПриЗагрузкеПользовательскихНастроекНаСервере"

//    в синтакс-помощнике.

//

Процедура ПриЗагрузкеПользовательскихНастроекНаСервере(Форма, НовыеПользовательскиеНастройкиКД) Экспорт

// Обработка события.

КонецПроцедуры

* ● **ПередЗаполнениемПанелиБыстрыхНастроек** – если установить в значение **Истина**, то в модуле объекта отчета следует определить процедуру **ПередЗаполнениемПанелиБыстрыхНастроек** по шаблону:

// Вызывается до перезаполнения панели настроек формы отчета.

//

// Параметры:

//   Форма - УправляемаяФорма - Форма отчета.

//   ПараметрыЗаполнения - Структура - Параметры, которые будут загружены в отчет.

//

Процедура ПередЗаполнениемПанелиБыстрыхНастроек(Форма, ПараметрыЗаполнения) Экспорт

// Обработка события.

КонецПроцедуры

* ● **ПослеЗаполненияПанелиБыстрыхНастроек** – если установить в значение **Истина**, то в модуле объекта отчета следует определить процедуру **ПослеЗаполненияПанелиБыстрыхНастроек** по шаблону:

// Вызывается после перезаполнения панели настроек формы отчета.

//

// Параметры:

//   Форма - УправляемаяФорма - Форма отчета.

//   ПараметрыЗаполнения - Структура - Параметры, которые будут загружены в отчет.

//

Процедура ПослеЗаполненияПанелиБыстрыхНастроек(Форма, ПараметрыЗаполнения) Экспорт

// Обработка события.

КонецПроцедуры

* ● **КонтекстныйВызовСервера** – если установить в значение **Истина**, то в модуле объекта отчета следует определить процедуру **КонтекстныйВызовСервера** по шаблону:

// Обработчик контекстного вызова сервера.

//   Позволяет выполнить контекстный вызов сервера когда это требуется из клиентского общего модуля.

//   Например, из ОтчетыКлиентПереопределяемый.ОбработчикКоманды().

//

// Параметры:

//   Форма  - УправляемаяФорма

//   Ключ      - Строка    - Ключ операции, которую необходимо выполнить в контекстном вызове.

//   Параметры - Структура - Параметры вызова сервера.

//   Результат - Структура - Результат работы сервера, возвращается на клиент.

//

// См. также:

//   ОбщаяФорма.ФормаОтчета.ВыполнитьКонтекстныйВызовСервера().

//

Процедура КонтекстныйВызовСервера(Форма, Ключ, Параметры, Результат) Экспорт

// Обработка события.

КонецПроцедуры

* ● **ПриОпределенииПараметровВыбора** – если установить в значение **Истина**, то в модуле объекта отчета следует определить процедуру **ПриОпределенииПараметровВыбора** по шаблону:

// Вызывается в форме отчета перед выводом настройки.

//   Подробнее - см. ОтчетыПереопределяемый.ПриОпределенииПараметровВыбора().

//

Процедура ПриОпределенииПараметровВыбора(Форма, СвойстваНастройки) Экспорт

// Обработка события.

КонецПроцедуры

* ● **ДополнитьСвязиОбъектовМетаданных** – если установить в значение **Истина**, то в модуле объекта отчета следует определить процедуру **ДополнитьСвязиОбъектовМетаданных** по шаблону:

// Дополнительные связи настроек этого отчета.

//   Подробнее - см. ОтчетыПереопределяемый.ДополнитьСвязиОбъектовМетаданных().

//

Процедура ДополнитьСвязиОбъектовМетаданных(СвязиОбъектовМетаданных) Экспорт

// Обработка события.

КонецПроцедуры

### Подключить обработчики полного обновления

При выпуске новых версий конфигурации с существенными изменениями в метаданных (добавлены/удалены/переименованы объекты метаданных) рекомендуется проводить полное обновление всех вспомогательных данных по отчетам. В противном случае поиск по отчетам не будет работать или будет работать не точно. Для полного обновления необходимо сделать следующее.

В процедуре **ПриДобавленииОбработчиковОбновления** модуля прикладного решения (который указывается при внедрении подсистемы «Обновление версии ИБ» в процедуре **ПриДобавленииПодсистем** модуля **ПодсистемыКонфигурацииПереопределяемый**) вставить вызов по шаблону:

ВариантыОтчетов.ДобавитьОбработчикиПолногоОбновления(Обработчики, "<ВерсияКонфигурации>");

где **ВерсияКонфигурации** – номер последнего функционального релиза прикладного решения, в котором были изменения в представлениях объектов метаданных или их реквизитах, данные которых могут выводиться в отчетах.

### Настройка прав доступа пользователей

Для настройки прав доступа пользователей к данным подсистемы нужно использовать роли, приведенные ниже.

Таблица 3.16.

| **№** | **Роли и их назначение** |
| --- | --- |
| 1. | **ПолныеПрава**(из подсистемы «Базовая функциональность»)  Администрирование вариантов отчетов |
| 2. | **ДобавлениеИзменениеВариантовОтчетов**  Администрирование вариантов отчетов.  Если у пользователя нет права **СохранениеДанныхПользователя**, то отключаются все возможности по сохранению настроек подсистемы |
| 3. | **ДобавлениеИзменениеЛичныхВариантовОтчетов**  Добавление и изменение своих вариантов отчетов (доступных только автору).  Просмотр общих вариантов отчетов (доступных всем пользователям).  Если у пользователя нет права **СохранениеДанныхПользователя**, то отключаются все возможности по сохранению настроек подсистемы |
| 4. | **ЧтениеВариантовОтчетов**  Аудит вариантов отчетов.  Просмотр любых вариантов отчетов без права изменения |
| 5. | **БазовыеПрава или БазовыеПраваВнешнегоПользователя**(из подсистемы «Базовая функциональность»)  Просмотр общих вариантов отчетов (доступных всем пользователям) |
| 6. | **ИспользованиеУниверсальногоОтчета**  Доступ к отчету**Универсальный отчет** |

## Использование при разработке конфигурации

Обновление данных справочника **Варианты отчетов** происходит автоматически при смене версии в следующих случаях:

● начальное заполнение;

● добавление или удаление вариантов отчетов;

● добавление или удаление вариантов отчетов из подсистем;

● изменения наименования варианта отчета;

● изменение ключа варианта отчета. Регистрация изменений выполняется в процедуре **ЗарегистрироватьИзмененияКлючейВариантовОтчетов** общего модуля **ВариантыОтчетовПереопределяемый**.

Для всех остальных случаев обновления данных справочника в прикладной конфигурации следует разрабатывать свои обработчики обновления.

При необходимости перезаполнения предопределенных вариантов отчетов можно воспользоваться процедурой **Обновить** общего модуля **ВариантыОтчетов**. Например, это может потребоваться при изменении настроек программы, если некоторые варианты отчетов отключаются по условиям в зависимости от настроек программы.

При необходимости сбросить пользовательские настройки отчетов можно воспользоваться процедурой **СброситьПользовательскиеНастройки** общего модуля **ВариантыОтчетов**. Например, это может потребоваться чтобы «пробросить» изменения в настройках вариантов отчетов пользователям этого варианта.

### Контекстные варианты отчетов

Для контекстных вариантов отчетов в подсистеме предусмотрены две возможности:

● Отключение варианта отчета. Более подробно об отключении написано выше в разделе «[Изменить настройки вариантов отчетов](https://its.1c.ru/db/content/bsp23doc/src/3.8.%20%D0%B2%D0%B0%D1%80%D0%B8%D0%B0%D0%BD%D1%82%D1%8B%20%D0%BE%D1%82%D1%87%D0%B5%D1%82%D0%BE%D0%B2.htm?_=1504708101#_изменить_настройки_вариантов)».

● При использовании поставляемой формы отчета – отключение кнопок работы с вариантом отчета (выбор, изменение и сохранение). Для этого в параметрах открытия формы необходимо передать ключ **ВидимостьКомандВариантовОтчетов** со значением **Ложь**.

### Настройка обмена данными

Для настройки обмена данными следует руководствоваться общими правилами.

В планы обмена распределенной информационной базы (РИБ) рекомендуется включать все объекты метаданных подсистемы, кроме:

● справочника **ПредопределенныеВариантыОтчетовРасширений**,

● регистра сведений **ПредопределенныеВариантыОтчетовВерсийРасширений**.

Из остальных планов обмена (не РИБ) рекомендуется исключать следующие объекты метаданных:

● справочник **ПредопределенныеВариантыОтчетовРасширений**,

● регистр сведений **ПредопределенныеВариантыОтчетовВерсийРасширений**.

# 3.9. Версионирование объектов

Подсистема «Версионирование объектов» предназначена для учета истории изменений объектов (кто, когда и что изменил). Также она позволяет получать отчеты по версиям или по конкретной версии объекта. Версионируемыми объектами могут быть справочники, документы и бизнес-процессы конфигурации.

## Настройка

Для использования подсистемы в конфигурации необходимо:

● если в конфигурации не используется подсистема «Настройки программы», то необходимо разместить в командном интерфейсе администратора системы регистр сведений **НастройкиВерсионированияОбъектов** и поместить константу **ИспользоватьВерсионированиеОбъектов** в основную форму редактирования констант конфигурации или в любую другую форму, предназначенную для администрирования системы;

● создать подписку **ЗаписатьВерсиюДокумента** на событие **ПередЗаписью**, обработчик **ВерсионированиеОбъектовСобытия.ЗаписатьВерсиюДокумента**.

Принять решение по поводу объектов метаданных конфигурации ссылочного типа, которые следует версионировать. Для каждого объекта метаданных, который определен для встраивания подсистемы версионирования объектов, необходимо:

● все версионируемые объекты перечислить в свойстве **Тип** определяемого типа **ВерсионируемыеДанные**(типы **Ссылка** – например, **СправочникСсылка** или **ДокументСсылка**);

● в свойстве **Тип** определяемого типа **ВерсионируемыеДанныеОбъект** перечислить все версионируемые объекты, кроме документов (типы **Объект** – например, **СправочникОбъект** или **БизнесПроцессОбъект**);

● перечислить все версионируемые документы в свойстве **Источник** подписок **ЗаписатьВерсиюДокумента**(типы **ДокументОбъект**);

● во всех формах объектов (элементов), для которых встраивается версионирование, в обработчике **ПриСозданииНаСервере** добавить фрагмент кода:

// СтандартныеПодсистемы.ВерсионированиеОбъектов

ВерсионированиеОбъектов.ПриСозданииНаСервере(ЭтотОбъект);

// Конец СтандартныеПодсистемы.ВерсионированиеОбъектов

● во всех модулях менеджера объектов (элементов), для которых встраивается версионирование, добавить фрагмент кода:

// СтандартныеПодсистемы.ВерсионированиеОбъектов

// Определяет настройки объекта для подсистемы ВерсионированиеОбъектов.

//

// Параметры:

// Настройки - Структура - настройки подсистемы.

Процедура ПриОпределенииНастроекВерсионированияОбъектов(Настройки) Экспорт

КонецПроцедуры

// Конец СтандартныеПодсистемы.ВерсионированиеОбъектов

● принять решение о необходимости скрытия из отчетов по версиям реквизитов и табличных частей, имеющих техническое назначение. Для скрытия реквизитов и табличных частей необходимо в процедуре **ПриОпределенииНастроекВерсионированияОбъектов** модуля менеджера объекта добавить следующий код:

Настройки.ПриПолученииСлужебныхРеквизитов = Истина;

и добавить в модуле менеджера процедуру **ПриПолученииСлужебныхРеквизитов**, в которой перечислить список скрываемых реквизитов и табличных частей:

// СтандартныеПодсистемы.ВерсионированиеОбъектов

// Определяет настройки объекта для подсистемы ВерсионированиеОбъектов.

//

// Параметры:

// Настройки - Структура - настройки подсистемы.

Процедура ПриОпределенииНастроекВерсионированияОбъектов(Настройки) Экспорт

Настройки.ПриПолученииСлужебныхРеквизитов = Истина;

КонецПроцедуры

// Ограничивает видимость реквизитов объекта в отчете по версии.

//

// Параметры:

// Реквизиты - Массив - список имен реквизитов объекта.

Процедура ПриПолученииСлужебныхРеквизитов(Реквизиты) Экспорт

Реквизиты.Добавить("ИмяРеквизита"); // реквизит объекта

Реквизиты.Добавить("ИмяТабличнойЧасти.\*"); // табличная часть объекта

КонецПроцедуры

// Конец СтандартныеПодсистемы.ВерсионированиеОбъектов

### Настройка прав доступа пользователей

Для настройки прав доступа пользователей к данным подсистемы «Версионирование объектов» следует использовать роли, приведенные ниже.

Таблица 3.17.

| **№** | **Роли и их назначение** |
| --- | --- |
| 1. | **ПолныеПрава** (из подсистемы «Базовая функциональность»)  Включение и отключение использования версионирования. Чтение версий объектов. Просмотр отчета по версиям. Возврат к предыдущим версиям объектов. Изменение настроек хранения объектов. Очистка устаревших версий объектов |
| 2. | **ЧтениеВерсийОбъектов**  Чтение версий объектов. Просмотр отчета по версиям |

## Использование при разработке конфигурации

### Разработка макетов отчета по конкретной версии объекта

Подсистема «Версионирование объектов» позволяет использовать произвольный макет отчета по версии объекта вместо макета по умолчанию.

Для создания макета отчета для версионируемого объекта можно воспользоваться конструктором печати объекта:

● открыть конструктор печати объекта;

● выбрать **Создать новую команду**, ввести имя команды (например, **Печать**; при этом необходимо убедиться, что процедуры с таким именем нет в модуле менеджера объекта метаданных) и нажать кнопку **Далее**;

● выбрать необходимые реквизиты, которые будут выводиться в отчете, и нажать кнопку **Далее**;

● выбрать необходимые поля в табличных частях объекта (если есть) и последовательно нажать кнопку **Далее**для каждой табличной части;

● не выбирать реквизиты для подвала документа и нажать кнопку **Далее**;

● в итоговом диалоге выбора группы оставить все значения по умолчанию и нажать **ОК**;

● удалить созданную команду печати и процедуру формирования данных печати из модуля менеджера объекта, которые были созданы на предыдущем шаге (в примере **Печать**);

● переименовать созданный макет в **МакетОбъекта** и при необходимости настроить его оформление.

### Настройка обмена данными

Все объекты метаданных подсистемы, содержащие данные, рекомендуется включать в планы обмена распределенной ИБ (РИБ).

# 3.10. Взаимодействия

Подсистема «Взаимодействия» предназначена для планирования, регистрации, упорядочивания взаимодействий и работы с результатами взаимодействий. Взаимодействия включают переписку по электронной почте, регистрацию звонков и встреч. Подсистема обеспечивает подбор и создание новых контактов взаимодействий.

## Настройка

Если в конфигурации не используется подсистема «Настройки программы», то на рабочем месте администратора программы необходимо разместить константы:

● **ИспользоватьПочтовыйКлиент**,

● **ОтправлятьПисьмаВФорматеHTML**,

● **ИспользоватьПрочиеВзаимодействия**,

● **ИспользоватьПризнакРассмотрено**.

См. пример в форме **Органайзер** обработки **ПанельАдминистрированияБСП**.

В форме персональных настроек разместить вызов команды **ЖурналДокументов.Взаимодействия.Команда.НастройкиРаботыСПочтой**. Пример размещения см. в демонстрационной конфигурации в общей форме **\_ДемоМоиНастройки**.

Разместить в командном интерфейсе пользователей, использующих подсистему, журнал документов **Взаимодействия**.

### Определение типов предметов и контактов взаимодействий

Необходимо принять решение по поводу состава объектов метаданных, которые могут выступать в качестве контактов взаимодействий. Например, это могут быть физические лица, партнеры, контактные лица партнеров и т. п. Расширить тип определяемого типа **КонтактВзаимодействия** допустимыми типами контактов.

Если предусматривается создание контактов напрямую из форм документов взаимодействий, то для форм объектов метаданных, определенных как контакты необходимо:

● добавить реквизит формы **НеобходимоОповещение** типа **Булево**,

● добавить реквизит формы **ОбъектОснование** типа **Произвольный**,

● добавить в обработчик события **ПриСозданииНаСервере** вызов следующей процедуры:

Взаимодействия.ПодготовитьОповещения(ЭтотОбъект, Параметры, Ложь);

● добавить в обработчик события формы **ПослеЗаписи** вызов следующей процедуры:

ВзаимодействияКлиент.КонтактПослеЗаписи(Форма, Объект, ПараметрыЗаписи, ИмяОбъектаОтправителяСообщения);

Необходимо принять решение по поводу состава объектов метаданных, которые могут выступать в качестве предметов взаимодействий. Например, это могут быть проекты, заказы клиентов, заказ поставщику и т. п. Расширить тип определяемого типа **ПредметВзаимодействия** допустимыми типами предметов.

Если предусматривается создание предметов на основании документов взаимодействий, то для форм объектов метаданных, определенных как контакты, необходимо:

● добавить реквизит формы **НеобходимоОповещение** типа **Булево**;

● добавить реквизит формы **ВзаимодействиеОснование** составного типа, включающего в себя типы тех документов взаимодействий, на основании которых создаются предметы взаимодействий;

● добавить в обработчик события **ПриСозданииНаСервере** вызов следующей процедуры:

Взаимодействия.ПодготовитьОповещения(ЭтотОбъект,Параметры,Ложь);

● добавить в обработчик события формы **ПослеЗаписи** вызов следующей процедуры:

ВзаимодействияКлиент.ВзаимодействиеПредметПослеЗаписи(Форма, Объект, ПараметрыЗаписи, ИмяОбъектаОтправителяСообщения);

### Настройка модуля менеджера предметов взаимодействий

В модуле менеджера каждого объекта метаданных, определенного как «предмет взаимодействий», необходимо реализовать экспортную функцию **ТекстЗапросаПоКонтактам**, в которой сформировать текст запроса по контактам, содержащимся в предмете взаимодействий. Например, ссылки на контакты взаимодействий могут иметься в реквизитах шапки и табличных частей предметов взаимодействий.

Функция принимает два необязательных параметра. В первом параметре **ТекстВременнаяТаблица** типа **Строка**может находиться часть текста запроса, отвечающая за помещение результата запроса во временную таблицу. Второй параметр **Объединить** типа **Булево** указывает на режим формирования запроса. Если данный параметр имеет значение **Истина**, то формируемый в функции запрос является частью другого запроса и должен начинаться с конструкции **ОБЪЕДИНИТЬ**. Возвращаемое значение – **Строка**, содержащая в себе текст запроса по контактам предмета взаимодействий.

Пример реализации такой функции можно посмотреть в модуле менеджера документа **\_ДемоЗаказПокупателя** в демонстрационной конфигурации.

Для тех предметов взаимодействий, для которых предполагается использовать механизм активности предметов и текущий статус подразумевает активность предмета, необходимо устанавливать признак активности в регистр сведений **СостоянияПредметовВзаимодействий**. Например, как это сделано в процедуре **ПриЗаписи** модуля объекта документа **\_ДемоЗаказПокупателя**.

### Настройка переопределяемых модулей

При необходимости можно вписать реализацию в функции переопределяемых модулей подсистемы:

● **ВзаимодействияКлиентПереопределяемый**,

● **ВзаимодействияКлиентСерверПереопределяемый**,

● **ВзаимодействияПереопределяемый**.

### Настройка форм списков предметов взаимодействий.

Для того чтобы появилась возможность устанавливать предмет взаимодействий путем множественного перетаскивания в форму списка предмета, необходимо выполнить следующие действия:

● у элемента формы списка установить флаг **РазрешитьПеретаскивание**;

● в обработчике события **ПроверкаПеретаскивания** элемента формы списка разместить следующий код:

ВзаимодействияКлиент.СписокПредметПроверкаПеретаскивания(Элемент, ПараметрыПеретаскивания, СтандартнаяОбработка, Строка, Поле);

● в обработчике события **Перетаскивание** элемента формы списка разместить следующий код:

ВзаимодействияКлиент.СписокПредметПеретаскивание(Элемент, ПараметрыПеретаскивания, СтандартнаяОбработка, Строка, Поле);

### Настройка прав доступа пользователей

Для настройки прав доступа пользователей к данным подсистемы «Взаимодействия» следует использовать роль, приведенную ниже.

Таблица 3.18.

| **№** | **Роли и их назначение** |
| --- | --- |
| 1. | **ДобавлениеИзменениеВзаимодействий**  Чтение, добавление, изменение взаимодействий |

Пример настройки прав доступа пользователей приведен ниже.

Таблица 3.19.

|  |  |  |
| --- | --- | --- |
| **№** | **Группа пользователей и ее функции** | **Состав ролей** |
| 1. | Ответственный за ведение взаимодействий | ● **БазовыеПрава** (из подсистемы «Базовая функциональность»),  ● **ЗапускТонкогоКлиента** (из подсистемы «Базовая функциональность»),  ● **ЧтениеУчетныхЗаписейЭлектроннойПочты** (из подсистемы «Работа с почтовыми сообщениями»),  **● ДобавлениеИзменениеВзаимодействий** |

### Особые случаи внедрения подсистемы

● [Внедрение подсистем «Взаимодействия» и «Управление доступом»](https://its.1c.ru/db/content/bsp23doc/src/3.59.%20%D0%BE%D1%81%D0%BE%D0%B1%D1%8B%D0%B5%20%D1%81%D0%BB%D1%83%D1%87%D0%B0%D0%B8%20%D0%B2%D0%BD%D0%B5%D0%B4%D1%80%D0%B5%D0%BD%D0%B8%D1%8F%20%D0%BF%D0%BE%D0%B4%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC.htm?_=1504708101#_внедрение_подсистем_).

● [Внедрение подсистемы «Взаимодействия» без подсистемы «Электронная подпись»](https://its.1c.ru/db/content/bsp23doc/src/3.59.%20%D0%BE%D1%81%D0%BE%D0%B1%D1%8B%D0%B5%20%D1%81%D0%BB%D1%83%D1%87%D0%B0%D0%B8%20%D0%B2%D0%BD%D0%B5%D0%B4%D1%80%D0%B5%D0%BD%D0%B8%D1%8F%20%D0%BF%D0%BE%D0%B4%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC.htm?_=1504708101#_внедрение_подсистемы_).

## Использование при разработке конфигурации

### Настройка обмена данными

Все объекты метаданных подсистемы рекомендуется включать в планы обмена распределенной ИБ (РИБ), планы обмена автономных рабочих мест, а также в планы обмена по правилам конвертации, за исключением следующих регистров сведений:

● **СостоянияПапокПисем**,

● **СостоянияПредметовВзаимодействий**,

● **СостоянияКонтактовВзаимодействий**.

В этих регистрах хранятся данные по состояниям документов взаимодействий (количество нерассмотренных и дата последнего взаимодействия), выводимые на панель навигации рабочего места ответственного за ведение взаимодействия. Для того чтобы эти данные были корректными, в процессе получения данных необходимо накапливать информацию о папках, предметах и контактах загруженных документов подсистемы взаимодействий.

А при завершении обмена выполнить перерасчет состояний при помощи следующих процедур:

● **Взаимодействия.РассчитатьРассмотреноПоКонтактам**,

● **Взаимодействия.РассчитатьРассмотреноПоПредметам**,

● **Взаимодействия.РассчитатьРассмотреноПоПапкам**.

Еще один регистр сведений, не включаемый в план обмена – **ЗаблокированныеДляПолученияУчетныеЗаписи**. Регистр предназначен для предотвращения дублирования получаемых писем при конкурентной работе с коллективными учетными записями.

Не включается в план обмена регистр сведений **Удалить\_АктивныеПредметыВзаимодействий**.

# 3.11. Графики работы

Подсистема «Графики работы» предназначена для хранения графиков, по которым работает предприятие. Для каждого графика можно указать, какие дни являются рабочими. Графики работы можно использовать как для указания работы компании в целом, так и отдельных ее частей. Например, график работы склада по выдаче товаров.

## Настройка

В командном интерфейсе ответственного за ведение нормативно-справочной информации необходимо поместить справочник **Календари**. См. пример в демонстрационной конфигурации в разделе «Справочники».

### Настройка прав доступа пользователей

Для настройки прав доступа пользователей к данным подсистемы «Графики работы» следует использовать роли, приведенные ниже.

Таблица 3.20.

| **№** | **Роли и их назначение** |
| --- | --- |
| 1. | **ЧтениеГрафиковРаботы**(из подсистемы «Базовая функциональность»)  Чтение календарей и графиков |
| 2. | **ДобавлениеИзменениеГрафиковРаботы**  Добавление и изменение календарей и графиков |
| 3. | **ПолныеПрава** (из подсистемы «Базовая функциональность»)  Удаление помеченных на удаление объектов подсистемы |

## Использование при разработке конфигурации

Подсистема используется совместно с подсистемой «Календарные графики», которая, в свою очередь, представляет программный интерфейс экспортными функциями общего модуля **КалендарныеГрафики**.

### Настройка обмена данными

Все объекты метаданных подсистемы, содержащие данные, рекомендуется включать в планы обмена распределенной ИБ (РИБ).

# 3.12. Групповое изменение объектов

Подсистема «Групповое изменение объектов» предоставляет пользовательский интерфейс для множественного изменения объектов одного типа. Помимо изменения реквизитов имеется возможность изменения дополнительных реквизитов и сведений (подсистема «Свойства»), если они есть у объектов.

Также подсистема учитывает наличие у объектов реквизитов, запрещенных для редактирования (подсистема «Запрет редактирования реквизитов объектов»).

## Настройка

### Размещение команды изменения объектов в списках

Сначала нужно принять решение по поводу состава объектов конфигурации, для которых должна быть добавлена команда по групповому изменению. Как правило, к таким объектам относятся большинство объектов конфигурации ссылочного типа.

Затем надо поместить команду для группового изменения объектов на командной панели и в контекстном меню формы списка. Рекомендуемые значения свойств команды:

Таблица 3.21.

|  |  |
| --- | --- |
| **Свойство** | **Значение** |
| Имя | **ИзменитьВыделенные** |
| Действие | **ИзменитьВыделенные** |
| Подсказка | Изменить выделенные объекты |
| Изменяет сохраняемые данные | Нет |

Также рекомендуется управлять видимостью команды в форме в зависимости от текущих прав пользователя. Например:

&НаСервере

Процедура ПриСозданииНаСервере(Отказ, СтандартнаяОбработка)

МожноРедактировать = ПравоДоступа("Редактирование", Метаданные.Справочники.\_ДемоНоменклатура);

Элементы.ФормаИзменитьВыделенные.Видимость = МожноРедактировать;

КонецПроцедуры

Установить у поля **Ссылка** основного реквизита формы признак **Использовать всегда**.

Добавить в модуль формы обработчик:

&НаКлиенте

Процедура ИзменитьВыделенные(Команда)

ГрупповоеИзменениеОбъектовКлиент.ИзменитьВыделенные(Элементы.Список, Список);

КонецПроцедуры

### Ограничение редактируемых реквизитов объектов

Изменение некоторых реквизитов с помощью групповой обработки может быть нежелательным либо нарушать логику работы конфигурации. Такие реквизиты необходимо скрывать от пользователя.

Обработка группового изменения может быть вызвана как из формы списка для выбранных объектов, так и самостоятельно с последующим выбором изменяемых объектов в самой обработке, поэтому необходимо для всех объектов конфигурации рассмотреть необходимость ограничения доступности реквизитов для группового изменения.

Если для объекта метаданных определены реквизиты, которые можно изменять с помощью обработки группового изменения, следует объявить экспортную функцию **РеквизитыРедактируемыеВГрупповойОбработке** в модуле менеджера объекта, которая возвращает массив – имена реквизитов. При этом оставшиеся реквизиты скрываются в обработке группового изменения. В общем виде разрешенные для редактирования реквизиты объекта описываются следующим образом (пример см. в справочнике **\_ДемоКонтрагенты** демонстрационной конфигурации):

// Возвращает реквизиты объекта, которые разрешается редактировать

// с помощью обработки группового изменения реквизитов.

//

// Возвращаемое значение:

//  Массив - список имен реквизитов объекта.

Функция РеквизитыРедактируемыеВГрупповойОбработке() Экспорт

РедактируемыеРеквизиты = Новый Массив;

РедактируемыеРеквизиты.Добавить("ИмяРеквизита"); // реквизит объекта

РедактируемыеРеквизиты.Добавить("ИмяТабличнойЧасти.ИмяРеквизита"); // реквизит табличной части объекта

РедактируемыеРеквизиты.Добавить("ИмяТабличнойЧасти.\*"); // все реквизиты табличной части объекта

РедактируемыеРеквизиты.Добавить("\*"); // все реквизиты и табличные части объекта

Возврат РедактируемыеРеквизиты;

КонецФункции

Если стоит обратная задача: известны имена реквизитов, которые не должны редактироваться в обработке группового изменения, – следует объявить в модуле менеджера объекта функцию **РеквизитыНеРедактируемыеВГрупповойОбработке**, которая возвращает массив – имена реквизитов (аналогично функции **РеквизитыРедактируемыеВГрупповойОбработке**). Пример см. в справочнике **\_ДемоНоменклатура** демонстрационной конфигурации.

Если объект не предназначен для использования в групповом изменении, то необходимо скрывать все его реквизиты.

**Важно!**

Служебные реквизиты объектов, такие как **Ссылка**, **Наименование**, **ПометкаУдаления** и т. д., а также некоторые табличные части не требуют дополнительных действий для скрытия в обработке группового изменения, так как они скрыты по умолчанию. Подробнее со списком таких реквизитов и табличных частей можно ознакомиться в макете **ФильтрРеквизитов** обработки **ГрупповоеИзменениеОбъектов**.

При необходимости также следует перенести логику проверки заполнения объектов и логику «при записи» из модулей форм в модули объектов.

Все объекты метаданных, в которых определены функции **РеквизитыРедактируемыеВГрупповойОбработке** и **РеквизитыНеРедактируемыеВГрупповойОбработке**, следует перечислить в процедуре **ПриОпределенииОбъектовСРедактируемымиРеквизитами** общего модуля **ГрупповоеИзменениеОбъектовПереопределяемый**.

### Настройка прав доступа пользователей

Таблица 3.22.

| **№** | **Роли и их назначение** |
| --- | --- |
| 1. | **ПолныеПрава**  Редактирование любых объектов конфигурации (при открытии формы группового изменения реквизитов из раздела командного интерфейса «Администрирование») |
| 2. | **Другие роли**  Для группового изменения реквизитов в списках объектов не требуются какие-либо дополнительные роли. Эта возможность будет доступна, если у пользователя есть права на редактирование самих объектов |

### Настройка обмена данными

Для настройки обмена данными никаких действий не требуется, так как подсистема не предоставляет собственных данных.

# 3.13. Даты запрета изменения

Подсистема «Даты запрета изменения» позволяет заблокировать изменения любых данных информационной базы (документов, записей регистров, элементов справочников и др.) ранее определенной даты. Администратор системы может настроить как одну общую дату для всей информационной базы в целом, так и несколько дат по разделам и/или отдельным объектам разделов учета.

Подсистема всегда предоставляет возможность настройки дат запрета как для всех пользователей, так и для конкретных пользователей, групп пользователей (опционально – для внешних пользователей, групп внешних пользователей).

Интерфейс настройки разработан от минимальных возможностей «нет запрета» до максимальных возможностей, определенных при внедрении, но не более чем **По разделам и объектам** в сочетании с настройкой **По пользователям** (и группам пользователей).

Подсистема позволяет устанавливать как произвольные даты запрета, так и относительные даты запрета (конец прошлого месяца, конец прошлого квартала и другие).

## Настройка

1. [Определить состав типов, для которых будет выполняться запрет изменения данных](https://its.1c.ru/db/content/bsp23doc/src/3.13.%20%D0%B4%D0%B0%D1%82%D1%8B%20%D0%B7%D0%B0%D0%BF%D1%80%D0%B5%D1%82%D0%B0%20%D0%B8%D0%B7%D0%BC%D0%B5%D0%BD%D0%B5%D0%BD%D0%B8%D1%8F.htm?_=1504708101#_определить_состав_типов,).

2. [Определить состав типов, для которых будут настраиваться запреты загрузки данных](https://its.1c.ru/db/content/bsp23doc/src/3.13.%20%D0%B4%D0%B0%D1%82%D1%8B%20%D0%B7%D0%B0%D0%BF%D1%80%D0%B5%D1%82%D0%B0%20%D0%B8%D0%B7%D0%BC%D0%B5%D0%BD%D0%B5%D0%BD%D0%B8%D1%8F.htm?_=1504708101#_определить_состав_типов,_1).

3. [Определить максимальные возможности настройки дат запрета (общая дата, по разделам, по объектам, по разделам и объектам)](https://its.1c.ru/db/content/bsp23doc/src/3.13.%20%D0%B4%D0%B0%D1%82%D1%8B%20%D0%B7%D0%B0%D0%BF%D1%80%D0%B5%D1%82%D0%B0%20%D0%B8%D0%B7%D0%BC%D0%B5%D0%BD%D0%B5%D0%BD%D0%B8%D1%8F.htm?_=1504708101#_определить_максимальные_возможности).

4. [Определить поля объектов метаданных, которые будут использованы для получения проверяемых данных (даты, раздела, объекта)](https://its.1c.ru/db/content/bsp23doc/src/3.13.%20%D0%B4%D0%B0%D1%82%D1%8B%20%D0%B7%D0%B0%D0%BF%D1%80%D0%B5%D1%82%D0%B0%20%D0%B8%D0%B7%D0%BC%D0%B5%D0%BD%D0%B5%D0%BD%D0%B8%D1%8F.htm?_=1504708101#_определить_поля_объектов).

5. [Определить случаи условного отказа от проверки запрета изменения и определить случаи проверки запрета загрузки](https://its.1c.ru/db/content/bsp23doc/src/3.13.%20%D0%B4%D0%B0%D1%82%D1%8B%20%D0%B7%D0%B0%D0%BF%D1%80%D0%B5%D1%82%D0%B0%20%D0%B8%D0%B7%D0%BC%D0%B5%D0%BD%D0%B5%D0%BD%D0%B8%D1%8F.htm?_=1504708101#_определить_случаи_условного).

6. [Определить возможность настройки дат запрета для внешних пользователей и групп внешних пользователей](https://its.1c.ru/db/content/bsp23doc/src/3.13.%20%D0%B4%D0%B0%D1%82%D1%8B%20%D0%B7%D0%B0%D0%BF%D1%80%D0%B5%D1%82%D0%B0%20%D0%B8%D0%B7%D0%BC%D0%B5%D0%BD%D0%B5%D0%BD%D0%B8%D1%8F.htm?_=1504708101#_определить_возможность_настройки).

### Определить состав типов, для которых будет выполняться запрет изменения данных

Определить состав типов справочников, документов, регистров, для которых будет выполняться запрет изменения данных.

Создать подписки на события и указать в них выбранные типы согласно таблице, приведенной ниже.

Подписки на событие **ПередЗаписью** для проверки запрета изменения данных:

Таблица 3.23.

|  |  |  |
| --- | --- | --- |
| **Имя** | **Источник** | **Обработчик** |
| **ПроверитьДатуЗапретаИзмененияПередЗаписьюДокумента** | Элементы типа: **ДокументОбъект** | ДатыЗапретаИзменения.ПроверитьДатуЗапретаИзмененияПередЗаписьюДокумента |
| **ПроверитьДатуЗапретаИзмененияПередЗаписью** | Элементы типов: **СправочникОбъект**,**ПланВидовХарактеристикОбъект**, **ПланСчетовОбъект**, **ПланВидовРасчетаОбъект**,**БизнесПроцессОбъект**, **ЗадачаОбъект**,**ПланОбменаОбъект** | ДатыЗапретаИзменения. ПроверитьДатуЗапретаИзмененияПередЗаписью |
| **ПроверитьДатуЗапретаИзмененияПередЗаписьюНабораЗаписей** | Элементы типов: **РегистрСведенийНаборЗаписей**, **РегистрНакопленияНаборЗаписей** | ДатыЗапретаИзменения. ПроверитьДатуЗапретаИзмененияПередЗаписьюНабораЗаписей |
| **ПроверитьДатуЗапретаИзмененияПередЗаписьюНабораЗаписейРегистраБухгалтерии** | Элементы типа: **РегистрБухгалтерииНаборЗаписей** | ДатыЗапретаИзменения. ПроверитьДатуЗапретаИзмененияПередЗаписьюНабораЗаписейРегистраБухгалтерии |
| **ПроверитьДатуЗапретаИзмененияПередЗаписьюНабораЗаписейРегистраРасчета** | Элементы типа: **РегистрРасчетаНаборЗаписей** | ДатыЗапретаИзменения. ПроверитьДатуЗапретаИзмененияПередЗаписьюНабораЗаписейРегистраРасчета |

Подписки на событие **ПередУдалением** для проверки запрета изменения данных:

Таблица 3.24.

|  |  |  |
| --- | --- | --- |
| **Имя** | **Источник** | **Обработчик** |
| **ПроверитьДатуЗапретаИзмененияПередУдалением** | Элементы типа: **СправочникОбъект**,**ДокументОбъект**, **ПланВидовХарактеристикОбъект**,**ПланСчетовОбъект**, **ПланВидовРасчетаОбъект**,**БизнесПроцессОбъект**, **ЗадачаОбъект**,**ПланОбменаОбъект** | ДатыЗапретаИзменения. ПроверитьДатуЗапретаИзмененияПередУдалением |

Для выбранных объектов метаданных необходимо вставить в модуль формы объекта в процедуру обработчика события **ПриЧтенииНаСервере** вызов:

ДатыЗапретаИзменения.ОбъектПриЧтенииНаСервере(ЭтотОбъект, ТекущийОбъект).

При выполнении вызова будет выполнена проверка запрета изменения, и при наличии запрета свойство формы **ТолькоПросмотр** будет установлено **Истина**.

### Определить состав типов, для которых будут настраиваться запреты загрузки данных

Для запрета загрузки данных по узлам планов обмена необходимо в определяемый тип **АдресатЗапретаИзменения** добавить типы требуемых планов обмена. Если хотя бы один тип плана обмена добавлен в состав определяемого типа **АдресатЗапретаИзменения**, то по команде **Даты запрета загрузки**данных будет открыта форма **Даты запрета изменения данных** в режиме редактирования дат запрета загрузки.

### Определить максимальные возможности настройки дат запрета (общая дата, по разделам, по объектам, по разделам и объектам)

Для различных прикладных решений требования к настройке даты запрета сильно отличаются. Для простой однопользовательской конфигурации достаточно одной общей даты, а для корпоративного решения требуется множество разрезов дат запрета к различным областям планирования и учета. Для этих целей предусмотрены разделы – это разрезы областей планирования и учета для установки дат запрета. Например, **Складской учет**, **Бухгалтерская отчетность**, **Планирование**. В подсистеме разделы – это предопределенные элементы объекта метаданных **ПланВидовХарактеристик.РазделыДатЗапретаИзменения**, которые редактируются в конфигураторе, а в режиме **1С:Предприятие** не редактируются.

При необходимости разделы могут быть уточнены объектом – например, раздел **Складской учет** можно уточнить складом. Для этого разделу нужно установить тип, например **СправочникСсылка.Склады**. Если раздел не предполагается уточнять объектом, тогда нужно установить тип **ПланВидовХарактеристикСсылка.РазделыДатЗапретаИзменения**.

Возможны четыре варианта внедрения:

1. **Общая дата**. В этом варианте список разделов пуст, у пользователя будет возможность задавать общую дату на всю конфигурацию (для всех пользователей или по пользователям).

2. **По разделам**. В этом варианте список разделов состоит из двух и более разделов, типы разделов не определены (**ПланВидовХарактеристикСсылка.РазделыДатЗапретаИзменения**). В интерфейсе объекты не отображаются, их нельзя добавить. Например, разделы **Учет** и **Планирование**. При этом можно задать общую дату или две даты отдельно по каждому разделу.

3. **По объектам**. В этом варианте список разделов состоит из одного раздела. Этот раздел не отображается в интерфейсе, но используется при получении данных для проверки. У раздела задан непустой тип. Например, раздел **Бухгалтерский учет** с типом **СправочникСсылка.Организации**. При этом можно задать общую дату или даты по отдельным организациям.

4. **По разделам и объектам**. В этом самом общем варианте список разделов состоит из двух и более разделов, причем хотя бы один раздел должен иметь определенный тип. Например, **Учет по организациям** и **Планирование**.

### Определить поля объектов метаданных, которые будут использованы для получения проверяемых данных (даты, раздела, объекта)

Для подготовки данных к проверке на запрет изменения требуется для каждого из выбранных объектов метаданных определить поля – источники значений. Для этих целей используется процедура **ЗаполнитьИсточникиДанныхДляПроверкиЗапретаИзменения** переопределяемого модуля.

// Содержит описание таблиц и полей объектов для проверки запретов изменения данных.

//   Вызывается из процедуры ИзменениеЗапрещено общего модуля ДатыЗапретаИзменения,

//   используемой в подписке на событие ПередЗаписью объекта для проверки наличия

//   запретов и отказа от изменений запрещенного объекта.

//

// Параметры:

//  ИсточникиДанных - ТаблицаЗначений - с колонками:

//   \* Таблица     - Строка - полное имя объекта метаданных,

//                   например Метаданные.Документы.ПриходнаяНакладная.ПолноеИмя().

//   \* ПолеДаты    - Строка - имя реквизита объекта или табличной части,

//                   например, "Дата", "Товары.ДатаОтгрузки".

//   \* Раздел      - Строка - имя предопределенного элемента

//                   "ПланВидовХарактеристикСсылка.РазделыДатЗапрета".

//   \* ПолеОбъекта - Строка - имя реквизита объекта или реквизита табличной части,

//                   например, "Организация", "Товары.Склад".

//

//  Для добавления строки имеется процедура ДобавитьСтроку в общем модуле ДатыЗапретаИзменения.

//

Процедура ЗаполнитьИсточникиДанныхДляПроверкиЗапретаИзменения(ИсточникиДанных) Экспорт

ДатыЗапретаИзменения.ДобавитьСтроку(ИсточникиДанных,

Метаданные.Документы.\_ДемоОприходованиеТоваров.ПолноеИмя(),

"Дата", "\_ДемоСкладскойУчет", "МестоХранения");

ДатыЗапретаИзменения.ДобавитьСтроку(ИсточникиДанных,

Метаданные.Документы.\_ДемоСписаниеТоваров.ПолноеИмя(),

"Дата", "\_ДемоСкладскойУчет", "МестоХранения");

ДатыЗапретаИзменения.ДобавитьСтроку(ИсточникиДанных,

Метаданные.Документы.\_ДемоСписаниеБезналичныхДенежныхСредств.ПолноеИмя(),

"ДатаПроведенияБанком", "\_ДемоБанк", "БанковскийСчет");

КонецПроцедуры

Поле может быть задано «через точку», например, если в документе указана ссылка на документ-основание, содержащий объект. Тогда реквизит будет задан так: **ДокументОснование.Склад**, где **ДокументОснование** – реквизит документа, проверяемого на запрет изменения. Можно указать поле «через несколько точек».

В качестве поля можно указать поле табличной части, в этом случае используется тот же метод «через точку», например **Товары.Склад**. Указывать поле табличной части можно только в проверяемом документе, а не в документе-основании (такое описание не поддерживается).

Следует иметь в виду, что использование метода «через точку» потребует дополнительного обращения к базе данных для получения значения, а использование поля табличной части увеличит количество проверяемых данных.

Для объекта данных может потребоваться несколько проверок (например, для документа **ПеремещениеТоваров**требуется проверить запрет изменения по двум складам). В этом случае используются несколько строк для описания проверок по одному объекту метаданных (см. пример выше).

### Определить случаи условного отказа от проверки запрета изменения и определить случаи проверки запрета загрузки

Кроме стандартного случая отказа от проверки запрета изменения и запрета загрузки данных при обновлении информационной базы, прикладной разработчик может пропустить проверку запретов по прикладным причинам.

Если в прикладной конфигурации потребовалось пропустить проверку запрета изменения данных, например, если заказ не закрыт (или при особых случаях обмена данными пропустить проверку запрета загрузки данных) следует воспользоваться переопределяемой процедурой **ПередПроверкойЗапретаИзменения**.

// Позволяет переопределить выполнение проверок запретов по произвольному условию.

//

// Параметры:

//  Объект       - СправочникОбъект,

//                 ДокументОбъект,

//                 ПланВидовХарактеристикОбъект,

//                 ПланСчетовОбъект,

//                 ПланВидовРасчетаОбъект,

//                 БизнесПроцессОбъект,

//                 ЗадачаОбъект,

//                 ПланОбменаОбъект - объект данных (ПередЗаписью или ПриЧтенииНаСервере).

//               - РегистрСведенийНаборЗаписей,

//                 РегистрНакопленияНаборЗаписей,

//                 РегистрБухгалтерииНаборЗаписей,

//                 РегистрРасчетаНаборЗаписей - набор записей (ПередЗаписью или ПриЧтенииНаСервере).

//

//  ПроверкаЗапретаИзменения    - Булево - когда Истина, тогда проверка изменения выполняется.

//                                Если установить Ложь, тогда проверка запрета изменения будет пропущена.

//

//  УзелПроверкиЗапретаЗагрузки - Неопределено - проверка запрета загрузки не выполняется.

//                              - ПланыОбменаСсылка - проверка запрета загрузки выполняется. Если

//                                установить Неопределено, проверка запрета загрузки будет пропущена.

//

//  ВерсияОбъекта               - Строка - начальное значение "". Проверяются обе версии объекта.

//                                Если установить "СтараяВерсия" или "НоваяВерсия", тогда будет

//                                выполнена проверка только старой или только новой версии объекта.

//

Процедура ПередПроверкойЗапретаИзменения(Объект,

                                         ПроверкаЗапретаИзменения,

                                         УзелПроверкиЗапретаЗагрузки,

                                         ВерсияОбъекта) Экспорт

Если ТипЗнч(Объект) = Тип("ДокументОбъект.\_ДемоЗаказПокупателя") Тогда

СтатусЗаказа = ОбщегоНазначения.ЗначениеРеквизитаОбъекта(Объект.Ссылка, "СтатусЗаказа");

ЗаказЗакрытСтараяВерсия = (СтатусЗаказа        = Перечисления.\_ДемоСтатусыЗаказовПокупателей.Закрыт);

ЗаказЗакрытНоваяВерсия  = (Объект.СтатусЗаказа = Перечисления.\_ДемоСтатусыЗаказовПокупателей.Закрыт);

Если Не ЗаказЗакрытСтараяВерсия И Не ЗаказЗакрытНоваяВерсия Тогда

ПроверкаЗапретаИзменения = Ложь;

УзелПроверкиЗапретаЗагрузки = Неопределено;

ИначеЕсли Не ЗаказЗакрытНоваяВерсия Тогда

ВерсияОбъекта = "СтараяВерсия"; // Проверить только старую версию объекта.

ИначеЕсли Не ЗаказЗакрытСтараяВерсия Тогда

ВерсияОбъекта = "НоваяВерсия"; // Проверить только новую версию объекта.

КонецЕсли;

ИначеЕсли ТипЗнч(Объект) = Тип("ДокументОбъект.\_ДемоСписаниеБезналичныхДенежныхСредств") Тогда

// Отказ от проверки с учетом того, что ДатаПроведенияБанком, используемая в проверке не указывается,

// если документ не проведен банком, а указывается позже после проведения документа банком.

ПроведеноБанкомНоваяВерсия  = Объект.ПроведеноБанком;

ПроведеноБанкомСтараяВерсия = ОбщегоНазначения.ЗначениеРеквизитаОбъекта(

Объект.Ссылка, "ПроведеноБанком") = Истина;

Если Не ПроведеноБанкомНоваяВерсия И Не ПроведеноБанкомСтараяВерсия Тогда

ПроверкаЗапретаИзменения = Ложь;

УзелПроверкиЗапретаЗагрузки = Неопределено;

ИначеЕсли Не ПроведеноБанкомНоваяВерсия Тогда

ВерсияОбъекта = "СтараяВерсия"; // Проверить только старую версию объекта.

ИначеЕсли Не ПроведеноБанкомСтараяВерсия Тогда

ВерсияОбъекта = "НоваяВерсия"; // Проверить только новую версию объекта.

КонецЕсли;

КонецЕсли;

КонецПроцедуры

### Определить возможность настройки дат запрета для внешних пользователей и групп внешних пользователей

Если в прикладном решении требуется делать настройку по внешним пользователям и группам внешних пользователей, то нужно вписать код в процедуру **НастройкаИнтерфейса** переопределяемого модуля:

Процедура НастройкаИнтерфейса(Знач НастройкиРаботыИнтерфейса) Экспорт

НастройкиРаботыИнтерфейса.ИспользоватьВнешнихПользователей = Истина;

КонецПроцедуры

### Размещение в командном интерфейсе

Для использования подсистемы необходимо разместить в командном интерфейсе ответственного за даты запрета объекты метаданных:

● команду **ДатыЗапретаИзмененияДанных** – для просмотра и редактирования дат запрета изменения;

● команду **ДатыЗапретаЗагрузкиДанных** – для просмотра и редактирования дат запрета загрузки (если запреты загрузки не используются, команду размещать не требуется).

Для размещения команд требуется включить регистр сведений **ДатыЗапретаИзменения** в подсистему интерфейса (у регистра отключено использование стандартных команд).

### Настройка прав доступа пользователей

Для настройки прав доступа пользователей к данным подсистемы «Даты запрета изменения» следует использовать роли:

Таблица 3.25.

| **№** | **Роли и их назначение** |
| --- | --- |
| 1. | **ПолныеПрава** (из подсистемы «Базовая функциональность»)  Просмотр и редактирование дат запрета изменения и дат запрета загрузки (доступ к форме **Даты запрета изменения данных** с отчетами) |
| 2. | **Добавление изменение дат запрета изменения**  Просмотр и редактирование дат запрета изменения (доступ к форме **Даты запрета изменения данных** и отчету **Даты запрета изменения данных**) |

## Использование при разработке конфигурации

При создании новых прикладных объектов метаданных нужно повторно выполнять соответствующую часть процедуры настройки, описанной выше.

Для дополнительной проверки запрета изменения (вне подписок на события) используется функция **ИзменениеЗапрещено** модуля **ДатыЗапретаИзменения**.

Также может потребоваться выполнить программную проверку запрета изменения нестандартного вида. Для этого предусмотрена функция **НайденЗапретИзмененияДанных** модуля **ДатыЗапретаИзменения**, которая выполнит проверку и вернет ее результат, а также отправит стандартное сообщение пользователю об ошибке.

Для работы функции нужно подготовить данные, шаблон которых легко получить (см. комментарий функции). Эта функция используется в подписке на событие перед записью. Данные, полученные из объекта данных, передаются в функцию дважды – для старого и нового экземпляров объекта.

// Выполняет поиск дат запрета по проверяемым данным

// для авторизованного пользователя или узла плана обмена.

//

// Параметры:

//  ДанныеДляПроверки - ТаблицаЗначений - возвращается функцией ШаблонДанныхДляПроверки

//                      общего модуля ДатыЗапретаИзменения.

//

//  ОписаниеДанных    - Неопределено - текст сообщения о запрете не формируется.

//                    - Структура - со свойствами:

//                      \* НоваяВерсия - Булево - если Истина, то сообщение о запрете

//                                   формировать для новой версии, иначе для старой версии.

//                      \* Данные - Ссылка,Объект - ссылка или объект данных для получения

//                                   представления, которое будет использовано в сообщении о запрете.

//                               - НаборЗаписей - набор записей регистра для получения

//                                   представления, которое будет использовано в сообщении о запрете.

//                               - Структура - со свойствами для сообщения о запрете:

//                                   \* Регистр - Строка - полное имя регистра;

//                                   \* Отбор   - Отбор - отбор набора записей.

//                               - Строка - подготовленное представление данных,

//                                 которое будет использовано в сообщении о запрете.

//

//  ОписаниеОшибки    - Строка - (возвращаемое значение) описание запрета изменения или загрузки.

//                      Не заполняется, если параметр ОписаниеДанных не указан.

//                    - Null - в этом случае текст сообщения об ошибке не будет подготовлен,

//                      что ускоряет выполнение проверки, когда запрет найден.

//

//  УзелПроверкиЗапретаЗагрузки - Неопределено - выполнить проверку изменения данных.

//                              - ПланыОбменаСсылка.<Имя плана обмена> - выполнить проверку

//                                загрузки данных для указанного узла.

//

// Возвращаемое значение:

//  Булево - если Истина, значит найден хотя бы один запрет изменения.

//

Функция НайденЗапретИзмененияДанных(Знач ДанныеДляПроверки,

                                    ОписаниеДанных = Неопределено,

                                    ОписаниеОшибки = Null,

                                    УзелПроверкиЗапретаЗагрузки = Неопределено) Экспорт

### Взаимодействие при обмене данными

Для проверки запрета загрузки следует использовать функцию **ЗагрузкаЗапрещена** модуля **ДатыЗапретаИзменения**.

Если при обмене данными требуется выполнить запись, когда признак **ОбменДанными.Загрузка** не установлен, но без проверки запрета изменения, то в свойства объекта **ДополнительныеСвойства** нужно вставить свойство **ПропуститьПроверкуЗапретаИзменения**.

### Настройка обмена данными

Для настройки обмена данными следует руководствоваться общими правилами.

В планы обмена распределенной информационной базы (РИБ) рекомендуется включать все объекты метаданных подсистемы, кроме константы **ВерсияДатЗапретаИзменения**.

Из остальных планов обмена (не РИБ) рекомендуется исключать все объекты метаданных подсистемы.

# 3.14. Дополнительные отчеты и обработки

Подсистема «Дополнительные отчеты и обработки» предназначена для подключения и использования дополнительных (внешних) отчетов и обработок к информационной базе в режиме **1С:Предприятие**.

В зависимости от назначения дополнительные отчеты и обработки бывают глобальными, если используются в целом для конфигурации, либо назначаемыми, если используются с конкретными типами объектов информационной базы. Назначаемые обработки делятся по своей функциональности на четыре типа: для заполнения объекта, печатные формы, для создания на основании и отчеты.

## Настройка

Для использования подсистемы в конфигурации необходимо поместить справочник **ДополнительныеОтчетыИОбработки** в командный интерфейс администратора. Затем выполнить шаги, описанные ниже.

### Настройка назначаемых дополнительных отчетов и обработок

Состав объектов метаданных, к которым можно подключать дополнительные печатные формы, определяется при внедрении подсистемы «[Печать](https://its.1c.ru/db/content/bsp23doc/src/3.33.%20%D0%BF%D0%B5%D1%87%D0%B0%D1%82%D1%8C.htm?_=1504708101#_подготовительные_действия)». Для остальных типов назначаемых дополнительных отчетов и обработок (**Заполнение**, **Ввод на основании** и **Отчет**) необходимо принять решение по поводу состава объектов метаданных, к которым они должны подключаться. Например, это могут быть все справочники и документы конфигурации. Эти объекты метаданных необходимо указать в свойстве **Тип**определяемого типа **ОбъектСДополнительнымиКомандами**.

Для каждого определенного выше объекта метаданных необходимо внести изменения во все его формы объекта и списка. В обработчик **ПриСозданииНаСервере** формы добавить фрагмент кода:

// СтандартныеПодсистемы.ДополнительныеОтчетыИОбработки

ДополнительныеОтчетыИОбработки.ПриСозданииНаСервере(ЭтотОбъект);

// Конец СтандартныеПодсистемы.ДополнительныеОтчетыИОбработки

В модулях форм объектов, включенных в состав определяемого типа **ОбъектСДополнительнымиКомандами**, дополнительно добавить код подсистемы:

#Область ОбработчикиКомандФормы

// СтандартныеПодсистемы.ДополнительныеОтчетыИОбработки

&НаКлиенте

Процедура Подключаемый\_ВыполнитьНазначаемуюКоманду(Команда)

Если Не ДополнительныеОтчетыИОбработкиКлиент.ВыполнитьНазначаемуюКомандуНаКлиенте(ЭтотОбъект, Команда.Имя) Тогда

ДополнительныеОтчетыИОбработкиВыполнитьНазначаемуюКомандуНаСервере(Команда.Имя);

КонецЕсли;

КонецПроцедуры

// Конец СтандартныеПодсистемы.ДополнительныеОтчетыИОбработки

#КонецОбласти

#Область СлужебныеПроцедурыИФункции

// СтандартныеПодсистемы.ДополнительныеОтчетыИОбработки

&НаСервере

Процедура ДополнительныеОтчетыИОбработкиВыполнитьНазначаемуюКомандуНаСервере(ИмяЭлемента)

ДополнительныеОтчетыИОбработки.ВыполнитьНазначаемуюКомандуНаСервере(ЭтотОбъект, ИмяЭлемента);

КонецПроцедуры

// Конец СтандартныеПодсистемы.ДополнительныеОтчетыИОбработки

#КонецОбласти

При необходимости можно переопределить расположение подменю команд типа **ЗаполнениеОбъекта**. Для этого на командную панель формы объекта следует добавить подменю с именем **ПодменюДополнительныхОбработокЗаполнения** и расположить его там, где требуется выводить команды заполнения.

### Настройка глобальных дополнительных отчетов и обработок

1. Необходимо принять решение по поводу места размещения в командном интерфейсе конфигурации команд для вызова глобальных отчетов и обработок. Рекомендуется размещать такие команды в подсистемах верхнего уровня, участвующих в формировании командного интерфейса (разделы глобального командного интерфейса).

2. Затем для каждого места размещения глобальных дополнительных обработок создать общую команду вида **ДополнительныеОбработки<место\_размещения>**, включить ее в состав функциональной опции **ИспользоватьДополнительныеОтчетыИОбработки** и добавить в нее код по шаблону:

&НаКлиенте

Процедура ОбработкаКоманды(ПараметрКоманды, ПараметрыВыполненияКоманды)

ДополнительныеОтчетыИОбработкиКлиент.ОткрытьФормуКомандДополнительныхОтчетовИОбработок(

ПараметрКоманды,

ПараметрыВыполненияКоманды,

ДополнительныеОтчетыИОбработкиКлиентСервер.ВидОбработкиДополнительнаяОбработка(),

"<Имя раздела>");

КонецПроцедуры

где **<имя раздела>**– имя раздела командного интерфейса, в котором размещена команда. Для получения имени рабочего стола следует использовать функцию **ИдентификаторРабочегоСтола** общего модуля **ДополнительныеОтчетыИОбработкиКлиентСервер**.

В качестве примера такой команды см. общую команду **ДополнительныеОбработкиАдминистрирование** в демонстрационной конфигурации.

3. Аналогичным образом для каждого места размещения глобальных дополнительных отчетов надо создать общую команду вида **ДополнительныеОтчеты<место\_размещения>**, включить ее в состав функциональной опции **ИспользоватьДополнительныеОтчетыИОбработки** и вставить в нее следующий код:

&НаКлиенте

Процедура ОбработкаКоманды(ПараметрКоманды, ПараметрыВыполненияКоманды)

ДополнительныеОтчетыИОбработкиКлиент.ОткрытьФормуКомандДополнительныхОтчетовИОбработок(

ПараметрКоманды,

ПараметрыВыполненияКоманды,

ДополнительныеОтчетыИОбработкиКлиентСервер.ВидОбработкиДополнительныйОтчет(),

"<Имя раздела>");

КонецПроцедуры

где **<имя раздела>** – имя раздела командного интерфейса, в котором размещена команда. Для получения имени рабочего стола следует использовать функцию **ИдентификаторРабочегоСтола** общего модуля **ДополнительныеОтчетыИОбработкиКлиентСервер**.

В качестве примера такой команды см. общую команду **ДополнительныеОтчетыАдминистрирование** в демонстрационной конфигурации.

4. В переопределяемом модуле **ДополнительныеОтчетыИОбработкиПереопределяемый** в процедурах **ОпределитьРазделыСДополнительнымиОбработками** и **ОпределитьРазделыСДополнительнымиОтчетами** перечислить в виде объектов метаданных разделы, в которых были размещены общие команды, созданные на предыдущих шагах. См. пример кода в демонстрационной конфигурации.

### Настройка необходимости проведения документов перед формированием внешних печатных форм

По умолчанию перед формированием любой внешней печатной формы проверяется проведенность печатаемых объектов-документов, и если найдется хотя бы один непроведенный документ, то пользователю будет выдано соответствующее предложение его провести. Если пользователь отказывается это делать, печать не выполняется.

Такую проверку можно отключить при внедрении подсистемы. Для этого необходимо в общем модуле **ДополнительныеОтчетыИОбработкиКлиентПереопределяемый** в процедуре **ПередВыполнениемКомандыПечатиВнешнейПечатнойФормы** установить параметру **СтандартнаяОбработка** значение **Ложь**:

Процедура ПередВыполнениемКомандыПечатиВнешнейПечатнойФормы(ПечатаемыеОбъекты, СтандартнаяОбработка) Экспорт

СтандартнаяОбработка = Ложь;

КонецПроцедуры

### Размещение в командном интерфейсе

Если в конфигурации не используется подсистема «Настройки программы», то в рабочем месте администратора программы необходимо разместить:

* ● константу **ИспользоватьДополнительныеОтчетыИОбработки**. При переключении значения константы регулируется возможность использования в информационной базе механизмов подсистемы. Если конфигурация адаптируется для использования в модели сервиса, отображать команду в форме настроек команды нужно только в локальном режиме работы – в модели сервиса управление значением константы выполняется централизованно из менеджера сервиса;
* ● команду открытия списка справочника **ДополнительныеОтчетыИОбработки**.

См. пример в форме **ПечатныеФормыОтчетыИОбработки** обработки **ПанельАдминистрированияБСП**.

## Особые случаи внедрения подсистемы

* ● [Внедрение подсистем «Дополнительные отчеты и обработки» и «Управление доступом»](https://its.1c.ru/db/content/bsp23doc/src/3.59.%20%D0%BE%D1%81%D0%BE%D0%B1%D1%8B%D0%B5%20%D1%81%D0%BB%D1%83%D1%87%D0%B0%D0%B8%20%D0%B2%D0%BD%D0%B5%D0%B4%D1%80%D0%B5%D0%BD%D0%B8%D1%8F%20%D0%BF%D0%BE%D0%B4%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC.htm?_=1504708101#_внедрение_подсистем_).

### Настройка прав доступа пользователей

Для настройки прав доступа пользователей к данным подсистемы «Дополнительные отчеты и обработки» следует использовать роли, указанные ниже.

Таблица 3.26.

| **№** | **Роли и их назначение** |
| --- | --- |
| 1. | **ДобавлениеИзменениеДополнительныхОтчетовИОбработок**  Добавление дополнительных обработок и отчетов в информационную базу; изменение параметров публикации и объектов назначения обработок и отчетов |
| 2. | **ЧтениеДополнительныхОтчетовИОбработок**  Возможность запуска внешних обработок.  При совместном использовании с подсистемой «**Управление доступом**» для ограничения доступа к отдельным дополнительным отчетам и обработкам используется вид доступа **ДополнительныеОтчетыИОбработки** |

Дополнительно следует создать вспомогательные роли или использовать подходящие роли, существующие в конфигурации, для обеспечения доступа к данным, которые не относятся к подсистеме «Дополнительные отчеты и обработки», но требуются для работы с ней.

Таблица 3.27.

| **№** | **Вспомогательные роли и их назначение** |
| --- | --- |
| 1. 1. | **<ЧтениеГлобальныхДополнительныхОтчетовИОбработок>**  Роль с установленным правом просмотра общих команд доступа к глобальным отчетам и обработкам (созданных на шагах 2 и 3 раздела настройки глобальных дополнительных отчетов и обработок). Назначается пользователям вместе с ролью **ЧтениеДополнительныхОтчетовИОбработок** |

Примеры настройки прав доступа пользователей приведены ниже.

Таблица 3.28.

| **№** | **Группа пользователей и ее функции** | **Состав ролей** |
| --- | --- | --- |
| 1. 1. | Администратор дополнительных отчетов и обработок.  Имеет возможность добавлять новые обработки в конфигурацию, публиковать их для общего использования, задавать объекты назначения и использовать внешние обработки в конфигурации | ● **БазовыеПрава** (из подсистемы «Базовая функциональность»),  ● **ЗапускТонкогоКлиента** (из подсистемы «Базовая функциональность»),  **●ДобавлениеИзменениеДополнительныхОтчетовИОбработок** |
| 1. 2. | Пользователь с правом выполнения дополнительных отчетов и обработок.  Имеет возможность запускать обработки на исполнение | ● **БазовыеПрава** или **БазовыеПраваВнешнегоПользователя**(из подсистемы «Базовая функциональность»),  ● **ЗапускТонкогоКлиента** (из подсистемы «Базовая функциональность»),  ● **ЧтениеДополнительныхОтчетовИОбработок**,  **●<ЧтениеГлобальныхДополнительныхОтчетовИОбработок>** |

## Использование при разработке конфигурации

### Создание нового отчета или обработки

Дополнительные отчеты и обработки могут быть глобальными (виды «ДополнительнаяОбработка» и «ДополнительныйОтчет») и назначаемыми (виды «ЗаполнениеОбъекта», «Отчет», «ПечатнаяФорма» и «СозданиеСвязанныхОбъектов»). Глобальные команды размещаются в глобальном командном интерфейсе. Назначаемые команды используются вместе с объектами информационной базы и размещаются в контексте тех объектов, для которых они предназначаются. Примеры назначаемых обработок см. в справочнике **Демо: Контрагенты** и в документе **Демо: Счета на оплату покупателям** в демонстрационной конфигурации. Общую информацию о работе с внешними отчетами и обработками можно найти в разделе [5.12.2. «Внешние отчеты и обработки»](https://its.1c.ru/db/v83doc/bookmark/dev/TI000000332) документации к платформе.

В общем виде последовательность создания дополнительной обработки (отчета) следующая:

1. Создать внешнюю обработку (отчет).

2. Регистрация дополнительной обработки в информационной базе происходит на основании сведений, которые поставляет сама обработка. Эти сведения должны возвращаться в виде структуры в функции **СведенияОВнешнейОбработке** модуля обработки. Поля структуры имеют следующее назначение:

Таблица 3.29.

| **Ключ** | **Содержание** |
| --- | --- |
| **Вид** | Строка, вид обработки, один из возможных: «ДополнительнаяОбработка», «ДополнительныйОтчет», «ЗаполнениеОбъекта», «Отчет», «ПечатнаяФорма» или «СозданиеСвязанныхОбъектов» |
| **Назначение** | Массив строк имен объектов метаданных в формате: <ИмяКлассаОбъектаМетаданного>.[ \* | <ИмяОбъектаМетаданных>].  Например, «Документ.СчетЗаказ» или «Справочник.\*». Примечание: параметр имеет смысл только для назначаемых обработок, для глобальных – не используется |
| **Наименование** | Наименование обработки, которым будет заполнено наименование элемента справочника, краткое описание для идентификации обработки администратором |
| **Версия** | Версия обработки в формате <старший номер>.<младший номер> используется при загрузке обработок в информационную базу. Например, 1.0 |
| **БезопасныйРежим** | Принимает значение **Истина** или **Ложь** в зависимости от того, требуется устанавливать или отключать безопасный режим при исполнении обработки (отчета). Если **Истина**, обработка (отчет) будет запущена в безопасном режиме. При этом, в частности, не гарантируется корректная работа дополнительных обработок под пользователями без полных прав в тех случаях, когда обработка вызывает код конфигурации, который не рассчитан на работу в безопасном режиме (в том числе неявно, когда срабатывает код модуля объекта и подписок на события при записи данных).  Более подробно о безопасном режиме см. документацию к платформе «1С:Предприятие» |
| **Информация** | Краткая информация по обработке, описание обработки |
| **Команды** | Команды, поставляемые обработкой. Таблица значений с колонками:  ● **Представление** – представление команды в пользовательском интерфейсе.  ● **Идентификатор** – идентификатор команды; любая строка, уникальная в пределах данной обработки (отчета)\*.  ● **Использование** – вариант запуска команды:  ● **ОткрытиеФормы** – открыть форму обработки;  ● **ВызовКлиентскогоМетода** – вызвать клиентскую экспортную процедуру из модуля формы обработки;  ● **ВызовСерверногоМетода** – вызвать серверную экспортную процедуру из модуля объекта обработки;  ● **СценарийВБезопасномРежиме** – вызвать серверную экспортную функцию из модуля объекта для формирования сценария выполнения (специальный вариант запуска для использования программного интерфейса, расширяющего доступные разработчику дополнительной обработки операции, при выполнении дополнительной обработки в безопасном режиме);  ● **ЗаполнениеФормы** – из модуля формы объекта вызвать серверную процедуру обработки для заполнения данных формы без записи объекта в базу данных.  ● **ПоказыватьОповещение – Истина**, если требуется показать оповещение при начале и при завершении работы обработки. Например, при запуске обработки без открытия формы.  ● **Модификатор** – дополнительный модификатор команды. Используется для дополнительных обработок печатных форм на основе табличных макетов. Для таких команд должен содержать строку **ПечатьMXL** (см. пример в демонстрационной конфигурации).  ● **ЗаменяемыеКоманды** – используется для дополнительных обработок печатных форм, содержит идентификаторы стандартных команд печати (строкой, через запятую), которые заменяются обработкой |
| **ВерсияБСП** | Минимальная версия БСП, на которую рассчитывает код дополнительной обработки. Должна быть задана не ниже 1.2.1.4.  Номер версии БСП задается в формате «РР.ПП.ВВ.СС» (РР – старший номер редакции; ПП – младший номер редакции; ВВ – номер версии; СС – номер сборки) |
| **Разрешения** | Массив разрешений, предоставляемых дополнительной обработке при использовании программного интерфейса, расширяющего доступные разработчику дополнительной обработки операции, при выполнении дополнительной обработки в безопасном режиме (подробнее см. раздел «[Расширение безопасного режима](https://its.1c.ru/db/content/bsp23doc/src/3.14.%20%D0%B4%D0%BE%D0%BF%D0%BE%D0%BB%D0%BD%D0%B8%D1%82%D0%B5%D0%BB%D1%8C%D0%BD%D1%8B%D0%B5%20%D0%BE%D1%82%D1%87%D0%B5%D1%82%D1%8B%20%D0%B8%20%D0%BE%D0%B1%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%BA%D0%B8.htm?_=1504708101#_расширение_безопасного_режима)») |
| **ОпределитьНастройкиФормы** | **Булево**. Используется для дополнительных отчетов, подключенных к подсистеме «Варианты отчетов» и использующих общую форму отчета (подробнее см. шаг 3).  Когда **Истина**, то дополнительный отчет имеет программный интерфейс для тесной интеграции с формой отчета, в том числе может переопределять некоторые настройки формы и подписываться на ее события. В этом случае в модуле объекта отчета следует определить процедуру по шаблону:  // Настройки общей формы отчета подсистемы "Варианты отчетов".  //  // Параметры:  //   Форма - УправляемаяФорма, Неопределено - Форма отчета или форма настроек отчета.  //       Неопределено, когда вызов без контекста.  //   КлючВарианта - Строка, Неопределено - Имя предопределенного  //       или уникальный идентификатор пользовательского варианта отчета.  //       Неопределено, когда вызов без контекста.  //   Настройки - Структура - см. возвращаемое значение  //       ОтчетыКлиентСервер.ПолучитьНастройкиОтчетаПоУмолчанию().  //  Процедура ОпределитьНастройкиФормы(Форма, КлючВарианта, Настройки) Экспорт  // Код процедуры.  КонецПроцедуры |

\* Для обработок с печатными формами на основе макета табличного документа должен содержать список макетов, на основе которых нужно получить печатную форму (см. описание параметра **ИменаМакетов**процедуры **УправлениеПечатьюКлиент.ВыполнитьКомандуПечати** в разделе «[Печать](https://its.1c.ru/db/content/bsp23doc/src/3.33.%20%D0%BF%D0%B5%D1%87%D0%B0%D1%82%D1%8C.htm?_=1504708101#_печать)»).

Пример реализации функции **СведенияОВнешнейОбработке** с использованием программного интерфейса:

Функция СведенияОВнешнейОбработке() Экспорт

ПараметрыРегистрации = ДополнительныеОтчетыИОбработки.СведенияОВнешнейОбработке("2.2.2.1");

ПараметрыРегистрации.Вид = ДополнительныеОтчетыИОбработкиКлиентСервер.ВидОбработки<...>();

ПараметрыРегистрации.Версия = "...";

Команда = ПараметрыРегистрации.Команды.Добавить();

Команда.Представление = НСтр("ru = '<Представление команды>'");

Команда.Идентификатор = "<Имя команды>";

Команда.Использование = ДополнительныеОтчетыИОбработкиКлиентСервер.ТипКоманды<...>();

Команда.ПоказыватьОповещение = <Истина/Ложь>;

Возврат ПараметрыРегистрации;

КонецФункции

В этом примере также демонстрируется возможность автоматического заполнения параметров **Наименование** и **Информация** из метаданных внешней обработки (берутся синоним и комментарий соответственно).

Пример реализации функции **СведенияОВнешнейОбработке** с ручным определением структуры параметров:

Функция СведенияОВнешнейОбработке() Экспорт

ПараметрыРегистрации = Новый Структура;

ПараметрыРегистрации.Вставить("Вид", ...);

ПараметрыРегистрации.Вставить("Назначение", ...);

ПараметрыРегистрации.Вставить("Наименование", ...);

ПараметрыРегистрации.Вставить("Версия", ...);

ПараметрыРегистрации.Вставить("Информация", ...);

ПараметрыРегистрации.Вставить("ВерсияБСП", "1.2.1.4");

Команды = Новый ТаблицаЗначений;

Команды.Колонки.Добавить("Представление", Новый ОписаниеТипов("Строка"));

Команды.Колонки.Добавить("Идентификатор", Новый ОписаниеТипов("Строка"));

Команды.Колонки.Добавить("Использование", Новый ОписаниеТипов("Строка"));

Команды.Колонки.Добавить("ПоказыватьОповещение", Новый ОписаниеТипов("Булево"));

Команды.Колонки.Добавить("Модификатор", Новый ОписаниеТипов("Строка"));

ПараметрыРегистрации.Вставить("Команды", Команды);

Команда = ПараметрыРегистрации.Команды.Добавить();

Команда.Представление = НСтр("ru = '<Представление команды>'");

Команда.Идентификатор = "<Имя команды>";

Команда.Использование = ДополнительныеОтчетыИОбработкиКлиентСервер.ТипКоманды<...>();

Команда.ПоказыватьОповещение = <Истина/Ложь>;

Возврат ПараметрыРегистрации;

КонецФункции

3. Для дополнительных отчетов принять решение по поводу подключения к подсистеме «Варианты отчетов» (если подсистема она внедрена в конфигурацию).

Подробнее про возможности подсистемы «Варианты отчетов» см. в [соответствующем разделе](https://its.1c.ru/db/content/bsp23doc/src/3.8.%20%D0%B2%D0%B0%D1%80%D0%B8%D0%B0%D0%BD%D1%82%D1%8B%20%D0%BE%D1%82%D1%87%D0%B5%D1%82%D0%BE%D0%B2.htm?_=1504708101#_варианты_отчетов). В частности, варианты дополнительного отчета, подключенного к подсистеме «Варианты отчетов», можно размещать в панелях отчетов программы, рядом с отчетами конфигурации.

Для подключения отчета к подсистеме «Варианты отчетов» в свойствах дополнительного отчета (меню **Действия**, команда **Свойства**) в поле **Хранилище вариантов** следует выбрать объект **ХранилищеВариантовОтчетов**.

Если дополнительный отчет подключен к подсистеме «Варианты отчетов» и использует общие формы **ФормаОтчета**, **ФормаНастроекОтчета** и **ФормаВариантаОтчета**, то в функции **СведенияОВнешнейОбработке** появляется возможность подключить обработчики, расширяющие поведение формы отчета. Для этого в свойстве **ПараметрыРегистрации.ОпределитьНастройкиФормы** следует указать значение **Истина** и в модуле объекта отчета определить процедуру по шаблону:

// Настройки общей формы отчета подсистемы "Варианты отчетов".

//

// Параметры:

//   Форма - УправляемаяФорма, Неопределено - Форма отчета или форма настроек отчета.

//       Неопределено, когда вызов без контекста.

//   КлючВарианта - Строка, Неопределено - Имя предопределенного

//       или уникальный идентификатор пользовательского варианта отчета.

//       Неопределено, когда вызов без контекста.

//   Настройки - Структура - см. возвращаемое значение

//       ОтчетыКлиентСервер.ПолучитьНастройкиОтчетаПоУмолчанию().

//

Процедура ОпределитьНастройкиФормы(Форма, КлючВарианта, Настройки) Экспорт

// Код процедуры.

КонецПроцедуры

#### Вариант запуска «Открытие формы»

Для этого варианта запуска необходимо создать форму обработки, которая будет открыта при выполнении команды:

* ● для глобальных отчетов и обработок – не требуется дополнительных действий;
* ● для назначаемых отчетов и обработок – в форме обработки добавить параметр **ОбъектыНазначения** типа**Произвольный**. В этот параметр передается массив ссылок на объекты, для которых выполняется дополнительная обработка.

**Примечание**

При использовании варианта запуска **Открытие формы** требуется самостоятельно вызвать метод **ОповеститьОбИзменении** после изменения объекта(ов), для того чтобы отразить изменения в пользовательском интерфейсе.

В форму передаются два параметра: **ИдентификаторКоманды** и **ДополнительнаяОбработкаСсылка** (ссылка на элемент справочника **ДополнительныеОтчетыИОбработки**, который связан с данной дополнительной обработкой).

#### Вариант запуска «Вызов клиентского метода»

Для этого варианта запуска необходимо создать форму обработки и в модуле формы обработки завести экспортную процедуру определенного вида.

Для глобальных отчетов и обработок – реализовать экспортную процедуру **ВыполнитьКоманду** с параметром**ИдентификаторКоманды** (**Строка**):

&НаКлиенте

Процедура ВыполнитьКоманду(ИдентификаторКоманды)

  // Реализация логики команды

  // ...

КонецПроцедуры

Для назначаемых обработок типа **ПечатнаяФорма** реализовать экспортную процедуру **Печать** с двумя параметрами: **ИдентификаторКоманды** и **ОбъектыНазначенияМассив**, где **ИдентификаторКоманды** – строка, идентификатор команды; **ОбъектыНазначенияМассив** – массив ссылок на объекты информационной базы, для которых выполняется дополнительная обработка:

&НаКлиенте

Процедура Печать(ИдентификаторКоманды, ОбъектыНазначенияМассив)

  // Реализация логики команды печати

  // ...

КонецПроцедуры

Для назначаемых обработок типа **Создание связанных объектов** – реализовать экспортную процедуру **ВыполнитьКоманду** с параметрами **ИдентификаторКоманды**, **ОбъектыНазначенияМассив** и **СозданныеОбъекты**, где **СозданныеОбъекты** – массив ссылок на созданные объекты:

&НаКлиенте

Процедура ВыполнитьКоманду(ИдентификаторКоманды, ОбъектыНазначенияМассив, СозданныеОбъекты)

  // Реализация логики команды по созданию связанных объектов

  // ...

КонецПроцедуры

Для назначаемых обработок типа **Заполнение объекта** и **Отчет** реализовать экспортную процедуру **ВыполнитьКоманду** с параметрами **ИдентификаторКоманды**, **ОбъектыНазначенияМассив**:

&НаКлиенте

Процедура ВыполнитьКоманду(ИдентификаторКоманды, ОбъектыНазначенияМассив)

  // Реализация логики команды по заполнению объекта

  // ...

КонецПроцедуры

Для всех типов дополнительных отчетов и обработок в форму передается параметр **ДополнительнаяОбработкаСсылка** – ссылка на элемент справочника **ДополнительныеОтчетыИОбработки**, который связан с данной дополнительной обработкой или отчетом.

#### Вариант запуска «Вызов серверного метода»

Для этого варианта запуска необходимо в модуле обработки требуется завести экспортную процедуру определенного вида.

Для глобальных отчетов и глобальных обработок реализовать экспортную процедуру **ВыполнитьКоманду** с параметрами **ИдентификаторКоманды** и **ПараметрыВыполненияКоманды**, где **ПараметрыВыполненияКоманды** – структура со свойством **ДополнительнаяОбработкаСсылка** (ссылка на элемент справочника **ДополнительныеОтчетыИОбработки**, который связан с данной дополнительной обработкой):

Процедура ВыполнитьКоманду(ИдентификаторКоманды, ПараметрыВыполненияКоманды) Экспорт

  // Реализация логики команды

  Если ИдентификаторКоманды = ... Тогда

    ...

  ИначеЕсли ...

КонецПроцедуры

Для назначаемых обработок типа **Создание связанных объектов** реализовать экспортную процедуру **ВыполнитьКоманду** с параметрами **ИдентификаторКоманды**, **ОбъектыНазначения**, **СозданныеОбъекты** и **ПараметрыВыполненияКоманды**:

Процедура ВыполнитьКоманду(ИдентификаторКоманды, ОбъектыНазначения, СозданныеОбъекты, ПараметрыВыполненияКоманды) Экспорт

  // Реализация логики команды по созданию связанных объектов

  Если ИдентификаторКоманды = ... Тогда

    ...

  ИначеЕсли ...

КонецПроцедуры

Для назначаемых обработок типа **ПечатнаяФорма** реализовать экспортную процедуру **Печать** с параметрами **МассивОбъектов**, **КоллекцияПечатныхФорм**, **ОбъектыПечати** и **ПараметрыВывода**. Описание параметров см. в разделе «[Печать](https://its.1c.ru/db/content/bsp23doc/src/3.33.%20%D0%BF%D0%B5%D1%87%D0%B0%D1%82%D1%8C.htm?_=1504708101#_печать)». При этом в структуре **ПараметрыВывода** содержится свойство **ДополнительнаяОбработкаСсылка** (ссылка на элемент справочника **ДополнительныеОтчетыИОбработки**, который связан с данной дополнительной обработкой).

Процедура Печать(МассивОбъектов, КоллекцияПечатныхФорм, ОбъектыПечати, ПараметрыВывода) Экспорт

  // Реализация логики команды печати

  Если ИдентификаторКоманды = ... Тогда

    ...

  ИначеЕсли ...

КонецПроцедуры

#### Пример: внешняя печатная форма

Обработка с одной печатной формой:

Функция СведенияОВнешнейОбработке() Экспорт

ПараметрыРегистрации = ДополнительныеОтчетыИОбработки.СведенияОВнешнейОбработке("2.3.1.73");

ПараметрыРегистрации.Вид = ДополнительныеОтчетыИОбработкиКлиентСервер.ВидОбработкиПечатнаяФорма();

ПараметрыРегистрации.Версия = "1.3";

// Определение объектов, к которым подключается эта обработка.

ПараметрыРегистрации.Назначение.Добавить("Документ.\_ДемоСчетНаОплатуПокупателю");

// Добавление команды печати "Счет на оплату".

НоваяКоманда = ПараметрыРегистрации.Команды.Добавить();

НоваяКоманда.Представление = НСтр("ru = 'Счет на оплату (внешняя печатная форма)'");

НоваяКоманда.Идентификатор = "СчетНаОплату";

НоваяКоманда.Использование = ДополнительныеОтчетыИОбработкиКлиентСервер.ТипКомандыВызовСерверногоМетода();

НоваяКоманда.Модификатор = "ПечатьMXL";

Возврат ПараметрыРегистрации;

КонецФункции

Процедура Печать(МассивОбъектов, КоллекцияПечатныхФорм, ОбъектыПечати, ПараметрыВывода) Экспорт

ПечатнаяФорма = УправлениеПечатью.СведенияОПечатнойФорме(КоллекцияПечатныхФорм, "СчетНаОплату");

Если ПечатнаяФорма <> Неопределено Тогда

ПечатнаяФорма.ТабличныйДокумент = СформироватьСчетНаОплатуПокупателю(МассивОбъектов, ОбъектыПечати);

ПечатнаяФорма.СинонимМакета = НСтр("ru = 'Счет на оплату'");

КонецЕсли;

КонецПроцедуры

Функция СформироватьСчетНаОплатуПокупателю(МассивОбъектов, ОбъектыПечати)

ТабличныйДокумент = Новый ТабличныйДокумент;

 ....

Возврат ТабличныйДокумент;

КонецФункции

Для подмены встроенной команды печати необходимо указать ее идентификатор в параметре **ЗаменяемыеКоманды**:

// Добавление команды печати "Счет на оплату".

НоваяКоманда = ПараметрыРегистрации.Команды.Добавить();

...

НоваяКоманда.ЗаменяемыеКоманды = "СчетЗаказ";

Список имеющихся команд печати с идентификаторами находится, как правило, в процедуре **ДобавитьКомандыПечати** модуля менеджера объекта. Если подмену делать не надо, параметр можно не указывать.

Обработка может содержать несколько печатных форм:

Функция СведенияОВнешнейОбработке() Экспорт

ПараметрыРегистрации = ДополнительныеОтчетыИОбработки.СведенияОВнешнейОбработке("2.3.1.73");

ПараметрыРегистрации.Вид = ДополнительныеОтчетыИОбработкиКлиентСервер.ВидОбработкиПечатнаяФорма();

ПараметрыРегистрации.Версия = "1.3";

// Определение объектов, к которым подключается эта обработка.

ПараметрыРегистрации.Назначение.Добавить("Документ.\_ДемоСчетНаОплатуПокупателю");

// Добавление команды печати "Счет на оплату".

НоваяКоманда = ПараметрыРегистрации.Команды.Добавить();

НоваяКоманда.Представление = НСтр("ru = 'Счет на оплату (внешняя печатная форма)'");

НоваяКоманда.Идентификатор = "СчетНаОплату";

НоваяКоманда.Использование = ДополнительныеОтчетыИОбработкиКлиентСервер.ТипКомандыВызовСерверногоМетода();

НоваяКоманда.Модификатор = "ПечатьMXL";

НоваяКоманда.ЗаменяемыеКоманды = "СчетЗаказ";

// Добавление команды печати "Заказ покупателя".

НоваяКоманда = ПараметрыРегистрации.Команды.Добавить();

НоваяКоманда.Представление = НСтр("ru = 'Заказ покупателя (внешняя печатная форма)'");

НоваяКоманда.Идентификатор = "ЗаказПокупателя";

НоваяКоманда.Использование = ДополнительныеОтчетыИОбработкиКлиентСервер.ТипКомандыВызовСерверногоМетода();

НоваяКоманда.Модификатор = "ПечатьMXL";

Возврат ПараметрыРегистрации;

КонецФункции

Процедура Печать(МассивОбъектов, КоллекцияПечатныхФорм, ОбъектыПечати, ПараметрыВывода) Экспорт

ПечатнаяФорма = УправлениеПечатью.СведенияОПечатнойФорме(КоллекцияПечатныхФорм, "СчетНаОплату");

Если ПечатнаяФорма <> Неопределено Тогда

ПечатнаяФорма.ТабличныйДокумент = СформироватьСчетНаОплатуПокупателю(МассивОбъектов, ОбъектыПечати);

ПечатнаяФорма.СинонимМакета = НСтр("ru = 'Счет на оплату'");

КонецЕсли;

ПечатнаяФорма = УправлениеПечатью.СведенияОПечатнойФорме(КоллекцияПечатныхФорм, "ЗаказПокупателя");

Если ПечатнаяФорма <> Неопределено Тогда

ПечатнаяФорма.ТабличныйДокумент = СформироватьЗаказПокупателя(МассивОбъектов, ОбъектыПечати);

ПечатнаяФорма.СинонимМакета = НСтр("ru = 'Заказ покупателя (внешняя печатная форма)'");

КонецЕсли;

КонецПроцедуры

Функция СформироватьСчетНаОплатуПокупателю(МассивОбъектов, ОбъектыПечати)

ТабличныйДокумент = Новый ТабличныйДокумент;

....

Возврат ТабличныйДокумент;

КонецФункции

Функция СформироватьЗаказПокупателя(МассивОбъектов, ОбъектыПечати)

ТабличныйДокумент = Новый ТабличныйДокумент;

....

Возврат ТабличныйДокумент;

КонецФункции

Обработка может содержать команду печати комплекта печатных форм. Описание команды печати комплекта отличается от команды печати одной печатной формы тем, что параметр **Идентификатор** содержит список идентификаторов печатных форм (этой обработки), входящих в комплект:

// Добавление команды печати "Комплект документов".

НоваяКоманда = ПараметрыРегистрации.Команды.Добавить();

НоваяКоманда.Представление = НСтр("ru = 'Комплект документов (внешняя печатная форма)'");

НоваяКоманда.Идентификатор = "СчетНаОплату,ЗаказПокупателя,ГарантийноеПисьмо";

НоваяКоманда.Использование = ДополнительныеОтчетыИОбработкиКлиентСервер.ТипКомандыВызовСерверногоМетода();

НоваяКоманда.Модификатор = "ПечатьMXL";

Пример обработки с командой печати комплекта:

Функция СведенияОВнешнейОбработке() Экспорт

ПараметрыРегистрации = ДополнительныеОтчетыИОбработки.СведенияОВнешнейОбработке("2.3.1.73");

ПараметрыРегистрации.Вид = ДополнительныеОтчетыИОбработкиКлиентСервер.ВидОбработкиПечатнаяФорма();

ПараметрыРегистрации.Версия = "1.3";

// Определение объектов, к которым подключается эта обработка.

ПараметрыРегистрации.Назначение.Добавить("Документ.\_ДемоСчетНаОплатуПокупателю");

// Добавление команды печати "Комплект документов".

НоваяКоманда = ПараметрыРегистрации.Команды.Добавить();

НоваяКоманда.Представление = НСтр("ru = 'Комплект документов (внешняя печатная форма)'");

НоваяКоманда.Идентификатор = "СчетНаОплату,ЗаказПокупателя,ГарантийноеПисьмо";

НоваяКоманда.Использование = ДополнительныеОтчетыИОбработкиКлиентСервер.ТипКомандыВызовСерверногоМетода();

НоваяКоманда.Модификатор = "ПечатьMXL";

Возврат ПараметрыРегистрации;

КонецФункции

Процедура Печать(МассивОбъектов, КоллекцияПечатныхФорм, ОбъектыПечати, ПараметрыВывода) Экспорт

ПечатнаяФорма = УправлениеПечатью.СведенияОПечатнойФорме(КоллекцияПечатныхФорм, "СчетНаОплату");

Если ПечатнаяФорма <> Неопределено Тогда

ПечатнаяФорма.ТабличныйДокумент = СформироватьСчетНаОплатуПокупателю(МассивОбъектов, ОбъектыПечати);

ПечатнаяФорма.СинонимМакета = НСтр("ru = 'Счет на оплату'");

КонецЕсли;

ПечатнаяФорма = УправлениеПечатью.СведенияОПечатнойФорме(КоллекцияПечатныхФорм, "ЗаказПокупателя");

Если ПечатнаяФорма <> Неопределено Тогда

ПечатнаяФорма.ТабличныйДокумент = СформироватьЗаказПокупателя(МассивОбъектов, ОбъектыПечати);

ПечатнаяФорма.СинонимМакета = НСтр("ru = 'Заказ покупателя (внешняя печатная форма)'");

КонецЕсли;

ПечатнаяФорма = УправлениеПечатью.СведенияОПечатнойФорме(КоллекцияПечатныхФорм, "ГарантийноеПисьмо");

Если ПечатнаяФорма <> Неопределено Тогда

ПечатнаяФорма.ТабличныйДокумент = СформироватьГарантийноеПисьмо(МассивОбъектов, ОбъектыПечати);

ПечатнаяФорма.СинонимМакета = НСтр("ru = 'Гарантийное письмо (внешняя печатная форма)'");

КонецЕсли;

КонецПроцедуры

Функция СформироватьСчетНаОплатуПокупателю(МассивОбъектов, ОбъектыПечати)

ТабличныйДокумент = Новый ТабличныйДокумент;

....

Возврат ТабличныйДокумент;

КонецФункции

Функция СформироватьЗаказПокупателя(МассивОбъектов, ОбъектыПечати)

ТабличныйДокумент = Новый ТабличныйДокумент;

....

Возврат ТабличныйДокумент;

КонецФункции

Функция СформироватьГарантийноеПисьмо(МассивОбъектов, ОбъектыПечати)

ТабличныйДокумент = Новый ТабличныйДокумент;

....

Возврат ТабличныйДокумент;

КонецФункции

Для назначаемых обработок типа **Заполнение объекта** и других типов обработок необходимо реализовать экспортную процедуру **ВыполнитьКоманду** с параметрами **ИдентификаторКоманды**, **ОбъектыНазначения** и **ПараметрыВыполненияКоманды**:

Процедура ВыполнитьКоманду(ИдентификаторКоманды, ОбъектыНазначения, ПараметрыВыполненияКоманды) Экспорт

  // Реализация логики команды по заполнению объекта

  Если ИдентификаторКоманды = ... Тогда

    ...

  ИначеЕсли ...

КонецПроцедуры

#### Пример: обработка заполнения данных формы без записи объекта

Для типа обработок **ЗаполнениеОбъекта** предусмотрена возможность работать напрямую с данными формы (а не со ссылкой на объект), что позволяет не выполнять принудительную запись объекта в форме ни до, ни после выполнения обработки.

Пример команды заполнения данных формы:

Функция СведенияОВнешнейОбработке() Экспорт

ПараметрыРегистрации =ДополнительныеОтчетыИОбработки.СведенияОВнешнейОбработке("2.2.2.1");

ПараметрыРегистрации.Вид =ДополнительныеОтчетыИОбработкиКлиентСервер.ВидОбработкиЗаполнениеОбъекта();

ПараметрыРегистрации.Версия = "1.2";

ПараметрыРегистрации.Назначение.Добавить("Справочник.\_ДемоКонтрагенты");

НоваяКоманда = ПараметрыРегистрации.Команды.Добавить();

НоваяКоманда.Представление = НСтр("ru = 'Заполнить реквизит ""ИНН"" не записывая объект (заполнение формы)'");

НоваяКоманда.Идентификатор = "ЗаполнитьИНН";

НоваяКоманда.Использование = ДополнительныеОтчетыИОбработкиКлиентСервер.ТипКомандыЗаполнениеФормы();

Возврат ПараметрыРегистрации;

КонецФункции

Процедура ВыполнитьКоманду(ИмяКоманды, ОбъектыНазначения, ПараметрыВыполнения) Экспорт

Если ИмяКоманды = "ЗаполнитьИНН" Тогда

Генератор = Новый ГенераторСлучайныхЧисел;

ЭтаФорма = ПараметрыВыполнения.ЭтаФорма;

ЭтаФорма.Объект.ИНН = Формат(Генератор.СлучайноеЧисло(1, 999999999), "ЧЦ=12; ЧДЦ=0; ЧВН=; ЧГ=");

ЭтаФорма.Модифицированность = Истина;

Сообщение = Новый СообщениеПользователю();

Сообщение.Поле = "Объект.ИНН";

Сообщение.Текст = НСтр("ru = 'Поле ""ИНН"" успешно заполнено'");

Сообщение.Сообщить();

...

КонецПроцедуры

Подробнее см. дополнительную обработку **Демо: Обработка заполнения** в справочнике **Дополнительные отчеты и обработки** демонстрационной конфигурации.

#### Пример: сохраняемые параметры дополнительной обработки

При необходимости задать параметры для дополнительной обработки можно воспользоваться реквизитом **ХранилищеНастроек** справочника **ДополнительныеОтчетыИОбработки**. Например, для заполнения документа **Распределение прочих затрат** необходимо получать цены по определенному, специально созданному для этих целей виду цен. Этот вид цен можно задавать с помощью сохраняемого параметра. В общем виде в реквизите **ХранилищеНастроек** можно сохранять произвольные параметры дополнительной обработки .(например, в виде структуры). Для доступа к реквизиту **ХранилищеНастроек** в команду выполнения обработки передается ссылка на связанный с ней элемент справочника **ДополнительныеОтчетыИОбработки**.

Пример кода для чтения параметра:

ХранилищеНастроек = ОбщегоНазначения.ЗначениеРеквизита(ОбработкаСсылка,"ХранилищеНастроек");

НастройкиСтруктура = ХранилищеНастроек.Получить();

для записи:

ДополнительнаяОбработкаОбъект = ОбъектСсылка.ПолучитьОбъект();

ДополнительнаяОбработкаОбъект.ХранилищеНастроек = НовыйХранилищеЗначения(НастройкиСтруктура);

ДополнительнаяОбработкаОбъект.Записать();

Подробнее см. дополнительные обработки **Демо: Загрузка номенклатуры из прайс-листа** в справочнике**Дополнительные отчеты и обработки** демонстрационной конфигурации.

### Фоновое выполнение длительных операций

Для того чтобы длительные операции дополнительного объекта запускались в фоновом режиме, необходимо запустить выполнение команды в фоновом задании и после его завершения принять результат.

Программный интерфейс, помогающий запускать длительные операции в фоне, представлен следующими процедурами:

1. Процедура **ВыполнитьКомандуВФоне** общего модуля **ДополнительныеОтчетыИОбработкиКлиент**.

2. В общем модуле **ДополнительныеОтчетыИОбработки**: **ВыполнитьКомандуИзФормыВнешнегоОбъекта** и **ВыполнитьКоманду**.

Рассмотрим подключение по шагам.

Шаг 1. Добавить реквизит управляемой формы для хранения ссылки внешнего объекта с типом **СправочникСсылка.ДополнительныеОтчетыИОбработки**. Например, **ОбъектСсылка**.

Шаг 2. В обработчике события управляемой формы **ПриСозданииНаСервере сохранить ссылку внешнего объекта**:

&НаСервере

Процедура ПриСозданииНаСервере(Отказ, СтандартнаяОбработка)

Параметры.Свойство("ДополнительнаяОбработкаСсылка", ОбъектСсылка);

КонецПроцедуры

Шаг 3. Добавить функцию для выполнения команды в файловом режиме по шаблону (в блок служебных процедур и функций):

&НаСервере

Функция ВыполнитьКомандуНапрямую(ИдентификаторКоманды, ПараметрыКоманды)

Возврат ДополнительныеОтчетыИОбработки.ВыполнитьКомандуИзФормыВнешнегоОбъекта(ИдентификаторКоманды, ПараметрыКоманды, ЭтотОбъект);

КонецФункции

Шаг 4. В обработчике команды управляемой формы, которая инициирует длительную операцию, добавить код по шаблону:

&НаКлиенте

Процедура ВыполнитьКомандуОбработки(Команда)

ИдентификаторКоманды = Команда.Имя;

ПараметрыКоманды = Новый Структура;

ПараметрыКоманды.Вставить("ДополнительнаяОбработкаСсылка", ОбъектСсылка);

ПараметрыКоманды.Вставить("СопровождающийТекст", НСтр("ru = 'Вставить наименование выполняемой команды'"));

Состояние(ПараметрыКоманды.СопровождающийТекст);

Если СтандартныеПодсистемыВызовСервера.ПараметрыРаботыКлиента().ИнформационнаяБазаФайловая Тогда

РезультатВыполнения = ВыполнитьКомандуНапрямую(ИдентификаторКоманды, ПараметрыКоманды);

// Добавить код обработки результата в РезультатВыполнения

// ...

Иначе

ДополнительныеОтчетыИОбработкиКлиент.ВыполнитьКомандуВФоне(ИдентификаторКоманды, ПараметрыКоманды, ЭтотОбъект);

КонецЕсли;

КонецПроцедуры

Шаг 5. В обработчике события управляемой формы **ОбработкаВыбора** добавить код для приемки результата:

&НаКлиенте

Процедура ОбработкаВыбора(ВыбранноеЗначение, ИсточникВыбора)

Если ИсточникВыбора.ИмяФормы = ДополнительныеОтчетыИОбработкиКлиент.ИмяФормыДлительнойОперации() Тогда

// Добавить код приемки результата в ВыбранноеЗначение

// ...

КонецЕсли;

КонецПроцедуры

**Примечание**

Длительные операции могут быть только серверными.

В демонстрационной базе БСП данная возможность представлена в обработках **Демо: Управление полнотекстовым поиском**, **Демо: Загрузка номенклатуры из прайс-листа (профили безопасности)**, **Демо: Обработка заполнения** и **Демо: Создание связанных объектов** в справочнике **Дополнительные отчеты и обработки**.

### Расширение безопасного режима

Подсистема «Дополнительные отчеты и обработки» содержит специальный программный интерфейс, предназначенный для расширения возможностей, доступных разработчику обработки в том случае, если требуется обеспечить работоспособность дополнительной обработки при подключении в безопасном режиме.

Следует учесть, что использование данного программного интерфейса усложняет процесс разработки дополнительного отчета или обработки и использовать его рекомендуется только в том случае, если будущий потребитель обработки предъявляет повышенные требования к безопасности (например, если обработка предназначается для размещения в сервисе «[1С:Предприятие 8 через Интернет»](https://1cfresh.com/)).

В демонстрационной базе БСП возможность расширения безопасного режима представлена в обработке **Демо: Загрузка номенклатуры из прайс-листа (сценарное выполнение)** в справочнике **Дополнительные отчеты и обработки**. Эта обработка является аналогом обработки **Демо: Загрузка номенклатуры из прайс-листа (профили безопасности)** и адаптирована к выполнению в безопасном режиме.

#### Состав программного интерфейса

Функции данного программного интерфейса расположены в общих модулях **ДополнительныеОтчетыИОбработкиВБезопасномРежиме** и **ДополнительныеОбработкиВБезопасномРежимеВызовСервера**.

В состав программного интерфейса входит следующий набор функций:

* ● Функции для преобразования полученного от пользователя файла вызовом метода глобального контекста **ПоместитьФайл** (или из другого источника, поскольку в безопасном режиме код дополнительной обработки не может взаимодействовать с файловой системой – требуется, чтобы дополнительная обработка оперировала адресом во временном хранилище, по которому расположены двоичные данные файла) в объект технологической платформы, позволяющий выполнить чтение и анализ содержимого файла:
* ● **ДополнительныеОтчетыИОбработкиВБезопасномРежиме**:
* ● **ЧтениеXMLИзДвоичныхДанных**;
* ● **ЧтениеHTMLИзДвоичныхДанных**;
* ● **ЧтениеFastInfosetИзДвоичныхДанных**;
* ● **ДополнительныеОтчетыИОбработкиВБезопасномРежимеВызовСервера**:
* ● **ТекстовыйДокументИзДвоичныхДанных**;
* ● **ТабличныйДокументИзДвоичныхДанных**;
* ● **РаспаковатьАрхив**.
* ● Функции для аналогичного обратного преобразования:
* ● **ДополнительныеОтчетыИОбработкиВБезопасномРежиме**:
* ● **ЗаписьXMLВДвоичныеДанные**;
* ● **ЗаписьHTMLВДвоичныеДанные**;
* ● **ЗаписьFastInfosetВДвоичныеДанные**;
* ● **ДополнительныеОтчетыИОбработкиВБезопасномРежимеВызовСервера**:
* ● **ТекстовыйДокументВДвоичныеДанные**;
* ● **ТабличныйДокументВДвоичныеДанные**;
* ● **ФорматированныйДокументВДвоичныеДанные**;
* ● **УпаковатьФайлыВАрхив**;
* ● Функции для интеграции с внешним программным обеспечением:
* ● **ДополнительныеОтчетыИОбработкиВБезопасномРежиме**:
* ● **СоздатьComОбъект**;
* ● **ПодключитьВнешнююКомпонентуИзОбщегоМакетаКонфигурации**;
* ● **ПодключитьВнешнююКомпонентуИзМакетаКонфигурации**;
* ● **ПолучитьФайлИзВнешнегоОбъекта**;
* ● **ПередатьФайлВоВнешнийОбъект**.
* ● Функции для взаимодействия с сетевыми ресурсами:
* ● **ДополнительныеОтчетыИОбработкиВБезопасномРежиме**:
* ● **ПолучитьФайлИзИнтернета**;
* ● **ПередатьФайлВИнтернет**;
* ● **WSСоединение**;
* ● Функции для выполнения операций, недоступных в безопасном режиме из-за того, что код конфигурации был разработан в расчете на выполнение в привилегированном режиме, автоматически установленном платформой:
* ● **ДополнительныеОтчетыИОбработкиВБезопасномРежиме**:
* ● **ПроведениеДокументов**.

Подробное описание параметров функций и возвращаемых ими значений приведено в комментариях к самим функциям.

#### Разрешения

Ряд функций описываемого программного интерфейса требуют передачи параметра **КлючСессии**. Для использования этих функций дополнительная обработка должна предварительно «запросить» разрешения на их выполнение.

«Запрос» разрешений реализуется путем заполнения свойства Разрешения структуры, возвращаемой функцией **СведенияОВнешнейОбработке()**, массивом разрешений, которые нужны дополнительной обработке для выполнения.

Для формирования массива необходимо использовать специальные функции общего модуля **ДополнительныеОтчетыИОбработкиВБезопасномРежимеИнтерфейс**. Эти функции названы по шаблону **Разрешение\*** и предполагают передачу в качестве параметров ограничений, достаточных для работы дополнительной обработки. Например, для функции **РазрешениеПолучениеДанныхИзИнтернет()** это протокол, через который будет осуществляться подключение, адрес сервера, к которому будет выполняться подключение, и номер сетевого порта, через который будет выполняться подключение.

Следует учесть, что при предоставлении сформированных разрешений они будут предоставляться именно с учетом запрошенных ограничений. Если необходимо получить разрешение без ограничений (например, адрес сервера, к которому будет осуществляться подключение, не известен заранее, а будет введен пользователем), следует передавать в качестве значения параметра-ограничителя результат выполнения функции **ДополнительныеОтчетыИОбработкиВБезопасномРежимеИнтерфейс.ЛюбоеЗначение()**.

Если дополнительной обработке предоставлены разрешения, для нее при каждом запуске генерируется ключ сессии расширения безопасного режима, связанный с разрешениями, предоставленными конкретной обработке.

Для команд со способом вызова **ОткрытиеФормы** ключ сессии передается в качестве параметра формы **КлючСессии**.

#### Способы использования

Из клиентского контекста дополнительной обработки доступны для вызова все экспортируемые функции общего модуля **ДополнительныеОтчетыИОбработкиВБезопасномРежимеВызовСервера**.

Из серверного контекста дополнительной обработки невозможен непосредственный вызов функций описываемого программного интерфейса (при попытке выполнить такой вызов будет сгенерировано исключение).

Для использования программного интерфейса из серверного контекста дополнительных обработок реализован механизм **выполнения сценария дополнительной обработки**. При использовании данного механизма предполагается, что дополнительная обработка должна сформировать сценарий своего выполнения, который может содержать как вызов процедур и функций самой обработки (в этом случае выполнение в любом случае будет происходить с установкой безопасного режима), так и вызов процедур и функций конфигурации (некоторые из которых могут выполняться без установки безопасного режима – в частности, в таком режиме могут выполняться все функции описываемого программного интерфейса). Важно, что для выполнения через данный механизм потенциально опасных действий (например, передача данных в Интернет) обработка должна предварительно «запросить» разрешения на выполнение этих действий.

Для использования механизма выполнения сценария дополнительной обработки в командах обработки, выполняющихся в серверном контексте, предназначен способ вызова команды дополнительной обработки **СценарийВБезопасномРежиме**. При выполнении команды с таким вариантом запуска подсистема **ДополнительныеОтчетыИОбработки** вызовет функцию объекта дополнительной обработки **СформироватьСценарий()**. Параметры функции **СформироватьСценарий()** аналогичны параметрам, передаваемым в процедуру **ВыполнитьКоманду** при выполнении команд со способом вызова **ВызовСерверногоМетода**. Результатом функции должна выступать таблица значений, содержащая сценарий выполнения дополнительной обработки.

Для сохранения контекста выполнения дополнительной обработки в ходе выполнения сценария предлагается использовать специальную коллекцию сохраняемых значений – любой этап сценария может сохранить результат выполнения в эту коллекцию под определенным именем, а также запросить передачу в качестве параметра процедуры или функции любого из сохраненных значений либо же самой коллекции.

При реализации функции **СформироватьСценарий()** не следует вручную создавать структуру таблицы значений. Необходимо использовать следующие функции общего модуля **ДополнительныеОтчетыИОбработкиВБезопасномРежимеИнтерфейс**:

* ● **НовыйСценарий()** – функция возвращает объект, который в дальнейшем должен рассматриваться как сценарий выполнения дополнительной обработки.
* ● Функции по добавлению этапов в сценарий:
* ● **ДобавитьМетодКонфигурации()** – добавляет в сценарий выполнения дополнительной обработки вызов процедуры или функции конфигурации. Результат, возвращаемый функцией, должен рассматриваться как этап сценария выполнения дополнительной обработки;
* ● **ДобавитьМетодОбработки()** – добавляет в сценарий выполнения дополнительной обработки вызов процедуры или функции объекта дополнительной обработки. Результат, возвращаемый функцией, должен рассматриваться как этап сценария выполнения дополнительной обработки.
* ● Функции по заполнению параметров, которые будут переданы в процедуру или функцию конфигурации/объекта внешней обработки:
* ● **ДобавитьКлючСессии()** – добавляет в массив параметров текущий ключ сессии расширения безопасного режима;
* ● **ДобавитьЗначение()** – добавляет в массив параметров фиксированное значение произвольного типа;
* ● **ДобавитьСохраняемоеЗначение()** – добавляет в массив параметров сохраняемое значение;
* ● **ДобавитьКоллекциюСохраняемыхЗначений()** – добавляет в массив параметров коллекцию сохраняемых значений;
* ● **ДобавитьПараметрВыполненияКоманды()** – добавляет в массив параметров элемент структуры **ПараметрыВыполненияКоманды**;
* ● **ДобавитьОбъектыНазначения()** – добавляет в массив параметров массив объектов назначения (актуально только для назначаемых дополнительных обработок).

При необходимости процесс выполнения сценария также может быть запущен с клиентского контекста дополнительной обработки (для команд со способом запуска **ОткрытиеФормы**) с помощью функции **ДополнительныеОтчетыИОбработкиВБезопасномРежимеВызовСервера.ВыполнитьСценарийВБезопасномРежиме()**.

### Настройка обмена данными

Все объекты метаданных подсистемы, содержащие данные, рекомендуется включать в планы обмена распределенной ИБ.

Для планов обмена автономного рабочего места (АРМ) в модели сервиса константу **ИспользоватьДополнительныеОтчетыИОбработки** следует включать в состав планов обмена, но не включать в состав подписок на события (выполняющих регистрацию данных при записи). Данное требование объясняется тем, что в автономном рабочем месте использование дополнительных отчетов и обработок настраивается независимо от их использования в экземпляре сервиса и должно передаваться только при создании начального образа.

# 3.15. Завершение работы пользователей

Подсистема «Завершение работы пользователей» позволяет завершать существующие соединения с информационной базой и устанавливать блокировку новых соединений с информационной базой на определенный период времени.

## Настройка

Для того чтобы использовать подсистему, в командный интерфейс конфигурации необходимо поместить обработки **БлокировкаРаботыПользователей** и **АктивныеПользователи**.

### Настройка прав доступа пользователей

Для настройки прав доступа пользователей к данным подсистемы «Завершение работы пользователей» следует использовать роль, указанную ниже.

Таблица 3.30.

| **№** | **Роли и их назначение** |
| --- | --- |
| 1. | **ПолныеПрава** (из подсистемы «Базовая функциональность»):  ● блокировка сеансов и завершение работы пользователей (с помощью формы **Блокировка работы пользователей**),  ● отключение сеансов пользователей из формы **Активные пользователи** |
| 2. | **АдминистраторСистемы**(из подсистемы «Базовая функциональность»):  ● блокировка запуска регламентных заданий,  ● настройка параметров администрирования кластера «1С:Предприятия» |

## Использование при разработке конфигурации

Программный интерфейс подсистемы представлен экспортными функциями общих модулей **СоединенияИБ** и **СоединенияИБКлиент**.

### Настройка обмена данными

Объекты метаданных подсистемы не следует включать в планы обмена распределенной ИБ (РИБ), а также в планы обмена по правилам конвертации, поскольку для различных узлов в них должны храниться разные данные:

● константа **ПараметрыАдминистрированияИБ**,

● регистр сведений **БлокировкиСеансовОбластейДанных**,

● регистр сведений **УдалитьБлокировкиСеансовОбластиДанных**.

# 3.16. Загрузка данных из файла

Подсистема «Загрузка данных из файла» предоставляет пользовательский и программный интерфейс для загрузки табличных данных из файлов в произвольные справочники и табличные части документов. Например, с ее помощью можно предусмотреть перенос нормативно-справочной информации при переходе с других программ, а также быстрое заполнение табличных частей при вводе документов.

## Настройка

### Загрузка данных в справочники

Если в конфигурации используется подсистема «Настройки программы», рекомендуется разместить обработку **ЗагрузкаДанныхИзФайла** на панели администрирования **Перенос данных** в группе **Загрузка данных**. См. пример размещения в демонстрационной конфигурации в форме **ПереносДанных** обработки **\_ДемоПанельАдминистрирования**.

В противном случае необходимо разместить обработку **ЗагрузкаДанныхИзФайла** в рабочем месте администратора программы.

По умолчанию загрузка данных становится доступной для большинства справочников конфигурации. Однако это не всегда целесообразно. Рекомендуется запрещать загрузку данных в справочники:

● содержащие служебную, технологическую или другую информацию, изменение которой может привести к некорректной работе программы;

● которые имеют собственные механизмы автоматического обновления или заполнения. Например, классификаторы **КлассификаторБанковРФ**, **СтраныМира** и пр.;

● пакетная загрузка в которые небезопасна (например, **ВнешниеПользователи**, **Пользователи**, **ТомаХраненияФайлов**);

● «технологические» справочники, которые не редактируют пользователи (например, **ГруппыИсполнителейЗадач**, **ИдентификаторыОбъектовМетаданных**, **ОчередьЗаданий** и т. д.).

В таких случаях в модуле **ЗагрузкаДанныхИзФайлаПереопределяемый** в методе **ПриОпределенииСправочниковДляЗагрузкиДанных** следует исключать такие справочники:

Процедура ПриОпределенииСправочниковДляЗагрузкиДанных(ЗагружаемыеСправочники) Экспорт

// Загрузка в классификатор валюты запрещена

СтрокаТаблицы = ЗагружаемыеСправочники.Найти(Метаданные.Справочники.Валюты.ПолноеИмя(), "ПолноеИмя");

Если СтрокаТаблицы <> Неопределено Тогда

ЗагружаемыеСправочники.Удалить(СтрокаТаблицы);

КонецЕсли;

КонецПроцедуры

Кроме того, по умолчанию загрузка данных запрещена для справочников:

● с одной или несколькими табличными частями (кроме табличных частей с именами **КонтактнаяИнформация**, **ДополнительныеРеквизиты**, **СертификатыШифрования**);

● с реквизитами типа **ХранилищеЗначений** (например, все справочники с присоединенными файлами или с форматированными документами).

Это ограничение обусловлено тем, что стандартный механизм загрузки не позволяет загружать данные в элементы таких справочников, а загрузка только оставшейся части реквизитов может привести к некорректным результатам.

Однако если известно, что загрузка в такой справочник без заполнения этих элементов не вызовет проблем в работе программы, то можно разрешить загрузку данных, разместив в модуле менеджера справочника экспортную функцию **ИспользоватьЗагрузкуДанныхИзФайла**, которая должна возвращать значение **Истина**.

Также имеется возможность переопределить стандартный механизм загрузки данных в справочники. Подробнее см. в разделе «[Особые случаи внедрения подсистемы](https://its.1c.ru/db/content/bsp23doc/src/3.16.%20%D0%B7%D0%B0%D0%B3%D1%80%D1%83%D0%B7%D0%BA%D0%B0%20%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85%20%D0%B8%D0%B7%20%D1%84%D0%B0%D0%B9%D0%BB%D0%B0.htm?_=1504708101#_особые_случаи_внедрения_1)».

### Загрузка данных в табличную часть документа или справочника

Нужно принять решение по поводу табличных частей объектов (документов и справочников), в которых востребована пакетная загрузка данных. Например, это могут быть табличные части **Товары** документов поступления и реализации товаров и услуг, строки которых часто заполняются на основании первичных документов: из таблиц Microsoft Excel или файлов другого формата.

Для объектов, в которые требуется добавить возможность загрузки данных из файла в табличную часть, необходимо произвести следующие действия.

#### Команда загрузки данных

В форме объекта создать команду и разместить ее на командной панели табличной части. Например, для табличной части **Товары** – команду **ЗагрузитьТоварыИзФайла** с синонимом **Загрузить из файла…** Установить свойство **Изменяет сохраняемые данные** в **Истина**. Если в объекте требуется предусмотреть загрузку в несколько табличных частей, то для каждой табличной части создается отдельная команда.

Далее в форме объекта реализовать обработку команды и процедуру добавления данных в табличную часть. Пример реализации для табличной части **Товары** документа **\_ДемоСчетНаОплатуПокупателю**:

&НаКлиенте

Процедура ЗагрузитьТоварыИзФайла(Команда)

ПараметрыЗагрузки = ЗагрузкаДанныхИзФайлаКлиент.ПараметрыЗагрузкиДанных();

ПараметрыЗагрузки.ПолноеИмяТабличнойЧасти = "\_ДемоПоступлениеТоваров.Товары";

ПараметрыЗагрузки.Заголовок = НСтр("ru = 'Загрузка списка товаров из файла'");

ДополнительныеПараметры = Новый Структура();

ДополнительныеПараметры.Вставить("Контрагент", Объект.Контрагент);

ДополнительныеПараметры.Вставить("Организация", Объект.Организация);

ПараметрыЗагрузки.ДополнительныеПараметры = ДополнительныеПараметры;

Оповещение = Новый ОписаниеОповещения("ЗагрузитьТоварыИзФайлаЗавершение", ЭтотОбъект);

ЗагрузкаДанныхИзФайлаКлиент.ПоказатьФормуЗагрузки(ПараметрыЗагрузки, Оповещение);

КонецПроцедуры

&НаКлиенте

Процедура ЗагрузитьТоварыИзФайлаЗавершение(АдресЗагруженныхДанных, ДополнительныеПараметры) Экспорт

Если АдресЗагруженныхДанных = Неопределено Тогда

Возврат;

КонецЕсли;

ЗагрузитьТоварыИзФайлаНаСервере(АдресЗагруженныхДанных);

КонецПроцедуры

&НаСервере

Процедура ЗагрузитьТоварыИзФайлаНаСервере(АдресЗагруженныхДанных)

ЗагруженныеДанные = ПолучитьИзВременногоХранилища(АдресЗагруженныхДанных);

ТоварыДобавлены = Ложь;

Для каждого СтрокаТаблицы Из ЗагруженныеДанные Цикл

Если Не ЗначениеЗаполнено(СтрокаТаблицы.Номенклатура) Тогда

Продолжить;

КонецЕсли;

НоваяСтрокаТовары = Объект.Товары.Добавить();

НоваяСтрокаТовары.Номенклатура = СтрокаТаблицы.Номенклатура;

НоваяСтрокаТовары.Цена = СтрокаТаблицы.Цена;

НоваяСтрокаТовары.Количество = СтрокаТаблицы.Количество;

НоваяСтрокаТовары.ЕдиницаИзмерения = СтрокаТаблицы.ЕдиницаИзмерения;

ТоварыДобавлены = Истина;

КонецЦикла;

Если ТоварыДобавлены Тогда

Модифицированность = Истина;

КонецЕсли;

КонецПроцедуры

#### Табличный макет для загрузки данных

Далее при объекте необходимо создать табличный макет с именем **ЗагрузкаИзФайла**. Если в объекте предусмотрена загрузка в несколько табличных частей, то к макету добавляется имя табличной части: **ЗагрузкаИзФайла<ИмяТабличнойЧасти>**.

В колонках первой строки макета перечисляются представления имен колонок загружаемых данных, а в свойствах ячейки **Имя** следует указать имя колонки таблицы значений с загружаемыми данными.

Возможны случаи, когда один реквизит табличной части может соответствовать сразу нескольким колонкам загружаемых данных. Например, значение реквизита **Номенклатура** табличной части может быть заполнено по одной из колонок: **Штрихкод**, **Артикул** или **Наименование**. В таких случаях в свойствах ячейки макета **ПараметрРасшифровки** требуется указать имя реквизита табличной части, с которым связана эта колонка. Таким образом, для колонок **Штрихкод**, **Артикул** и **Наименование** в свойстве ячейки **ПараметрРасшифровки**указывается **Номенклатура**.

В свойстве **Примечание** каждой колонки первой строки макета рекомендуется написать подсказку о назначении и типе этой колонки. Например, в колонке **Артикул** может быть указано примечание: «Артикул товара (не более 20 символов)».

См. пример макета для загрузки данных в документе **\_ДемоСчетНаОплатуПокупателю**.

Также имеется возможность динамически создавать шаблон с нужным набором колонок для загрузки данных в табличные части объектов. Подробнее см. в разделе «[Особые случаи внедрения подсистемы](https://its.1c.ru/db/content/bsp23doc/src/3.16.%20%D0%B7%D0%B0%D0%B3%D1%80%D1%83%D0%B7%D0%BA%D0%B0%20%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85%20%D0%B8%D0%B7%20%D1%84%D0%B0%D0%B9%D0%BB%D0%B0.htm?_=1504708101#_особые_случаи_внедрения_1)».

#### Процедуры в модуле менеджера документа

В модуле менеджера документа добавить экспортную процедуру **СопоставитьЗагружаемыеДанныеИзФайла**, в которой реализовать логику сопоставления загружаемых данных с данными в информационной базе:

// Производит сопоставление данных, загружаемых в табличную часть ПолноеИмяТабличнойЧасти,

// с данными в ИБ, и заполняет параметры АдресТаблицыСопоставления и СписокНеоднозначностей.

//

// Параметры:

//   ПолноеИмяТабличнойЧасти   – Строка – полное имя табличной части, в которую загружаются данные.

//   АдресЗагружаемыхДанных    – Строка – адрес временного хранилища с таблицей значений, в которой

//       находятся загруженные данные из файла. Состав колонок:

//     \* Идентификатор – Число – порядковый номер строки;

//     \* остальные колонки соответствуют колонкам макета ЗагрузкаИзФайла.

//   АдресТаблицыСопоставления – Строка – адрес временного хранилища с пустой таблицей значений,

//             являющейся копией табличной части документа,

//                                        которую необходимо заполнить из таблицы АдресЗагружаемыхДанных.

//   СписокНеоднозначностей – ТаблицаЗначений – список неоднозначных значений, для которых в ИБ имеется несколько подходящих вариантов.

//    \* Колонка       – Строка – имя колонки, в которой была обнаружена неоднозначность;

//    \* Идентификатор – Число  – идентификатор строки, в которой была обнаружена неоднозначность.

//   ДополнительныеПараметры   – ЛюбойТип – Любые дополнительные сведения.

//

Процедура СопоставитьЗагружаемыеДанные(ПолноеИмяТабличнойЧасти, АдресЗагружаемыхДанных, АдресТаблицыСопоставления, СписокНеоднозначностей, ДополнительныеПараметры) Экспорт

КонецПроцедуры

При сопоставлении данных возможны неоднозначные ситуации, когда по значениям колонок исходных (загружаемых) данных находятся сразу несколько подходящих объектов информационной базы и поэтому невозможно однозначно определить, какой именно объект информационной базы должен быть присвоен реквизиту текущей строки табличной части. Для разрешения неоднозначностей требуется реализовать экспортную процедуру **ЗаполнитьСписокНеоднозначностей**:

// Возвращает список подходящих объектов ИБ для неоднозначного значения ячейки.

//

// Параметры:

//   ПолноеИмяТабличнойЧасти  – Строка – Полное имя табличной части, в которую загружаются данные.

// ИмяКолонки   – Строка – Имя колонки, в который возникла неоднозначность.

//  СписокНеоднозначностей    – Массив  – Массив для заполнения с неоднозначными данными.

//  ЗагружаемыеЗначенияСтрока – Строка  – Загружаемые данные, на основании которых возникла неоднозначность.

//  ДополнительныеПараметры   – ЛюбойТип – Любые дополнительные сведения

//

Процедура ЗаполнитьСписокНеоднозначностей(ПолноеИмяТабличнойЧасти, СписокНеоднозначностей, ИмяКолонки, ЗагружаемыеЗначенияСтрока, ДополнительныеПараметры) Экспорт

КонецПроцедуры

См. пример реализации в документе **\_ДемоСчетНаОплатуПокупателю**.

### Особые случаи внедрения подсистемы

#### Собственный алгоритм загрузки данных

Если для загрузки элементов справочника недостаточно стандартных средств, можно предусмотреть собственный алгоритм загрузки. Для этого требуется:

1. В справочнике создать табличный макет с именем **ЗагрузкаИзФайла**, в котором определить состав колонок. Для быстрого создания макета рекомендуется запустить конфигурацию и в режиме **1С:Предприятие** выполнить команду по загрузке данных в этот справочник. Затем на шаге ввода данных в таблицу сохранить бланк для заполнения в MXL-файл и использовать в качестве заготовки для макета. Для загрузки контактной информации и дополнительных сведений в справочник в макете следует определить специальные колонки **<Контактная информация>** и **<Дополнительные сведения>** обозначающие место, где будут автоматически добавлены колонки с видами контактной информации объекта и его дополнительными реквизитами и свойствами.

2. В модуль менеджера справочника реализовать три экспортные процедуры: **ОпределитьПараметрыЗагрузкиДанныхИзФайла**, **СопоставитьЗагружаемыеДанныеИзФайла** и **ЗагрузитьИзФайла**. В них реализовать логику сопоставления и загрузки данных в реквизиты справочника.

// Устанавливает параметры загрузки данных из файла

//

// Параметры:

//     Параметры – Структура – Список параметров. Поля:

//         \* Заголовок – Строка – Заголовок окна

//         \* ОбязательныеКолонки -  Массив – Список имен колонок, обязательных для заполнения

//         \* ТипДанныхКолонки – Соответствие, Ключ – Имя колонки, Значение – Описание типа данных

//

Процедура ОпределитьПараметрыЗагрузкиДанныхИзФайла(Параметры) Экспорт

КонецПроцедуры

// Производит сопоставление загружаемых данных с данными в ИБ.

//

// Параметры:

//   ЗагружаемыеДанные – ТаблицаЗначений – таблица значений с загружаемыми данными:

//     \* СопоставленныйОбъект – СправочникСсылка – Ссылка на сопоставленный объект. Заполняется внутри процедуры

//     \* <другие колонки>     – Произвольный – Состав колонок соответствует макету "ЗагрузкаИзФайла"

//

Процедура СопоставитьЗагружаемыеДанныеИзФайла(ЗагружаемыеДанные) Экспорт

КонецПроцедуры

// Загрузка данных из файла

//

// Параметры:

//   ЗагружаемыеДанные – ТаблицаЗначений с колонками:

//     \* СопоставленныйОбъект         – СправочникСсылка – Ссылка на сопоставленный объект

//     \* РезультатСопоставленияСтроки – Строка       – Cтатусом загрузки, возможны варианты: Создан, Обновлен, Пропущен

//     \* ОписаниеОшибки               – Строка       – расшифровка ошибки загрузки данных

//     \* Идентификатор                – Число        – Уникальный номер строки

//     \* <другие колонки>             – Произвольный – Строки загружаемого файла в соответствии с макетом

// ПараметрыЗагрузки                  – Структура    – Параметры загрузки

//     \* СоздаватьНовые               – Булево       – Требуется ли создавать новые элементы справочника

//     \* ОбновлятьСуществующие        – Булево       – Требуется ли обновлять элементы справочника

// Отказ                              – Булево       – Отмена загрузки

Процедура ЗагрузитьИзФайла(ЗагружаемыеДанные, ПараметрыЗагрузки, Отказ) Экспорт

КонецПроцедуры

См. пример реализации в справочнике **\_ДемоНоменклатура** в демонстрационной конфигурации.

#### Программное создание табличного макета для загрузки данных в табличную часть

В случаях, когда набор колонок в шаблоне не статичен, а определяется динамически (например, зависит от текущих значений реквизитов объекта, статуса документа, значений функциональных опций и т.п.), есть возможность программно определить состав колонок табличного макета.

Для этого можно воспользоваться следующими функциями программного интерфейса:

● **СформироватьОписаниеКолонок** общего модуля **ЗагрузкаДанныхИзФайла**;

● **ОписаниеКолонкиМакета** и **КолонкаМакета** общего модуля **ЗагрузкаДанныхИзФайлаКлиентСервер**.

Пример реализации для табличной части **Аналоги** документа **\_ДемоНоменклатура**:

&НаКлиенте

Процедура ЗагрузитьТоварыИзФайла(Команда)

ПараметрыЗагрузки = ЗагрузкаДанныхИзФайлаКлиент.ПараметрыЗагрузкиДанных();

ПараметрыЗагрузки.ПолноеИмяТабличнойЧасти = "\_ДемоНоменклатура.Аналоги";

ПараметрыЗагрузки.Заголовок = НСтр("ru = 'Загрузка списка аналогов из файла'");

// Описание колонок для макета загрузки комплектации

ПараметрыЗагрузки.КолонкиМакета = ОписаниеКолонокМакетаДляЗагрузкиАналогов();

Оповещение = Новый ОписаниеОповещения("ЗагрузитьАналогиИзФайлаЗавершение", ЭтотОбъект);

ЗагрузкаДанныхИзФайлаКлиент.ПоказатьФормуЗагрузки(ПараметрыЗагрузки, Оповещение);

КонецПроцедуры

&НаСервере

Функция ОписаниеКолонокМакетаДляЗагрузкиАналогов();

КолонкиМакета = ЗагрузкаДанныхИзФайла.СформироватьОписаниеКолонок(Объект.Аналоги);

ЗагрузкаДанныхИзФайлаКлиентСервер.УдалитьКолонкуМакета("Аналог", КолонкиМакета);

Если Объект.ВидНоменклатуры.Наименование = "Услуга" Тогда

// У услуг нет штрихкода

Колонка = ЗагрузкаДанныхИзФайлаКлиентСервер.ОписаниеКолонкиМакета("ШтрихкодАртикул", ОбщегоНазначения.ОписаниеТипаСтрока(20), НСтр("ru = 'Артикул'"));

Иначе

Колонка = ЗагрузкаДанныхИзФайлаКлиентСервер.ОписаниеКолонкиМакета("ШтрихкодАртикул", ОбщегоНазначения.ОписаниеТипаСтрока(20), НСтр("ru = 'Штрихкод и Артикул'"));

КонецЕсли;

Колонка.ОбязательнаДляЗаполнения = Истина;

Колонка.Позиция = 1;

Колонка.Группа = "Номенклатура";

Колонка.Родитель = "Аналог";

КолонкиМакета.Добавить(Колонка);

Колонка = ЗагрузкаДанныхИзФайлаКлиентСервер.ОписаниеКолонкиМакета("Наименование", ОбщегоНазначения.ОписаниеТипаСтрока(100));

Колонка.Группа = "Номенклатура";

Колонка.Родитель = "Аналог";

Колонка.Позиция = 2;

КолонкиМакета.Добавить(Колонка);

Колонка = ЗагрузкаДанныхИзФайлаКлиентСервер.КолонкаМакета("Совместимость", КолонкиМакета);

Колонка.Позиция = 3;

Возврат КолонкиМакета;

КонецФункции

См. полностью пример реализации в справочнике **\_ДемоНоменклатура** в демонстрационной конфигурации.

### Настройка прав доступа пользователей

Таблица 3.31.

| **№** | **Роли и их назначение** |
| --- | --- |
| 1. | **ПолныеПрава**(из подсистемы «Базовая функциональность»)  Загрузка данных из файлов в справочники |

Для загрузки данных в табличные части документов не требуются какие-либо дополнительные роли. Команды загрузки доступны тем пользователям, у которых есть права на редактирование самого документа.

## Использование при разработке конфигурации

### Разработка дополнительных обработок для загрузки данных

Для расширения возможностей загрузки данных из файла в справочники можно также использовать средства подсистемы «Дополнительные отчеты и обработки».

Надо создать внешнюю обработку согласно требованиям раздела «[Дополнительные отчеты и обработки](https://its.1c.ru/db/content/bsp23doc/src/3.14.%20%D0%B4%D0%BE%D0%BF%D0%BE%D0%BB%D0%BD%D0%B8%D1%82%D0%B5%D0%BB%D1%8C%D0%BD%D1%8B%D0%B5%20%D0%BE%D1%82%D1%87%D0%B5%D1%82%D1%8B%20%D0%B8%20%D0%BE%D0%B1%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%BA%D0%B8.htm?_=1504708101#_дополнительные_отчеты_и)». Затем в функции **СведенияОВнешнейОбработке** модуля объекта обработки описать параметры команды. Например, для добавления варианта загрузки **Демо: Контрагенты (Юридические лица с контактной информацией)**:

// Возвращает сведения о внешней обработке.

Функция СведенияОВнешнейОбработке() Экспорт

ПараметрыРегистрации = ДополнительныеОтчетыИОбработки.СведенияОВнешнейОбработке("2.2.3.1");

ПараметрыРегистрации.Вид = ДополнительныеОтчетыИОбработкиКлиентСервер.ВидОбработкиДополнительнаяОбработка();

ПараметрыРегистрации.Версия = "1.0";

НоваяКоманда = ПараметрыРегистрации.Команды.Добавить();

НоваяКоманда.Представление = НСтр("ru = 'Демо: Контрагенты (Юридические лица с контактной информацией)'");

НоваяКоманда.Использование = ДополнительныеОтчетыИОбработкиКлиентСервер.ТипКомандыЗагрузкаДанныхИзФайла();

НоваяКоманда.Модификатор   = Метаданные.Справочники.\_ДемоКонтрагенты.ПолноеИмя();

НоваяКоманда.Идентификатор = "ЮрЛицаКонтрагентов";

Возврат ПараметрыРегистрации;

КонецФункции

В модуль объекта обработки разместить экспортные процедуры**ОпределитьПараметрыЗагрузкиДанныхИзФайла**,где переопределяются настройки загрузки по умолчанию, и**СопоставитьЗагружаемыеДанныеИзФайла** и **ЗагрузитьИзФайла**, в которых реализовать логику сопоставления и загрузки данных в реквизиты справочника:

// Определяет параметры загрузки данных из файла.

//

// Параметры:

//   ИдентификаторКоманды – Строка – Имя команды, определенное в функции СведенияОВнешнейОбработке().

//   ПараметрыЗагрузки – Структура – Настройки загрузки данных:

//       \* ИмяМакетаСШаблоном – Строка – Имя макета с шаблоном загружаемых данных.

//           По умолчанию используется макет "ЗагрузкаИзФайла".

//       \* ОбязательныеКолонкиМакета - Массив - Список имен колонок обязательных для заполнения.

//

Процедура ОпределитьПараметрыЗагрузкиДанныхИзФайла(ИдентификаторКоманды, ПараметрыЗагрузки) Экспорт

Если ИдентификаторКоманды = "ЮрЛицаКонтрагентов" Тогда

ПараметрыЗагрузки.ИмяМакетаСШаблоном = "ЗагрузкаИзФайлКонтрагенты";

КонецЕсли;

КонецПроцедуры

// Производит сопоставление данных, загружаемых из файла, с данными в информационной базе

//

//   ИдентификаторКоманды – Строка – Идентификатор команды

//   ЗагружаемыеДанные – ТаблицаЗначений – таблица значений с загружаемыми данными:

//     \* СопоставленныйОбъект – СправочникСсылка.Ссылка – Ссылка на сопоставленный объект. Заполняется внутри процедуры

//     \* <другие колонки>    – Строка – Состав колонок соответствует макету "ЗагрузкаИзФайла"

//

Процедура СопоставитьЗагружаемыеДанныеИзФайла(ИдентификаторКоманды, ЗагружаемыеДанные) Экспорт

КонецПроцедуры

// Производит загрузку строки сопоставленных данных в элемент справочника.

//

//   ИдентификаторКоманды – Строка – Идентификатор команды

//   ЗагружаемыеДанные – Структура – Ключ – название колонки, Значение – Данные ячейки

//     \* СопоставленныйОбъект – СправочникСсылка.Ссылка – Ссылка на сопоставленный объект

//     \* <другие колонки>    – Строка – Строки загружаемого файла

//   ПараметрыЗагрузки – Структура – Параметры с пользовательскими установками загрузки данных

// \* Обновлять – Булево – Требуется ли создавать новые объекты

// \* Создавать – Булево – Требуется ли обновлять новые объекты

// Отказ                      – Булево       – Отмена загрузки

//

Процедура ЗагрузитьИзФайла(ИдентификаторКоманды, ЗагружаемыеДанные, ПараметрыЗагрузки, Отказ) Экспорт

КонецПроцедуры

См. пример обработки загрузки данных в демонстрационной конфигурации: раздел **Администрирование – Печатные формы, отчеты и обработки – Дополнительные отчеты и обработки – Загрузка юридических лиц (с контактной информацией)**.

### Настройка обмена данными

Для настройки обмена данными никаких действий не требуется, так как подсистема не предоставляет собственных данных.

# 3.17. Заметки пользователя

Подсистема «Заметки пользователя» предназначена для хранения персональных заметок (различной неструктурированной информации, которая недоступна для других пользователей информационной базы). Заметки можно отмечать цветом, помещать на рабочий стол и объединять в группы.

## Настройка

Для использования подсистемы «Заметки пользователя» необходимо:

1. Определить состав объектов конфигурации, по поводу которых пользователи должны иметь возможность вводить заметки. Как правило, это практически все справочники.

Отметить выбранные типы объектов в свойстве **Тип** определяемыхтипов**ПредметЗаметок** (ссылки)и **ПредметЗаметокОбъект** (объекты).

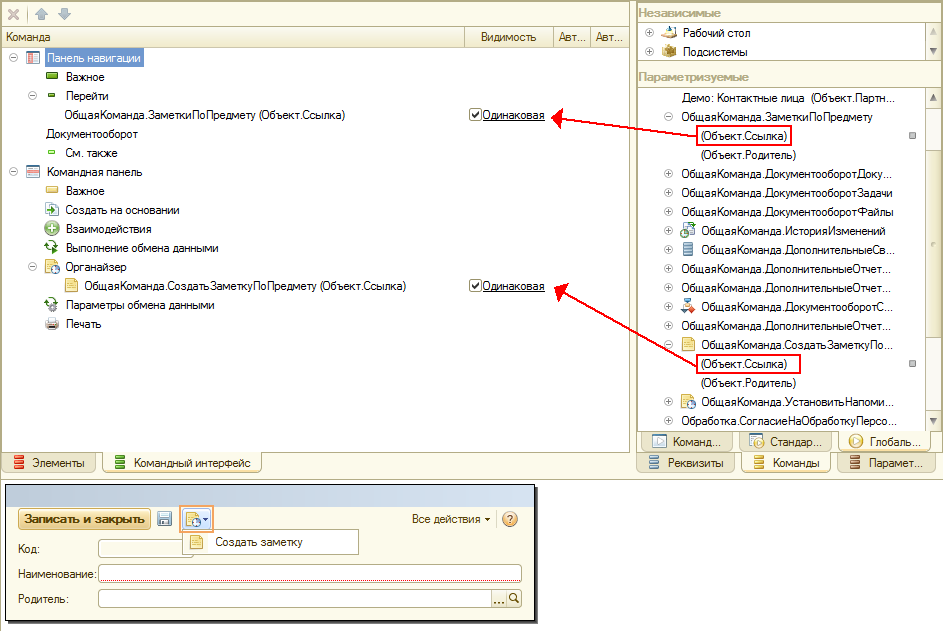
Создать подписку **УстановитьСтатусИзмененияПометкиУдаленияОбъекта** на событие **ПередЗаписью**, указав в качестве источника выбранные типы объектов (кроме документов), обработчик – **ЗаметкиПользователя.УстановитьСтатусИзмененияПометкиУдаленияОбъекта**.

2. Создать подписку **УстановитьСтатусИзмененияПометкиУдаленияДокумента** на событие **ПередЗаписью**, указав в качестве источника выбранные типы документов, обработчик – **ЗаметкиПользователя.УстановитьСтатусИзмененияПометкиУдаленияДокумента**.

3. Разместить форму **МоиЗаметки** справочника **Заметки** на рабочем столе, команду **ВсеЗаметки** справочника **Заметки** в командном интерфейсе пользователя.

**Примечание**

Общие команды **СоздатьЗаметкуПоПредмету** и **МоиЗаметкиПоПредмету** не появятся в формах групп объектов автоматически. При такой потребности эти команды необходимо поместить в командный интерфейс форм вручную:



### Настройка прав доступа пользователей

Таблица 3.32.

| **№** | **Роли и их назначение** |
| --- | --- |
| 1. | **ПолныеПрава**(из подсистемы «Базовая функциональность»)  Создание, редактирование своих заметок, просмотр заметок пользователей, удаление заметок |
| 2. | **ДобавлениеИзменениеЗаметок**  Создание, просмотр и пометка на удаление только своих заметок |

## Использование при разработке конфигурации

Для отображения информации о наличии заметок непосредственно в форме списка объекта рекомендуется предусмотреть в конфигурации вспомогательный регистр сведений, в котором программно поддерживается информация о наличии заметок по объекту. У всех пользователей без ограничений должен быть предусмотрен доступ к этому регистру на просмотр. По сравнению с конструкцией **Когда…Тогда** с непосредственной выборкой из справочника **Заметки**, при таком подходе к запросу не будут добавлены «тяжеловесные» ограничения доступа к данным на уровне записей (RLS), что позволяет повысить скорость открытия списка.

Для реализации такого подхода реквизит формы с динамическим списком необходимо переключить в режим **Произвольный запрос** и изменить текст запроса, добавив поле, как в примере в форме списка справочника **\_ДемоКонтрагенты**:

ВЫБРАТЬ

Справочник\_ДемоКонтрагенты.Ссылка,

Справочник\_ДемоКонтрагенты.ПометкаУдаления,

Справочник\_ДемоКонтрагенты.Предопределенный,

Справочник\_ДемоКонтрагенты.Код,

Справочник\_ДемоКонтрагенты.Наименование,

ВЫБОР

КОГДА ЕСТЬNULL(\_ДемоНаличиеЗаметокПоПредмету.ЕстьЗаметки, ЛОЖЬ)

ТОГДА 0

ИНАЧЕ -1

КОНЕЦ КАК ЕстьЗаметки

ИЗ

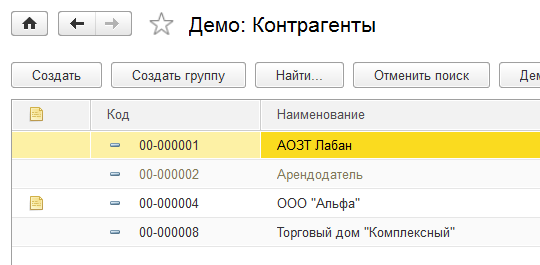
Справочник.\_ДемоКонтрагенты КАК Справочник\_ДемоКонтрагенты

ЛЕВОЕ СОЕДИНЕНИЕ РегистрСведений.\_ДемоНаличиеЗаметокПоПредмету КАК \_ДемоНаличиеЗаметокПоПредмету

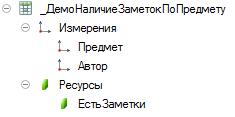
ПО (Справочник\_ДемоКонтрагенты.Ссылка = \_ДемоНаличиеЗаметокПоПредмету.Предмет

И \_ДемоНаличиеЗаметокПоПредмету.Автор = &Пользователь)

Пример списка с колонкой-индикатором наличия заметок:



Пример структуры вспомогательного регистра сведений:



### Настройка обмена данными

Все объекты метаданных подсистемы, содержащие данные, рекомендуется включать в планы обмена распределенной ИБ (РИБ).

# 3.18. Заполнение объектов

Подсистема «Заполнение объектов» предназначена для вывода команд заполнения в карточках, списках и журналах объектов программы.

Список команд определяется в составе конфигурации и автоматически выводится в подменю. Также имеется возможность переопределять список команд (добавлять новые команды и изменять существующие) при помощи расширений.

## Настройка

Необходимо принять решение, для каких объектов программы и в каких их формах требуется выводить команды заполнения.

При этом стоит учитывать сценарии подключения команд при помощи расширений. Т. е. если для объекта программы не планируется поставлять команды заполнения в составе конфигурации, но есть высокая вероятность, что партнеры захотят подключать к этому объекту свои команды заполнения (при помощи расширений), то рекомендуется подключить команды заполнения для этого объекта.

### Подключить объекты программы

В модуле менеджера каждого объекта, подключенного к подсистеме «Заполнение объектов», в области программного интерфейса должна быть определена процедура **ДобавитьКомандыЗаполнения** по шаблону:

#Область ПрограммныйИнтерфейс

// Определяет список команд заполнения.

//

// Параметры:

//   КомандыЗаполнения - ТаблицаЗначений - Таблица с командами заполнения. Для изменения.

//       См. описание 1 параметра процедуры ЗаполнениеОбъектовПереопределяемый.ПередДобавлениемКомандЗаполнения().

//   Параметры - Структура - Вспомогательные параметры. Для чтения.

//       См. описание 2 параметра процедуры ЗаполнениеОбъектовПереопределяемый.ПередДобавлениемКомандЗаполнения().

//

Процедура ДобавитьКомандыЗаполнения(КомандыЗаполнения, Параметры) Экспорт

КонецПроцедуры

#КонецОбласти

### Подключить формы объектов программы

В формах объектов, подключенных на предыдущем шаге, в которых требуется выводить подменю с командами заполнения, надо:

1. В процедуре **ПриСозданииНаСервере** (обработчик события формы) вставить вызов по шаблону:

// СтандартныеПодсистемы.БазоваяФункциональность

ПодключаемыеКоманды.ПриСозданииНаСервере(ЭтотОбъект);

// Конец СтандартныеПодсистемы.БазоваяФункциональность

2. В модуле формы вставить две процедуры (подключаемые обработчики команд):

// СтандартныеПодсистемы.БазоваяФункциональность

&НаКлиенте

Процедура Подключаемый\_ВыполнитьКоманду(Команда)

ПодключаемыеКомандыКлиент.ВыполнитьКоманду(ЭтотОбъект, Команда, <ОбъектИлиТаблицаФормы>);

КонецПроцедуры

&НаСервере

Процедура Подключаемый\_ВыполнитьКомандуНаСервере(Контекст, Результат)

ПодключаемыеКоманды.ВыполнитьКоманду(ЭтотОбъект, Контекст, <ОбъектИлиТаблицаФормы>, Результат);

КонецПроцедуры

// Конец СтандартныеПодсистемы.БазоваяФункциональность

● Если форма является формой объекта, то в параметре **<ОбъектИлиТаблицаФормы>** следует передавать основной реквизит формы типа **ДанныеФормыСтруктура**. Например:

&НаКлиенте

Процедура Подключаемый\_ВыполнитьКоманду(Команда)

ПодключаемыеКомандыКлиент.ВыполнитьКоманду(ЭтотОбъект, Команда, Объект);

КонецПроцедуры

● Если форма является формой списка, то в параметре **<ОбъектИлиТаблицаФормы>** следует передавать элемент формы типа **ТаблицаФормы**, связанный с динамическим списком. Например:

&НаКлиенте

Процедура Подключаемый\_ВыполнитьКоманду(Команда)

ПодключаемыеКомандыКлиент.ВыполнитьКоманду(ЭтотОбъект, Команда, Элементы.Список);

КонецПроцедуры

3. В командную панель формы добавить основное подменю для вывода команд заполнения по шаблону:

* ● Имя: **ПодменюЗаполнить**,
* ● Заголовок: **Заполнить**,
* ● Вид: **Подменю**,
* ● Отображение: **Картинка**.
* ● Картинка: **ЗаполнитьФорму** (картинка из конфигурации).

4. Помимо основного подменю в форме могут быть размещены дополнительные подменю (например, для заполнения табличных частей). Имена этих подменю указываются в свойстве **Группа** тех команд, которые должны выводиться в этих подменю. Имена подменю заполнения табличных частей рекомендуется образовывать путем добавления префикса с именем табличной части, например: **ТоварыПодменюЗаполнить**.

5. Если планируется, что в одном подменю будет выводиться более 10 команд, то рекомендуется добавлять вложенные группы кнопок с суффиксами **Важное**, **Обычное** и **СмТакже**. Например: **ПодменюЗаполнитьВажное**, **ПодменюЗаполнитьОбычное** и **ПодменюЗаполнитьСмТакже**. Имена этих групп указываются в свойстве **Группа** тех команд, которые должны выводиться в этих группах. Подробнее про это свойство см. раздел «[Добавить команды заполнения](https://its.1c.ru/db/content/bsp23doc/src/3.18.%20%D0%B7%D0%B0%D0%BF%D0%BE%D0%BB%D0%BD%D0%B5%D0%BD%D0%B8%D0%B5%20%D0%BE%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BE%D0%B2.htm?_=1504708101#_добавить_команды_заполнения)».

Если в форме нет групп **ПодменюЗаполнить** или **ПодменюЗаполнитьОбычное**, то в модуле менеджера не вызывается процедура **ДобавитьКомандыЗаполнения**.

### Настройка прав доступа пользователей

Для настройки прав доступа дополнительных действий не требуется.

Таблица 3.33.

| **№** | **Роли и их назначение** |
| --- | --- |
| 1. | **БазовыеПрава** или**БазовыеПраваВнешнегоПользователя**  Просмотр информации |

## Использование при разработке конфигурации

### Добавить команды заполнения

Команды заполнения, используемые сразу во всех (или во многих) объектах метаданных, описываются в процедуре **ПередДобавлениемКомандЗаполнения** модуля **ЗаполнениеОбъектовПереопределяемый**, например:

Процедура ПередДобавлениемКомандЗаполнения(КомандыЗаполнения, НастройкиОбъекта, СтандартнаяОбработка) Экспорт

...

// Универсальная команда заполнения для справочников.

Если Метаданные.Справочники.Содержит(НастройкиОбъекта.Метаданные) Тогда

Команда = КомандыЗаполнения.Добавить();

...

КонецЕсли;

...

КонецПроцедуры

Команды заполнения объекта описываются в процедуре **ДобавитьКомандыЗаполнения** модуля менеджера этого объекта, например:

// Определяет список команд заполнения.

//

// Параметры:

//   КомандыЗаполнения - ТаблицаЗначений - Таблица с командами заполнения. Для изменения.

//       См. описание 1 параметра процедуры ЗаполнениеОбъектовПереопределяемый.ПередДобавлениемКомандЗаполнения().

//   Параметры - Структура - Вспомогательные параметры. Для чтения.

//       См. описание 2 параметра процедуры ЗаполнениеОбъектовПереопределяемый.ПередДобавлениемКомандЗаполнения().

//

Процедура ДобавитьКомандыЗаполнения(КомандыЗаполнения, Параметры) Экспорт

Команда = КомандыЗаполнения.Добавить();

...

КонецПроцедуры

Также команды заполнения могут описываться в обработках конфигурации и расширений. Для этого обработка должна входить в состав подсистемы **ПодключаемыеОтчетыИОбработки** и в ее модуле менеджера в области **ПрограммныйИнтерфейс** должна быть определена процедура **ПриОпределенииНастроек** по шаблону [подключения отчетов и обработок к другим объектам метаданных](https://its.1c.ru/db/content/bsp23doc/src/3.4.%20%D0%B1%D0%B0%D0%B7%D0%BE%D0%B2%D0%B0%D1%8F%20%D1%84%D1%83%D0%BD%D0%BA%D1%86%D0%B8%D0%BE%D0%BD%D0%B0%D0%BB%D1%8C%D0%BD%D0%BE%D1%81%D1%82%D1%8C.htm?_=1504708101#_подключение_отчетов_и), а также процедура **ДобавитьКомандыЗаполнения**. Например:

#Область ПрограммныйИнтерфейс

// Определяет состав программного интерфейса для интеграции с конфигурацией.

//

// Параметры:

//   Настройки - Структура - Настройки интеграции этого объекта.

//       См. возвращаемое значение функции ПодключаемыеКоманды.НастройкиПодключаемыхОтчетовИОбработок().

//

Процедура ПриОпределенииНастроек(Настройки) Экспорт

Настройки.Размещение.Добавить(Метаданные.Документы.ИмяДокумента);

Настройки.ДобавитьКомандыЗаполнения = Истина;

КонецПроцедуры

// Определяет список команд заполнения.

//

// Параметры:

//   КомандыЗаполнения - ТаблицаЗначений - Таблица с командами заполнения. Для изменения.

//       См. описание 1 параметра процедуры ЗаполнениеОбъектовПереопределяемый.ПередДобавлениемКомандЗаполнения().

//   Параметры - Структура - Вспомогательные параметры. Для чтения.

//       См. описание 2 параметра процедуры ЗаполнениеОбъектовПереопределяемый.ПередДобавлениемКомандЗаполнения().

//

Процедура ДобавитьКомандыЗаполнения(КомандыЗаполнения, Параметры) Экспорт

Команда = КомандыЗаполнения.Добавить();

...

КонецПроцедуры

#КонецОбласти

Процедура **ДобавитьКомандыЗаполнения** обработок, подключенных по такой технологии, вызывается для всех объектов метаданных, указанных в параметре **Размещение** процедуры **ПриОпределенииНастроек**.

Таблица 3.34. Колонки таблицы «КомандыЗаполнения»

| **Параметр** | **Тип** | **Описание** |
| --- | --- | --- |
| **Идентификатор**(необязательный) | **Строка** | Идентификатор команды, который используется для идентификации команды и определения имени команды в форме.  Если идентификатор не указан, то имя команды будет определено автоматически |
| **Представление**(обязательный) | **Строка** | Представление команды в форме. Пример:  Команда.Представление = НСтр("ru = 'Заполнить ТТН'"); |
| **Группа** (необязательный) | **Строка** | Имя подменю или группы кнопок, в которой выводится эта команда |
| **Порядок**  (необязательный) | **Число** | Значение от 1 до 100, указывающее порядок размещения команды по отношению к другим командам. Сортировка команд осуществляется сначала по полю **Порядок**, затем по представлению.  Значение по умолчанию: 50 |
| **СочетаниеКлавиш**(необязательный) | **СочетаниеКлавиш** | Сочетание клавиш для быстрого вызова команды |
| **Настройки видимости:** | | |
| **ТипПараметра**(необязательный) | **ОписаниеТипов** | Типы объектов, для которых предназначена эта команда. Используется, когда обработка (или другой поставщик команд) подключена к нескольким объектам конфигурации, для уточнения списка объектов, к которым подключена конкретная команда |
| **ВидимостьВФормах**(необязательный) | **Строка** | Имена форм, в которых должна отображаться команда, через запятую. Используется для уточнения состава форм, к которым подключена конкретная команда. Если параметр не указан, то команда будет отображаться во всех формах. Пример:  Команда.ВидимостьВФормах = "ФормаДокумента" |
| **ФункциональныеОпции**  (необязательный) | **Строка** | Имена функциональных опций, определяющих видимость команды, через запятую |
| **Настройки процесса выполнения:** | | |
| **ИмяСписка**(необязательный) | **Строка** | Имя таблицы формы, связанной с динамическим списком, для которого выводится команда. Используется, когда в форме несколько списков и команда предназначена для второстепенного списка |
| **МножественныйВыбор**(необязательный) | **Булево** | Если **Истина**, то команда поддерживает множественный выбор и в 1 параметре обработчика команды будет передан массив ссылок.  Если **Ложь**, то команда поддерживает только одиночный выбор и в 1 параметре обработчика команды будет передана 1 ссылка |
| **РежимЗаписи**(необязательный) | **Строка** | Настройки дополнительных проверок и действий, связанных к записью объекта, выполняемых перед обработчиком команды:  ● **НеЗаписывать** – объект не записывается, а в параметрах обработчика вместо ссылок передается вся форма. В этом режиме рекомендуется работать напрямую с формой, которая передается в структуре 2 параметра обработчика команды.  ● **ЗаписыватьТолькоНовые** – записывать только новые объекты.  ● **Записывать** – записывать новые и модифицированные объекты.  ● **Проводить** – проводить документы.  Например:  Команда.РежимЗаписи = "НеЗаписывать";  Перед записью и проведением у пользователя запрашивается подтверждение. Значение по умолчанию: **Записывать** |
| **ТребуетсяРаботаСФайлами**(необязательный) | **Булево** | Если **Истина**, то в веб-клиенте перед выполнением команды предлагается установить расширение работы с файлами |
| **Настройки обработчика:** | | |
| **Менеджер**(необязательный) | **Строка** | Полное имя объекта метаданных, отвечающего за выполнение команды. Заполняется автоматически полным именем объекта, в модуле менеджера которого добавлена команда |
| **ИмяФормы**(обязательный, если не указан **Обработчик**) | **Строка** | Имя формы, которую требуется получить для выполнения команды. Если **Обработчик** не указан, то у формы вызывается метод **Открыть** |
| **ПараметрыФормы**(необязательный) | **Структура**, **Неопределено** | Параметры формы, указанной в **ИмяФормы** |
| **Обработчик**(обязательный, если не указано **ИмяФормы**) | **Строка** | Имя процедуры, обрабатывающей основное действие команды.  Например:  Команда.Обработчик = "ЗаполнитьТТН";  Если **ИмяФормы** заполнено, то в модуле указанной формы ожидается клиентская процедура по шаблону:  &НаКлиенте  Процедура <ИмяПроцедуры>(<ИмяПараметраКоманды>, ПараметрыВыполнения) Экспорт  // Код обработчика команды.  КонецПроцедуры  Если **ИмяФормы** не заполнено, то в модуле менеджера объекта, указанного в **Менеджер**, ожидается серверная процедура по шаблону:  Процедура <ИмяПроцедуры>(<ИмяПараметраКоманды>, ПараметрыВыполнения) Экспорт  // Код обработчика команды.  КонецПроцедуры  Также обработчик может быть размещен в общем модуле, например:  Команда.Обработчик = "<ИмяМодуля>.ЗаполнитьТТН";  В этом случае выбор шаблона процедуры зависит от того, серверный это модуль или клиентский.  Имя 1-го параметра обработчика команды **<ИмяПараметраКоманды>** рекомендуется задавать в зависимости от типов объектов, для которых предназначена команда. Также следует учитывать, что тип этого параметра зависит от настройки **МножественныйВыбор**. Если **МножественныйВыбор = Истина**, то передается массив ссылок, в противном случае передается ссылка. Например, если команда предназначена для заполнения документов **Счет на оплату** и **МножественныйВыбор** включен, то параметр можно называть **МассивСчетовНаОплату** или **СчетаНаОплату**. Если **МножественныйВыбор** отключен, то параметр можно назвать **СчетНаОплату** или **СсылкаСчетаНаОплату**.  2-й параметр обработчика команды **ПараметрыВыполнения** имеет тип **Структура** со следующими полями:  ● **ОписаниеКоманды – Структура.**Описание команды.  Структура аналогична этой таблице.  ● **Идентификатор – Строка**. Идентификатор команды.  ● **Представление – Строка**. Представление команды в форме.  ● **ДополнительныеПараметры – Неопределено**, **ФиксированнаяСтруктура**. Дополнительные параметры команды.  ● **Форма – УправляемаяФорма**. Форма, из которой вызвана команда.  ● **ЭтоФормаОбъекта – Булево**. **Истина**, если команда вызвана из формы объекта.  ● **Источник – ТаблицаФормы**,**ДанныеФормыСтруктура**. Объект или список формы с полем **Ссылка** |
| **ДополнительныеПараметры**(необязательный) | **Структура**, **Неопределено** | Дополнительные параметры, которые могут быть считаны в обработчике команды |

Примеры команд и их обработчиков см. в обработке **\_ДемоПодключаемыеКомандыЗаполнениеКонтрагентов**расширения **\_ДемоПодключаемыеКоманды** демонстрационной конфигурации.

### Настройка обмена данными

Для настройки обмена данными никаких действий не требуется, так как подсистема не предоставляет собственных данных.

# 3.19. Запрет редактирования реквизитов объектов

Подсистема «Запрет редактирования реквизитов объектов» предназначена для запрета редактирования некоторых реквизитов объектов, которые определяют характер данного объекта и которые условно можно назвать ключевыми реквизитами данного объекта. Как правило, ключевыми реквизитами являются реквизиты объекта, влияющие на поведение других объектов, например на проведение документов. Необдуманное изменение таких реквизитов может привести к рассогласованию данных учета.

Примером может служить реквизит **Валюта** справочника **Кассы**, который определяет валюту остатков по данной кассе, или признак **Является услугой** справочника **Номенклатура**, который влияет на проведение документов с такой номенклатурой. Такие реквизиты нельзя изменять после появления остатков по данной кассе или после появления документов с данной номенклатурой соответственно.

При этом подсистема допускает программное изменение ключевых реквизитов объектов. В этом случае необходимые проверки должен выполнять вызывающий код.

## Настройка

Для использования подсистемы необходимо выполнить следующие шаги:

1. Определить список объектов, для которых применяется подсистема, и состав их ключевых реквизитов (как минимум один реквизит для каждого объекта). Перечислить список этих объектов в процедуре **ПриОпределенииОбъектовСЗаблокированнымиРеквизитами** общего модуля **ЗапретРедактированияРеквизитовОбъектовПереопределяемый**.

2. Принять решение по поводу состава объектов метаданных, наличие ссылок из которых на данный объект метаданных не должно влиять на возможность редактирования его ключевых реквизитов. Такие объекты следует включить в фильтр поиска исключений ссылок (см. раздел «[Настройка исключений поиска ссылок на объекты](https://its.1c.ru/db/content/bsp23doc/src/3.4.%20%D0%B1%D0%B0%D0%B7%D0%BE%D0%B2%D0%B0%D1%8F%20%D1%84%D1%83%D0%BD%D0%BA%D1%86%D0%B8%D0%BE%D0%BD%D0%B0%D0%BB%D1%8C%D0%BD%D0%BE%D1%81%D1%82%D1%8C.htm?_=1504708101#_настройка_исключений_поиска)»).

Для каждого объекта метаданных, для которого применяется подсистема, надо выполнить следующие шаги:

1. Добавить в модуль менеджера объекта функцию **ПолучитьБлокируемыеРеквизитыОбъекта**, в которой вернуть список имен ключевых реквизитов/табличных частей объекта или реквизитов формы. После имени реквизита можно указать точку с запятой и через запятую перечислить имена дополнительных элементов формы, имеющих отношение к реквизиту, которые тоже нужно заблокировать:

// Возвращает описание блокируемых реквизитов.

//

// Возвращаемое значение:

//  Массив - содержит строки в формате ИмяРеквизита[;ИмяЭлементаФормы,...]

//           где ИмяРеквизита - имя реквизита объекта, ИмяЭлементаФормы - имя элемента формы,

//           связанного с реквизитом.

//

Функция ПолучитьБлокируемыеРеквизитыОбъекта() Экспорт

Результат = Новый Массив;

Результат.Добавить("ИмяРеквизитаОбъекта1");

Результат.Добавить("ИмяТабличнойЧастиОбъекта1");

Результат.Добавить("ИмяРеквизитаФормы1");

Результат.Добавить("ИмяРеквизитаОбъекта2; ИмяЭлементаФормы1, ИмяЭлементаФормы2");

...

Возврат Результат;

КонецФункции

2. Добавить в модуль формы объекта процедуру:

&НаКлиенте

Процедура Подключаемый\_РазрешитьРедактированиеРеквизитовОбъекта(Команда)

ЗапретРедактированияРеквизитовОбъектовКлиент.РазрешитьРедактированиеРеквизитовОбъекта(ЭтотОбъект);

КонецПроцедуры

3. Добавить в обработчики событий **ПриСозданииНаСервере** и **ПослеЗаписиНаСервере** код:

// Обработчик подсистемы запрета редактирования реквизитов объектов

ЗапретРедактированияРеквизитовОбъектов.ЗаблокироватьРеквизиты(ЭтотОбъект);

**Примечание**

Вторым параметром функции может выступать группа формы, на которую будет прикреплена команда разрешения редактирования ключевых реквизитов.

4. Для ключевых реквизитов в свойствах объекта рекомендуется выставлять свойство **Проверка заполнения** как **Выдавать ошибку**либо вручную осуществлять проверку ключевых реквизитов на заполненность.

Пример использования подсистемы можно посмотреть в демонстрационной конфигурации в справочнике **\_ДемоНоменклатура**.

### Переопределение логики, разрешающей изменение ключевых реквизитов

Существует возможность задать нестандартные условия разрешения изменения ключевых реквизитов. Для этого нужно сделать следующее.

Во-первых, переопределить содержимое процедуры модуля формы объекта **Подключаемый\_РазрешитьРедактированиеРеквизитовОбъекта**. Пример:

&НаКлиенте

Процедура Подключаемый\_РазрешитьРедактированиеРеквизитовОбъекта(Команда)

ЗаблокированныеРеквизиты = ЗапретРедактированияРеквизитовОбъектовКлиент.Реквизиты(ЭтотОбъект);

Если ЗаблокированныеРеквизиты.Количество() > 0 Тогда

ПараметрыФормы = Новый Структура;

ПараметрыФормы.Вставить("Ссылка", Объект.Ссылка);

ПараметрыФормы.Вставить("ЗаблокированныеРеквизиты", ЗаблокированныеРеквизиты);

ОткрытьФорму("Документ.\_ДемоЗаказПокупателя.Форма.РазблокированиеРеквизитов", ПараметрыФормы,

ЭтотОбъект,,,, Новый ОписаниеОповещения("ПослеВыбораРеквизитовДляРазблокирования", ЭтотОбъект));

Иначе

ЗапретРедактированияРеквизитовОбъектовКлиент.ПоказатьПредупреждениеВсеВидимыеРеквизитыРазблокированы();

КонецЕсли;

КонецПроцедуры

&НаКлиенте

Процедура ПослеВыбораРеквизитовДляРазблокирования(РазблокируемыеРеквизиты, Контекст) Экспорт

Если ТипЗнч(РазблокируемыеРеквизиты) <> Тип("Массив") Тогда

Возврат;

КонецЕсли;

ЗапретРедактированияРеквизитовОбъектовКлиент.УстановитьДоступностьЭлементовФормы(ЭтотОбъект,

РазблокируемыеРеквизиты);

КонецПроцедуры

Во-вторых, создать свою форму **РазблокированиеРеквизитов**. В ней позволить пользователю пометить нужные реквизиты и вернуть результат с помощью метода **Закрыть**. Пример:

&НаКлиенте

Процедура РазрешитьРедактирование(Команда)

РазблокируемыеРеквизиты = Новый Массив;

РазблокируемыеРеквизиты.Добавить("<Имя реквизита>");

…

Закрыть(РазблокируемыеРеквизиты);

КонецПроцедуры

В случаях, когда необходимо самостоятельно рассчитывать и устанавливать значения свойств **ТолькоПросмотр**или **Доступность** соответствующих полей разблокируемых реквизитов с учетом разных причин, вместо процедуры **УстановитьДоступностьЭлементовФормы** нужно вызвать процедуру **УстановитьРазрешенностьРедактированияРеквизитов**. Она не изменяет доступность полей реквизитов, а только отмечает реквизиты как разрешенные для редактирования, что нужно для корректной работы функции **Реквизиты**. Кроме того, если какие-то из реквизитов нужно считать недоступными для разблокировки, например, если для реквизита формы нет права редактирования, заданного через роли, или по другой причине, можно переопределить значение свойства **ПравоРедактирования** с помощью процедуры **УстановитьРазрешенностьРедактированияРеквизитов**. Это необходимо для корректного отключения доступности команды разблокировки реквизитов, когда все реквизиты уже разблокированы.

Форму, которая будет разрешать редактирование реквизитов, следует назвать **РазблокированиеРеквизитов**, чтобы поддержать групповое изменение объектов информационной базы. Форма должна возвращать массив, содержащий имена реквизитов, которые нужно разблокировать. Для возврата массива следует использовать метод **Закрыть**. Кроме того, если форма обслуживает более одного реквизита, то рекомендуется использовать параметр формы **ЗаблокированныеРеквизиты**, в который передается **Массив** имен реквизитов для разблокирования.

### Размещение в командном интерфейсе

Все действия по изменению командного интерфейса выполняются при интеграции подсистемы (см. выше).

### Настройка прав доступа пользователей

Для настройки прав доступа пользователей к данным подсистемы «Запрет редактирования реквизитов объектов» следует использовать роль, указанную ниже.

Таблица 3.35.

| **№** | **Роли и их назначение** |
| --- | --- |
| 1. | **РедактированиеРеквизитовОбъектов**  Редактирование заблокированных значений реквизитов объектов |

Пример настройки прав доступа пользователей приведен ниже.

Таблица 3.36.

| **№** | **Группа пользователей и ее функции** | **Состав ролей** |
| --- | --- | --- |
| 1. | Ответственный за ведение документов | **<ДобавлениеИзменениеДокументов>** – основная роль, предоставляющая право редактирования документа.  **РедактированиеРеквизитовОбъектов** |

## Использование при разработке конфигурации

### Настройка обмена данными

Для настройки обмена данными никаких действий не требуется, так как подсистема не предоставляет собственных данных.

# 3.20. Защита персональных данных

Подсистема «Защита персональных данных» предназначена для соответствия информационной системы, построенной на основе конфигурации, требованиям Федерального закона № 152-ФЗ от 27.06.2006 «О персональных данных».

## Настройка

Для внедрения подсистемы «Защита персональных данных» необходимо принять решение по поводу того, какие сущности в конфигурации являются субъектами персональных данных, т. е. к каким сущностям относятся защищаемые персональные данные. Как правило, это такие справочники, как **Физические лица**, **Сотрудники**,**Контрагенты**, **Партнеры** и др.

Затем необходимо задать состав типов субъектов персональных данных в качестве определяемого типа **СубъектПерсональныхДанных**.

В общем модуле **ЗащитаПерсональныхДанныхПереопределяемый** в процедуре **ДополнитьДанныеСубъектовПерсональныхДанных** описать определение данных субъекта, необходимых для заполнения документа согласия на обработку персональных данных. См. пример реализации этой процедуры в демонстрационной конфигурации.

Если в конфигурации не используется подсистема «Настройки программы», то в рабочем месте администратора программы необходимо разместить обработку **ЗащитаПерсональныхДанных** и ее команду **ЗащитаПерсональныхДанных**. См. пример в форме **ПоддержкаИОбслуживание** обработки **ПанельАдминистрированияБСП**.

После этого необходимо подключить механизм настройки регистрации доступа к персональным данным, который включает регистрацию событий журнала регистрации типа **Доступ.Доступ**. Для этого в подсистемах конфигурации определяются процедуры, в которых заполняются сведения о персональных данных и областях персональных данных. Вызовы таких процедур осуществляются из методов, расположенных в переопределяемом общем модуле **ЗащитаПерсональныхДанныхПереопределяемый**, –**ЗаполнитьСведенияОПерсональныхДанных** и **ЗаполнитьОбластиПерсональныхДанных** соответственно.

Описание сведений о персональных данных представляет собой таблицу значений с колонками типа **Строка**:

● **Объект** – полное имя объекта метаданных;

● **ПоляРегистрации** – имена полей регистрации, разделенные запятой. При необходимости задать альтернативные поля их можно перечислить, разделяя символом «**|**»;

● **ПоляДоступа** – имена полей доступа, разделенные запятой;

● **ОбластьДанных** – имя области персональных данных, к которой относятся описываемые сведения (заполнение не обязательно).

Описание областей персональных данных является обязательным, только если при заполнении сведений определена их принадлежность к конкретной области. Описание областей также заполняется в виде таблицы значений с колонками типа **Строка**:

● **Имя** – имя области данных (так, как указано в таблице сведений);

● **Представление** – пользовательское представление области;

● **Родитель** – данные, в состав которых входит описываемая область.

Логика заполнения сведений о персональных данных и областей должна определяться исходя из состава субъектов персональных данных, а также состава данных информационной базы, относящихся к персональным. Пример такого заполнения см. в процедурах **ЗаполнитьСведенияОПерсональныхДанных** и **ЗаполнитьОбластиПерсональныхДанных** общего модуля**ЗащитаПерсональныхДанныхПереопределяемый**.

События **Доступ.Доступ** и **Доступ.Отказ** **в доступе** выводятся в форме обработки **ЗащитаПерсональныхДанных** в том случае, если в информационной базе включена их регистрация. При этом следует иметь в виду, что событие **Доступ.Отказ в доступе** по умолчанию включено.

Для управления регистрацией событий доступа к персональным данным в форме настройки системы необходимо разместить реквизит и элементы для управления:

● добавить реквизит формы **ОбластиПерсональныхДанных** типа **ДеревоЗначений**,

● добавить таблицу формы **ОбластиПерсональныхДанных**, связанную с этим реквизитом.

Разместить вызовы процедур общего модуля **ЗащитаПерсональныхДанных**:

● в обработчике **ПриСозданииНаСервере – ПриСозданииФормыНастройкиРегистрацииСобытий**,

● в обработчике **ПриЗаписиНаСервере – ПриЗаписиФормыНастройкиРегистрацииСобытий**.

Далее необходимо определить место в командном интерфейсе для включения объектов работы с согласиями на обработку персональных данных субъектов:

● журнала **СогласияНаОбработкуПерсональныхДанных**,

● отчетов **СогласияНаОбработкуПерсональныхДанныхДействующие** и **СогласияНаОбработкуПерсональныхДанныхИстекающие**.

### Настройка прав доступа пользователей

Для настройки прав доступа пользователей к инструментам подсистемы «Защита персональных данных» следует использовать роли, приведенные ниже.

Таблица 3.37.

| **№** | **Роли и их назначение** |
| --- | --- |
| 1. | **ПросмотрЖурналаРегистрации** (из подсистемы «Базовая функциональность»)  Право просмотра журнала регистрации |
| 2. | **Администрирование** (из подсистемы «Базовая функциональность») |

Для доступа к согласиям на обработку персональных данных необходимо создать в конфигурации роли по аналогии с демонстрационными ролями **\_ДемоДобавлениеИзменениеСогласийНаОбработкуПерсональныхДанных** и **\_ДемоЧтениеСогласийНаОбработкуПерсональныхДанных**, описать в них текст необходимого ограничения доступа на уровне записей.

### Настройка обмена данными

Все объекты метаданных подсистемы, содержащие данные, рекомендуется включать в планы обмена распределенной ИБ (РИБ).

Исключением являются перечисленные ниже объекты. Их не следует включать в планы обмена распределенной ИБ (РИБ), а также в планы обмена по правилам конвертации, поскольку для различных узлов в них должны храниться разные данные: регистр сведений **ОбластиПерсональныхДанных**.

# 3.21. Интеграция с «1С-Коннект (Бухфон)»

Подсистема предоставляет возможность пользователям инфраструктуры «1С-Коннект» (Бухфон) запускать приложение из конфигурации. В интерфейсе пользователя представляет собой кнопку на рабочем столе и форму для настроек. Разработчик может по своему усмотрению настроить расположение кнопки, а также место вызова настроек. Использование всей подсистемы регулируется функциональной опцией. Имеется возможность в переопределяемом модуле указать свои действия при изменении настроек пользователем. Например, скрыть форму, если кнопка не выводится на форму и является единственным элементом формы.

Настройки кнопки включают в себя признак отображения, логин/пароль и разрешение на их подстановку в программу при запуске. Настройки хранятся в разрезе пользователей и клиента «1С» (при запуске пользователем базы с другого ПК требуется указать настройки повторно, так как расположение файла заведомо неизвестно).

## Настройка

Нужно определить форму на рабочем столе для размещения кнопки запуска «1С-Коннект». Добавить общую команду **Запустить1СБухфон** на выбранную форму.

В процедуру модуля формы **ПриСозданииНаСервере** добавить следующий код:

// СтандартныеПодсистемы.Интеграция1СБухфон

Если ПравоДоступа("Просмотр", Метаданные.ОбщиеКоманды.Запустить1СБухфон) Тогда

Интеграция1СБухфон.ПриСозданииНаСервере(Элементы.<Имя элемента>);

КонецЕсли;

// Конец СтандартныеПодсистемы.Интеграция1СБухфон

В процедуру модуля формы **ОбработкаОповещения** добавить код:

// СтандартныеПодсистемы.Интеграция1СБухфон

Если Элементы.Найти("<Имя элемента>") <> Неопределено Тогда

Интеграция1СБухфонКлиент.ОбработкаОповещения(ИмяСобытия, Элементы.<Имя элемента>);

КонецЕсли;

// Конец СтандартныеПодсистемы.Интеграция1СБухфон

Пример размещения см. в общей форме **\_ДемоПоддержкаПользователей**.

Далее необходимо разместить общую команду **Настройка1СБухфон** в форме персональных настроек пользователя. Пример размещения см. в общей форме **\_ДемоМоиНастройки**.

Если в конфигурации не используется подсистема «Настройки программы», то в рабочем месте администратора программы необходимо разместить константу **ИспользоватьИнтеграцию1СБухфон**. См. пример в форме **ОбщиеНастройки** обработки **ПанельАдминистрированияБСП**.

### Настройка прав доступа пользователей

Необходимо принять решение по поводу категории пользователей, для которой должна быть доступна подсистема, и добавить роль **Интеграция1СБухфон** в необходимые профили групп доступа.

Таблица 3.38.

| **№** | **Роли и их назначение** |
| --- | --- |
| 1. | **БазовыеПрава**  Обязательная роль для работы с подсистемой |
| 2. | **Интеграция1СБухфон**  Персональная настройка и запуск «1С-Коннект» |

Примеры настройки прав доступа пользователей приведены ниже.

Таблица 3.39.

| **№** | **Группа пользователей и ее функции** | **Состав ролей** |
| --- | --- | --- |
| 1. | Администратор.  Включение или запрет интеграции с «1С-Коннект» | **ПолныеПрава** (из подсистемы «Базовая функциональность»).  **ЗапускТонкогоКлиента** (из подсистемы «Базовая функциональность») |
| 2. | Бухгалтер.  Персональная настройка и запуск «1С-Коннект» | **БазовыеПрава** (из подсистемы «Базовая функциональность»).  **ЗапускТонкогоКлиента** (из подсистемы «Базовая функциональность»).  **Интеграция1СБухфон** |

## Использование при разработке конфигурации

### Настройка обмена данными

Все объекты метаданных подсистемы, содержащие данные, рекомендуется включать в планы обмена распределенной ИБ (РИБ).

# 3.22. Информация при запуске

Подсистема «Информация при запуске» отображает HTML-страницы с различной информацией при запуске программы (например, рекламу). Страницы содержатся в макетах обработки **ИнформацияПриЗапуске**. Каждый макет содержит стартовую страницу, а также может содержать другие страницы, ссылки которых указаны в стартовой странице. Для каждого макета на командной панели обработки автоматически создается команда, по щелчку по которой осуществляется переход к странице с информацией.

## Настройка

Определить набор страниц с информацией, которая должна выводиться при запуске программы.

Каждый набор страниц запаковать в ZIP-архив таким образом, чтобы стартовая страница была в корне этого архива.

Загрузить архивы в обработку **ИнформацияПриЗапуске**, добавив их в виде макетов из двоичных данных.

Описать настройки каждого макета (архива) в макете **Описатель** обработки **ИнформацияПриЗапуске**, указав свойства:

● **Имя макета** – имя макета, как оно задано в метаданных;

● **Раздел** – наименование подменю, в котором будет размещена команда вызова информации, размещенной в макете. Если наименование подменю не указано, то команда будет размещена прямо в командной панели;

● **Наименование** – наименование команды, по щелчку по которой будет осуществлен переход к странице с информацией;

● **Имя стартового файла** – имя стартовой HTML-страницы в ZIP-архиве. Указывается только имя, без слешей и имени архива;

● **Дата начала показа** – дата начала действия этой строки;

● **Дата окончания показа** – дата окончания действия этой строки;

● **Включать в показ при открытии** – используется для определения стартовой страницы, отображаемой при открытии обработки. Если «0» или "" (не указано), то не показывать при открытии. «1» – включать в первую очередь, «2» – включать во вторую очередь. При прочих равных (в случае возникновения конкуренции) страница выбирается случайным образом;

● **Показывать в ПРОФ** – если **Истина**, то информация будет показана в локальном режиме в версии ПРОФ;

● **Показывать в базовой** – если **Истина**, то информация будет показана в локальном режиме в базовой версии;

● **Показывать в модели сервиса** – если **Истина**, то информация будет показана в модели сервиса.

### Настройка прав доступа пользователей

Для настройки прав доступа дополнительных действий не требуется.

Таблица 3.40.

| **№** | **Роли и их назначение** |
| --- | --- |
| 1. | **БазовыеПрава или БазовыеПраваВнешнегоПользователя**  Просмотр информации |

### Настройка обмена данными

Регистр сведений **ПакетыИнформацииПриЗапуске** рекомендуется включать только в состав начального образа подчиненного узла распределенной ИБ (см. раздел «[Особенности создания начального образа подчиненного узла распределенной ИБ](https://its.1c.ru/db/content/bsp23doc/src/3.28.%20%D0%BE%D0%B1%D0%BC%D0%B5%D0%BD%20%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D0%BC%D0%B8.htm?_=1504708101#_особенности_создания_начального)»).

# 3.23. Календарные графики

Подсистема «Календарные графики» предназначена для хранения графиков, по которым работает предприятие.

С подсистемой поставляются готовые функции, которые позволяют получать различную информацию по рабочим дням, входящим в график.

## Настройка

Для использования подсистемы в конфигурации необходимо поместить справочник **ПроизводственныеКалендари** в командный интерфейс конфигурации.

### Настройка прав доступа пользователей

Для настройки прав доступа пользователей к данным подсистемы «Календарные графики» следует использовать роли, приведенные ниже.

Таблица 3.41.

| **№** | **Роли и их назначение** |
| --- | --- |
| 1. | **БазовыеПрава** (из подсистемы «Базовая функциональность»)  Чтение календарей и графиков |
| 2. | **ДобавлениеИзменениеКалендарныхГрафиков**  Добавление и изменение календарей и графиков |
| 3. | **ПолныеПрава** (из подсистемы «Базовая функциональность»)  Удаление помеченных на удаление объектов подсистемы |

## Использование при разработке конфигурации

Программный интерфейс подсистемы представлен экспортными функциями общего модуля **КалендарныеГрафики**.

### Настройка обмена данными

Все объекты метаданных подсистемы, содержащие данные, рекомендуется включать в планы обмена распределенной ИБ (РИБ).

# 3.24. Контактная информация

Подсистема «Контактная информация» позволяет расширять состав реквизитов объекта произвольным набором реквизитов контактной информации (КИ). Реквизиты КИ могут быть как предопределенными (адреса, телефоны, факс, e-mail, www), так и пользовательскими (например, «телефон с 18:00 до 24:00»). Справочник со встроенной контактной информацией называется владельцем КИ.

**Важно!**

Вид КИ **Адрес** использует возможности подсистемы «Адресный классификатор». Основной режим использования подсистемы предполагает, что в конфигурацию также переносится подсистема «Адресный классификатор».

## Настройка

После выполнения переноса объектов библиотеки необходимо включить в командный интерфейс конфигурации справочники:

* ● **ВидыКонтактнойИнформации**,
* ● **Страны мира**.

Настройка объектов подсистемы сводится к созданию в справочнике **ВидыКонтактнойИнформации**предопределенных элементов и вызову обработчика обновления информационной базы, использующего процедуру **УстановитьСвойстваВидаКонтактнойИнформации** из модуля **УправлениеКонтактнойИнформацией**, для установки значения реквизитов этих элементов.

1. Принять решение по поводу состава объектов – владельцев контактной информации и видов контактной информации у каждого такого объекта. Например, у справочника **КонтактныеЛица** необходимо разместить телефон и адрес, а у справочника **Контрагент** – адрес электронной почты.

2. Добавить ссылки на объекты-владельцы (кроме документов) в состав определяемого типа **ВладелецКонтактнойИнформации**.

3. Добавить ссылки на документы-владельцы в подписку на событие **ЗаполнитьКонтактнуюИнформациюДокумента**.

4. Перечислить виды контактной информации в справочнике **ВидыКонтактнойИнформации**. Справочник видов контактной информации – иерархический.

* ● На первом уровне вводятся предопределенные группы: для каждого вида объекта – владельца контактной информации – своя группа. У группы должно быть задано имя в формате **Справочник<ИмяСправочника>**или **Документ<ИмяДокумента>**. Например, для справочника физических лиц – **СправочникФизическиеЛица**. Наименование группы рекомендуется задавать в виде: **Контактная информация справочника "Физические лица"**.
* ● На втором уровне вводятся элементы – виды контактной информации, список которых определяет разработчик конфигурации. На втором уровне могут быть также заведены предопределенные группе, если предполагается хранение контактной информации для табличной части объекта. В этом случае у группы должно быть задано имя в формате **Документ<ИмяСправочника><ИмяТабличнойЧасти>**.
* ● На третьем уровне вводятся элементы – виды контактной информации для табличной части объекта. Список видов контактной информации определяет ответственный за конфигурацию.

5. Добавить обработчик обновления информационной базы, предназначенный для первоначального заполнения и обновления значения реквизитов, предопределенных видов КИ при помощи процедуры **УстановитьСвойстваВидаКонтактнойИнформации** модуля **УправлениеКонтактнойИнформацией**.

Пример процедуры обновления информационной базы:

// Процедура обновления ИБ для справочника видов контактной информации

Процедура КонтактнаяИнформацияОбновлениеИБ() Экспорт

// Справочник "Контрагенты"

ПараметрыВида = УправлениеКонтактнойИнформацией.ПараметрыВидаКонтактнойИнформации(Перечисления.ТипыКонтактнойИнформации.Адрес);

ПараметрыВида.Вид = Справочники.ВидыКонтактнойИнформации.\_ДемоАдресКонтрагента;

ПараметрыВида.Используется = Истина;

ПараметрыВида.МожноИзменятьСпособРедактирования = Истина;

ПараметрыВида.Порядок = 1;

ПараметрыВида.НастройкиПроверки.АдресТолькоРоссийский = Истина;

ПараметрыВида.НастройкиПроверки.ПроверятьКорректность = Истина;

ПараметрыВида.НастройкиПроверки.ПроверятьПоФИАС = Истина;

УправлениеКонтактнойИнформацией.УстановитьСвойстваВидаКонтактнойИнформации(ПараметрыВида);

КонецПроцедуры

### Настройка объектов – владельцев контактной информации

1. У объекта – владельца контактной информации необходимо создать табличную часть **КонтактнаяИнформация**со следующими обязательными реквизитами:

Таблица 3.42.

| **Имя** | **Тип значения** | **Подсказка** |
| --- | --- | --- |
| **Тип** | **ПеречислениеСсылка.ТипыКонтактнойИнформации** | Тип контактной информации (телефон, адрес и т. п.) |
| **Вид** | **СправочникСсылка.ВидыКонтактнойИнформации** | Вид контактной информации |
| **Представление** | Строка (500) | Представление контактной информации для отображения в формах |
| **ЗначенияПолей** | Строка неограниченной длины | Служебное поле для хранения контактной информации |
| **Страна** | Строка (100) | Страна (заполняется для адреса) |
| **Регион** | Строка (50) | Регион (заполняется для адреса) |
| **Город** | Строка (50) | Город (заполняется для адреса) |
| **АдресЭП** | Строка (100) | Адрес электронной почты |
| **ДоменноеИмяСервера** | Строка (100) | Доменное имя сервера электронной почты или веб-страницы |
| **НомерТелефона** | Строка (20) | Полный номер телефона |
| **НомерТелефонаБезКодов** | Строка (20) | Номер телефона без кодов и добавочного номера |

Для реквизитов **Вид** и **Тип** необходимо включить индексирование.

Для всех строковых реквизитов ограниченной длины следует устанавливать индексы по необходимости. Т. е. индексирование должно быть включено только для тех реквизитов, которые используются для выборки данных.

Кроме того, для задействования дополнительных возможностей подсистемы список реквизитов табличной части **КонтактнаяИнформация** может также включать в себя:

Таблица 3.43.

| **Имя** | **Тип значения** | **Подсказка** |
| --- | --- | --- |
| **ИдентификаторСтрокиТабличнойЧасти**\* | **Число** (7,0) | Идентификатор строки табличной части |
| **ВидДляСписка**\*\* | **СправочникСсылка.ВидыКонтактнойИнформации** | Вид контактной информации отображаемый в списках |
| **ДействуетС**\*\*\* | **Дата** | Дата, с которой действует запись контактной информации |

\* если предполагается хранение контактной информации для табличной части объекта. В этом случае необходимо также добавить этот же реквизит в табличную часть объекта. Пример реализации см. в документе **\_ДемоЗаказПокупателя** демонстрационной конфигурации.

\*\* если предполагается [вывод полей контактной информации в списках и отчетах](https://its.1c.ru/db/content/bsp23doc/src/3.24.%20%D0%BA%D0%BE%D0%BD%D1%82%D0%B0%D0%BA%D1%82%D0%BD%D0%B0%D1%8F%20%D0%B8%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%86%D0%B8%D1%8F.htm?_=1504708101#_вывод_полей_контактной).

\*\*\* если предполагается [хранение истории изменения контактной информации](https://its.1c.ru/db/content/bsp23doc/src/3.24.%20%D0%BA%D0%BE%D0%BD%D1%82%D0%B0%D0%BA%D1%82%D0%BD%D0%B0%D1%8F%20%D0%B8%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%86%D0%B8%D1%8F.htm?_=1504708101#_хранение_истории_изменений_1).

2. Изменить форму элемента:

* ● Если предполагается хранение контактной информации для объекта, то в форме элемента создать группу или страницу с именем **ГруппаКонтактнаяИнформация** и задать ей заголовок – **Адреса, телефоны** (или другой).
* ● Если предполагается хранение контактной информации для табличной части объекта, то создать группу колонок с именем **<ИмяТабличнойЧасти>ГруппаКонтактнаяИнформация** в нужной табличной части объекта. Например, для табличной части **Партнеры** группа контактной информации должна иметь вид **ПартнерыГруппаКонтактнаяИнформация**.

3. Изменить модуль формы элемента.

* ● В обработчики событий формы **ПриСозданииНаСервере**, **ПриЧтенииНаСервере**,**ОбработкаПроверкиЗаполненияНаСервере** и **ПередЗаписьюНаСервере** вставить строки из соответствующих процедур, располагающихся ниже. Если предполагается хранение контактной информации для табличной части объекта, то в обработчик события формы **ПослеЗаписиНаСервере** вставить строки из соответствующей процедуры ниже. Добавляемые строки должны располагаться после проверок, которые могут вызвать отказ в событии.

#Область ОбработчикиСобытийФормы

&НаСервере

Процедура ПриСозданииНаСервере(Отказ, СтандартнаяОбработка)

// СтандартныеПодсистемы.КонтактнаяИнформация

УправлениеКонтактнойИнформацией.ПриСозданииНаСервере(ЭтотОбъект, Объект, "ГруппаКонтактнаяИнформация", ПоложениеЗаголовкаЭлементаФормы.Лево);

// Конец СтандартныеПодсистемы.КонтактнаяИнформация

КонецПроцедуры

&НаСервере

Процедура ПриЧтенииНаСервере(ТекущийОбъект)

// СтандартныеПодсистемы.КонтактнаяИнформация

УправлениеКонтактнойИнформацией.ПриЧтенииНаСервере(ЭтотОбъект, ТекущийОбъект);

// Конец СтандартныеПодсистемы.КонтактнаяИнформация

КонецПроцедуры

&НаСервере

Процедура ОбработкаПроверкиЗаполненияНаСервере(Отказ, ПроверяемыеРеквизиты)

// СтандартныеПодсистемы.КонтактнаяИнформация

УправлениеКонтактнойИнформацией.ОбработкаПроверкиЗаполненияНаСервере(ЭтотОбъект, Объект, Отказ);

// Конец СтандартныеПодсистемы.КонтактнаяИнформация

КонецПроцедуры

&НаСервере

Процедура ПередЗаписьюНаСервере(Отказ, ТекущийОбъект, ПараметрыЗаписи)

// СтандартныеПодсистемы.КонтактнаяИнформация

УправлениеКонтактнойИнформацией.ПередЗаписьюНаСервере(ЭтотОбъект, ТекущийОбъект);

// Конец СтандартныеПодсистемы.КонтактнаяИнформация

КонецПроцедуры

&НаСервере

Процедура ПослеЗаписиНаСервере(ТекущийОбъект, ПараметрыЗаписи)

// СтандартныеПодсистемы.КонтактнаяИнформация

УправлениеКонтактнойИнформацией.ПослеЗаписиНаСервере(ЭтотОбъект, ТекущийОбъект);

// Конец СтандартныеПодсистемы.КонтактнаяИнформация

КонецПроцедуры

#КонецОбласти

где **<ГруппаКонтактнаяИнформация>**, ссылка на элемент формы – контейнер, в который будут добавляться элементы для редактирования контактной информации. Например:

УправлениеКонтактнойИнформацией.ПриСозданииНаСервере(ЭтотОбъект, Объект, "ГруппаКонтактнаяИнформация");

* ● В вызове процедуры **УправлениеКонтактнойИнформацией.ПриСозданииНаСервере** в качестве третьего параметра нужно указать имя элемента, в котором будут располагаться создаваемые поля ввода. Четвертый (необязательный) параметр позволяет задать положение заголовка элементов контактной информации. Заголовки могут располагаться сверху и слева (значения параметра **ПоложениеЗаголовкаЭлементаФормы.Лево** и **ПоложениеЗаголовкаЭлементаФормы.Верх**соответственно).

4. Перенести блок процедур контактной информации, расположенный ниже, в модуль формы объекта – владельца контактной информации.

// СтандартныеПодсистемы.КонтактнаяИнформация

&НаКлиенте

Процедура Подключаемый\_КонтактнаяИнформацияПриИзменении(Элемент)

УправлениеКонтактнойИнформациейКлиент.ПриИзменении(ЭтотОбъект, Элемент);

КонецПроцедуры

&НаКлиенте

Процедура Подключаемый\_КонтактнаяИнформацияНачалоВыбора(Элемент, ДанныеВыбора, СтандартнаяОбработка)

УправлениеКонтактнойИнформациейКлиент.НачалоВыбора(ЭтотОбъект, Элемент, , СтандартнаяОбработка);

КонецПроцедуры

&НаКлиенте

Процедура Подключаемый\_КонтактнаяИнформацияПриНажатии(Элемент, СтандартнаяОбработка)

УправлениеКонтактнойИнформациейКлиент.НачалоВыбора(ЭтотОбъект, Элемент, , СтандартнаяОбработка);

КонецПроцедуры

&НаКлиенте

Процедура Подключаемый\_КонтактнаяИнформацияОчистка(Элемент, СтандартнаяОбработка)

УправлениеКонтактнойИнформациейКлиент.Очистка(ЭтотОбъект, Элемент.Имя);

КонецПроцедуры

&НаКлиенте

Процедура Подключаемый\_КонтактнаяИнформацияВыполнитьКоманду(Команда)

УправлениеКонтактнойИнформациейКлиент.ВыполнитьКоманду(ЭтотОбъект, Команда.Имя);

КонецПроцедуры

&НаСервере

Процедура Подключаемый\_ОбновитьКонтактнуюИнформацию(Результат)

УправлениеКонтактнойИнформацией.ОбновитьКонтактнуюИнформацию(ЭтотОбъект, Объект, Результат);

КонецПроцедуры

// Конец СтандартныеПодсистемы.КонтактнаяИнформация

### Настройка прав доступа пользователей

Для настройки прав доступа пользователей к данным подсистемы «Контактная информация» следует использовать роли, перечисленные ниже.

Таблица 3.44.

| **№** | **Роли и их назначение** |
| --- | --- |
| 1. | **БазовыеПрава**(из подсистемы «Базовая функциональность»)  Чтение видов контактной информации и списка стран |
| 2. | **ДобавлениеИзменениеВидовКонтактнойИнформации**  Ведение списка стран |
| 3. | **ПолныеПрава** (из подсистемы «Базовая функциональность»)  Ведение видов контактной информации, списка стран, а также удаление помеченных на удаление объектов подсистемы |

Примеры настройки прав доступа пользователей приведены ниже.

Таблица 3.45.

| **№** | **Группа пользователей и ее функции** | **Состав ролей** |
| --- | --- | --- |
| 1. | Администратор | ● **ПолныеПрава** (из подсистемы «Базовая функциональность») |
| 2. | Ответственный за нормативно-справочную информацию (НСИ): ведение видов контактной информации, списка стран, а также всей остальной НСИ | ● **БазовыеПрава** (из подсистемы «Базовая функциональность»),  ● **ЗапускТонкогоКлиента**(из подсистемы «Базовая функциональность»),  ●**ДобавлениеИзменениеВидовКонтактнойИнформации** |
| 3. | Менеджер по закупкам: ведение списка номенклатуры, ведение списка стран мира (для ввода стран происхождения товаров) | ● **БазовыеПрава** (из подсистемы «Базовая функциональность»),  ● **ЗапускТонкогоКлиента** (из подсистемы «Базовая функциональность»),  ●**ДобавлениеИзменениеВидовКонтактнойИнформации**,  **● <ДобавлениеИзменениеНоменклатуры>** |

### Особые случаи внедрения подсистемы

#### Вывод полей контактной информации в списках и отчетах

Для вывода контактной информации в динамических списках и отчетах необходимо выполнить следующую настройку. В объекте-владельце контактной информации в табличную часть **КонтактнаяИнформация** добавить реквизит **ВидДляСписка** с типом **СправочникСсылка**.**ВидКонтактнойИнформации**, затем в форме **Дополнительные характеристики объекта данных** добавить дополнительную характеристику:

Таблица 3.46.

|  |  |  |
| --- | --- | --- |
| **Виды характеристик** | | |
| **Поле ключа** | **Поле отбора видов** | **Значение отбора видов** |
| **Справочник.ВидыКонтактнойИнформации** | | |
| **Ссылка** | **Родитель** | **<Предопределенная группа справочника  ВидыКонтактнойИнформации>** |

Таблица 3.47.

|  |  |  |
| --- | --- | --- |
| **Значения характеристик** | | |
| **Поле объекта** | **Поле вида** | **Поле значения** |
| **Справочник.<Имя объекта с контактной информацией>.ТабличнаяЧасть.КонтактнаяИнформация** | | |
| **Ссылка** | **ВидДляСписка** | **Представление** |

где **<Предопределенная группа (элемент) справочника ВидыКонтактнойИнформации>** – предопределенная группа(элемент), созданная на этапе формирования видов контактной информации.

Для отображения контактной информации в списках, в форме списка в меню **Еще** выбрать пункт **Изменить форму**. В таблице **Элементы формы** выбрать реквизит **Ссылка**, нажать кнопку **Добавить поля** в командной панели формы и далее в открывшемся списке полей выбрать нужные поля контактной информации для отображения.

См. пример реализации в справочнике **\_ДемоКонтрагенты** демонстрационной конфигурации.

#### Создание статических элементов управления

Создание статических реквизитов контактной информации позволяет:

* ● добавлять элементы контактной информации в различные части формы (например, вынести самые важные виды контактной информации в шапку);
* ● изменять внешний вид и свойства элементов контактной информации без внесения изменений в типовую конфигурацию;
* ● повысить скорость открытия формы – владельца контактной информации.

Для создания статических элементов контактной информации необходимо:

* ● принять решение о видах контактной информации, элементы управления которых должны создаваться вручную;
* ● в обработчике **ПриСозданииНаСервере** добавить выбранные виды в массив и передать его в качестве пятого параметра в процедуру **УправлениеКонтактнойИнформацией.ПриСозданииНаСервере**;
* ● добавить в форму реквизит **ПараметрыКонтактнойИнформации**, тип – произвольный;
* ● добавить в форму реквизит **КонтактнаяИнформацияОписаниеДополнительныхРеквизитов**, тип – таблица значений с колонками:
* ● **ИмяРеквизита – Строка**;
* ● **Вид – СправочникСсылка.ВидыКонтактнойИнформации**;
* ● **Тип – ПеречислениеСсылка.ТипыКонтактнойИнформации**;
* ● **ЗначенияПолей – СписокЗначений**, **Строка**;
* ● **Представление – Строка**;
* ● **Комментарий – Строка**;
* ● **ЭтоРеквизитТабличнойЧасти – Булево**;
* ● **ЭтоИсторическаяКонтактнаяИнформация – Булево**;
* ● **ИмяЭлементаДляРазмещения – Строка**;
* ● для каждого вида контактной информации добавить реквизит с именем вида **КонтактнаяИнформацияПоле<ИмяВидаКонтактнойИнформации>**;
* ● добавить на форму элементы:
* ● для каждого вида контактной информации добавить на форму группу с именем **ГруппаКонтактнаяИнформацияПоле <ИмяВидаКонтактнойИнформации>**. Свойства: **Отображение** – **Нет**, **Группировка** – **Горизонтальная**, **ОтображатьЗаголовок** – **Ложь**;
* ● в созданную группу перетащить ранее созданный реквизит с именем вида **КонтактнаяИнформацияПоле<ИмяВидаКонтактнойИнформации>**, установив соответствующее отображение заголовка (**Лево** или **Верх**) в зависимости от положения заголовка, передаваемого в процедуру **УправлениеКонтактнойИнформацией.ПриСозданииНаСервере**;
* ● для добавления действия необходимо создать команду с именем вида **КомандаКонтактнаяИнформацияПоле<ИмяВидаКонтактнойИнформации>**, отображение **Картинка**. Затем эту команду необходимо перетащить в ранее созданную группу **ГруппаКонтактнаяИнформацияПоле <ИмяВидаКонтактнойИнформации>**. Картинка команды, заголовок и подисказка устанавливаются в зависимости от типа контактной информации:
* ● для типа **Адрес** действие добавляется только в том случае, если имеется несколько видов контактной информации типа **Адрес**. В этом случае для команды устанавливается картинка **ПереместитьВлево**, заголовок **Заполнить**, подсказка **Заполнить адрес из другого поля**;
* ● для типа **ВебСтраница** картинка **КонтактнаяИнформацияПерейтиПоСсылке**, заголовок **Перейти**, подсказка **Перейти по ссылке**;
* ● для типа **АдресЭлектроннойПочты** действие устанавливается только при наличии подсистемы «[**РаботаСПочтовымиСообщениями**](https://its.1c.ru/db/content/bsp23doc/src/3.43.%20%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%B0%20%D1%81%20%D0%BF%D0%BE%D1%87%D1%82%D0%BE%D0%B2%D1%8B%D0%BC%D0%B8%20%D1%81%D0%BE%D0%BE%D0%B1%D1%89%D0%B5%D0%BD%D0%B8%D1%8F%D0%BC%D0%B8.htm?_=1504708101#_работа_с_почтовыми)**»**. При этом устанавливается картинка **ОтправитьЭлектронноеПисьмо**, заголовок **Написать письмо**, подсказка **Написать письмо**;
* ● для типа **Телефон** при наличии подсистемы «[ОтправкаSMS](https://its.1c.ru/db/content/bsp23doc/src/3.30.%20%D0%BE%D0%B1%D0%BD%D0%BE%D0%B2%D0%BB%D0%B5%D0%BD%D0%B8%D0%B5%20%D0%BA%D0%BE%D0%BD%D1%84%D0%B8%D0%B3%D1%83%D1%80%D0%B0%D1%86%D0%B8%D0%B8.htm?_=1504708101#_отправка_sms)» используется картинка **ПозвонитьИлиОтправитьSMS**, заголовок **Позвонить или отправить SMS**, подсказка **Позвонить или отправить SMS**. Если подсистемы «[ОтправкаSMS](https://its.1c.ru/db/content/bsp23doc/src/3.30.%20%D0%BE%D0%B1%D0%BD%D0%BE%D0%B2%D0%BB%D0%B5%D0%BD%D0%B8%D0%B5%20%D0%BA%D0%BE%D0%BD%D1%84%D0%B8%D0%B3%D1%83%D1%80%D0%B0%D1%86%D0%B8%D0%B8.htm?_=1504708101#_отправка_sms)» нет, то используется картинка **Позвонить**, заголовок**Позвонить**, подсказка **Позвонить по телефону**;
* ● для типа **Skype** необходимо установить картинку, заголовок и подсказку **Skype**.

Виды контактной информации, добавляемые в форму вручную, рекомендуется сделать недоступными для редактирования пользователем. Это можно сделать в обработчике обновления при помощи процедуры **УстановитьСвойстваВидаКонтактнойИнформации** из общего модуля **УправлениеКонтактнойИнформацией**. За недоступность редактирования пользователем отвечает свойство **ЗапретитьРедактированиеПользователем**.

Пример использования см. в форме элемента справочника **\_ДемоКонтрагенты** демонстрационной конфигурации.

#### Отложенная инициализация элементов контактной информации

Если элементы контактной информации (или их часть) располагаются на отдельной закладке и работа с ними не является основным сценарием работы с формой, то можно увеличить скорость открытия формы путем переноса создания элементов контактной информации с момента открытия формы на момент перехода на закладку контактной информации.

Использование отложенной инициализации также накладывает следующие ограничения:

* ● невозможно перетаскивать элементы со страницы контактной информации,
* ● в некоторых случаях возможны «скачки» формы.

Для использования отложенной инициализации необходимо:

* ● добавить в форму реквизит **ПараметрыКонтактнойИнформации**, тип – произвольный;
* ● добавить в форму реквизит **КонтактнаяИнформацияОписаниеДополнительныхРеквизитов**, тип – таблица значений. Описание колонок и их типов см. в предыдущем разделе;
* ● в обработчик **УправлениеКонтактнойИнформацией.ПриСозданииНаСервере** передать в качестве шестого параметра (**ОтложеннаяИнициализация**) значение **Истина**;
* ● для страницы контактной информации установить свойство **РазрешитьИзменениеСостава** в значение **Ложь**;
* ● для группы страниц, содержащей страницу контактной информации, добавить обработчик **ПриСменеСтраницы**, в который добавить код:

// СтандартныеПодсистемы.КонтактнаяИнформация

Если ТекущаяСтраница.Имя = ЭтотОбъект.ПараметрыКонтактнойИнформации.ГруппаДляРазмещения И Не

     ЭтотОбъект.ПараметрыКонтактнойИнформации.ВыполненаОтложеннаяИнициализация Тогда

  КонтактнаяИнформацияПриСменеСтраницы();

КонецЕсли;

// Конец СтандартныеПодсистемы.КонтактнаяИнформация

* ● в блок процедур контактной информации добавить процедуру:

&НаСервере

Процедура КонтактнаяИнформацияПриСменеСтраницы()

УправлениеКонтактнойИнформацией.ВыполнитьОтложеннуюИнициализацию(ЭтотОбъект, Объект);

КонецПроцедуры

* ● если на странице контактной информации нет статических элементов, то добавить декорацию с именем **ПустаяДекорацияКонтактнаяИнформация**.

Пример использования см. в форме элемента справочника **\_ДемоКонтрагенты** демонстрационной конфигурации.

#### Хранение истории изменений контактной информации

В случае если для ведения учета важно хранить не только текущие значения полей контактной информации, но и их историю, а также получать значение на указанную дату, необходимо настроить хранение истории изменения контактной информации:

1. Принять решение по поводу состава объектов – владельцев контактной информацией и видов контактной информации, у которых необходимо хранить историю изменения контактной информации. Например, у справочника **Организации** это может быть юридический адрес.

2. Добавить в табличную часть **КонтактнаяИнформация** этих объектов реквизит **ДействуетС** с типом **Дата**.

3. Написать обработчик обновления, устанавливающий флаг о необходимости хранения история изменений для вида контактной информации. Пример кода обработчика:

ПараметрыВида = УправлениеКонтактнойИнформацией.ПараметрыВидаКонтактнойИнформации("Адрес");

ПараметрыВида.ХранитьИсториюИзменений = Истина;

  УправлениеКонтактнойИнформацией.УстановитьСвойстваВидаКонтактнойИнформации(ПараметрыВида);

Для получения контактной информации на заданную дату необходимо в процедурах и функциях **КонтактнаяИнформацияОбъекта**, **КонтактнаяИнформацияОбъектов**, **СоздатьВТКонтактнаяИнформация**, **ТаблицаКонтактнойИнформацииОбъекта** указывать параметр **Дата** (например, дату документа в коде печатных форм). Для записи контактной информации на заданную дату следует использовать процедуры **ЗаполнитьКонтактнуюИнформациюОбъектов** и **ЗаполнитьКонтактнуюИнформациюОбъекта** аналогичным образом. При включенной истории хранения изменений не допускается возможность ввода нескольких значений контактной информации. Более подробную информацию см. в комментариях к процедурам.

Пример реализации см. в демонстрационной конфигурации в справочнике **\_ДемоОрганизации** для вида контактной информации **Юридический адрес**.

#### Заполнение и редактирование контактной информации одного объекта в форме другого объекта

Имеется возможность выводить контактную информацию объекта в отдельной форме или форме другого объекта. Такая возможность позволяет редактировать контактную информацию объекта без необходимости открытия формы владельца контактной информации.

Например, если необходимо для вывода к уже ранее размещенной контактной информации элемента справочника **Демо: Контакные лица партнеров** выводить контактную информацию реквизита **ФизическоеЛицо**ссылающегося на элемент справочника **Демо: Физические лица**, то следует сделать следующие:

* ● создать реквизит формы **ФизическоеЛицо** с типом **СправочникОбъект.\_ДемоФизическиеЛица**, содержащий объект владельца редактируемой контактной информацией;
* ● на форме создать группу или страницу с именем **ГруппаКонтактнаяИнформацияФизическоеЛицо**– контейнер, в который будут добавляться элементы формы для редактирования контактной информации физического лица;
* ● в обработчиках событий формы **ПриСозданииНаСервере**, **ПриЧтенииНаСервере** и **ПередЗаписьюНаСервере**для заполнения и сохранения контактной информации физического лица добавить аналогичные вызовы, где в качестве параметров указать реквизит формы **ФизическоеЛицо** и группу размещения контактной информации:

#Область ОбработчикиСобытийФормы

&НаСервере

Процедура ПриСозданииНаСервере(Отказ, СтандартнаяОбработка)

// СтандартныеПодсистемы.КонтактнаяИнформация

УправлениеКонтактнойИнформацией.ПриСозданииНаСервере(ЭтотОбъект, Объект,, ПоложениеЗаголовкаЭлементаФормы.Лево);

УправлениеКонтактнойИнформацией.ПриСозданииНаСервере(ЭтотОбъект, ФизическоеЛицо, "ГруппаКонтактнаяИнформацияФизическогоЛица", ПоложениеЗаголовкаЭлементаФормы.Лево);

// Конец СтандартныеПодсистемы.КонтактнаяИнформация

КонецПроцедуры

&НаСервере

Процедура ПриЧтенииНаСервере(ТекущийОбъект)

// СтандартныеПодсистемы.КонтактнаяИнформация

Если ЗначениеЗаполнено(ТекущийОбъект.ФизическоеЛицо) Тогда

ЗначениеВРеквизитФормы(ТекущийОбъект.ФизическоеЛицо.ПолучитьОбъект(), "ФизическоеЛицо");

КонецЕсли;

УправлениеКонтактнойИнформацией.ПриЧтенииНаСервере(ЭтотОбъект, ФизическоеЛицо, "ГруппаКонтактнаяИнформацияФизическогоЛица");

УправлениеКонтактнойИнформацией.ПриЧтенииНаСервере(ЭтотОбъект, Объект);

// Конец СтандартныеПодсистемы.КонтактнаяИнформация

КонецПроцедуры

&НаСервере

Процедура ПередЗаписьюНаСервере(Отказ, ТекущийОбъект, ПараметрыЗаписи)

// СтандартныеПодсистемы.КонтактнаяИнформация

УправлениеКонтактнойИнформацией.ПередЗаписьюНаСервере(ЭтотОбъект, ТекущийОбъект);

УправлениеКонтактнойИнформацией.ПередЗаписьюНаСервере(ЭтотОбъект, ФизическоеЛицоОбъект);

// Конец СтандартныеПодсистемы.КонтактнаяИнформация

КонецПроцедуры

&НаСервере

Процедура ПриЗаписиНаСервере(Отказ, ТекущийОбъект, ПараметрыЗаписи)

ФизическоеЛицоОбъект = РеквизитФормыВЗначение("ФизическоеЛицо");

ФизическоеЛицоОбъект.Записать();

ЗначениеВРеквизитФормы(ФизическоеЛицоОбъект, "ФизическоеЛицо");

КонецПроцедуры

#КонецОбласти

* ● добавить код блокирующий для изменения элемент справочника **Демо: Физические лица**, контактная информация которого выведна на форму:
* ● для этого в обработчики события **ПриОткрытии** подключаем обработчик проверяющий необходимость блокировки объекта физического лица:

&НаКлиенте

Процедура ПриОткрытии(Отказ)

ПодключитьОбработчикОжидания("ПроверитьНеобходимостьБлокировкиФизическогоЛица", 1, Ложь);

КонецПроцедуры

* ● и процедуру, вызываемую этим обработчиком для попытки блокировки элемента справочника физического лица, если информация на форме была изменена:

&НаКлиенте

Процедура ПроверитьНеобходимостьБлокировкиФизическогоЛица()

Если Модифицированность Тогда

Попытка

ЗаблокироватьДанныеДляРедактирования(ФизическоеЛицо.Ссылка, ФизическоеЛицо.ВерсияДанных, ФормаУникальныйИдентификатор);

ОтключитьОбработчикОжидания("ПроверитьНеобходимостьБлокировкиФизическогоЛица");

Исключение

ВызватьИсключение НСтр("ru = 'Данные контактого лица не могут быть записаны, т.к. личные данные физического лица недоступны для изменения.

|Возможно, эти данные физического лица редактируются другим пользователем.'");

Прочитать();

КонецПопытки;

КонецЕсли;

КонецПроцедуры

Пример использования см. в форме элемента справочника **\_ДемоКонтактныеЛицаПартнеров** демонстрационной конфигурации.

#### Добавление произвольных реквизитов с контактной информацией

Имеется возможность хранить значения полей контактной информации не только в стандартной табличной части **КонтактнаяИнформация**, но и в произвольных реквизитах произвольных объектов конфигурации. Такая возможность может быть востребована для добавления к объектам отдельных полей с контактной информацией.

Например, необходимо хранить данные адреса доставки в документе **ЗаказПокупателя**. При этом в форме документа должно быть поле **Адрес доставки**, редактируемое с помощью стандартной формы ввода адреса, и поля **Дата доставки**, **Метро**, **Комментарий**. При этом в заказе покупателя не требуются остальные сервисы подсистемы по заведению других видов контактной информации.

Для решения этой задачи необходимо:

1. Добавить к документу **ЗаказПокупателя** основные реквизиты:

* ● **АдресДоставки** типа «строка неограниченной длины». Этот реквизит будет хранить служебные значения контактной информации адреса доставки;
* ● **ДатаДоставки** типа **Дата** для хранения даты доставки;
* ● **Метро**. Ссылка на справочник **СтанцииМетро**.

2. Добавить к документу **ЗаказПокупателя** вспомогательные реквизиты, данные в которых будут вычисляться при записи документа по данным реквизита **АдресДоставки**. Эти реквизиты предназначены для оперативного поиска документов по данным контактной информации, вывода данных на печать и т. п.

* ● **АдресДоставкиСтрокой** – строка длиной 200 символов;
* ● **ГородАдресаДоставки** – строка длиной 100 символов.

3. Поместить в форму вспомогательный реквизит **ВидКонтактнойИнформацииАдресаДоставки** для контроля работы с адресом и инициализировать его при создании формы:

ВидКонтактнойИнформацииАдресаДоставки = Новый Структура;

ВидКонтактнойИнформацииАдресаДоставки.Вставить("Тип", Перечисления.ТипыКонтактнойИнформации.Адрес );

ВидКонтактнойИнформацииАдресаДоставки.Вставить("АдресТолькоРоссийский", Ложь);

ВидКонтактнойИнформацииАдресаДоставки.Вставить("ВключатьСтрануВПредставление", Ложь);

ВидКонтактнойИнформацииАдресаДоставки.Вставить("СкрыватьНеактуальныеАдреса", Ложь);

Эта структура заменяет собой соответствующий элемент справочника **ВидыКонтактнойИнформации**.

4. Поместить в форму документа вспомогательный строковый реквизит **ПредставлениеАдресаДоставки**. Инициализировать значение реквизита при создании формы:

ПредставлениеАдресаДоставки = УправлениеКонтактнойИнформацией.ПредставлениеКонтактнойИнформации(Объект.АдресДоставки);

* ● создать соответствующее вспомогательному реквизиту поле ввода, установить у него флажок**КнопкаВыбора** в **Истина**;
* ● определить обработчики событий для этого элемента:

&НаКлиенте

Процедура ПредставлениеАдресаДоставкиПриИзменении(Элемент)

Текст = Элемент.ТекстРедактирования;

Если ПустаяСтрока(Текст) Тогда

// Очистка данных.

// Cбрасываем как представления,

// так и внутренние значения полей

ПредставлениеАдресаДоставки = "";

КомментарийАдресаДоставки   = "";

Объект.АдресДоставки        = "";

Возврат;

КонецЕсли;

// Формируем внутренние значения полей по тексту

// и параметрам

ПредставлениеАдресаДоставки = Текст;

Объект.АдресДоставки = ЗначенияПолейКонтактнойИнформацииСервер(Текст,

ВидКонтактнойИнформацииАдресаДоставки, КомментарийАдресаДоставки);

КонецПроцедуры

&НаКлиенте

Процедура ПредставлениеАдресаДоставкиНачалоВыбора(Элемент, ДанныеВыбора, СтандартнаяОбработка)

// Если представление было изменено в поле и сразу нажата

// кнопка выбора, то необходимо привести данные в соответствие

// и сбросить внутренние поля для повторного репарсинга

Если Элемент.ТекстРедактирования<>ПредставлениеАдресаДоставки Тогда

ПредставлениеАдресаДоставки = Элемент.ТекстРедактирования;

Объект.АдресДоставки = "";

КонецЕсли;

// Данные для редактирования

ПараметрыОткрытия = Новый Структура;

ПараметрыОткрытия.Вставить("ВидКонтактнойИнформации", ВидКонтактнойИнформацииАдресаДоставки);

ПараметрыОткрытия.Вставить("ЗначенияПолей", Объект.АдресДоставки);

ПараметрыОткрытия.Вставить("Представление", ПредставлениеАдресаДоставки);

ПараметрыОткрытия.Вставить("Комментарий", КомментарийАдресаДоставки);

// Переопределямый заголовок формы,

// по умолчанию отобразятся данные по ВидКонтактнойИнформации

ПараметрыОткрытия.Вставить("Заголовок", НСтр("ru='Адрес доставки'"));

УправлениеКонтактнойИнформациейКлиент.ОткрытьФормуКонтактнойИнформации(ПараметрыОткрытия, Элемент );

КонецПроцедуры

&НаКлиенте

Процедура ПредставлениеАдресаДоставкиОчистка(Элемент, СтандартнаяОбработка)

// Сбрасываем как представления,

//  так и внутренние значения полей

ПредставлениеАдресаДоставки = "";

КомментарийАдресаДоставки   = "";

Объект.АдресДоставки        = "";

КонецПроцедуры

&НаКлиенте

Процедура ПредставлениеАдресаДоставкиОбработкаВыбора(Элемент, ВыбранноеЗначение, СтандартнаяОбработка)

СтандартнаяОбработка = Ложь;

Если ТипЗнч(ВыбранноеЗначение)<>Тип("Структура") Тогда

// Отказ от выбора, данные неизменны

Возврат;

КонецЕсли;

ПредставлениеАдресаДоставки = ВыбранноеЗначение.Представление;

КомментарийАдресаДоставки   = ВыбранноеЗначение.Комментарий;

Объект.АдресДоставки        = ВыбранноеЗначение.КонтактнаяИнформация;

КонецПроцедуры

Следует заметить, что если функционал ввода адреса строкой не требуется, то можно обойтись без этого вспомогательного поля ввода, заменив его, например, на гиперссылку. Текст гиперссылки при этом будет представлением адреса, а обработчик нажатия аналогичен обработчику **НачалоВыбора**.

5. Поместить в форму вспомогательный реквизит **КомментарийАдресаДоставки** для редактирования комментария, не попадающего в основное представление адреса. Инициализировать его при создании формы:

КомментарийАдресаДоставки =

УправлениеКонтактнойИнформацией.КомментарийКонтактнойИнформации(

Объект.АдресДоставки

);

6. Поместить в форму поле ввода и связать его с реквизитом **КомментарийАдресаДоставки**. Определить следующие обработчики изменений:

&НаКлиенте

Процедура КомментарийАдресаДоставкиПриИзменении(Элемент)

ЗаполнитьКомментарийАдресаДоставкиСервер();

КонецПроцедуры

// Устанавливаем новый комментарий для адреса доставки

//

&НаСервере

Процедура ЗаполнитьКомментарийАдресаДоставкиСервер()

Если ПустаяСтрока(Объект.АдресДоставки) Тогда

// Необходимо инициализировать данные

Объект.АдресДоставки = ЗначенияПолейКонтактнойИнформацииСервер(ПредставлениеАдресаДоставки,

ВидКонтактнойИнформацииАдресаДоставки, КомментарийАдресаДоставки);

Возврат;

КонецЕсли;

УправлениеКонтактнойИнформацией.КомментарийКонтактнойИнформации(Объект.АдресДоставки,

  КомментарийАдресаДоставки);

КонецПроцедуры

7. Поместить в форму документа поля ввода, связанные с реквизитами объекта **ДатаДоставки** и **Метро**.

8. В модуле объекта документа, в обработчике события **ПередЗаписью** заполнить вспомогательные реквизиты:

ГородДоставки = УправлениеКонтактнойИнформацией.ГородАдресаКонтактнойИнформации(АдресДоставки);

АдресДоставкиСтрокой = УправлениеКонтактнойИнформацией.ПредставлениеКонтактнойИнформации(АдресДоставки);

БСП обеспечивает программный интерфейс для оперирования данными:

* ● методы для разбора контактной информации из строки представления;
* ● методы для формирования представления контактной информации;
* ● методы для получения и установки значений частей контактной информации (например, комментария, страны адреса, города адреса и т. п.);
* ● методы для сравнения объектов контактной информации.

Пример реализации см. в демонстрационной конфигурации в реквизите **АдресДоставки** документа **\_ДемоЗаказПокупателя**.

#### Отключение неиспользуемых видов контактной информации

В случае если объект – владелец контактной информации отключен по функциональной опции, имеется возможность отключить неиспользуемый вид контактной информации. Для этого можно воспользоваться свойством **Используется** вида контактной информации. При установке значения свойства **Используется** в значение **Ложь**вид контактной информации перестает отображаться в форме списка, а также в форме объекта – владельца контактной информации. Устанавливать свойство **Используется** можно как для отдельных видов контактной информации, так и для всей контактной информации объекта владельца (группа видов контактной информации). Для отключения неиспользуемого вида контактной информации необходимо:

* ● Написать обработчик обновления, отключающий неиспользуемые виды контактной информации при переходе на новую версию. Пример кода обработчика:

КонтактныеЛицПартнеров = Справочники.ВидыКонтактнойИнформации.Справочник\_ДемоКонтактныеЛицаПартнеров.ПолучитьОбъект();

КонтактныеЛицПартнеров.Используется = ПолучитьФункциональнуюОпцию("\_ДемоИспользоватьКонтактныеЛицаПартнеров");

ОбновлениеИнформационнойБазы.ЗаписатьДанные(КонтактныеЛицПартнеров);

* ● Добавить аналогичный код в событие при изменении значения функциональной опции. См. пример в событии **ПриЗаписи** в модуле менеджера значения константы \_**ДемоИспользоватьКонтактныеЛицаПартнеров**.

#### Добавление реквизитов типа «СправочникСсылка.СтраныМира»

В общем случае при добавлении реквизита типа **СправочникСсылка.СтраныМира** можно выбирать любой элемент справочника. Но если список выбираемых стран мира должен быть регламентирован (т. е. разрешен выбор стран только из классификатора), то дополнительно следует в свойстве **ПараметрыВыбора** элемента формы (поля ввода) указать:

* ● **РазрешитьДанныеКлассификатора – Булево**, **Истина**;
* ● **ТолькоДанныеКлассификатора – Булево**, **Истина**.

Также есть возможность автоматически создавать новые элементы справочника **СтраныМира** при выборе автоподбором. Для этого необходимо в модуле формы в обработчике события **ОбработкаВыбора** вставить вызов:

УправлениеКонтактнойИнформациейКлиент.СтранаМираОбработкаВыбора(Элемент, ВыбранноеЗначение, СтандартнаяОбработка);

Пример использования см. в форме элемента справочника **\_ДемоНоменклатура** демонстрационной конфигурации.

#### Совместное внедрение с другими подсистемами

* ● [Внедрение подсистемы «Пользователи» без подсистемы «Контактная информация»](https://its.1c.ru/db/content/bsp23doc/src/3.59.%20%D0%BE%D1%81%D0%BE%D0%B1%D1%8B%D0%B5%20%D1%81%D0%BB%D1%83%D1%87%D0%B0%D0%B8%20%D0%B2%D0%BD%D0%B5%D0%B4%D1%80%D0%B5%D0%BD%D0%B8%D1%8F%20%D0%BF%D0%BE%D0%B4%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC.htm?_=1504708101#_внедрение_подсистемы_).

## Использование при разработке конфигурации

При разработке новых или доработке существующих предопределенных видов контактной информации необходимо предусмотреть их обновление при помощи обработчика обновления.

### Автоматическое добавление элементов в справочник «СтраныМира» из классификатора стран мира (ОКСМ)

По умолчанию справочник **СтраныМира** содержит только один предопределенный элемент **Россия**. Пользователи с соответствующими правами могут добавлять в справочник новые элементы – как вручную, так и с помощью подбора из классификатора стран мира. При этом новые элементы справочника могут быть автоматически созданы по данным классификатора в процессе выбора и непосредственно при автоподборе в поле ввода страны.

Таким образом, в справочнике **СтраныМира** содержатся только те страны, которые используются в учете.

### Настройка обмена данными

Для настройки обмена данными следует руководствоваться общими правилами. В план обмена необходимо включить все объекты метаданных, содержащие данные, кроме регистра сведений **УдалитьСтраныМира**.

При включении справочника **СтраныМира** в план обмена между различными прикладными решениями с использованием правил конвертации необходимо настроить в правиле конвертации объекта (ПКО) поиск по ссылке, а затем по полям поиска – составному ключу **Код** и **Наименование**.

#### Настройка обмена данными с предыдущими версиями

* ● При синхронизации контактной информации с конфигурациями на базе БСП версии 2.1.2 и меньше необходимо при выгрузке в эту конфигурацию производить конвертацию реквизита **ЗначенияПолей** табличной части **КонтактнаяИнформация** при помощи вызова:

Значение = УправлениеКонтактнойИнформацией.ПредыдущийФорматКонтактнойИнформацииXML(Значение, Истина)

При использовании правил конвертации для этого необходимо вписать приведенный выше код в обработчик **ПриВыгрузке** правила конвертации свойства (ПКС) для реквизита **ЗначенияПолей** табличной части **КонтактнаяИнформация**.

* ● При синхронизации контактной информации с конфигурациями на базе БСП редакции 2.2 и меньше, необходимо в правилах конвертации настроить правила выгрузки из БСП 2.3.1 в БСП 2.2 таким образом, чтобы для объектов – владельцев контактной информации исключать выгрузку:
* ● реквизита **ВидДляСписка** табличной части **КонтактнаяИнформация**;
* ● всех строк табличной части **КонтактнаяИнформация** с типом контактной информации **Skype**.
* ● Если табличная часть **КонтактнаяИнформация** содержит реквизит **ДействуетС**, то при синхронизации контактной информации с конфигурациями на базе БСП версии 2.3.1 и меньше для видов контактной информации с установленным свойством **ХранитьИсториюИзменений** необходимо настроить правила обмена следующим образом:
* ● при выгрузке данных – если у вида контактной информации существует несколько записей с разными датами в реквизите **ДействуетС**, то в выгрузку включается только актуальная (действующая) запись. Все исторические записи игнорируются и в выгрузку не попадают;
* ● при загрузке данных – все записи этого вида контактной информации сохраняются без изменений. Но в случае, когдазагружаемая запись отличается от данных текущей (действующей) записи этого вида контактной информации, в табличную часть **КонтактнаяИнформация** добавляется новая строка с загружаемыми данными, а реквизиту **ДействуетС**новой строки контактной информации присваивается текущая дата сеанса, т. к. эта запись становится актуальной(действующей).

Пример использования см. в демонстрационной конфигурации, в макетах **ПравилаОбмена** и **ПравилаОбменаКорреспондента** плана обмена \_**ДемоОбменСБиблиотекойСтандартныхПодсистем225**.

# 3.25. Напоминания пользователя

Подсистема «Напоминания пользователя» предназначена для установки персональных напоминаний по поводу какого-либо объекта системы и оповещения пользователя в назначенное время.

## Настройка

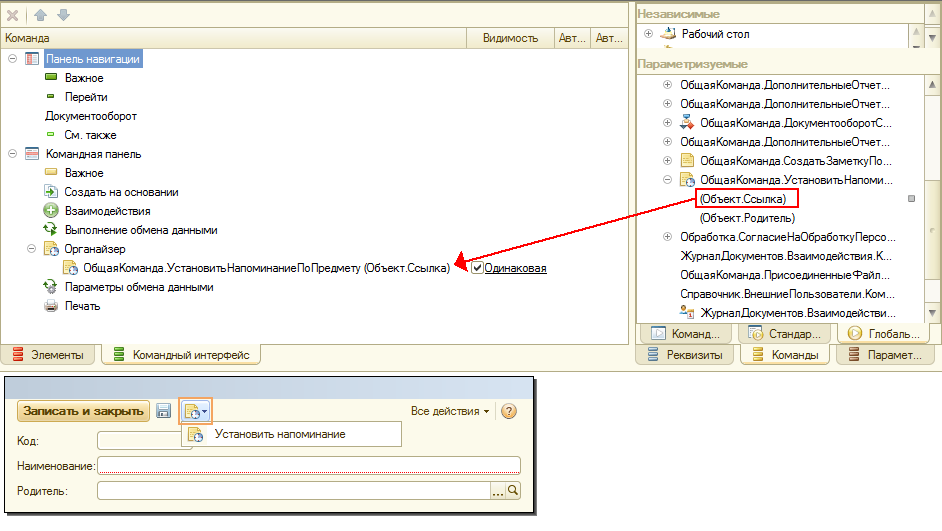
Для использования подсистемы необходимо разместить в командном интерфейсе пользователя форму **МоиНапоминания** регистра сведений **НапоминанияПользователя**, а также выполнить следующие действия:

1. Определить состав объектов ссылочного типа, по поводу которых пользователь должен иметь возможность вводить напоминания. Как правило, это большинство объектов конфигурации, с которыми работает пользователь.

2. В определяемых типах **ПредметНапоминания** и **ПредметНапоминанияОбъект** отметить выбранные типы объектов в свойстве **Тип**.

**Примечание**

По умолчанию общая команда **Напомнить** не видна в формах групп объектов. При необходимости эту команду нужно поместить на командную панель форм группы вручную:



### Определение состава реквизитов объекта для относительной настройки срока напоминания

По умолчанию пользователю предлагается задать срок напоминания относительно любого реквизита типа **Дата**. В списке доступных для выбора реквизитов доступны только реквизиты, значения дат которых находятся в будущем. Этот список можно переопределить – например, убрать из него служебные реквизиты. Для этого необходимо вписать свою реализацию в процедуру **ПриЗаполненииСпискаРеквизитовИсточникаСДатамиДляНапоминания** модуля **НапоминанияПользователяПереопределяемый**.

### Настройка прав доступа пользователей

Таблица 3.48.

| **№** | **Роли и их назначение** |
| --- | --- |
| 1. | **ПолныеПрава** (из подсистемы «Базовая функциональность»)  Создание, редактирование напоминаний пользователей, удаление напоминаний |
| 2. | **ДобавлениеИзменениеНапоминаний**  Создание, просмотр и удаление своих напоминаний |

## Использование при разработке конфигурации

### Программный интерфейс для создания напоминаний

Для неинтерактивного создания напоминаний предусмотрен программный интерфейс: процедуры **НапомнитьВУказанноеВремя** и **НапомнитьДоВремениПредмета** общего модуля **НапоминанияПользователяКлиент**. Пример его использования можно посмотреть в демонстрационной конфигурации:

● в разделе **Бизнес процессы и задачи – Исполнение задач – Мои задачи** – в любой задаче на командной панели – **Органайзер – Демо: Напомнить за 5 минут**;

● в разделе **Органайзер – Заметки пользователя – Демо: Контрагенты** – в карточке контрагента на командной панели – **Органайзер – Демо: Напомнить через 10 минут**.

Общая команда **\_ДемоНапомнитьЗа5Минут**, справочник **\_ДемоКонтрагенты**, команда **\_ДемоНапомнитьЧерез10Минут**.

### Настройка обмена данными

Все объекты метаданных подсистемы, содержащие данные, рекомендуется включать в планы обмена распределенной ИБ (РИБ).

# 3.26. Настройка порядка элементов

Подсистема «Настройка порядка элементов» предназначена для изменения порядка следования элементов справочника, плана видов характеристик или плана видов расчета в динамических списках. Если справочник подчиненный, то нумерация элементов выполняется в пределах владельца. Если справочник или ПВХ иерархический, то нумерация элементов выполняется в пределах родителя.

Подсистема накладывает ограничение на использование отборов в динамических списках: в качестве отборов можно использовать только поля основной таблицы списка. В противном случае в момент перемещения элемента будет вызвано исключение.

**Важно!**

Перемещение элементов вверх/вниз в динамическом списке выполняется без учета пользовательского отбора, установленного стандартной командой **Найти…**

## Настройка

Определить состав справочников и планов видов характеристик, для которых требуется настраивать порядок элементов, указав их в составе определяемого типа **ОбъектСНастраиваемымПорядком**.

Для каждого объекта, для которого предусмотрена настройка порядка элементов:

1. Добавить реквизит **РеквизитДопУпорядочивания**.

Таблица 3.49.

| **Свойство** | **Значение** |
| --- | --- |
| Тип значения | **Число**, длина – 5, точность – 0 (длина может быть изменена) |
| Индексировать | **Индексировать с доп. упорядочиванием** |
| Синоним | **Порядок** |

2. В модуль формы списка добавить фрагмент кода:

////////////////////////////////////////////////////////////////////////////////

// ПРОЦЕДУРЫ ПОДСИСТЕМЫ НАСТРОЙКИ ПОРЯДКА ЭЛЕМЕНТОВ

&НаКлиенте

Процедура ПереместитьЭлементВверх(Команда)

НастройкаПорядкаЭлементовКлиент.ПереместитьЭлементВверхВыполнить(Список, Элементы.Список);

КонецПроцедуры

&НаКлиенте

Процедура ПереместитьЭлементВниз(Команда)

НастройкаПорядкаЭлементовКлиент.ПереместитьЭлементВнизВыполнить(Список, Элементы.Список);

КонецПроцедуры

3. В форме списка создать две команды и добавить их на командную панель списка:

* ● **ПереместитьЭлементВверх**, картинка – **ПереместитьВверх**, обработчик – **ПереместитьЭлементВверх**;
* ● **ПереместитьЭлементВниз**, картинка – **ПереместитьВниз**, обработчик – **ПереместитьЭлементВниз**.

4. В форме списка для динамического списка установить сортировку по полю **Порядок**.

5. В форме списка для полей **Порядок** и **Ссылка** динамического списка установить свойство **Использовать всегда**.

Если у объекта есть предопределенные элементы, то в процедуре обновления информационной базы необходимо перезаписать эти элементы в том порядке, в каком они должны отображаться в списке.

**Важно!**

В списках, у которых есть отборы, а также для объектов с включенным ограничением прав на уровне записей подсистема может использоваться в ограниченном количестве случаев. Например, если каждый пользователь списка видит только свои элементы, то перемещение любого элемента будет всегда корректным. В остальных случаях возможна ситуация, когда один пользователь двигает элемент в своей области видимости на одну позицию, а для другого пользователя это «движение» выглядит, как «прыжок» элемента сразу на несколько позиций.

### Настройка прав доступа пользователей

Настройка прав доступа пользователей к данным подсистемы «Настройка порядка элементов» не требуется.

## Использование при разработке конфигурации

### Настройка обмена данными

Для настройки обмена данными никаких действий не требуется, так как подсистема не предоставляет собственных данных.

# 3.27. Настройки программы

Подсистема «Настройки программы» предоставляет панели настроек для всех подсистем библиотеки, размещаемые в разделе командного интерфейса **Администрирование**. Панели настроек представлены формами обработки **ПанельАдминистрированияБСП**.

## Настройка

Расположить подсистему **Администрирование** последней (крайней права) в списке разделов командного интерфейса (свойства конфигурации – **Командный интерфейс**).

Если состав раздела **Администрирование** для объектов БСП не был загружен из поставки БСП (см. таблицы 3 и 4), то в разделе **Администрирование** следует разместить все панели администрирования (формы обработки **ПанельАдминистрированияБСП**), а также частотные команды, которые должны быть на виду (например, удаление помеченных объектов). Пример размещения см. в демонстрационной базе.

Если в конфигурацию не встроена ни одна подсистема, входящая в форму (в этом случае форма панели администрирования открывается пустой), то рекомендуется отключить видимость команды этой панели. Например, если не внедрена подсистема **СинхронизацияДанных**, то необходимо отключить в разделе **Администрирование** видимость команды **НастройкиСинхронизацииДанных**.

Необходимо принять решение по поводу состава настроек конфигурации и места их размещения на панелях. При выборе панели настроек нужно руководствоваться следующими соображениями:

● форма **ПоддержкаИОбслуживание** предназначена для операций, которые периодически должен выполнять администратор программы:

* ● мониторинг состояния системы;
* ● резервное копирование и восстановление;
* ● оптимизация быстродействия;
* ● другие операции;

● форма **ОбщиеНастройки** предназначена для изменения общих для программы параметров работы (например, заголовка программы, полнотекстового поиска и т. п.);

● форма **НастройкиПользователейИПрав** предназначена для администрирования пользователей:

* ● ведение списка пользователей;
* ● настройка подсистемы прав доступа;
* ● управление настройками пользователей;

● форма **ПоискИУстановкаОбновлений** предназначена для обновления версии программы и просмотра информации о предыдущих обновлениях;

● форма **НастройкиРаботыСФайлами** предназначена для настройки параметров подсистем, работающих с файлами;

● форма **НастройкиСинхронизацииДанных** предназначена для настройки синхронизации данных с другими программами и организации совместной работы в распределенной информационной базе;

● форма **Органайзер** предназначена для настройки электронной почты, заметок, напоминаний и бизнес-процессов;

● форма **ПечатныеФормыОтчетыИОбработки** предназначена для настройки печатных форм, вариантов отчетов, рассылок отчетов и дополнительных отчетов и обработок.

На панелях настроек могут быть размещены поля редактирования констант, гиперссылки для перехода к связанным формам настроек, поясняющие надписи и другие элементы управления.

### Размещение констант

В формах настроек предусмотрен код, позволяющий упростить процесс размещения констант. Для размещения константы необходимо:

1. Перетащить константу из реквизита формы **НаборКонстант** в элементы формы. Полю ввода должно быть назначено то же имя, которое имеет константа.

2. Определить обработчик поля ввода **ПриИзменении**, разместив в нем вызов процедуры **Подключаемый\_ПриИзмененииРеквизита**, например:

&НаКлиенте

Процедура ИмяКонстантыПриИзменении(Элемент)

Подключаемый\_ПриИзмененииРеквизита(Элемент);

КонецПроцедуры

### Размещение гиперссылок

Для размещения гиперссылки необходимо:

1. Включить глобальную команду или объект с командами в состав раздела **Администрирование**.

2. Выключить видимость команды (объекта) в этом разделе для того, чтобы системный администратор имел возможность включить ее обратно для часто используемых команд.

3. В форме настройки перетащить команду в элементы формы. Установить вид поля **Гиперссылка**.

### Размещение поясняющих надписей

Для всех полей ввода и гиперссылок рекомендуется добавлять расширенные подсказки, поясняющие назначение элемента управления. При добавлении расширенной подсказки необходимо установить у владельца свойство **Отображение подсказки** в значение **Отображать снизу**, а также отключить у расширенной подсказки свойство **АвтомаксимальнаяШирина**.

### Поддержка различных режимов работы конфигурации

Если конфигурация рассчитана на различные режимы работы, следует управлять видимостью элементов управления на формах настроек. Например, гиперссылка для перехода к форме **Параметры администрирования ИБ** должна быть видна только в клиент-серверном режиме работы.

Элементы формы, связанные с функциональными опциями, скрываются автоматически. Если элементы формы не скрываются автоматически (например, если это поясняющий текст, который не может быть включен в состав функциональной опции), то в обработчике **ПриСозданииНаСервере** следует написать код, который будет скрывать всю группу.

Для определения режима работы конфигурации в поставляемых формах настроек для реквизита формы **РежимРаботы** можно использовать свойства: **МодельСервиса**, **Локальный**, **Файловый**, **КлиентСерверный**, **ЛокальныйФайловый**, **ЛокальныйКлиентСерверный** (подробнее – см. описание процедуры **ПанельНастроекЗаполнитьРежимРаботы** общего модуля **НастройкиПрограммыСервер**).

Например:

&НаСервере

Процедура ПриСозданииНаСервере(Отказ, СтандартнаяОбработка)

// Настройки видимости при запуске

Элементы.ГруппаПараметрыАдминистрированияИБ.Видимость = РежимРаботы.КлиентСерверный;

КонецПроцедуры

### Взаимосвязанные элементы управления

Некоторые элементы могут быть связаны друг с другом, например:

● гиперссылка для перехода к списку **Настройки версионирования объектов** должна быть доступна, только когда установлен флажок **Версионирование объектов**;

● флажок **Подчиненные Бизнес-процессы** должен быть доступен только тогда, когда установлен флажок **Бизнес-процессы и задачи**.

В таких ситуациях подчиненные элементы следует делать недоступными (но оставлять видимыми) в зависимости от главного элемента. Для этого нужно поместить обслуживающий код в процедуре **УстановитьДоступность**. Например:

&НаСервере

Процедура УстановитьДоступность(РеквизитПутьКДанным = "")

Если РеквизитПутьКДанным = "НаборКонстант.ИспользоватьБизнесПроцессыИЗадачи" ИЛИ РеквизитПутьКДанным = "" Тогда

Элементы.ОткрытьРолиИИсполнителиБизнесПроцессов.Доступность = НаборКонстант.ИспользоватьБизнесПроцессыИЗадачи;

КонецЕсли;

КонецПроцедуры

### Ролевая видимость

В некоторых сценариях с одной формой настроек могут работать пользователи с различными правами доступа. Например, в модели сервиса настройку программы могут выполнять и администратор абонента, и администратор сервиса.

В таких случаях формы настроек должны «разрезаться» по правам доступа таким образом, чтобы пользователю (администратору) были видны только те настройки, к которым он имеет доступ.

Для этого необходимо добавить проверку ролей перед вызовом кода, который изменяет свойства элементов из скрываемой группы в процедурах **ПриСозданииНаСервере** или **Доступность** в процедуре **УстановитьДоступность**. Например, для проверки того, что с формой работает администратор сервиса:

&НаСервере

Процедура УстановитьДоступность(РеквизитПутьКДанным = "")

Если РежимРаботы.ЭтоАдминистраторСистемы Тогда

<Обслуживающий код>

КонецЕсли;

КонецПроцедуры

### Настройка прав доступа пользователей

Для настройки прав доступа пользователей следует использовать роли, приведенные ниже.

Таблица 3.50.

| **№** | **Роли и их назначение** |
| --- | --- |
| 1. | **ПолныеПрава** (из подсистемы «Базовая функциональность»)  Доступ ко всем прикладным настройкам программы и настройкам, специфичным только в локальном режиме работы |
| 2. | **АдминистраторСистемы** (из подсистемы «Базовая функциональность»)  Доступ ко всем общим настройкам программы при работе в модели сервиса, а также ко всем административным настройкам в локальном режиме |

### Настройка обмена данными

Для настройки обмена данными никаких действий не требуется, так как подсистема не предоставляет собственных данных.

# 3.28. Обмен данными

Подсистема «Обмен данными» объединяет функциональность, связанную с обменом информацией между различными информационными базами:

* ● распределенные информационные базы (РИБ);
* ● обмен данными через универсальный формат;
* ● обмен данными по правилам обмена (правила обмена создаются при помощи конфигурации «Конвертация данных», редакция 2.1);
* ● обмен данными без правил обмена.

Поддерживается обмен данными между конфигурациями, работающими в модели сервиса, а также между конфигурациями, работающими в модели сервиса и в локальном режиме. Обмен данными может быть односторонним или двусторонним и выполняться автоматически (по некоторому расписанию) или вручную по требованию пользователя. Для транспорта сообщений обмена могут быть использованы различные каналы связи: сетевой каталог, электронная почта, FTP, обмен через Интернет (веб-сервис). При использовании правил обмена доступно выполнение обмена через внешнее соединение к информационной базе-корреспонденту.

Подсистема «Обмен данными» не является самостоятельно функционирующей подсистемой. Для ее использования необходимо разработать в конфигурации прикладные планы обмена согласно предлагаемой методике и провести дополнительную настройку библиотечных объектов.

Если требуется выполнять обмен данными в модели сервиса, то дополнительно необходимо встроить подсистему «Обмен данными в модели сервиса».

Подсистема «Обмен данными» совместно с подсистемой «Обмен данными в модели сервиса» позволяет поддержать пять сценариев обмена данными:

1. Обмен данными между различающимися конфигурациями, работающими в локальном режиме.

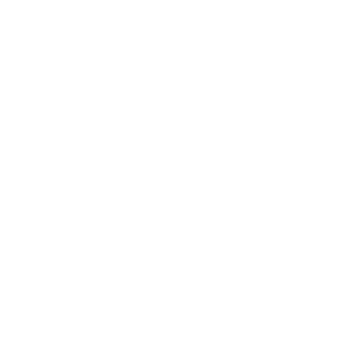
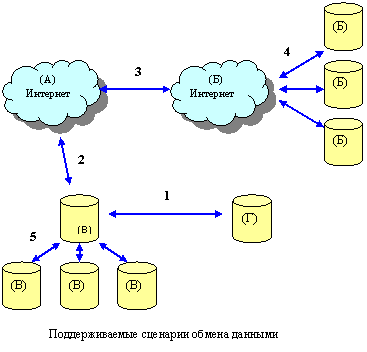
2. Обмен данными между различающимися конфигурациями, одна из которых работает в модели сервиса, а вторая – в локальном режиме.

3. Обмен данными между различающимися конфигурациями, работающими в модели сервиса.

4. Автономная работа в модели сервиса.

5. Обмен в распределенной информационной базе.

Поддерживаемые сценарии обмена данными наглядно представлены на рисунке ниже.



Подсистема предоставляет в распоряжение разработчика четыре вида обмена данными:

* ● обмен данными в распределенной информационной базе (**РИБ**) – используется стандартный механизм платформы для организации обмена в распределенной информационной базе;
* ● обмен данными через универсальный формат (**ОУФ**) – позволяет организовать обмен между различающимися конфигурациями с использованием модуля (**менеджер обмена**), в котором прописана логика приведения данных конфигурации к структуре, описанной в формате (**конфигурация – формат**), а также обратный процесс преобразования (**формат – конфигурация**);
* ● универсальный обмен данными по правилам (**УОП**) – позволяет организовать обмен между различающимися конфигурациями с использованием правил обмена данными, в которых прописана логика преобразования данных одной конфигурации в данные другой конфигурации;
* ● универсальный обмен данными без использования правил обмена (**УО**) – позволяет организовать обмен между различающимися конфигурациями. При выборе данного вида обмена следует учитывать основное требование: необходимо, чтобы объекты, которые участвуют в обмене, в обеих базах имели идентичную структуру метаданных.

## Настройка

### Принятие решения о виде обмена данными

Перед внедрением подсистемы следует принять решение о виде обмена данными, который необходимо реализовать. Возможности и особенности поддерживаемых видов обмена приведены в таблице ниже.

Таблица 3.51.

| **Характеристика** | **ОУФ** | **УОП** | **УО** | **РИБ** |
| --- | --- | --- | --- | --- |
| **Режим использования** | | | | |
| Работа в локальном режиме | + | + | + | + |
| Работа в модели сервиса | + | + | + | + |
| **Поддерживаемые каналы связи** | | | | |
| Через сетевой или локальный каталог | + | + | + | + |
| Через **ftp-сервер** | + | + | + | + |
| Через электронную почту (**e-mail**) | + | + | + | + |
| Обмен через Интернет (**web-сервис**) | + | + | + | + |
| Обмен через прямое подключение к информационной базе-корреспонденту (**COM-соединение**) | – | + | – | – |
| **Особенности взаимодействия конфигураций** | | | | |
| Существует разделение баз на главную и подчиненную | – | – | – | + |
| Конфигурации обменивающихся информационных баз идентичны как по структуре метаданных, так и по прикладной бизнес-логике | – | – | –\* | + |
| Требуется разработка логики преобразования данных | + | + | – | – |
| Логика конвертации данных на стороне конфигурации № 1 не зависит от внутреннего устройства и бизнес-логики конфигурации № 2 | + | – | – | – |
| **Прочие возможности** | | | | |
| Настройка ограничения и направления миграции данных по узлам | + | + | + | + |
| Предварительный просмотр загружаемых данных и сопоставление объектов информационных баз | + | + | – | – |

**Примечание**

\* Конфигурации обменивающихся информационных баз должны быть идентичны только в части структуры метаданных данных, которые участвуют в обмене.

В зависимости от вида обмена технологии внедрения и настройки подсистемы в конфигурации различаются.

### Настройка для работы в локальном режиме

Для использования подсистемы в конфигурации необходимо:

* ● разместить в командном интерфейсе общую команду **СинхронизацияДанных**;
* ● в форме редактирования настроек системы поместить поле, связанное с константами **ИспользоватьСинхронизациюДанных**, **ПрефиксУзлаРаспределеннойИнформационнойБазы**;
* ● выполнить публикацию веб-сервиса **Exchange** (если предполагается использование обмена через Интернет).

Пример размещения объектов подсистемы в командном интерфейсе можно посмотреть в демонстрационной конфигурации.

В общем модуле **ОбменДаннымиПереопределяемый** необходимо задать возвращаемое значение функции **ПрефиксИнформационнойБазыПоУмолчанию**. В качестве возвращаемого значения функции следует указать строку префикса – **Строка, 2**. Длина префикса не должна превышать двух символов. Значение функции будет использоваться для задания префикса информационной базы по умолчанию в помощнике настройки обмена данными.

### Настройка для работы в модели сервиса

* ● Разместить в командном интерфейсе общую команду **АвтономнаяРабота**;
* ● Разместить в командном интерфейсе общую команду **АвтономнаяРаботаВМоделиСервиса**;
* ● Разместить в командном интерфейсе общую команду **СинхронизацияДанныхВМоделиСервиса**;
* ● Выполнить внутреннюю неразделенную публикацию веб-сервисов **RemoteAdministrationOfExchange**, **RemoteAdministrationOfExchange\_2\_0\_1\_6**;
* ● Выполнить внутреннюю неразделенную и внешнюю разделенную публикацию веб-сервисов **Exchange**, **Exchange\_2\_0\_1\_6 и InterfaceVersion**.

Пример файла разделенной публикации веб-сервисов **Exchange**, **Exchange\_2\_0\_1\_6 и InterfaceVersion**:

<?xml version="1.0" encoding="UTF-8"?>

<point xmlns="http://v8.1c.ru/8.2/virtual-resource-system"

xmlns:xs="http://www.w3.org/2001/XMLSchema"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

base="/demo"

ib="Srvr=&quot;server1C&quot;;Ref=&quot;demo&quot;;">

<zones>

<zone specify="false" safe="true"/>

<zone specify="true" safe="true"/>

</zones>

<ws>

<point name="Exchange"

alias="exchange.1cws"/>

<point name="Exchange\_2\_0\_1\_6"

alias="exchange\_2\_0\_1\_6.1cws"/>

<point name="InterfaceVersion"

alias="InterfaceVersion.1cws"/>

</ws>

</point>

Пример файла неразделенной публикации веб-сервисов **RemoteAdministrationOfExchange**, **RemoteAdministrationOfExchange\_2\_0\_1\_6**, **Exchange**, **Exchange\_2\_0\_1\_6**:

<?xml version="1.0" encoding="UTF-8"?>

<point xmlns="http://v8.1c.ru/8.2/virtual-resource-system"

xmlns:xs="http://www.w3.org/2001/XMLSchema"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

base="/demo\_ws"

ib="Srvr=&quot;server1C&quot;;Ref=&quot;demo&quot;;"

enable="false">

<ws>

<point name="Exchange"

alias="exchange.1cws"/>

<point name="Exchange\_2\_0\_1\_6"

alias="exchange\_2\_0\_1\_6.1cws"/>

<point name="RemoteAdministrationOfExchange"

alias="RemoteAdministrationOfExchange.1cws"/>

<point name="RemoteAdministrationOfExchange\_2\_0\_1\_6"

alias="RemoteAdministrationOfExchange\_2\_0\_1\_6.1cws"/>

</ws>

</point>

### Автономная работа в модели сервиса

Одним из возможных сценариев использования подсистемы обмена данными является сценарий автономной работы в модели сервиса. Этот сценарий позволяет работать с информационной базой без постоянного подключения к Интернету и при необходимости выполнять синхронизацию данных с Интернетом.

Для организации автономной работы в конфигурации необходимо:

* ● создать план обмена РИБ согласно этой документации;
* ● план обмена должен быть разделенным (входить в состав реквизита-разделителя конфигурации **ОбластьДанныхОсновныеДанные**);
* ● для плана обмена должен быть установлен признак **Распределенная ИБ**;
* ● тип кода плана обмена: **Строка**, 36, переменная;
* ● в модуле менеджера плана обмена функция **ПланОбменаИспользуетсяВМоделиСервиса** должна возвращать значение **Истина**.

**Важно!**

В конфигурации может быть создан только один план обмена, который предназначен для обслуживания автономной работы в модели сервиса.

В состав плана обмена автономной работы не рекомендуется включать объекты метаданных классификаторов с потенциально большим объемом данных (например, адресный классификатор), т. к. обновление таких классификаторов требует много времени на синхронизацию и порождает большой трафик данных. Данные таких классификаторов рекомендуется загружать в автономное рабочее место из альтернативных источников, например с сайта «1С», и только при необходимости.

Обмен данными с автономным рабочим местом имеет следующие особенности:

* ● информация о пользователях и доступе к данным мигрирует только сверху вниз – из сервиса в автономное рабочее место. Все изменения, сделанные в автономном рабочем месте, игнорируются и приводятся в соответствие со значениями в сервисе;
* ● информация общих данных (классификаторов банков, классификаторов валют и пр.) мигрирует только сверху вниз – из сервиса в автономное рабочее место. Все изменения, сделанные в автономном рабочем месте, игнорируются и приводятся в соответствие со значениями в сервисе.

### Разработка планов обмена

Для организации обмена в локальном режиме и обмена в модели сервиса используется одна технология. Поэтому разработка планов обмена и других объектов метаданных будет одинаковой для двух режимов использования. Особенности и различия будут отражены отдельно.

План обмена может содержать реквизиты и табличные части. Состав плана обмена определяет набор объектов метаданных, которые будут участвовать в обмене данными.

В общем модуле **ОбменДаннымиПереопределяемый** в тело процедуры **ПолучитьПланыОбмена** следует добавить код, относящийся к новому плану обмена:

Процедура ПолучитьПланыОбмена(ПланыОбменаПодсистемы) Экспорт

ПланыОбменаПодсистемы.Добавить(Метаданные.ПланыОбмена.<ИмяПланаОбмена>);

КонецПроцедуры

### Реквизиты и свойства плана обмена

Наименование для плана обмена необходимо назначать согласно общим правилам наименования объектов метаданных. Свойства плана обмена необходимо задать согласно приведенной ниже таблице.

Таблица 3.52.

| **Свойство** | **План обмена ОУФ, УОП и УО** | **План обмена РИБ** |
| --- | --- | --- |
| **Имя** | **Обмен<Источник><Приемник>** | Не регламентируется |
| **Распределенная ИБ** | **Ложь** | **Истина** |
| **Состав** | См. описание ниже | См. описание ниже |
| **Код** (тип, длина) | **Строка**, 9, переменная | **Строка**, 9, переменная  (**Строка**, 36, переменная)\* |
| **Наименование** | **Строка**, 150 | **Строка**, 150 |
| **Макеты** | См. описание ниже | См. описание ниже |

\* Если план обмена используется для организации автономной работы в модели сервиса.

Для плана обмена допустимо создавать произвольное количество реквизитов и табличных частей. Все реквизиты и табличные части плана обмена можно использовать в правилах регистрации объектов на узлах. Правила регистрации настраиваются при помощи конфигурации «Конвертация данных». Правила регистрации используются как для универсальных обменов, так и для обмена в РИБ.

Для планов обмена может быть предусмотрен набор реквизитов для переключения режимов выгрузки объектов метаданных (реквизиты-переключатели). Например, для переключения режима выгрузки информации, связанной с контрагентами, создается реквизит-переключатель, который управляет режимом выгрузки сразу для нескольких объектов метаданных: справочники **Контрагенты**, **Договоры контрагентов**. Количество таких реквизитов-переключателей не ограничено. Тип реквизита-переключателя строго регламентирован – **ПеречислениеСсылка.РежимыВыгрузкиОбъектовОбмена**.

Если план обмена используется для организации обменов данными в модели сервиса и является разделенным, то для такого плана обмена следует создать реквизит **РегистрироватьИзменения**:

Таблица 3.53.

| **Имя** | **Тип** | **Индексировать** |
| --- | --- | --- |
| **РегистрироватьИзменения** | **Булево** | Да |

### Состав плана обмена

Для планов обмена **УОП** рекомендуется исключить все объекты метаданных подсистемы, кроме регистра сведений **СоответствияОбъектовИнформационныхБаз**.

**Важно!**

Все элементы состава плана обмена должны иметь признак авторегистрации **Запретить** вне зависимости от наличия правил регистрации для объектов.

Для остальных планов обмена рекомендуется исключить все объекты метаданных подсистемы.

Если для объекта метаданных требуется ограничить миграцию элементов при обмене, то для этого объекта необходимо создать правила регистрации при помощи конфигурации «Конвертация данных». Если правило регистрации для объекта метаданных не создано, то считается, что миграция объектов такого типа не ограничена, даже если признак авторегистрации установлен в значение **Запретить**.

Рекомендации по включению объектов других подсистем библиотеки в состав планов обмена приведены в разделах «Использование при разработке конфигурации», «Настройка обмена данными» соответствующих подсистем.

### Режим совместимости для планов обмена УОП

Этот режим предназначен для поддержания обратной совместимости обмена данными между конфигурациями на базе БСП версии 2.0 и ниже – с одной стороны и конфигурациями на базе БСП 2.1 и выше – с другой.

Для конфигураций, рассчитанных только на локальный режим работы, необходимо принять решение по поводу включения режима совместимости.

Для конфигураций, рассчитанных на работу в модели сервиса, режим совместимости отключается принудительно (его нельзя включить).

По умолчанию для новых правил конвертации режим совместимости с конфигурациями на базе БСП версии 2.0 и более ранних отключен. При необходимости поддержать обмен данными с такими конфигурациями следует установить режим совместимости в значение «Версия БСП 2.0». Режим совместимости устанавливается в конфигурации «Конвертация данных» версии 2.1.6 и выше в свойствах конвертации на закладке **Дополнительно**. Значение сохраняется в правилах обмена.

Для существующих правил конвертации по умолчанию режим совместимости установлен в значение «Версия БСП 2.0».

При включенном режиме совместимости становится недоступным [режим отладки](https://its.1c.ru/db/content/bsp23doc/src/3.28.%20%D0%BE%D0%B1%D0%BC%D0%B5%D0%BD%20%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D0%BC%D0%B8.htm?_=1504708101#_режим_отладки), а также [режим безопасного выполнения кода обработчиков](https://its.1c.ru/db/content/bsp23doc/src/3.28.%20%D0%BE%D0%B1%D0%BC%D0%B5%D0%BD%20%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D0%BC%D0%B8.htm?_=1504708101#_безопасное_выполнение_кода).

В таблице приведены варианты использования режимов обмена данными.

Таблица 3.54.

| **Режим совместимости** | **Режим отладки** | **Выполнение кода обработчиков** |
| --- | --- | --- |
| Отключен\* | Отключен\* | Код обработчиков выгрузки и загрузки выполняется из обработок в составе конфигурации |
| Включен | Код обработчиков выгрузки и загрузки выполняется из внешних обработок |
| Включен | Недоступен  (Отключен) | Код обработчиков выполняется:  ● при загрузке – из файла сообщения обмена,  ● при выгрузке – из правил обмена |

\* Режимы совместимости и отладки всегда отключены в модели сервиса.

### Безопасное выполнение кода обработчиков для планов обмена УОП

При отключенных режимах совместимости и [отладки](https://its.1c.ru/db/content/bsp23doc/src/3.28.%20%D0%BE%D0%B1%D0%BC%D0%B5%D0%BD%20%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D0%BC%D0%B8.htm?_=1504708101#_режим_отладки) код обработчиков выгрузки и загрузки выполняется из обработок в составе конфигурации. Это обеспечивает высокий уровень безопасности обмена данными, прежде всего в модели сервиса.

При включенном режиме совместимости код обработчиков загрузки зачитывается и выполняется из сообщения обмена данными, что является небезопасным.

При отключенном режиме совместимости необходимо:

* ● в инструменте «Конвертация данных» версии 2.1.6 и выше:
* ● отключить режим совместимости в свойствах конвертации;
* ● сохранить правила конвертации объектов, установить флажок сохранения кода обработчиков выгрузки в текстовый документ и указать путь к нему (при этом модуль загрузки необходимо сформировать из второго комплекта правил аналогичным образом);
* ● в конфигурации:
* ● загрузить правила конвертации, сохраненные в «Конвертации данных»;
* ● создать обработки в составе конфигурации. Рекомендуемые имена обработок: **ОбработчикиЗагрузкиИз<ИмяКонфигурации>** и **ОбработчикиВыгрузкиВ<ИмяКонфигурации>**;
* ● добавить функции в модуль менеджера плана обмена (см. ниже);
* ● скопировать текст модулей из выгруженных ранее файлов в модули созданных обработок.

### Модуль менеджера плана обмена

В модуле менеджера плана обмена должны быть обязательно объявлены следующие экспортные процедуры:

* ● **ОпределитьНастройки(Настройки, ИдентификаторНастройки**) – процедура-обработчик, позволяющая переопределить настройки плана обмена, задаваемые по умолчанию в функции **ОбменДаннымиСервер.НастройкиПланаОбменаПоУмолчанию**. Параметры:
* ● **Настройки – Структура** – ключом является имя настройки, значением – значение настройки:
* ● **ПредупреждатьОНесоответствииВерсийПланаОбмена – Булево** – признак необходимости проверки на расхождение версий в правилах конвертации, используется только для **УОП**, проверка выполняется при загрузке комплекта правил, при отправке данных и при получении данных;
* ● **ПутьКФайлуКомплектаПравилНаПользовательскомСайте – Строка**– содержит путь к файлу комплекта правил в виде архива на пользовательском сайте в разделе конфигурации;
* ● **ПутьКФайлуКомплектаПравилВКаталогеШаблонов – Строка**– содержит относительный путь к файлу комплекта правил в каталоге шаблонов «1С:Предприятия»;
* ● **ЗаголовокКомандыДляСозданияНовогоОбменаДанными – Строка**– содержит представление команды, выводимое в пользовательском интерфейсе при создании новой настройки обмена данными;
* ● **ЗаголовокПомощникаСозданияОбмена – Строка**– содержит представление заголовка формы помощника создания обмена данными, выводимое в пользовательском интерфейсе;
* ● **НаименованиеКонфигурацииКорреспондента – Строка**– содержит наименование конфигурации корреспондента выводимое в пользовательском интерфейсе;
* ● **ЗаголовокУзлаПланаОбмена – Строка**– содержит представление узла плана обмена, выводимое в пользовательском интерфейсе;
* ● **ВариантыНастроекОбмена – Массив**– список возможных настроек плана обмена, используется для создания заготовленных шаблонов с заполненными настройками планов обмена;
* ● **ОтображатьНастройкуОтборовНаУзле – Булево** – признак отображения настроек отборов на узле в помощнике создания обмена;
* ● **ОтображатьЗначенияПоУмолчаниюНаУзле – Булево** – признак отображения значений по умолчанию в помощнике создания обмена;
* ● **ОтображатьНастройкуОтборовНаУзлеБазыКорреспондента – Булево** – признак отображения настроек отборов на узле базы-корреспондента в помощнике создания обмена;
* ● **ОтображатьЗначенияПоУмолчаниюНаУзлеБазыКорреспондента – Булево** – признак отображения значений по умолчанию базы-корреспондента в помощнике создания обмена;
* ● **ИдентификаторНастройки – Строка** – уточняет вариант настройки, список возможных значений которых был объявлен в параметре **Настройки** - **ВариантыНастроекОбмена**.
* ● **ОбработчикПроверкиПараметровУчета(Отказ, Получатель, Сообщение)** – обработчик вызывается помощником настройки обмена при настройке обмена данными через веб-сервис. В обработчике необходимо определить алгоритм проверки параметров учета в конфигурации. Если в результате проверки выяснится, что параметры учета не настроены или настроены не полностью, то в обработчике необходимо переменной **Отказ**установить значение **Истина**. Дополнительно можно назначить переменной **Сообщение** описание для пользователя, какие именно настройки необходимо заполнить. Параметры:
* ● **Отказ**. Тип: **Булево**. Если в информационной базе заданы не все настройки параметров учета, необходимые для правильной работы обмена данными, то нужно установить этот флажок;
* ● **Получатель**. Тип: **ПланОбменаСсылка**. Ссылка на узел плана обмена, для которого выполняется настройка обмена данными;
* ● **Сообщение**. Тип: **Строка**. Сообщение пользователю о действиях, которые необходимо выполнить для правильной настройки параметров учета в конфигурации.
* ● **ПолучитьДополнительныеДанныеДляКорреспондента(ДополнительныеДанные)** – процедура-обработчик получения произвольных дополнительных данных, используемых при настройке обмена в базе-корреспонденте. Полученные данные могут быть использованы в обработчике **ПриСозданииНаСервере** в форме настройки узла плана обмена **ФормаНастройкиЗначенийПоУмолчаниюБазыКорреспондента** при настройке обмена данными с использованием помощника. Параметры:
* ● **ДополнительныеДанные – Структура** – дополнительные данные, которые будут использованы в базе-корреспонденте при настройке обмена; в качестве значений структуры применимы только значения, поддерживающие XDTO-сериализацию.
* ● **ПриПодключенииККорреспонденту(ВерсияКорреспондента)** – обработчик события при подключении к корреспонденту. Событие возникает при успешном подключении к корреспонденту и получении версии конфигурации корреспондента при настройке обмена с использованием помощника через прямое подключение или при подключении к корреспонденту через Интернет. В обработчике можно проанализировать версию корреспондента и, если настройка обмена не поддерживается с корреспондентом указанной версии, вызвать исключение. Параметры:
* ● **ВерсияКорреспондента** (только чтение) – **Строка** – версия конфигурации корреспондента, например "2.1.5.1".
* ● **ПриОтправкеДанныхОтправителя(Отправитель, Игнорировать)** – обработчик события при отправке данных узла-отправителя. Событие возникает при отправке данных узла-отправителя из текущей базы в базу-корреспондент, до помещения данных узла в сообщение обмена. В обработчике можно изменить отправляемые данные или вовсе отказаться от отправки данных узла. Параметры:
* ● **Отправитель** – тип значения зависит от текущего состояния обмена данными с другой программой:
* ● при настройке нового обмена данными – **Структура** – содержит настройки отборов на узле;
* ● при выполнении обмена данными – **ПланОбменаОбъект** – узел плана обмена, от имени которого выполняется отправка данных;
* ● **Игнорировать – Булево** – признак отказа от выгрузки данных узла. Если в обработчике установить значение этого параметра в значение **Истина**, то отправка данных узла выполнена не будет. Значение по умолчанию – **Ложь**. При настройке нового обмена параметр не используется.
* ● **ПриПолученииДанныхОтправителя(Отправитель, Игнорировать)** – обработчик события при получении данных узла-отправителя. Событие возникает при получении данных узла-отправителя, когда данные узла прочитаны из сообщения обмена, но не записаны в информационную базу. В обработчике можно изменить полученные данные или вовсе отказаться от получения данных узла. Параметры:
* ● **Отправитель** – тип значения зависит от текущего состояния обмена данными с другой программой:
* ● при настройке нового обмена данными – **Структура** – содержит настройки отборов на узле;
* ● при выполнении обмена данными – **ПланОбменаОбъект** – узел плана обмена, от имени которого выполняется отправка данных;
* ● **Игнорировать – Булево** – признак отказа от выгрузки данных узла. Если в обработчике установить значение этого параметра в значение **Истина**, то отправка данных узла выполнена не будет. Значение по умолчанию – **Ложь**. При настройке нового обмена параметр не используется.

В модуле менеджера плана обмена должны быть обязательно объявлены следующие экспортные функции:

* ● **НастройкаОтборовНаУзле(ВерсияКорреспондента, ИмяФормы, ИдентификаторНастройки)** – возвращает структуру отборов на узле плана обмена со значениями по умолчанию; структура настроек повторяет состав реквизитов шапки и табличных частей плана обмена; для реквизитов шапки используются аналогичные по ключу и значению элементы структуры, а для табличных частей используются структуры, содержащие массивы значений полей табличных частей плана обмена. Возвращает **Неопределено**, если отборы на узле не предусмотрены. Параметры:
* ● **ВерсияКорреспондента –**см. описание для процедуры **ПриПодключенииККорреспонденту**;
* ● **ИмяФормы – Строка** – имя используемой формы настройки узла. Возможно, например, использование различных форм для разных версий корреспондента;
* ● **ИдентификаторНастройки –**см. описание для процедуры **ОпределитьНастройки**.
* ● **НастройкаОтборовНаУзлеБазыКорреспондента(ВерсияКорреспондента, ИмяФормы, ИдентификаторНастройки)** – возвращает структуру отборов на узле плана обмена со значениями по умолчанию для базы корреспондента обмена. Структура настроек повторяет состав реквизитов шапки и табличных частей плана обмена базы корреспондента; для реквизитов шапки используются аналогичные по ключу и значению элементы структуры, а для табличных частей используются структуры, содержащие массивы значений полей табличных частей плана обмена (только для обмена через внешнее соединение). Возвращает **Неопределено**, если отборы на узле не предусмотрены. Параметры: см. описание для функции **НастройкаОтборовНаУзле**.
* ● **ЗначенияПоУмолчаниюНаУзле(ВерсияКорреспондента, ИмяФормы, ИдентификаторНастройки**) – возвращает структуру отборов на узле плана обмена со значениями по умолчанию. Структура настроек повторяет состав реквизитов шапки плана обмена; для реквизитов шапки используются аналогичные по ключу и значению элементы структуры (только для обмена по правилам). Возвращает **Неопределено**, если значения по умолчанию на узле не предусмотрены. Параметры: см. описание для функции **НастройкаОтборовНаУзле**.
* ● **ЗначенияПоУмолчаниюНаУзлеБазыКорреспондента(ВерсияКорреспондента, ИмяФормы, ИдентификаторНастройки**) – возвращает структуру отборов на узле плана обмена со значениями по умолчанию для базы корреспондента обмена. Структура настроек повторяет состав реквизитов шапки плана обмена; для реквизитов шапки используются аналогичные по ключу и значению элементы структуры (только для обмена по правилам через внешнее соединение). Возвращает **Неопределено**, если значения по умолчанию на узле не предусмотрены. Параметры: см. описание для функции **НастройкаОтборовНаУзле**.
* ● **ОписаниеОграниченийПередачиДанных(НастройкаОтборовНаУзле, ВерсияКорреспондента, ИдентификаторНастройки)** – возвращает строку описания ограничений миграции данных для пользователя; прикладной разработчик на основе установленных отборов на узле должен сформировать строку описания ограничений, удобную для восприятия пользователем. Возвращает пустую строку, если ограничения миграции данных для узла не предусмотрены. Параметры:
* ● **НастройкаОтборовНаУзле – Структура** – структура отборов на узле плана обмена полученная при помощи функции **НастройкаОтборовНаУзле**;
* ● **ВерсияКорреспондента –**см. описание для процедуры **ПриПодключенииККорреспонденту**;
* ● **ИдентификаторНастройки –**см. описание для процедуры **ОпределитьНастройки**.
* ● **ОписаниеОграниченийПередачиДанныхБазыКорреспондента(НастройкаОтборовНаУзле, ВерсияКорреспондента, ИдентификаторНастройки)** – возвращает строку описания ограничений миграции данных для базы корреспондента, которая отображается пользователю; прикладной разработчик на основе установленных отборов на узле базы корреспондента обмена должен сформировать строку описания ограничений, удобную для восприятия пользователем (только для обмена через внешнее соединение). Возвращает пустую строку, если ограничения миграции данных для узла не предусмотрены. Параметры: см. описание для функции **ОписаниеОграниченийПередачиДанных**.
* ● **ОписаниеЗначенийПоУмолчанию(ЗначенияПоУмолчаниюНаУзле, ВерсияКорреспондента, ИдентификаторНастройки)** – возвращает пользовательское представление значений по умолчанию в виде строки. Возвращает пустую строку, если значения по умолчанию на узле не предусмотрены. Параметры:
* ● **ЗначенияПоУмолчаниюНаУзле – Структура** – структура значений по умолчанию на узле плана обмена, полученная при помощи функции **ЗначенияПоУмолчаниюНаУзле**;
* ● **ВерсияКорреспондента** –см. описание для процедуры **ПриПодключенииККорреспонденту**;
* ● **ИдентификаторНастройки** –см. описание для процедуры **ОпределитьНастройки**.
* ● **ОписаниеЗначенийПоУмолчаниюБазыКорреспондента (ЗначенияПоУмолчаниюНаУзле**, **ВерсияКорреспондента**, **ИдентификаторНастройки)** – возвращает пользовательское представление значений по умолчанию в виде строки для базы корреспондента обмена (только для обмена через внешнее соединение). Возвращает пустую строку, если значения по умолчанию на узле не предусмотрены. Параметры: см. описание для функции **ЗначенияПоУмолчаниюНаУзле**.
* ● **ИмяФайлаНастроекДляПриемника** – имя файла настроек по умолчанию. В этот файл будут выгружены настройки обмена для приемника; значение должно быть одинаковым в плане обмена источника и приемника.
* ● **ИспользуемыеТранспортыСообщенийОбмена** – возвращает массив используемых транспортов сообщений для текущего плана обмена; значение должно быть одинаковым в плане обмена источника и приемника.
* ● **ИмяФормыСозданияНачальногоОбраза** – возвращает пользовательскую форму для создания начального образа базы. Эта форма будет открыта после завершения настройки обмена с помощью помощника для планов обмена РИБ. Для планов обмена универсального обмена функция должна возвращать пустую строку.
* ● **ИспользоватьПомощникСозданияОбменаДанными** – определяет, будет ли использоваться помощник для создания новых узлов плана обмена.
* ● **КорреспондентВМоделиСервиса** – возвращает признак того, что план обмена поддерживает обмен данными с корреспондентом, работающим в модели сервиса. Если признак установлен, то становится возможным создать обмен данными, когда текущая информационная база работает в локальном режиме, а корреспондент – в модели сервиса.
* ● **КраткаяИнформацияПоОбмену(ИдентификаторНастройки)** – возвращает строку с кратким описанием обмена данными, которое выводится на первой странице «Помощника создания обмена данными». Параметры:
* ● **ИдентификаторНастройки –**см. описание для процедуры **ОпределитьНастройки**.
* ● **ПодробнаяИнформацияПоОбмену(ИдентификаторНастройки)** – возвращает ссылку на веб-страницу или полный путь к форме внутри конфигурации строкой. Параметры:см. описание для процедуры **КраткаяИнформацияПоОбмену**.
* ● **ОбщиеДанныеУзлов(ВерсияКорреспондента, ИмяФормы)** – возвращает имена реквизитов и табличных частей плана обмена, перечисленные через запятую, которые являются общими для пары обменивающихся конфигураций. Например, если для плана обмена предусмотрено ограничение миграции данных по организациям в обе стороны, то табличная часть плана обмена, в которой перечислены разрешенные организации, считается общей. Возвращает пустую строку, если общие данные узлов не предусмотрены. Параметры:
* ● **ВерсияКорреспондента –**см. описание для процедуры **ПриПодключенииККорреспонденту**;
* ● **ИмяФормы – Строка** – имя используемой формы настройки узла. Возможно, например, использование различных форм для разных версий корреспондента.
* ● **ПланОбменаИспользуетсяВМоделиСервиса** – возвращает признак использования плана обмена для организации обмена в модели сервиса. Если признак установлен, то в сервисе можно включить обмен данными с использованием этого плана обмена, а также можно использовать этот план обмена для организации автономной работы в модели сервиса. Если признак не установлен, то план обмена будет использоваться только для обмена в локальном режиме работы конфигурации.
* ● **ПояснениеДляНастройкиПараметровУчета** – возвращает пояснение о последовательности действий пользователя для настройки параметров учета в текущей информационной базе.
* ● **ПояснениеДляНастройкиПараметровУчетаБазыКорреспондента(ВерсияКорреспондента)** – возвращает пояснение о последовательности действий пользователя для настройки параметров учета в базе-корреспонденте. Параметры: см. описание для процедуры **ПриПодключенииККорреспонденту**.

Для всех видов обмена кроме РИБ в модуле менеджера плана обмена должны быть обязательно объявлены следующие процедуры и функции:

* ● Процедура **НастроитьИнтерактивнуюВыгрузку(Получатель, Параметры),** предназначенная для настройки параметров работы помощника обмена данными на этапе интерактивного дополнения выборки выгружаемых данных. Для настройки доступны три типовых варианта добавления данных к выгрузке («Не добавлять», «Добавить все документы за период», «Добавить данные с произвольным отбором») и один дополнительный вариант по переопределяемому сценарию узла. Каждый вариант можно отключить, изменить название, подсказку и т. д. При отключении всех вариантов этап интерактивного дополнения выборки выгружаемых данных будет целиком пропущен. Подробное описание и пример использования см. в демонстрационной конфигурации, модуль менеджера плана обмена **\_ДемоОбменСБиблиотекойСтандартныхПодсистем**. Параметры:
* ● **Получатель – ПланОбменаСсылка** – узел, для которого производится настройка;
* ● **Параметры – Структура** – параметры для изменения. Для настройки необходимо установить значения свойств в необходимые значения.
* ● Функция**ПредставлениеОтбораИнтерактивнойВыгрузки(Получатель, Параметры)** – возвращает представление отбора для варианта дополнения выгрузки по сценарию узла. Параметры:
* ● **Получатель – ПланОбменаСсылка** – узел, для которого запрашивается представление;
* ● **Параметры – Структура** – описывает текущие параметры отбора, для которых запрашивается представление, поля структуры соответствуют аналогичным полям, заполняемым в процедуре **НастроитьИнтерактивнуюВыгрузку**. Подробное описание и пример использования см. в демонстрационной конфигурации, модуль менеджера плана обмена **\_ДемоОбменСБиблиотекойСтандартныхПодсистем**.
* ● Функция**ИмяКонфигурацииИсточника** – имя конфигурации-источника (как задано в свойстве **Имя** в метаданных) для точной идентификации имени этой конфигурации при настройке синхронизации данных в модели сервиса. Значение должно различаться в плане обмена источника и приемника.

В модуле менеджера плана обмена дополнительно могут содержаться процедуры и функции, наличие которых зависит от вида обмена данными.

Пример реализации функций можно посмотреть в демонстрационной конфигурации в планах обмена **\_ДемоОбменСБиблиотекойСтандартныхПодсистем**,**\_ДемоОбменВРаспределеннойИнформационнойБазе**.

Для обмена через универсальный формат в модуле менеджера плана обмена дополнительно должны быть обязательно объявлены следующие экспортные функции и процедуры:

* ● **ФорматОбмена** – функция, возвращает имя используемого формата в виде строки (пример: http://v8.1c.ru/edi/edi\_stnd/EnterpriseData);
* ● **ПолучитьВерсииФорматаОбмена(ВерсииФормата)** – процедура, заполняет соответствие номеров поддерживаемых версий формата данных и ссылок на общие модули, реализующих логику обмена через конкретную версию формата.

Пример реализации функций можно посмотреть в демонстрационной конфигурации в плане обмена **\_ДемоСинхронизацияДанныхЧерезУниверсальныйФормат**.

### Формы плана обмена

Формы узла и списка плана обмена создаются по усмотрению разработчика. Если для плана обмена предполагается использовать ограничение миграции данных по узлам, то для этого следует в форме узла предусмотреть редактирование настроек ограничения миграции данных.

В форме узла должен быть определен обработчик **ПриЗакрытии**:

&НаКлиенте

Процедура ПриЗакрытии()

Оповестить("Запись\_УзелПланаОбмена");

КонецПроцедуры

Если предполагается использование формы узла для обмена данными в модели сервиса, то при работе в этом режиме рекомендуется скрывать от пользователя код и наименование узла.

Дополнительно необходимо разработать произвольные формы плана обмена. Эти формы используются на этапе настройки обмена данными помощником настройки.

Возможность использования форм для планов обмена в зависимости от вида обмена представлена в таблице 3.55.

Таблица 3.55.

|  |  |  |  |
| --- | --- | --- | --- |
| **Имя формы узла плана обмена** | **УОП** | **УО** | **РИБ** |
| **ФормаНастройкиУзлов** | + | + | – |
| **ФормаНастройкиУзла** | + | + | + |
| **ФормаНастройкиУзлаБазыКорреспондента** | + | – | – |
| **ФормаНастройкиЗначенийПоУмолчанию** | + | – | – |
| **ФормаНастройкиЗначенийПоУмолчаниюБазы-Корреспондента** | + | – | – |
| **ФормаСозданияНачальногоОбраза** | – | – | + |

Необходимость использования форм для планов обмена в зависимости от поддерживаемых транспортов сообщений представлена в таблице ниже. Если необходимо поддерживать несколько транспортов сообщений, то признаки использования нужно складывать по условию **ИЛИ**. В общем случае, когда поддерживаются все транспорты, необходимо использовать все перечисленные формы.

Таблица 3.56.

|  |  |  |  |
| --- | --- | --- | --- |
| **Имя формы узла плана обмена** | **Обмен через сетевой каталог, e-mail, FTP** | **Обмен через внешнее соединение** | **Обмен через веб-сервис** |
| **ФормаНастройкиУзлов** | – | – | + |
| **ФормаНастройкиУзла** | + | + | – |
| **ФормаНастройкиУзлаБазыКорреспондента** | – | + | – |
| **ФормаНастройкиЗначенийПоУмолчанию** | + | + | + |
| **ФормаНастройкиЗначенийПоУмолчаниюБазыКорреспондента** | – | + | + |
| **ФормаСозданияНачальногоОбраза** | + | – | – |

Для реквизитов настроечных форм можно назначать функциональные опции, позволяющие оперативно изменять состав настроек узла в зависимости от общих установок приложения. Например, в форме значений по умолчанию можно указать ставку НДС для получаемых данных, если в них не проставлено необходимое значение. Необходимость заполнения этого реквизита, его видимость могут управляться соответствующей функциональной опцией.

#### Разработка формы «ФормаНастройкиУзлов»

Если для плана обмена планируется использовать обмен данными через веб-сервис или обмен данными в модели сервиса и для плана обмена предусмотрены фильтры ограничения миграции данных, то необходимо создать произвольную форму плана обмена с предопределенным именем **ФормаНастройкиУзлов**. Она вызывается помощником настройки обмена данными для задания ограничения миграции данных. Для случая, когда необходимо поддержать настройку обмена с базой-корреспондентом разных версий, можно создать несколько форм настройки узлов. Имя используемой формы в зависимости от версии корреспондента определяет функция, расположенная в модуле менеджера плана обмена **ОбщиеДанныеУзлов**.

Для реквизитов формы, которые размещены в форме и определяют признак модифицированности формы, следует установить свойство **Сохраняемые данные**.

Табличные части в форме должны именоваться как соответствующие табличные части планов обмена текущей базы и базы-корреспондента. Табличная часть должна содержать три поля: **Использовать** (**Булево**), **Представление** (**Строка**, неограниченной длины), **УникальныйИдентификаторСсылки** (**Строка**, неограниченной длины). Заполнение табличных частей в форме выполняется автоматически по данным текущей информационной базы и данным базы-корреспондента.

Форма условно разделяется на три части: общие данные узлов, данные этой информационной базы и данные базы-корреспондента. Общие данные узлов одинаковы для двух обменивающихся информационных баз. Реквизиты и табличные части плана обмена, которые относятся к общим данным, определяет функция, расположенная в модуле менеджера плана обмена **ОбщиеДанныеУзлов**.

Если необходимо раздельно вводить информацию об одноименных реквизитах планов обмена этой информационной базы и базы-корреспондента, то нужно добавить в форму реквизиты **ИменаРеквизитов** и **ИменаРеквизитовБазыКорреспондента** произвольного типа. В обработчике формы **ПриСозданииНаСервере**задать значения этих реквизитов. Они должны содержать значения типа **Структура**: ключом является имя реквизита плана обмена, а значением – имя реквизита формы, который используется для ввода значения этого реквизита.

Например, в этой информационной базе есть план обмена с табличной частью **Организации**, и у плана обмена базы-корреспондента тоже задан такой реквизит. Было принято решение, что отбор по организациям необходимо устанавливать независимо для каждой из информационных баз. То есть организации не являются общими данными узлов, и в форме должно быть два списка организаций. Для этого нужно добавить в форму реквизиты, описанные выше, и заполнить их значения следующим образом:

ИменаРеквизитов = Новый Структура;

ИменаРеквизитов.Вставить("Организации", "ОрганизацииЭтойПрограммы");

ИменаРеквизитовБазыКорреспондента = Новый Структура;

ИменаРеквизитовБазыКорреспондента.Вставить("Организации", "ОрганизацииДругойПрограммы");

Это означает, что для выбора организаций этой информационной базы будет использоваться реквизит формы **ОрганизацииЭтойПрограммы**, а для выбора организаций базы-корреспондента будет использоваться реквизит **ОрганизацииДругойПрограммы**.

В обработчике формы **ПриСозданииНаСервере** необходимо предусмотреть задание значений по умолчанию для реквизитов формы. Также нужно определить значение реквизита **ОписаниеКонтекста**.

В модуле формы обязательно наличие следующих обработчиков:

&НаСервере

Процедура ПриСозданииНаСервере(Отказ, СтандартнаяОбработка)

Если Параметры.Свойство("ПолучитьЗначенияПоУмолчанию") Тогда

// Задание значений по умолчанию

...

КонецЕсли;

ОбменДаннымиСервер.ФормаНастройкиУзловПриСозданииНаСервере(ЭтотОбъект, Отказ);

ПолучитьОписаниеКонтекста();

КонецПроцедуры

&НаКлиенте

Процедура ПередЗакрытием(Отказ, СтандартнаяОбработка)

ОбменДаннымиКлиент.ФормаНастройкиПередЗакрытием(Отказ, ЭтотОбъект);

КонецПроцедуры

В форме необходимо реализовать команду **ЗаписатьИЗакрыть** с обработчиком команды:

&НаКлиенте

Процедура ЗаписатьИЗакрыть(Команда)

ПолучитьОписаниеКонтекста();

ОбменДаннымиКлиент.ФормаНастройкиУзловКомандаЗакрытьФорму(ЭтотОбъект);

КонецПроцедуры

Описание контекста формы хранится в реквизите формы **ОписаниеКонтекста**. Значение реквизита **ОписаниеКонтекста** используется в помощнике настройки обмена для отображения значений настроек. Обновлять описание контекста следует при создании формы на сервере и перед закрытием формы.

#### Разработка формы «ФормаНастройкиУзла»

Если для плана обмена предусмотрены фильтры ограничения миграции данных, то должна быть создана произвольная форма с предопределенным именем **ФормаНастройкиУзла**. Она вызывается помощником настройки обмена данными для задания ограничения миграции данных. Для случая, когда необходимо поддержать настройку обмена с базой-корреспондентом разных версий, можно создать несколько форм настройки. Имя используемой формы в зависимости от версии корреспондента определяет функция, расположенная в модуле менеджера плана обмена, – **НастройкаОтборовНаУзле**.

Состав реквизитов формы повторяет состав реквизитов шапки и табличных частей плана обмена. Для формы определены обязательные реквизиты формы: **НастройкаОтборовНаУзле** произвольного типа и**ВерсияКорреспондента** (**Строка**, неограниченной длины). Для реквизитов формы, которые размещены в форме и определяют признак модифицированности формы, следует установить свойство **Сохраняемые данные**.

В модуле формы обязательно наличие следующих обработчиков:

&НаСервере

Процедура ПриСозданииНаСервере(Отказ, СтандартнаяОбработка)

ОбменДаннымиСервер.ФормаНастройкиУзлаПриСозданииНаСервере(ЭтотОбъект, "\_ДемоОбменСБиблиотекойСтандартныхПодсистем");

КонецПроцедуры

&НаКлиенте

Процедура ПередЗакрытием(Отказ, СтандартнаяОбработка)

ОбменДаннымиКлиент.ФормаНастройкиПередЗакрытием(Отказ, ЭтотОбъект);

КонецПроцедуры

&НаКлиенте

Процедура КомандаОК(Команда)

ОбменДаннымиКлиент.ФормаНастройкиУзлаКомандаЗакрытьФорму(ЭтотОбъект);

КонецПроцедуры

В качестве второго параметра процедуры **ФормаНастройкиУзлаПриСозданииНаСервере** следует использовать имя плана обмена (**Строка**), как оно задано в конфигураторе. В свойствах формы для обработчиков событий **ПриСозданииНаСервере** и **ПередЗакрытием** необходимо задать соответствующие процедуры модуля формы.

#### Разработка формы «ФормаНастройкиУзлаБазыКорреспондента»

Если для плана обмена планируется использовать обмен данными через внешнее соединение и для плана обмена предусмотрены фильтры ограничения миграции данных, то необходимо создать произвольную форму плана обмена с предопределенным именем **ФормаНастройкиУзлаБазыКорреспондента**. Она вызывается помощником настройки обмена данными для задания ограничения миграции данных на узле в базе-корреспонденте. Для случая, когда необходимо поддержать настройку обмена с базой-корреспондентом разных версий, можно создать несколько форм настройки. Имя используемой формы в зависимости от версии корреспондента определяет функция, расположенная в модуле менеджера плана обмена, – **НастройкаОтборовНаУзлеБазыКорреспондента**.

Состав реквизитов формы повторяет состав реквизитов шапки и табличных частей плана обмена базы-корреспондента. Для формы определены три обязательных реквизита формы: **НастройкаОтборовНаУзле** (тип **Произвольный**), **ПараметрыВнешнегоСоединения** (тип **Произвольный**) и **ВерсияКорреспондента** (**Строка**, неограниченной длины). Для реквизитов формы, которые размещены в форме и определяют признак модифицированности формы, следует установить свойство **Сохраняемые данные**.

В модуле формы обязательно наличие следующих обработчиков:

&НаСервере

Процедура ПриСозданииНаСервере(Отказ, СтандартнаяОбработка)

ОбменДаннымиСервер.ФормаНастройкиУзлаБазыКорреспондентаПриСозданииНаСервере(ЭтотОбъект, "\_ДемоОбменСБиблиотекойСтандартныхПодсистем");

КонецПроцедуры

&НаКлиенте

Процедура КомандаОК(Команда)

ОбменДаннымиКлиент.ФормаНастройкиУзлаКомандаЗакрытьФорму(ЭтотОбъект);

КонецПроцедуры

&НаКлиенте

Процедура ПередЗакрытием(Отказ, СтандартнаяОбработка)

ОбменДаннымиКлиент.ФормаНастройкиПередЗакрытием(Отказ, ЭтотОбъект);

КонецПроцедуры

В качестве второго параметра процедуры **ФормаНастройкиУзлаБазыКорреспондентаПриСозданииНаСервере** следует использовать имя плана обмена (**Строка**), как оно задано в конфигураторе. В свойствах формы для обработчиков событий **ПриСозданииНаСервере** и **ПередЗакрытием** необходимо задать соответствующие процедуры модуля формы.

Все реквизиты формы должны иметь примитивный тип, кроме описанных особо (см. выше). Реквизитам ссылочных типов плана обмена должны соответствовать реквизиты формы типа **Строка** (неограниченной длины). Реквизит формы, который отображает данные реквизита ссылочного типа плана обмена, будем условно называть отображаемым реквизитом. Для каждого отображаемого реквизита следует создать по одному дополнительному реквизиту. Дополнительный реквизит используется для сохранения уникального идентификатора ссылки объекта базы-корреспондента. Имя дополнительного реквизита строго регламентировано и складывается из имени отображаемого реквизита и постфикса **\_Ключ**. Тип дополнительного реквизита – **Строка** (неограниченной длины).

Отображаемые реквизиты предназначены для получения значений ссылочных типов базы-корреспондента. Сами данные ссылочного типа базы-корреспондента не могут быть отображены, поэтому отображаемый реквизит содержит только представление данных базы-корреспондента. В текущей реализации подсистемы обмена возможна работа только с объектами базы-корреспондента типа **Справочник**.

Наряду с отображаемыми реквизитами в форме могут использоваться реквизиты примитивных типов и перечислений. Для реквизитов с типом перечислений требуется, чтобы в конфигурации первой базы и конфигурации базы-корреспондента присутствовал один и тот же тип перечисления.

Интерактивный выбор и задание значений для отображаемых реквизитов не являются стандартными и переопределяются с использованием соответствующих обработчиков событий элементов формы. Для задания значения отображаемого реквизита, расположенного в форме, следует определить обработчик события **НачалоВыбора** для связанного элемента формы. Также следует определить обработчик **ОбработкаВыбора**. Если отображаемый реквизит является реквизитом формы, то обработчик **ОбработкаВыбора** следует определять для формы. Если отображаемый реквизит является колонкой реквизита формы, то обработчик **ОбработкаВыбора**следует определять для соответствующей таблицы формы.

Пример вызова обработчиков для случая, когда отображаемый реквизит является колонкой реквизита формы:

&НаКлиенте

Процедура ОрганизацииОрганизацияНачалоВыбора(Элемент, ДанныеВыбора, СтандартнаяОбработка)

ОбменДаннымиКлиент.ОбработчикВыбораЭлементовБазыКорреспондентаНачалоВыбора("Организация", "Справочник.Организации", Элементы.Организации, СтандартнаяОбработка, ПараметрыВнешнегоСоединения);

КонецПроцедуры

&НаКлиенте

Процедура ОрганизацииОбработкаВыбора(Элемент, ВыбранноеЗначение, СтандартнаяОбработка)

ОбменДаннымиКлиент.ОбработчикВыбораЭлементовБазыКорреспондентаОбработкаВыбора(Элемент, ВыбранноеЗначение, Организации);

КонецПроцедуры

Примеры вызова обработчиков для случая, когда отображаемый реквизит является реквизитом формы:

&НаКлиенте

Процедура НоменклатураНачалоВыбора(Элемент, ДанныеВыбора, СтандартнаяОбработка)

ОбменДаннымиКлиент.ОбработчикВыбораЭлементовБазыКорреспондентаНачалоВыбора("Номенклатура", "Справочник.\_ДемоНоменклатура", ЭтотОбъект, СтандартнаяОбработка, ПараметрыВнешнегоСоединения);

КонецПроцедуры

&НаКлиенте

Процедура ОбработкаВыбора(ВыбранноеЗначение, ИсточникВыбора)

ОбменДаннымиКлиент.ОбработчикВыбораЭлементовБазыКорреспондентаОбработкаВыбора(ЭтотОбъект, ВыбранноеЗначение);

КонецПроцедуры

* ● Процедура **ОбменДаннымиКлиент.ОбработчикВыбораЭлементовБазыКорреспондентаНачалоВыбора(ИмяРеквизита, ИмяТаблицы, Владелец, СтандартнаяОбработка, ПараметрыВнешнегоСоединения**). Параметры:
* ● **ИмяРеквизита – Строка** – имя отображаемого реквизита формы, для которого требуется установить значение;
* ● **ИмяТаблицы – Строка** – полное имя таблицы базы корреспондента, которая выступает источником данных для выбора значения;
* ● **Владелец – Управляемая форма**, **Таблица формы**. Если процедура вызывается из обработчика события поля формы, то в качестве значения следует использовать текущую управляемую форму. Если процедура вызывается из обработчика события поля таблицы формы, то следует указывать текущую таблицу формы;
* ● **СтандартнаяОбработка – Булево** – флажок стандартной обработки; параметр дублирует одноименный флажок обработчика события **НачалоВыбора**.
* ● **ПараметрыВнешнегоСоединения** – произвольный – параметр дублирует одноименный реквизит формы.
* ● Процедура **ОбменДаннымиКлиент.ОбработчикВыбораЭлементовБазыКорреспондентаОбработкаВыбора(Элемент, ВыбранноеЗначение, ДанныеФормыКоллекция)**. Параметры:
* ● **Элемент – Управляемая форма**, **Таблица формы** – элемент, которому адресовано событие обработки выбора. Если процедура вызывается из обработчика события формы, то в качестве значения следует использовать текущую управляемую форму. Если процедура вызывается из обработчика события таблицы формы, то следует указывать текущую таблицу формы;
* ● **ВыбранноеЗначение – Структура** – параметр дублирует одноименный параметр обработчика события **ОбработкаВыбора**;
* ● **ДанныеФормыКоллекция**(необязательный). Параметр указывается, когда процедура вызывается для таблицы формы. В параметр передается таблица формы, для которой вызывается событие **ОбработкаВыбора**.

#### Разработка формы «ФормаНастройкиЗначенийПоУмолчанию»

Если для плана обмена требуется задавать значения по умолчанию, то должна быть создана произвольная форма с предопределенным именем **ФормаНастройкиЗначенийПоУмолчанию**. Разработка формы аналогична разработке формы **ФормаНастройкиУзла**.

#### Разработка формы «ФормаНастройкиЗначенийПоУмолчаниюБазыКорреспондента»

Если для плана обмена планируется использовать обмен данными через внешнее соединение и есть потребность задавать значения по умолчанию для базы-корреспондента, то необходимо создать произвольную форму плана обмена с предопределенным именем **ФормаНастройкиЗначенийПоУмолчаниюБазыКорреспондента**. Методика создания формы аналогична методике для формы **ФормаНастройкиУзлаБазыКорреспондента**. Для случая, когда необходимо поддержать настройку обмена с базой-корреспондентом разных версий, можно создать несколько форм настройки. Имя используемой формы в зависимости от версии корреспондента определяет функция, расположенная в модуле менеджера плана обмена, – **ЗначенияПоУмолчаниюНаУзлеБазыКорреспондента**.

Для формы определены два обязательных реквизита формы: **ЗначенияПоУмолчаниюНаУзле** (тип **Произвольный**) и **ВерсияКорреспондента** (**Строка**, неограниченной длины). Для реквизитов формы, которые размещены в форме и определяют признак модифицированности формы, следует установить свойство **Сохраняемые данные**.

#### Разработка формы «ФормаСозданияНачальногоОбраза»

Дополнительно для плана обмена РИБ возможно определить форму создания первоначального образа подчиненного узла с произвольным именем. Эта форма не является обязательной. Она либо используется для вызова стандартного окна создания начального образа, либо может переопределять стандартный процесс создания первоначального образа. Имя формы должно быть указано в функции **ИмяФормыСозданияНачальногоОбраза**модуля менеджера плана обмена.

#### Обязательные для заполнения реквизиты формы

Если некоторые из реквизитов плана обмена являются обязательными для заполнения, то в формах настроек для этих реквизитов следует установить свойство **Проверка заполнения** в значение **Выдавать ошибку**. При создании обмена данными с использованием помощника система потребует заполнения этих реквизитов.

Пример реализации форм можно посмотреть в демонстрационной конфигурации в планах обмена **\_ДемоОбменСБиблиотекойСтандартныхПодсистем** и **\_ДемоОбменВРаспределеннойИнформационнойБазе**.

#### Настройка отправки данных в помощнике интерактивной выгрузки. Дополнительная форма настройки

При использовании помощника интерактивного обмена существует возможность программной настройки этапа отправки данных для включения в состав передаваемой информации объектов, определенных пользователем, например «все документы за прошлый месяц». При этом к данным пользователя автоматически применяются и общие ограничения узла, например отбор по организациям. По умолчанию предлагаются три варианта дополнения отправляемых данных: не добавлять, добавить документы с отбором за период, добавить произвольные данные (документы и НСИ).

Отключение, настройка предопределенных вариантов, изменение их порядка в форме, определение дополнительного сценария настройки производятся в процедуре **НастроитьИнтерактивнуюВыгрузку** менеджера плана обмена. Для настройки необходимо изменить параметры, предлагаемые по умолчанию. Также нужно создать форму для редактирования настроек отбора дополнительных данных по сценарию узла. Полное имя этой формы требуется указать в соответствующем поле параметров.

Эта форма будет открыта для выбора из помощника интерактивного обмена с установленными параметрами, описывающими текущий отбор. Форма должна вернуть результатом выбора структуру, описывающую отбор, измененный пользователем.

Состав параметров открытия, описание результата выбора и пример реализации настройки можно увидеть в демонстрационной конфигурации. В модуле менеджера плана обмена **\_ДемоОбменСБиблиотекойСтандартныхПодсистем** демонстрируется настройка поведения помощника интерактивного обмена. В форме **НастройкаВыгрузки** этого плана обмена демонстрируются прием, обработка и обратная передача помощнику отредактированных параметров отбора.

### Макет правил регистрации

План обмена может содержать только один макет правил регистрации. Наличие макета правил регистрации рекомендуется, но не является обязательным. Наличие макета свидетельствует о необходимости выполнять ограничение миграции данных при обмене.

Правила регистрации хранятся в текстовом макете, в который помещается информация из файла правил в формате XML. Исходный файл правил формируется при помощи конфигурации «Конвертация данных», редакция 2.1. Кодировка файла: UTF-8.

Для размещения/обновления правил регистрации в макете необходимо:

* ● открыть файл правил XML в конфигурации как текстовый документ (с помощью команды меню **Файл – Открыть**);
* ● скопировать в буфер обмена содержимое файла;
* ● вставить в макет плана обмена данные из буфера обмена.

Имя макета правил регистрации для всех планов обмена строго регламентировано и должно принимать значение **ПравилаРегистрации**.

### Макеты правил обмена для планов обмена УОП

План обмена может содержать один или несколько макетов правил обмена. Наличие макетов правил обмена рекомендуется, но не является обязательным. Правила обмена хранятся в текстовом макете, в который помещается информация из файла правил обмена в формате XML. Исходный файл правил формируется при помощи конфигурации «Конвертация данных», редакция 2.1. Кодировка файла: UTF-8.

Для размещения правил обмена в макете необходимо выполнить ту же последовательность действий, что и при добавлении правил регистрации (см. раздел «[Макет правил регистрации](https://its.1c.ru/db/content/bsp23doc/src/3.28.%20%D0%BE%D0%B1%D0%BC%D0%B5%D0%BD%20%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D0%BC%D0%B8.htm?_=1504708101#_макет_правил_регистрации)»).

Имя макета правил обмена должно соответствовать шаблону: **ПравилаОбмена<Приемник><ВерсияПриемника>**, где **<Приемник>** – имя конфигурации-приемника, **<ВерсияПриемника>** – номер версии конфигурации-приемника.

Например, **ПравилаОбменаБухгалтерияПредприятия\_1\_6\_18**.

Обязательным является наличие макета с именем **ПравилаОбмена**. Правила обмена из этого макета будут использоваться как правила обмена по умолчанию при настройке обмена данными с использованием помощника настройки.

### Создание подписок на события

Для работы подсистемы обмена данными необходимо создать подписки на события. Подписки на события создаются для работы механизма регистрации изменения данных.

Механизм регистрации изменения данных используется не только для реализации прикладной логики ограничений миграции данных. Он также обеспечивает соблюдение особого порядка регистрации данных при выполнении обновления информационной базы, а также позволяет оптимизировать процесс регистрации данных при выполнении загрузки данных при обмене по правилам (УОП) и через универсальный формат (ОУФ).

Для каждого плана обмена создается свой набор подписок на события. В общем случае для одного плана обмена может быть создано шесть подписок на события. Параметры подписок следует задать согласно таблице, приведенной ниже. Имя подписки на событие строго регламентировано и определяется по шаблону: **<ИмяПланаОбмена><ВидПодписки>**, где **<ИмяПланаОбмена>** – имя плана обмена, для которого создается подписка на событие; **<ВидПодписки>** – вид подписки на событие (выбирается из таблицы, приведенной ниже).

Подписки на события для обмена данными:

Таблица 3.57.

|  |  |  |
| --- | --- | --- |
| **Вид подписки** | **Источник** | **Событие** |
| **РегистрацияДокумента** | Элементы типа: **ДокументОбъект** | Перед записью |
| **Регистрация** | Элементы типа: **СправочникОбъект**,**ПланВидовХарактеристикОбъект**, **ПланСчетовОбъект**, **ПланВидовРасчетаОбъект**, **БизнесПроцессОбъект**, **ЗадачаОбъект** | Перед записью |
| **РегистрацияНабора** | Элементы типа: **РегистрСведенийНаборЗаписей**,**РегистрНакопленияНаборЗаписей**, **РегистрБухгалтерииНаборЗаписей** | Перед записью |
| **РегистрацияНабораРасчета** | Элементы типа: **РегистрРасчетаНаборЗаписей** | Перед записью |
| **РегистрацияКонстанты** | Элементы типа: **КонстантаМенеджерЗначения** | Перед записью |
| **РегистрацияУдаления** | Элементы типа: **ДокументОбъект**, **СправочникОбъект**, **ПланВидовХарактеристикОбъект**, **ПланСчетовОбъект**, **ПланВидовРасчетаОбъект**, **БизнесПроцессОбъект**, **ЗадачаОбъект** | Перед удалением |

Для назначения обработчиков событий подписок в конфигурации требуется создать общий модуль с атрибутами: **Сервер**, **Внешнее соединение**, **Клиент (обычное приложение)**, **Вызов сервера**. Имена и количество таких общих модулей не регламентированы.

В процедуре – обработчике события подписки следует прописать вызов процедуры регистрации из общего модуля **ОбменДаннымиСобытия**. Имя и параметры процедуры регистрации зависят от вида подписки. Для выбора процедуры регистрации следует воспользоваться таблицей, приведенной ниже. В качестве первого параметра процедуры регистрации следует указать имя плана обмена, все остальные параметры повторяют параметры процедуры – обработчика события подписки, из которой выполняется вызов.

Таблица 3.58.

|  |  |
| --- | --- |
| **Вид подписки** | **Процедура – обработчика события** |
| **РегистрацияДокумента** | ОбменДаннымиСобытия.МеханизмРегистрацииОбъектовПередЗаписьюДокумента("<ИмяПланаОбмена>", Источник, Отказ, РежимЗаписи, РежимПроведения); |
| **Регистрация** | ОбменДаннымиСобытия.МеханизмРегистрацииОбъектовПередЗаписью("<ИмяПланаОбмена>", Источник, Отказ); |
| **РегистрацияНабора** | ОбменДаннымиСобытия.МеханизмРегистрацииОбъектовПередЗаписьюРегистра("<ИмяПланаОбмена>", Источник, Отказ, Замещение); |
| **РегистрацияНабораРасчета** | ОбменДаннымиСобытия.МеханизмРегистрацииОбъектовПередЗаписьюРегистра("<ИмяПланаОбмена>", Источник, Отказ, Замещение); |
| **РегистрацияКонстанты** | ОбменДаннымиСобытия.МеханизмРегистрацииОбъектовПередЗаписьюКонстанты("<ИмяПланаОбмена>", Источник, Отказ); |
| **РегистрацияУдаления** | ОбменДаннымиСобытия.МеханизмРегистрацииОбъектовПередУдалением("<ИмяПланаОбмена>", Источник, Отказ); |

Ниже перечислены объекты метаданных, которые не должны использоваться в качестве источников событий подписок:

* ● регистр сведений **СоответствияОбъектовИнформационныхБаз**,
* ● константа **НастройкиПодчиненногоУзлаРИБ**.

### Настройка общих команд

Для перечисленных в таблице общих команд при первоначальном встраивании подсистемы необходимо задать свойство **Тип параметра команды**. Следует задать составной тип со ссылками на планы обмена согласно таблице 3.59.

Таблица 3.59.

| **Имя команды** | **ОУФ** | **УОП** | **УО** | **РИБ** |
| --- | --- | --- | --- | --- |
| **Синхронизировать** | + | + | + | + |
| **СинхронизироватьСДополнительнымиПараметрами** | + | + | – | – |
| **НастроитьПараметрыТранспортаСообщенийОбмена** | + | + | + | + |
| **ЗагрузитьПравилаКонвертацииОбъектов** | - | + | – | – |
| **ЗагрузитьПравилаРегистрацииОбъектов** | + | + | + | + |
| **ЗагрузитьКомплектПравил** | - | + | - | - |
| **СценарииСинхронизации** | + | + | + | + |
| **СобытияОтправки** | + | + | + | + |
| **СобытияПолучения** | + | + | + | + |
| **ПолучитьНастройкиСинхронизацииДляДругойПрограммы** | + | + | + | – |
| **СоставОтправляемыхДанных** | + | + | + | + |
| **УдалитьНастройкуСинхронизации** | + | + | + | + |

Для конфигураций, работающих в локальном режиме, дополнительно можно создать отдельные команды для открытия помощников создания обменов данными. Для этого необходимо создать общую команду. Имя команды следует задавать согласно прикладному смыслу команды, например, **ПомощникНастройкиОбменаДаннымиСБухгалтерией**. В модуле команды в процедуре – обработчике команды записать код вида:

ОбменДаннымиКлиент.ОткрытьПомощникНастройкиОбменаДанными("\_ДемоОбменСБиблиотекойСтандартныхПодсистем");

В качестве параметра процедуры **ОткрытьПомощникНастройкиОбменаДанными** надо использовать имя плана обмена, для которого требуется открыть помощник. Для каждого плана обмена рекомендуется создать по одной такой команде (создание команды не является обязательным: помощник настройки обмена можно также вызвать из меню **Создать** в форме управления обменами данных).

Для конфигураций, работающих в модели сервиса, создавать отдельные команды для открытия помощников настройки обмена данными не требуется.

### Настройка прав доступа пользователей

Для настройки прав доступа пользователей к данным подсистемы «Обмен данными» следует использовать роли:

Таблица 3.60.

| **№** | **Роли и их назначение** |
| --- | --- |
|  | **ВыполнениеСинхронизацииДанных**  ● Интерактивный обмен данными (запуск вручную).  ● Мониторинг обмена данными |
|  | **ПолныеПрава** (из подсистемы «Базовая функциональность»)  ● Включение и отключение подсистемы «Обмен данными».  ● Добавление и изменение обменов данными.  ● Включение и отключение обменов данными.  ● Изменение префикса информационной базы.  ● Интерактивный обмен данными (запуск вручную).  ● Мониторинг обмена данными.  ● Удаление помеченных на удаление объектов подсистемы |

Дополнительно следует создать вспомогательные роли или использовать подходящие роли, существующие в конфигурации, для обеспечения доступа к данным, которые не относятся к подсистеме «Обмен данными», но требуются для работы с ней.

Таблица 3.61.

| **№** | **Вспомогательные роли и их назначение** |
| --- | --- |
|  | **<ДобавлениеИзменениеУзловПлановОбмена>**  Роль для изменения планов обмена конфигурации |
|  | **<ЧтениеУзловПлановОбмена>**  Роль для чтения узлов планов обмена конфигурации |

Примеры настройки прав доступа пользователей при работе в локальном режиме приведены ниже.

Таблица 3.62.

| **№** | **Группа пользователей и ее функции** | **Состав ролей** |
| --- | --- | --- |
|  | Администратор:  ● включение/отключение подсистемы в конфигурации;  ● удаление помеченных элементов подсистемы;  ● назначение префикса узла распределенной информационной базы | ● **ПолныеПрава** (из подсистемы «Базовая функциональность»),  ● **АдминистраторСистемы** (из подсистемы «Базовая функциональность») |
|  | Администратор обмена:  ● создание новых обменов данными;  ● изменение существующих обменов данными;  ● выполнение обмена;  ● мониторинг обмена | ● **ЗапускТонкогоКлиента** (из подсистемы «Базовая функциональность»),  ● **ПросмотрЖурналаРегистрации** (из подсистемы «Базовая функциональность»),  ● **<ДобавлениеИзменениеУзловПлановОбмена>**,  ● Для создания начального образа узла РИБ также требуется наличие права **Администрирование данных** |
|  | Пользователь обмена:  ● выполнение обмена;  ● мониторинг обмена | ● **ЗапускТонкогоКлиента** (из подсистемы «Базовая функциональность»),  ● **ПросмотрЖурналаРегистрации** (из подсистемы «Базовая функциональность»),  ● **ВыполнениеСинхронизацииДанных**,  **● <ЧтениеУзловПлановОбмена>** |

Также следует учитывать, что при создании и выполнении обмена данными через внешнее соединение в базе-корреспонденте должен использоваться пользователь с ролями **АдминистраторСистемы** и **ПолныеПрава**.

#### Настройка прав доступа пользователей при внедрении подсистемы «Версионирование объектов»

При внедрении подсистемы «Версионирование объектов» она используется для регистрации конфликтов и запретов загрузки данных по установленной дате. Для их просмотра необходимо назначить роль **ЧтениеИнформацииОВерсияхОбъектов**.

#### Настройка прав доступа пользователей при работе в модели сервиса

При работе конфигурации в модели сервиса пользователям областей данных необходимо назначить роль **ВыполнениеСинхронизацииДанных**. Настройка обменов данными выполняется администратором абонента в контексте конфигурации «Менеджер сервиса».

## Использование при разработке конфигурации

В процессе обмена данными при чтении файла сообщения обмена важно не выполнять дополнительных проверок на заполнение реквизитов объектов. Для этого в обработчиках событий **ПередЗаписью**, **ПриЗаписи** и **ПередУдалением** для каждого объекта метаданных, который участвует в обмене, следует использовать конструкцию:

Если ОбменДанными.Загрузка Тогда

   Возврат;

КонецЕсли;

Этот код следует располагать в самом начале обработчиков **ПередЗаписью**, **ПриЗаписи** и **ПередУдалением** в модуле объекта, а также в подписках на события этих типов.

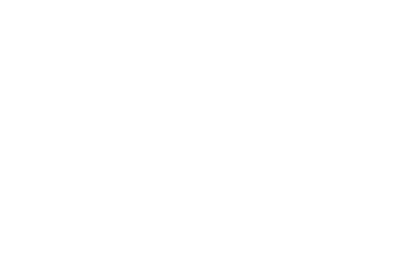
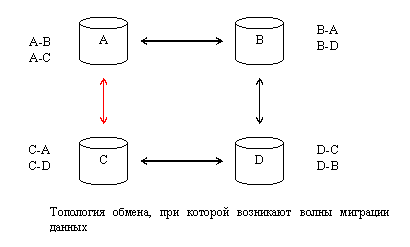
### Настройка обмена данными

Объекты метаданных подсистемы должны быть включены в состав планов обмена в соответствии с описанием в разделе «[Состав плана обмена](https://its.1c.ru/db/content/bsp23doc/src/3.28.%20%D0%BE%D0%B1%D0%BC%D0%B5%D0%BD%20%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D0%BC%D0%B8.htm?_=1504708101#_состав_плана_обмена)».

### Общие сведения для всех видов обмена данными

#### Топология обмена данными

Подсистема не накладывает ограничений на топологию обмена и на порядок настройки обмена в гетерогенных системах. Следует создавать обмены таким образом, чтобы избежать возникновения волн миграции данных. Волны миграции данных – это явление, при котором наблюдается бесконечная миграция изменений данных между информационными базами, участвующими в обмене. Они возникают в случае неправильной настройки топологии обмена (в разрезе информационных баз, планов обмена, объектов метаданных). Волны миграции данных имеют тенденцию к постоянному увеличению при работе пользователей в обменивающихся информационных базах.



На рисунке показаны четыре ИБ, между которыми настроен обмен. Каждая ИБ имеет свой уникальный префикс (код): A, B, C и D. Благодаря этому может быть настроена произвольная топология обмена. На рисунке показаны сочетания пар предопределенного узла и узла ИБ-корреспондента для каждой ИБ: A-B, A-C и др. Красным цветом показан обмен, который может спровоцировать возникновение волн миграции данных. При создании топологии обмена следует избегать таких обменов.

#### Использование значений по умолчанию в обмене данными

В процессе обмена данными может возникнуть необходимость дополнить данные, полученные от корреспондента. Например, документ не может быть проведен, пока в нем не указана валюта взаиморасчетов. В исходных данных, полученных от корреспондента, валюта не предусмотрена. Поэтому значение валюты необходимо подставить в документ на этапе загрузки данных. Это значение можно, например, запомнить в свойствах узла плана обмена. На этапе настройки обмена можно будет задать необходимые значения. Для выбора значений по умолчанию в помощнике используются формы **ФормаНастройкиЗначенийПоУмолчанию** и **ФормаНастройкиЗначенийПоУмолчаниюБазыКорреспондента**. В случае обмена по правилам использование значений по умолчанию производится в обработчиках правил обмена – например, в обработчике правила конвертации объекта «После загрузки».

Желательно проектировать обмен данными таким образом, чтобы при настройке обмена от пользователя не требовалось указывать эти значения. Все необходимые настройки должны устанавливаться системой или вообще отсутствовать. Только в исключительных случаях допускается выводить настройки значений по умолчанию на пользователя. Иными словами, в правильно спроектированном обмене данными формы **ФормаНастройкиЗначенийПоУмолчанию** и **ФормаНастройкиЗначенийПоУмолчаниюБазыКорреспондента** должны отсутствовать.

#### Коды узлов плана обмена

В локальном режиме работы коды узлов планов обмена совпадают с префиксами информационных баз, участвующих в обмене. Предопределенному узлу плана обмена назначается код, равный префиксу текущей информационной базы. Коды узлов назначаются системой в момент создания новой настройки обмена данными.

В модели сервиса коды узлов планов обмена соответствуют номерам областей данных, которым они принадлежат, с префиксом «S». Префикс «S» используется для разделения множества кодов узлов, работающих в модели сервиса, и множества кодов узлов, работающих в локальном режиме. Предопределенному узлу плана обмена назначается код, равный номеру текущей области данных, с префиксом «S». Код предопределенного узла назначается системой в момент создания области данных.

#### Настройка каталогов обмена данными для работы в модели сервиса

Для информационной базы, которая работает в модели сервиса, необходимо задать значение константы **КаталогСообщенийОбменаДаннымиДляWindows**или значение константы **КаталогСообщенийОбменаДаннымиДляLinux**, если сервер «1С:Предприятия» работает под управлением ОС MS Windows или ОС Linux соответственно. Если кластер сервера «1С:Предприятия» развернут на нескольких машинах с разными ОС, то необходимо задать значение обеих констант. В этом случае значения констант должны указывать на один и тот же сетевой каталог.

**Важно!**

Если две информационные базы, между которыми настроен обмен данными, расположены в одной локальной сети или на одной машине, то рекомендуется, чтобы константы **КаталогСообщенийОбменаДаннымиДляWindows**и **КаталогСообщенийОбменаДаннымиДляLinux**в обеих базах указывали на один и тот же каталог обмена. Это позволит повысить производительность обмена данными.

Если кластер сервера «1С:Предприятия» работает на одном физическом сервере, то задание значений констант не является обязательным. В качестве каталога временных файлов для сообщений обмена будет использоваться временный каталог пользователя, от имени которого запущен сервер «1С:Предприятия».

#### Обновление правил обмена

В процессе разработки конфигурации в нее могут быть внесены изменения, которые потребуют изменения правил обмена (только для **УОП**) и правил регистрации. Кроме того, правила обмена и правила регистрации могут быть изменены ввиду исправления ошибок в самих правилах. Правила следует также обновлять при изменении внутреннего формата правил. Для обновления правил необходимо выполнить обновление соответствующих макетов правил и обработок в конфигураторе. При изменении версии конфигурации обновление правил будет выполнено автоматически подсистемой обновления версии ИБ. Обновлению подлежат только типовые правила, загруженные из макета конфигурации. Нетиповые правила, загруженные из внешнего файла, не обновляются.

**Важно!**

Правила обмена загружаются в информационную базу. Поэтому при разработке правил обмена без изменения версии конфигурации автоматического обновления правил не произойдет. В таком случае необходимо выполнять обновление правил вручную. Для этого следует зайти в форму загрузки правил (команда **Открыть правила конвертации объектов** или **Открыть правила регистрации объектов**) и перечитать правила из макета конфигурации или из внешнего файла.

#### Сценарий выгрузки объектов «При необходимости»

Для универсального обмена данными по правилам обмена и для обмена через универсальный формат подсистема позволяет использовать сценарий обмена **При необходимости**. Суть сценария заключается в том, что объекты выгружаются не всегда, а только в случае, если на объект имеются ссылки из ранее выгруженных объектов. Например, справочник **Номенклатура** может содержать значительное количество элементов, при этом обмениваться необходимо не всеми элементами, а только теми, которые выгружаются вместе с документами прихода/расхода номенклатуры. Если элемент номенклатуры был выгружен хотя бы один раз, то изменения элемента постоянно синхронизируются между обменивающимися конфигурациями.

Для использования сценария на примере справочника номенклатуры в системе необходимо:

* ● создать в плане обмена отдельный реквизит шапки – переключатель режима выгрузки с именем **РежимВыгрузкиНоменклатуры**, тип **ПеречислениеСсылка.РежимыВыгрузкиОбъектовОбмена**;
* ● значение реквизита **РежимВыгрузкиНоменклатуры** установить в значение **Перечисления.РежимыВыгрузкиОбъектовОбмена.ВыгружатьПриНеобходимости**;
* ● создать правило регистрации объекта (ПРО) для справочника номенклатуры, в котором дополнительно задать значение поля переключателя режима равным **РежимВыгрузкиНоменклатуры**.

Для одного переключателя режима выгрузки **РежимВыгрузкиНоменклатуры** можно использовать несколько объектов метаданных. Например, при выгрузке номенклатуры также можно выгружать единицы измерения номенклатуры. Единицы измерения номенклатуры в этом случае будут тоже выгружаться только при необходимости.

#### Обработчик события «Регистрация изменений начальной выгрузки данных»

Для универсального обмена данными с использованием правил обмена, обмена через универсальный формат и для универсального обмена данными без правил обмена предусмотрен обработчик события «Регистрация изменений начальной выгрузки данных». Он не используется для обменов в РИБ.

Обработчик располагается в общем модуле **ОбменДаннымиПереопределяемый** и используется для переопределения стандартной обработки регистрации изменений начальной выгрузки данных. При стандартной обработке будут зарегистрированы изменения всех данных из состава плана обмена. Если для плана обмена предусмотрены фильтры ограничения миграции данных, то использование этого обработчика позволит повысить производительность начальной выгрузки данных (в некоторых случаях в среднем в 2–4 раза по сравнению со стандартной обработкой).

В обработчике можно реализовать регистрацию изменений с учетом фильтров ограничения миграции данных. Если для плана обмена используются ограничения миграции по дате или по дате и организациям, то можно воспользоваться универсальной процедурой **ОбменДаннымиСевер.ЗарегистрироватьДанныеПоДатеНачалаВыгрузкиИОрганизациям**. Пример реализации обработчика можно посмотреть в демонстрационной конфигурации.

#### Обработчик события «При выгрузке данных» для планов обмена ОУФ, УОП и УО

Обработчик располагается в общем модуле **ОбменДаннымиПереопределяемый** и используется для переопределения стандартной процедуры выгрузки данных в файл сообщения обмена. При стандартной обработке данные будут выгружены в файл в соответствии с видом плана обмена – с использованием правил конвертации или с использованием платформенной XML-сериализации данных. В данном обработчике может быть реализована произвольная логика выгрузки данных: выборка данных для выгрузки, сериализация данных в файл сообщения или сериализация данных в поток. После выполнения обработчика выгруженные данные будут отправлены получателю подсистемой обмена данными. Формат сообщения для выгрузки может быть произвольным.

#### Обработчик события «При загрузке данных» для планов обмена ОУФ, УОП и УО

Обработчик располагается в общем модуле **ОбменДаннымиПереопределяемый** и используется для переопределения стандартной процедуры загрузки данных из файла сообщения обмена. При стандартной обработке данные будут загружены из файла в соответствии с видом плана обмена – с использованием правил конвертации или с использованием платформенной XML-сериализации данных. В данном обработчике может быть реализована произвольная логика загрузки данных: необходимые проверки перед загрузкой данных, сериализация данных из файла сообщения или сериализация данных из потока. Формат сообщения для загрузки может быть произвольным.

#### Общие данные узлов

Для планов обмена двух обменивающихся информационных баз могут быть выделены так называемые общие данные узлов. Общие данные узлов – это реквизиты, которые присутствуют в планах обмена обеих конфигураций и значения которых должны быть одинаковыми в узлах обмена информационных баз-корреспондентов.

В качестве примера использования механизма общих данных узлов можно рассмотреть задачу разработки обмена с настройкой **Дата начала выгрузки документов**. Если в обеих конфигурациях в модуле менеджера плана обмена реквизит, управляющей данной настройкой, будет объявлен как общие данные узлов, то автоматически будет поддерживаться синхронность данной настройки в информационных базах-корреспондентах.

Не рекомендуется включать в общие данные узлов реквизиты ссылочного типа, если объекты данных, соответствующие этим типам, могут иметь разные уникальные идентификаторы в информационных базах, участвующих в обмене.

#### Коллизии изменения данных

При настроенном обмене между двумя базами возникают ситуации, когда одни и те же данные изменяются одновременно (в промежутке между выполнением сеанса обмена данными) в обеих информационных базах. Это приводит к возникновению двух различающихся версий одинаковых данных в обменивающихся информационных базах. Такая ситуация называется коллизией (конфликтом) изменения данных.

При возникновении коллизии изменения данных в информационную базу будет загружена та версия, которая имеет более высокий приоритет. При одинаковом приоритете обеих версий приоритет будет назначен автоматически, исходя из значений по умолчанию в зависимости от вида обмена данными:

* ● РИБ – изменения данных в главном узле имеют приоритет по отношению к изменениям данных в подчиненном узле;
* ● ОУФ, УО, УОП – приоритет имеют данные, загружаемые из другой информационной базы.

Разработчик имеет возможность переопределить приоритеты для возникающих коллизий при помощи процедуры **ОбменДаннымиПереопределяемый.ПриКоллизииИзмененийДанных** (подробнее см. комментарий к процедуре).

Следует учитывать, что для однозначного определения разрешения коллизии необходимо устанавливать приоритеты при возникновении коллизий изменения данных таким образом, чтобы они были противоположными для двух обменивающихся информационных баз. То есть если в первой информационной базе в случае коллизии данные принимаются, то во второй информационной базе при коллизии они должны быть отклонены.

### РИБ

#### Особенности создания начального образа подчиненного узла распределенной ИБ

Если объект метаданных нужно использовать только на момент создания начального образа подчиненного узла в распределенной ИБ, то этот объект метаданных требуется включить в состав плана обмена с отключенным признаком авторегистрации. Также необходимо исключить его из состава всех подписок на события подсистемы «Обмен данными».

Например, справочник номенклатуры не должен синхронизироваться между узлами РИБ. В каждом узле справочник номенклатуры специфичен и ведется независимо от других узлов. Однако вновь создаваемый подчиненный узел должен содержать заполненный по умолчанию справочник номенклатуры (обязательная структура папок и элементов).

### Обмен через универсальный формат

Процедуры и функции, полностью описывающие правила выгрузки данных из информационной базы в формат обмена и правила загрузки данных из формата обмена в информационную базу, разрабатываются в общем модуле – модуле менеджера обмена через универсальный формат.

Модуль создается автоматически с помощью конфигурации «Конвертация данных», редакция 3.0, на основе настроенных правил обмена либо вручную в конфигураторе.

При описании структуры модуля используются следующие сокращения:

* ● **ПОД** – правило обработки данных;
* ● **ПКО** – правило конвертации объекта;
* ● **ПКПД** – правило конвертации предопределенных данных;
* ● **ПКС** – правило конвертации свойства.

Модуль состоит из нескольких крупных разделов, каждый из которых содержит свою группу процедур и функций.

Комментарий. Первая строка модуля содержит комментарий с наименованием конвертации. Эта строка необходима для идентификации модуля при использовании команды **Загрузка обработчиков** в программе «Конвертация данных», редакция 3.0.

Процедуры конвертации. Содержит предопределенные процедуры, которые выполняются на разных этапах синхронизации данных: перед конвертацией, после конвертации, перед отложенным заполнением.

Правила обработки данных (ПОД). Содержит процедуры и функции, которые описывают правила обработки данных.

Правила конвертации объектов (ПКО). Содержит процедуры и функции, которые описывают правила конвертации объектов, а также правила конвертации свойств данных объектов.

Правила конвертации предопределенных данных (ПКПД). Содержит процедуру, заполняющую правила конвертации предопределенных данных.

Алгоритмы. Содержит произвольные алгоритмы, которые вызываются из других правил (ПОД или ПКО).

Параметры. Содержит логику заполнения параметров конвертации.

Общего назначения. Содержит процедуры и функции, которые широко используются в правилах и алгоритмах.

Ниже описаны параметры процедур и функций, которые используются в нескольких видах процедур модуля менеджера.

КомпонентыОбмена. **Тип – Структура**. Содержит параметры и правила обмена, инициализированные в рамках выполнения сеанса обмена.

НаправлениеОбмена. **Тип – Строка**. Либо «Отправка», либо «Получение».

ДанныеИБ. **Тип – СправочникОбъект** либо **ДокументОбъект**.

#### Процедуры, связанные с событиями конвертации

Предусмотрены три предопределенные процедуры, которые вызываются в процессе конвертации:

* ● **ПередКонвертацией**. Вызывается перед выполнением синхронизации данных. Обычно в этой процедуре размещается логика инициализации различных параметров конвертации, заполнения значений по умолчанию и т. д. Параметры: **КомпонентыОбмена**.
* ● **ПослеКонвертации**. Вызывается после выполнения синхронизации данных, но до выполнения отложенного заполнения. Параметры: **КомпонентыОбмена**.
* ● **ПередОтложеннымЗаполнением**. Вызывается перед выполнением отложенного заполнения. Здесь может быть расположена логика сортировки или корректировки таблицы объектов, подлежащих отложенному заполнению. Параметры: **КомпонентыОбмена**.

#### Процедуры ПОД

ЗаполнитьПравилаОбработкиДанных. Экспортная процедура, в которой располагается логика заполнения правил обработки данных. Содержит вызовы других процедур, которые добавляют в таблицу правил правило обработки конкретного объекта (см. ниже процедуры **ДобавитьПОД**). Параметры: **НаправлениеОбмена**,**ПравилаОбработкиДанных** (таблица значений, инициализированная в рамках выполнения сеанса обмена).

ДобавитьПОД\_<ИмяПОД>. Набор процедур, которые наполняют таблицу ПОД правилами для конкретных объектов. Количество таких процедур соответствует количеству ПОД, предусмотренных для данной конвертации в программе «Конвертация данных», редакция 3.0. Параметры: **ПравилаОбработкиДанных** (таблица значений, инициализированная в рамках выполнения сеанса обмена).

ПОД\_<ИмяПОД>\_ПриОбработке. Процедура содержит текст обработчика **ПриОбработке** для конкретного ПОД. Обработчик предназначен для реализации логики конвертации на уровне объектов. Например, назначить конкретному объекту определенное ПКО в зависимости от содержимого объекта. Параметры:

* ● **ДанныеИБ** либо **ДанныеXDTO** (в зависимости от направления обмена):
* ● при отправке – объект (**СправочникОбъект**, **ДокументОбъект**);
* ● при получении – структуру с описанием объекта XDTO.
* ● **ИспользованиеПКО**. Тип – **Структура**. Ключ содержит строку с именем ПКО, а значение типа **Булево**(**Истина** – ПКО используется, **Ложь** – ПКО не используется).
* ● **КомпонентыОбмена**.

ПОД\_<ИмяПОД>\_ВыборкаДанных. Функция содержит текст обработчика **ПриВыгрузке**. Обработчик предназначен для реализации произвольного алгоритма выборки объектов, подлежащих выгрузке. Возвращаемое значение: массив объектов, подлежащих выгрузке. В массиве могут содержаться как ссылки на объекты информационной базы, так и структура с данными для выгрузки. Параметры: **КомпонентыОбмена**.

#### Процедуры ПКО

ЗаполнитьПравилаКонвертацииОбъектов. Экспортная процедура, в которой располагается логика заполнения правил конвертации объектов. Содержит вызовы других процедур, которые добавляют в таблицу правил правило конвертации конкретного объекта (см. ниже процедуры **ДобавитьПКО**). Параметры: **НаправлениеОбмена**,**ПравилаКонвертации** (таблица значений, инициализированная в рамках выполнения сеанса обмена).

ДобавитьПКО\_<ИмяПКО>. Набор процедур, которые наполняют таблицу ПКО правилами для конкретных объектов. Количество таких процедур соответствует количеству ПКО, предусмотренных для данной конвертации в программе «Конвертация данных», редакция 3.0. Параметры: **ПравилаКонвертации** (таблица значений, инициализированная в рамках выполнения сеанса обмена).

ПКО\_<ИмяПКО>\_ПриОтправкеДанных. Процедура содержит текст обработчика **ПриОтправке** для конкретного ПКО. Обработчик используется при выгрузке данных. Предназначен для реализации логики конвертации данных, содержащихся в объекте информационной базы, в описание объекта XDTO. Параметры:

* ● **ДанныеИБ**. Тип – **СправочникОбъект**, **ДокументОбъект**. Обрабатываемый объект информационной базы.
* ● **ДанныеXDTO**. Тип – **Структура**. Предназначен для доступа к данным объекта XDTO.
* ● **КомпонентыОбмена**.
* ● **СтекВыгрузки**. Тип – **Массив**. Содержит ссылки на выгружаемые объекты с учетом вложенности.

ПКО\_<ИмяПКО>\_ПриКонвертацииДанныхXDTO. Процедура содержит текст обработчика **ПриКонвертацииДанныхXDTO** для конкретного ПКО. Обработчик используется при загрузке данных. Предназначен для реализации произвольной логики конвертации данных XDTO. Параметры:

* ● **ДанныеXDTO**. Тип – **Структура**. Свойства объекта XDTO, прошедшие предварительную обработку для упрощения доступа к ним.
* ● **ПолученныеДанные**. Тип – **СправочникОбъект**, **ДокументОбъект**. Объект информационной базы, сформированный путем конвертации данных XDTO. Не записан в информационную базу.
* ● **КомпонентыОбмена**.

ПКО\_<ИмяПКО>\_ПередЗаписьюПолученныхДанных. Процедура содержит текст обработчика **ПередЗаписьюПолученныхДанных** для конкретного ПКО. Обработчик используется при загрузке данных. Предназначена для реализации дополнительной логики, которую необходимо выполнить перед записью объекта в информационную базу. Например, нужно ли загрузить изменения в существующие данные ИБ либо следует загрузить их как новые данные. Параметры:

* ● **ПолученныеДанные**. Тип – **СправочникОбъект**, **ДокументОбъект**. Элемент данных, сформированный путем конвертации данных XDTO.

Записывается в случае, если эти данные являются для информационной базы новыми (параметр **ДанныеИБ**содержит значение **Неопределено**).

В противном случае **ПолученныеДанные** замещают собой **ДанныеИБ** (все свойства из **ПолученныеДанные**переносятся в **ДанныеИБ**).

Если стандартное замещение данных ИБ полученными данными не требуется, следует прописать свою логику переноса, после чего установить параметру **ПолученныеДанные** значение **Неопределено**:

* ● **ДанныеИБ**. Тип – **СправочникОбъект**, **ДокументОбъект**. Элемент данных информационной базы, соответствующий полученным данным. Если соответствующие данные не найдены, содержит **Неопределено**.
* ● **КонвертацияСвойств**. Тип – **Таблица значений**. Содержит правила конвертации свойств текущего объекта, инициализированные в рамках выполнения сеанса обмена.
* ● **КомпонентыОбмена**.

#### Процедуры ПКПД

**ЗаполнитьПравилаКонвертацииПредопределенныхДанных**. Экспортная процедура, в которой располагается логика заполнения правил конвертации предопределенных данных. Параметры: **НаправлениеОбмена**,**ПравилаКонвертации** (таблица значений, инициализированная в рамках выполнения сеанса обмена).

#### Алгоритмы

В программе «Конвертация данных», редакция 3.0 есть возможность создавать произвольные алгоритмы, которые вызываются из обработчиков ПОД и ПКПД. Наименование, параметры и содержимое алгоритмов определяются при разработке правил.

#### Параметры

ЗаполнитьПараметрыКонвертации. Экспортная процедура, в которой происходит заполнение структуры с параметрами конвертации. Параметры: **ПараметрыКонвертации** (тип – **Структура**).

#### Процедуры и функции общего назначения

ВыполнитьПроцедуруМодуляМенеджера. Параметры: **ИмяПроцедуры** (строка), **Параметры** (структура). Экспортная процедура, которая предназначена для вызова неэкспортной процедуры модуля, имя и параметры которой получены на вход. Позволяет выполнить вызов процедуры или функции по строке без использования метода **Выполнить**.

ВыполнитьФункциюМодуляМенеджера. Параметры: **ИмяПроцедуры** (строка), **Параметры** (структура). Функция, назначение аналогично **ВыполнитьПроцедуруМодуляМенеджера**. Отличие в том, что она вызывает функцию и возвращает ее значение.

### Универсальный обмен по правилам

#### Безопасное выполнение кода обработчиков при использовании обработки «УниверсальныйОбменДаннымиXML»

Для конфигураций, обменивающихся при помощи обработки **УниверсальныйОбменДаннымиXML** (УОД), можно повысить уровень безопасности путем выполнения кода обработчиков загрузки из обработки в составе конфигурации. Для того чтобы сгенерировать модуль с кодом обработчиков загрузки, необходимо:

1. Запустить базу-источник и сформировать файл выгрузки на основании правил обмена.

2. Запустить базу-приемник, запустить в ней УОД и на основании файла выгрузки сформировать текст модуля обработчиков загрузки.

3. Скопировать полученный текст модуля в обработку в составе конфигурации.

4. Подключить полученную обработку при помощи конструкции вида:

УОД = Обработки.УниверсальныйОбменДаннымиXML.Создать();

УОД.ФлагРежимОтладкиОбработчиков = Истина;

УОД.ИмяФайлаВнешнейОбработкиОбработчиковСобытий = «<ИмяОбработки>»;

УОД.ВыполнитьЗагрузку();

#### Режим отладки

Режим отладки позволяет разрабатывать и выполнять отладку кода обработчиков с использованием конфигуратора в случае обмена по правилам конвертации. В этом режиме код обработчиков выгрузки и/или загрузки выполняется из внешних обработок, что позволяет вносить в него изменения без перезапуска конфигурации. Для использования данного режима необходимо:

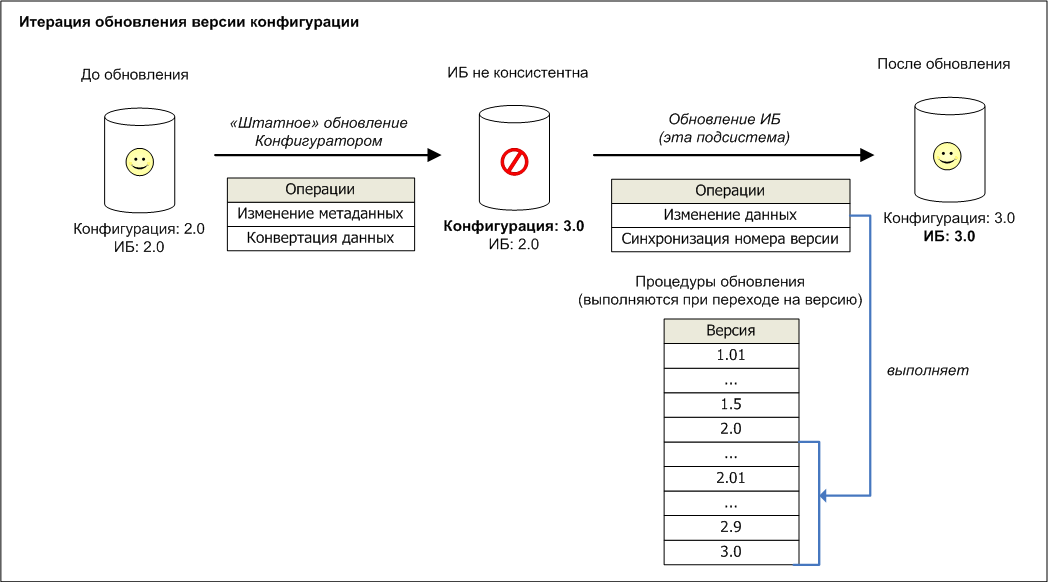
* ● В инструменте «Конвертация данных» версии 2.1.6 и выше:
* ● Отключить режим совместимости в свойствах конвертации.
* ● Сохранить правила конвертации объектов.
* ● Сформировать отладочные модули, содержащие код обработчиков, и скопировать их в модуль внешней обработки.
* ● В конфигурации:
* ● Загрузить правила конвертации, сохраненные в КД.
* ● В настройках правил конвертации включить режим отладки.
* ● Подключить внешние обработки, сформированные в КД.
* ● Запустить обмен данными.
* ● При этом в режиме **Конфигуратор** можно открыть подключенную внешнюю обработку, отлаживать и дорабатывать ее код.
* ● Имеется возможность перенести все сделанные в обработке изменения в правила обмена, используя инструмент «Конвертация данных» версии 2.1.6 и выше.

Режим отладки недоступен при работе в модели сервиса по соображениям безопасности.

# 3.29. Обновление версии ИБ

Подсистема «Обновление версии ИБ» предоставляет программный интерфейс для выполнения процедур-обработчиков по первоначальному заполнению и обновлению данных информационной базы (ИБ) при изменении версии конфигурации, а также позволяет выводить отчет об изменениях в новой версии конфигурации.

Принципиальная схема обновления конфигурации информационной базы представлена на рисунке.



При запуске клиентского приложения подсистема «Обновление версии ИБ» проверяет, не изменилась ли конфигурация. Если версия конфигурации отличается от версии, сохраненной в информационной базе, то при запуске системы с административными правами выполняется обновление информационной базы. В этом случае, подсистема «Обновление версии ИБ» последовательно выполняет процедуры – обработчики обновления в интервале от номера версии информационной базы до номера версии конфигурации включительно и записывает в информационную базу текущую версию конфигурации.

После обновления информационной базы подсистема «Обновление версии ИБ» выводит форму **Описание обновлений**, в которой администратор может ознакомиться с описанием изменений в текущей версии конфигурации.

Если у пользователя недостаточно прав (нет прав **МонопольныйРежим** и **Администрирование**) или при обновлении происходит ошибка, то запуск системы останавливается.

Если в конфигурации предусмотрены отложенные процедуры – обработчики обновления, то в клиент-серверной версии они запускаются в фоне с помощью регламентного задания **Отложенное обновление ИБ**, параллельно с началом работы пользователей с новой версией программы. В файловом режиме работы отложенные обработчики выполняются сразу, в основном цикле обновления.

## Настройка

### Подготовка к использованию подсистемы

Для библиотеки или конечной конфигурации создать отдельный общий модуль с именем **ОбновлениеИнформационнойБазы<Сокращение>**, где сокращение – короткое имя библиотеки или конфигурации. Пример: общий модуль **ОбновлениеИнформационнойБазыБСП**.

Затем добавить имя созданного модуля в процедуру **ПриДобавленииПодсистем** общего модуля **ПодсистемыКонфигурацииПереопределяемый**.

Из общего модуля **ОбновлениеИнформационнойБазыБСП** скопировать определения следующих процедур (очистив их содержимое) в созданный общий модуль:

Процедура ПриДобавленииПодсистемы(Описание) Экспорт

КонецПроцедуры

Процедура ПриДобавленииОбработчиковОбновления(Обработчики) Экспорт

КонецПроцедуры

Процедура ПередОбновлениемИнформационнойБазы() Экспорт

КонецПроцедуры

Процедура ПослеОбновленияИнформационнойБазы(Знач ПредыдущаяВерсия, Знач ТекущаяВерсия,

Знач ВыполненныеОбработчики, ВыводитьОписаниеОбновлений, МонопольныйРежим) Экспорт

КонецПроцедуры

Процедура ПриПодготовкеМакетаОписанияОбновлений(Знач Макет) Экспорт

КонецПроцедуры

Для конечных конфигураций также добавить три процедуры:

* ● **ПриДобавленииОбработчиковПереходаСДругойПрограммы**,
* ● **ПриОпределенииРежимаОбновленияДанных**,
* ● **ПриЗавершенииПереходаСДругойПрограммы**:

Процедура ПриДобавленииОбработчиковПереходаСДругойПрограммы(Обработчики) Экспорт

КонецПроцедуры

Процедура ПриОпределенииРежимаОбновленияДанных(РежимОбновленияДанных, СтандартнаяОбработка) Экспорт

КонецПроцедуры

Процедура ПриЗавершенииПереходаСДругойПрограммы(Знач ПредыдущееИмяКонфигурации, Знач ПредыдущаяВерсияКонфигурации, Параметры) Экспорт

КонецПроцедуры

Пример реализации см. в общем модуле **\_ДемоОбновлениеИнформационнойБазыБСП** демонстрационной конфигурации.

Затем в процедуру **ПриДобавленииПодсистемы** вписать имя и версию библиотеки или конфигурации, а также зависимости от других библиотек (если предусмотрены). По указанным зависимостям вычисляется порядок вызова обработчиков обновления данных библиотек. Если зависимости не указаны, порядок вызова обработчиков библиотек определяется порядком добавления имен модулей библиотек в общем модуле **ПодсистемыКонфигурацииПереопределяемый**. При этом библиотека с именем **СтандартныеПодсистемы**будет вызываться первой, а библиотека у которой имя совпадает со значением свойства **Метаданные.Имя** всегда будет вызываться последней.

// См. описание этой же процедуры в модуле ОбновлениеИнформационнойБазыБСП.

Процедура ПриДобавленииПодсистемы(Описание) Экспорт

Описание.Имя = "БиблиотекаСтандартныхПодсистемДемо";

Описание.Версия = "2.1.3.24";

// Требуется библиотека стандартных подсистем.

Описание.ТребуемыеПодсистемы.Добавить("СтандартныеПодсистемы");

КонецПроцедуры

При этом в коде процедуры **ПриДобавленииПодсистемы** не следует получать имя и версию напрямую из свойств конфигурации **Метаданные.Имя** и **Метаданные.Версия**. В противном случае при доработке конфигураций потребуется снимать с поддержки и вносить изменения в модуль обновления поставщика.

### Размещение в командном интерфейсе

Если в конфигурации не используется подсистема «Настройки программы», то в командном интерфейсе администратора необходимо разместить следующие объекты метаданных:

* ● константа **ДетализироватьОбновлениеИБВЖурналеРегистрации**,
* ● общая форма **ОписаниеИзмененийПрограммы**,
* ● обработка **РезультатыОбновленияПрограммы**.

См. пример размещения в демонстрационной конфигурации в группе **Обновление версии программы** формы **ПоискИУстановкаОбновлений** обработки **ПанельАдминистрированияБСП**.

### Создать общий макет «ОписаниеИзмененийСистемы»

Описание изменений в новой версии готовится разработчиками конфигурации к выпуску каждой версии конфигурации в общем табличном макете **ОписаниеИзмененийСистемы**. Табличный макет может включать в себя столько разделов, сколько версий содержится в истории версий продукта.

Каждый такой раздел состоит из двух областей:

* ● В область **ШапкаР\_П\_В\_С** помещается текст заголовка **Новое в версии Р.П.В.С**.
* ● Область **ВерсияР\_П\_В\_С** содержит описание изменений данной версии; может состоять из подразделов или содержать гиперссылки.

Разделы между собой отделяются отступом, который определяется областью **Отступ**.

Для специальной обработки нажатий на гиперссылки, содержащихся в тексте макета, предназначена процедура-обработчик **ПриНажатииНаГиперссылкуВДокументеОписанияОбновлений** общего модуля **ОбновлениеИнформационнойБазыКлиентПереопределяемый**.

Пример заполнения макета можно посмотреть в демонстрационной конфигурации.

**Важно!**

При первом внедрении макет **ОписаниеИзмененийСистемы** следует создать до выполнения первого запуска конфигурации в режиме **1С:Предприятие**.

### Настройка прав доступа пользователей

Для настройки прав доступа пользователей к данным и функциям подсистемы «Обновление версии ИБ» следует использовать роли, указанные ниже.

Таблица 3.63.

| **№** | **Роли и их назначение** |
| --- | --- |
| 1. | **АдминистраторСистемы** (из подсистемы «Базовая функциональность»)  Запуск программы для обновления версии ИБ |
| 2. | **БазовыеПрава** (из подсистемы «Базовая функциональность»)  Просмотр описания изменения системы |

### Настройка обмена данными

Для настройки обмена данными следует руководствоваться общими правилами.

В планы обмена распределенной информационной базы (РИБ) рекомендуется включать все объекты метаданных подсистемы, кроме константы **РазделыОписанияИзмененийСистемы**.

В планах обмена распределенной информационной базы (РИБ) рекомендуется отключать регистрацию изменений для следующих объектов метаданных подсистемы (см. также раздел «[Особенности создания начального образа подчиненного узла распределенной ИБ](https://its.1c.ru/db/content/bsp23doc/src/3.28.%20%D0%BE%D0%B1%D0%BC%D0%B5%D0%BD%20%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D0%BC%D0%B8.htm?_=1504708101#_особенности_создания_начального)»):

* ● константа **ДетализироватьОбновлениеИБВЖурналеРегистрации**,
* ● константа **СведенияОбОбновленииИБ**,
* ● регистр сведений **ВерсииПодсистем**.

Из остальных планов обмена (не РИБ) рекомендуется исключать следующие объекты метаданных:

* ● константа **ДетализироватьОбновлениеИБВЖурналеРегистрации**,
* ● константа **РазделыОписанияИзмененийСистемы**,
* ● константа **СведенияОбОбновленииИБ**,
* ● регистр сведений **ВерсииПодсистем**.

## Использование при разработке конфигурации

### Разработка обработчиков обновления

Для подключения своих обработчиков обновления необходимо в модуль обновления информационной базы библиотеки или конфигурации, созданный на предыдущем шаге, добавить описание обработчиков обновления в процедуру **ПриДобавленииОбработчиковОбновления**.

Для каждого обработчика обновления нужно добавить фрагмент кода по шаблону:

Обработчик = Обработчики.Добавить();

Обработчик.Версия = "<номер версии>";

Обработчик.Процедура = "<полное имя экспортной процедуры>";

Обработчик.НачальноеЗаполнение = {Истина|Ложь};

Обработчик.РежимВыполнения = {"Монопольно"|"Оперативно"|"Отложенно"};

Строка таблицы значений **Обработчик** имеет различный состав полей, который зависит от вида обработчика обновления: монопольный, оперативный или отложенный. Свойства, общие для всех видов обработчиков обновления:

* ● **Версия** (**Строка**) – номер версии конфигурации, при обновлении на которую должна быть вызвана процедура обновления, указанная в параметре **Процедура**.
* ● номер версии конфигурации указывается в формате **Р.П.В.С** (Р – старший номер редакции; П – младший номер редакции; В – номер версии; С – номер сборки);
* ● если в качестве версии указан символ «**\***», то обработчик обновления должен выполняться каждый раз при обновлении информационной базы, независимо от номера версии конфигурации;
* ● если свойство **Версия** не задано, то должно быть установлено в **Истина** свойство **НачальноеЗаполнение**(см. далее).
* ● **Процедура** (**Строка**) – имя процедуры обновления. Процедура обновления должна располагаться в серверном общем модуле и должна быть экспортной. Например, **ОбновлениеИБ.ПерейтиНаВерсию\_1\_2\_3\_4**.
* ● **НачальноеЗаполнение** (**Булево**) – если **Истина**, то процедура обновления будет вызвана при первом запуске на пустой информационной базе (версия «0.0.0.0»), созданной из файла поставки конфигурации и не содержащей данных. Это обработчики первоначального заполнения базы. По умолчанию – **Ложь**.
* ● **РежимВыполнения** (**Строка**) – принимает одно из значений: **Монопольно**, **Оперативно** и **Отложенно**. Если свойство не задано, то обработчик – монопольный.

Например, для выполнения двух экспортных процедур **ВыполнятьВсегдаПриСменеВерсии** и **ПерейтиНаВерсию\_1\_0\_0\_5** общего модуля **\_ДемоОбновлениеИнформационнойБазыБСП** при переходе с версии 1.0.0.1 на версию 1.0.0.5 необходимо разместить следующий фрагмент кода в процедуре **ПриДобавленииОбработчиковОбновления**:

Процедура ПриДобавленииОбработчиковОбновления(Обработчики) Экспорт

// Подключаются процедуры – обработчики обновления конфигурации.

Обработчик = Обработчики.Добавить();

Обработчик.Версия = "1.0.0.0";

Обработчик.Процедура = "\_ДемоОбновлениеИнформационнойБазыБСП.ПерейтиНаВерсию\_1\_0\_0\_0";

Обработчик = Обработчики.Добавить();

Обработчик.Версия = "1.0.0.5";

Обработчик.Процедура = "\_ДемоОбновлениеИнформационнойБазыБСП.ПерейтиНаВерсию\_1\_0\_0\_5";

Обработчик = Обработчики.Добавить();

Обработчик.Версия = "\*";

Обработчик.Процедура = "\_ДемоОбновлениеИнформационнойБазыБСП.ВыполнятьВсегдаПриСменеВерсии";

КонецПроцедуры

### Как выбрать способ обработки данных: монопольный, оперативный или отложенный

В зависимости от значения свойства **РежимВыполнения** можно выбрать наиболее эффективный способ обработки данных.

1. **Монопольно** – если обработчик обновления необходимо выполнять монопольно, в условиях отсутствия активных сеансов работы пользователей, регламентных заданий, внешних соединений и подключений по веб-сервисам. В противном случае обновление версии программы прерывается.

Монопольные обработчики предназначены для обновления тех данных, обработка которых должна быть обязательно завершена к моменту входа пользователей в программу. Для сокращения времени простоя (ожидания обработки данных), рекомендуется большие объемы данных обновлять отложено (см. ниже). Примеры монопольных обработчиков: обработка небольшого объема данных текущего периода, активных позиций номенклатуры и т. п.

Если хотя бы один обработчик обновления конфигурации – монопольный, то все оперативные обработчики (см. далее) выполняются в монопольном режиме.

2. **Оперативно** – если обработчик обновления необходимо выполнять не монопольно: при активных сеансах работы пользователей, регламентных заданий, внешних соединений и подключений через веб-сервисы.

Оперативные обработчики следует применять в редких случаях, когда важно сократить время ожидания пользователей на обновление информационной базы – например, для обработки неразделенных данных в модели сервиса, инициализации настроек пользователей и т. п. Как правило, их следует применять только при выпуске исправительных релизов.

Подробнее об оперативном обновлении см. раздел «[Оперативное обновление на исправительные релизы конфигураций](https://its.1c.ru/db/content/bsp23doc/src/3.29.%20%D0%BE%D0%B1%D0%BD%D0%BE%D0%B2%D0%BB%D0%B5%D0%BD%D0%B8%D0%B5%20%D0%B2%D0%B5%D1%80%D1%81%D0%B8%D0%B8%20%D0%B8%D0%B1.htm?_=1504708101#_оперативное_обновление_на)».

3. **Отложенно** – если обработчик обновления необходимо выполнять в фоне после того, как завершено выполнение монопольных (оперативных) обработчиков, и пользователям уже разрешен вход в программу.

Отложенные обработчики предназначены для обработки той части данных ИБ, которые не препятствуют пользователям начинать свою работу с новой версией программы, не дожидаясь завершения обработки этих данных. Примеры отложенных обработчиков: обработка больших архивов данных за закрытые/прошлые периоды, неактивных позиций номенклатуры, различных данных, отключенных в данный момент функциональными опциями и т. п.

Подробнее об отложенной обработке данных см. раздел «[Отложенное обновление больших архивов данных](https://its.1c.ru/db/content/bsp23doc/src/3.29.%20%D0%BE%D0%B1%D0%BD%D0%BE%D0%B2%D0%BB%D0%B5%D0%BD%D0%B8%D0%B5%20%D0%B2%D0%B5%D1%80%D1%81%D0%B8%D0%B8%20%D0%B8%D0%B1.htm?_=1504708101#_отложенное_обновление_больших)».

### Общие рекомендации по реализации обработчиков обновления

* ● Код обработчиков обновления выполняется на сервере, поэтому они не должны содержать никакой логики по интерактивному взаимодействию с пользователем.
* ● В случае критической ошибки при обновлении в коде обработчика необходимо вызвать исключение, которое приведет к остановке всей процедуры обновления. Остановка обновления информационной базы приведет к невозможности запуска до тех пор, пока причины ошибки не будут устранены.
* ● Код обработчика обновления должен быть рассчитан на неоднократное выполнение, чтобы его повторное выполнение не приводило, например, к дублированию данных в информационной базе.
* ● На одну и ту же версию конфигурации может быть написано сколь угодно много обработчиков. Например, это могут быть обработчики от разных подсистем. При этом порядок их вызова для одной версии может быть произвольным.
* ● Порядок вызова обработчиков обновления в пределах одной версии является случайным, т. е. нельзя ставить в зависимость работоспособность одного обработчика обновления от выполнения другого обработчика. Если подобные зависимости появляются, то такие обработчики необходимо объединять в один.

Кроме того, обработчик обновления не должен содержать лишних, избыточных действий с данными – должен выполняться максимально быстро. Для этого в большинстве случаев необходимо отключать бизнес-логику при обработке данных, а также отключать регистрацию изменений на узлах планов обмена, чтобы избежать отправки всего объема обработанных данных во всех узлы. Таким образом:

* ● в распределенной информационной базе (РИБ) обработка данных должна выполняться независимо в каждом из узлов;
* ● при обмене между произвольными конфигурациями (программами) обработка данных не должна приводить к их выгрузке в базы-получатели.

Исключение составляют случаи создания ссылочных объектов, которые должны быть перенесены механизмами обмена данными в другие узлы РИБ с тем же значением реквизита **Ссылка**.

Таким образом, в коде обработчика обновления вместо кода вида:

ДокументОбъект.Записать();

должно быть:

ДокументОбъект.ОбменДанными.Загрузка = Истина; // отключить всю бизнес-логику при записи

ДокументОбъект.ДополнительныеСвойства.Вставить("ОтключитьМеханизмРегистрацииОбъектов");

ДокументОбъект.ОбменДанными.Получатели.АвтоЗаполнение = Ложь;

ДокументОбъект.Записать();

Для сокращения объема кода рекомендуется использовать процедуру **ЗаписатьДанные** общего модуля **ОбновлениеИнформационнойБазы**:

ОбновлениеИнформационнойБазы.ЗаписатьДанные(ДокументОбъект);

Для удобной отладки монопольных и оперативных обработчиков обновления можно воспользоваться параметром запуска **РежимОтладки** (обновление будет выполнено без использования фоновых заданий) или флажком **Выполнять обновление версии ИБ без установки монопольного режима и без фонового задания (режим отладки)**в обработке **ИнструментыРазработчикаОбновлениеВерсииИБ**.

### Отложенное обновление больших архивов данных

В тех случаях, когда исчерпаны все остальные средства по оптимизации обработчиков обновления и можно выделить некоторые действия по обработке данных, выполнение которых не требуется обязательно для начала работы пользователей с программой, рекомендуется перенести эту обработку на более поздний момент времени и выполнять ее отложенно.

Отложенная обработка данных не блокирует вход пользователей в программу и позволяет избежать ситуаций, когда обновление больших баз занимает существенное время (сутки и более), что нарушает график работы компании (из-за большого времени простоя информационной системы).

**Важно!**

Отложенная обработка данных возможна только в клиент-серверном варианте работы. В файловом режиме работы отложенные обработчики обновления выполняются сразу, до начала работы пользователей с новой версией программы.

Рекомендуется реализовать отложенные обработчики обновления для обработки больших архивов данных за закрытые/прошлые периоды, неактивных позиций номенклатуры, закрытых договоров, различных данных, отключенных в данный момент функциональными опциями и т. п. В большинстве случаев отложенно следует обновлять документы, регистры, бизнес-процессы и задачи, которые имеют тенденцию накапливаться со временем.

Механизм отложенной обработки данных имеет два режима выполнения, которые настраиваются отдельно для каждой библиотеки и основной конфигурации:

* ● **Последовательно** (по умолчанию) – отложенные обработчики обновления выполняются последовательно в интервале от номера версии информационной базы до номера версии конфигурации включительно (по возрастанию номеров версий, которые указаны в обработчиках). До тех пор пока один из обработчиков не завершил обработку своей порции данных, следующий не запускается. Данный режим обновления подходит для конфигураций (и библиотек), в которых отложенные обработчики для новых версий обрабатывают те же данные, что обрабатывали обработчики более старых версий. Кроме того, за счет последовательного выполнения к ним предъявляются минимальные требования по «устойчивости» к обрабатываемым данным: при обновлении «через» несколько версий они гарантированно могут рассчитывать на определенное начальное состояние обрабатываемых данных, которое осталось после выполнения обработчиков предыдущей версии.
* ● **Параллельно** – отложенный обработчик после обработки первой порции данных передает управление следующему обработчику, а после выполнения последнего обработчика цикл повторяется заново, пока все данные не будут обработаны. Таким образом одновременно обновляются объекты информационной базы сразу всех типов, в отличие от последовательного режима, при котором объекты разных типов обрабатываются по очереди и многократно (при обновлении «через» несколько версий).

Разработка отложенных обработчиков обновления для обоих режимов существенно различается. Особенности разработки обработчиков для отложенного режима обновления описаны отдельно.

Для того чтобы указать, что обработчик обновления должен выполняться отложенно, необходимо свойству **РежимВыполнения** присвоить значение **Отложенно**, указать уникальный идентификатор и задать комментарий, который кратко поясняет пользователю, какие данные и как он обрабатывает.

* Свойства, специфичные для обоих видов отложенных обработчиков обновления:
* ● **Комментарий (Строка)**– комментарий заполняется обязательно и не должен совпадать с комментариями к другим обработчиками обновления. В нем рекомендуется описывать не только суть выполняемого действия, но и масштаб временно неработоспособного функционала, например:
* ● Подготовка индекса для поиска отчетов, предусмотренных в программе. Поиск отчетов временно недоступен.
* ● Реструктуризация дополнительных реквизитов и сведений. Рекомендуется воздержаться от их редактирования до завершения обработки.
* ● Первоначальный расчет количества нерассмотренных писем по папкам. До завершения обработки всех писем их количество может выводиться некорректно.
* ● Заполняются движения по новому регистру **Движения Номенклатура-Контрагент** по документам **Расчет себестоимости товаров**. После выполнения обработки появится возможность формировать отчеты по товарам.
* ● **Идентификатор (УникальныйИдентификатор)**– идентификатор отложенного обработчика, который необходимо заполнять для разрешения конфликтов при переименованиях или переносе в другой модуль процедуры обновления. В таких случаях по идентификатору будет определен новый путь к обработчику, и он успешно завершит обработку данных.
* ● **БлокируемыеОбъекты** (**Строка**) – полные имена объектов через запятую, которые следует блокировать в пользовательском интерфейсе от редактирования до завершения процедуры обработки данных. Подробнее см. [Блокировка необработанных данных в пользовательском интерфейсе](https://its.1c.ru/db/content/bsp23doc/src/3.29.%20%D0%BE%D0%B1%D0%BD%D0%BE%D0%B2%D0%BB%D0%B5%D0%BD%D0%B8%D0%B5%20%D0%B2%D0%B5%D1%80%D1%81%D0%B8%D0%B8%20%D0%B8%D0%B1.htm?_=1504708101#_блокировка_необработанных_данных).
* ● **ПроцедураПроверки** (**Строка**) – имя функции, которая дополнительно для переданного объекта определяет, завершена ли для него процедура обработки данных. Если переданный объект обработан, то следует вернуть значение **Истина**. Вызывается из процедуры **ПроверитьОбъектОбработан** общего модуля **ОбновлениеИнформационнойБазы**. Подробнее см. [Блокировка необработанных данных в пользовательском интерфейсе](https://its.1c.ru/db/content/bsp23doc/src/3.29.%20%D0%BE%D0%B1%D0%BD%D0%BE%D0%B2%D0%BB%D0%B5%D0%BD%D0%B8%D0%B5%20%D0%B2%D0%B5%D1%80%D1%81%D0%B8%D0%B8%20%D0%B8%D0%B1.htm?_=1504708101#_блокировка_еще_необработанных).

Например:

Обработчик = Обработчики.Добавить();

Обработчик.Версия = "1.2.3.4";

Обработчик.Процедура = "Заказы.ЗаполнитьСтатусЗаказовПокупателей";

Обработчик.РежимВыполнения = "Отложенно";

Обработчик.Идентификатор = Новый УникальныйИдентификатор("83d5c5dd-1462-4d72-ab98-f8f5dcc0664d");

Обработчик.Комментарий = НСтр("ru = 'Заполняет значение нового реквизита СтатусЗаказа у документов ""Заказ покупателя"" прошлых периодов.'");

Синтаксис процедуры-обработчика отложенного обновления:

Процедура ЗаполнитьСтатусыЗаказовПокупателей(Параметры) Экспорт

где **Параметры – Структура** со свойствами:

* ● **ОбработкаЗавершена** (**Булево**) – для того чтобы обработчик был вызван повторно для обработки следующей порции данных, следует записать в него значение **Ложь**;
* ● **ПрогрессВыполнения** (**Структура**) – необходимо заполнять для отображения прогресса обработки данных:
* ● **ВсегоОбъектов** (**Число**) – общее количество объектов, которое необходимо обработать;
* ● **ОбработаноОбъектов** (**Число**) – сколько объектов уже обработано.

Кроме того, в структуру **Параметры** можно добавить произвольное количество свойств произвольных типов, значения которых будут автоматически запоминаться между вызовами процедуры – обработчика отложенного обновления. Таким образом можно передавать контекст отложенной обработки между ее вызовами (например, дату, по которую был обработан архив документов). При этом не рекомендуется сохранять в свойствах структуры большие объемы данных, а использовать только примитивные типы (например, **Дата**).

Отложенную обработку данных необходимо выполнять порциями, чтобы не создавать длительную нагрузку на сервер предприятия и СУБД. По умолчанию размер порции – 1000 (документов, записей и т. п.). Размер порции можно увеличить для небольших объектов и уменьшить для документов, в которых большие табличные части (в среднем).

Также рекомендуется начинать обработку с самых свежих данных.

Например:

Запрос = Новый Запрос;

Запрос.Текст =

"ВЫБРАТЬ ПЕРВЫЕ 1000

| ЗаказПокупателя.Ссылка

|ИЗ

| Документ.ЗаказПокупателя КАК ЗаказПокупателя

|ГДЕ

| ЗаказПокупателя.СтатусЗаказа = &ПустаяСсылка

|

|УПОРЯДОЧИТЬ ПО

| ЗаказПокупателя.Дата УБЫВ";

Отложенные обработчики обновления должны самостоятельно заботиться о целостности обновляемых данных: при чтении данных с последующим изменением требуется выполнять эти действия в транзакции и устанавливать исключительную управляемую блокировку.

Например:

НачатьТранзакцию();

Попытка

Блокировка = Новый БлокировкаДанных;

ЭлементБлокировки = Блокировка.Добавить("Документ.\_ДемоЗаказПокупателя");

ЭлементБлокировки.УстановитьЗначение("Ссылка", ЗаказПокупателя.Ссылка);

Блокировка.Заблокировать();

ДокументОбъект = ЗаказПокупателя.Ссылка.ПолучитьОбъект();

// Если объект ранее был удален или обработан другими сеансами, пропускаем его.

Если ДокументОбъект = Неопределено Тогда

ОтменитьТранзакцию();

Возврат;

КонецЕсли;

Если ДокументОбъект.СтатусЗаказа = Перечисления.\_ДемоСтатусыЗаказовПокупателей.ПустаяСсылка() Тогда

ОтменитьТранзакцию();

Возврат;

КонецЕсли;

// Обрабатываем документ

// ...

ОбновлениеИнформационнойБазы.ЗаписатьДанные(ДокументОбъект);

ЗафиксироватьТранзакцию();

Исключение

ОтменитьТранзакцию();

ВызватьИсключение;

КонецПопытки;

При возникновении исключения в обработчике отложенного обновления следует всегда пробрасывать его в вызывающий код (см. блок **Исключение – КонецПопытки** в примере выше). В противном случае может возникнуть ситуация, когда обработчик будет выполняться бесконечно долго (если параметру **ОбработкаЗавершена** присвоено значение **Ложь**), а данные ИБ не будут обновлены.

При возникновении исключения в обработчике он помечается как ошибочный, и через определенный интервал времени выполняются еще две попытки его запуска. Тем самым обработчик сможет корректно обновить данные, временно заблокированные другими сеансами. Но если обработчик продолжает и дальше прерываться с исключением (допущена ошибка в самом обработчике), то он пропускается и ставится в очередь выполнения только при очередном обновлении версии программы. Это позволяет обновить данные при переходе на следующий исправительный релиз, в котором исправлена ошибка в обработчике.

В общем случае при обработке данных конкретной таблицы (документа, регистра и т. п.) некоторая часть ее данных требуется пользователям сразу к моменту начала работы в новой версии программы, а все остальное может быть обработано отложенно. В таких случаях рекомендуется реализовать два обработчика обновления: монопольный и отложенный.

В редких случаях, когда в новой версии конфигурации появился монопольный (или оперативный) обработчик обновления, данные которого обрабатывались в предыдущих версиях отложенно, следует:

* ● пересмотреть проектное решение и сделать такой обработчик отложенным,
* ● либо «старые» отложенные обработчики обновления сделать монопольными (оперативными).

В противном случае возникнет ситуация, когда данные будут обработаны в неправильном порядке: сначала выполнится монопольный (оперативный) обработчик, который рассчитывается на то, что данные были обработаны ранее отложенно.

### Блокировка необработанных данных в пользовательском интерфейсе

Если отложенный обработчик обрабатывает не только архивные данные, но и данные текущего периода, то для предотвращения некорректной работы программы рекомендуется блокировать еще не обработанные данные от изменения пользователями программы, а также отчеты и обработки, которые могут некорректно работать до завершения отложенного обновления этих данных.

Для этого необходимо:

* ● в свойстве **БлокируемыеОбъекты** отложенного обработчика обновления следует указать те объекты метаданных, с которыми он работает (читает или записывает), а также связанные с ними отчеты и обработки;
* ● в свойстве **ПроцедураПроверки** указать имя экспортной функции, которая дополнительно проверяет, нужно ли блокировать конкретный объект. Например:

Обработчик = Обработчики.Добавить();

Обработчик.Версия = "1.2.3.4";

Обработчик.Идентификатор = Новый УникальныйИдентификатор("b3be66c5-708d-42c8-a019-818036d09d06");

Обработчик.Процедура = "Заказы.ЗаполнитьСтатусыЗаказовПокупателей";

Обработчик.Комментарий = НСтр("ru = 'Заполнение значения нового реквизита ""Статус заказа"" у документов ""Демо: Заказ покупателя"" прошлых периодов.

|До завершения обработки ""Статус заказа"" данных документов будет отображаться некорректно.'");

Обработчик.РежимВыполнения = "Отложенно";

Обработчик.ПроцедураПроверки = "Заказы.ЗаказПокупателяОбработан";

Обработчик.БлокируемыеОбъекты = "Документ.ЗаказПокупателя,Отчет.СтатусыЗаказовПокупателей";

Затем для всех объектов, указанных в свойстве **БлокируемыеОбъекты**, необходимо добавить вызов процедуры **ОбновлениеИнформационнойБазы.ПроверитьОбъектОбработан**в обработчике события модуля формы объекта **ПриСозданииНаСервере** и в обработчике события модуля объекта **ПередЗаписью**:

Процедура ПриСозданииНаСервере(Отказ, СтандартнаяОбработка)

Если Параметры.Свойство("АвтоТест") Тогда // Возврат при получении формы для анализа.

Возврат;

КонецЕсли;

ОбновлениеИнформационнойБазы.ПроверитьОбъектОбработан(Объект, ЭтотОбъект);

…

КонецПроцедуры

Процедура ПередЗаписью(Отказ, РежимЗаписи, РежимПроведения)

Если ОбменДанными.Загрузка Тогда

Возврат;

КонецЕсли;

ОбновлениеИнформационнойБазы.ПроверитьОбъектОбработан(ЭтотОбъект);

…

КонецПроцедуры

Функция проверки, указанная в свойстве **ПроцедураПроверки**, возвращает **Истина** для объектов, если объект уже обработан и с ним могут работать пользователи, или **Ложь**, если объект следует заблокировать от редактирования пользователями. Она принимает на вход параметр типа **Структура**, свойства которого идентифицируют запрашиваемый объект:

* ● **Данные (ЛюбаяСсылка, НаборЗаписей, Объект, ДанныеФормыСтруктура)** – объект, который нужно проверить, обработан ли он данным отложенным обработчиком обновления или еще нет.
* ● **МетаданныеОбъекта (ОбъектМетаданных)** - объект метаданных, соответствующий параметру **Данные**.
* ● **ПолноеИмя (Строка)** - полное имя объекта метаданных.
* ● **Отбор (ЛюбаяСсылка, Структура)** – если **Данные** – это ссылочный объект, то значение ссылки; если регистр подчиненный регистратору – значение отбора по регистратору. Если **Данные** – это независимый регистр сведений, то в этом параметре предается структура, соответствующая установленным отборам по измерениям.
* ● **ЭтоНовый (Булево)** – если **Данные** – это ссылочный объект, то признак нового объекта. Для других типов – всегда **Ложь**.

Ее реализация должна быть достаточно простой, чтобы не сильно замедлять открытие формы, например:

Функция ЗаказПокупателяОбработан(Параметры) Экспорт

Если ТипЗнч(Параметры.Данные) = Тип("ДокументСсылка.\_ДемоЗаказПокупателя") Тогда

СтатусЗаказа = ОбщегоНазначения.ЗначениеРеквизитаОбъекта(Параметры.Данные, "СтатусЗаказа");

Иначе

СтатусЗаказа = Параметры.Данные.СтатусЗаказа;

КонецЕсли;

Возврат СтатусЗаказа <> Перечисления.\_ДемоСтатусыЗаказовПокупателей.ПустаяСсылка();

КонецФункции

Пример отложенного обработчика обновления с блокировкой необработанных данных в пользовательском интерфейсе см. в демонстрационной конфигурации: процедура **ПриДобавленииОбработчиковОбновления**общего модуля **\_ДемоОбновлениеИнформационнойБазыБСП**.

### Особенности параллельного режима отложенного обновления

Для установки параллельного режима выполнения отложенных обработчиков конфигурации (библиотеки) следует в процедуре **ПриДобавленииПодсистемы** общего модуля **ОбновлениеИнформационнойБазы<Сокращение>**указать:

Описание.РежимВыполненияОтложенныхОбработчиков = "Параллельно";

Данный режим обновления можно включить с определенной версии, чтобы не пересматривать реализацию старых отложенных обработчиков. Для этого необходимо заполнить свойство **ПараллельноеОтложенноеОбновлениеСВерсии**:

Описание.ПараллельноеОтложенноеОбновлениеСВерсии = "2.3.3.20";

Свойства, специфичные только для отложенных обработчиков обновления с режимом выполнения **Параллельно**:

* ● **ОчередьОтложеннойОбработки** (**Число**) – если обработчик обрабатывает данные совершенно независимо от других, то следует указывать значение очереди – «1». Если же при обработке данных обработчик опирается на результат выполнения другого обработчика, то ему следует указывать более высокое значение очереди. Это необходимо для оптимальной обработки данных, чтобы обработчик не отрабатывал вхолостую.
* ● **ПроцедураЗаполненияДанныхОбновления** (**Строка**) – указывается процедура, которая регистрирует данные, подлежащие обновлению данным обработчиком.
* ● **ЗапускатьТолькоВГлавномУзле** (**Булево**) – указать **Истина**, если обработчик обновления может выполняться только в главном узле РИБ. По умолчанию **Ложь**.
* ● **ЗапускатьИВПодчиненномУзлеРИБСФильтрами** (**Булево**) – указать **Истина**, если обработчик обновления может выполняться только в подчиненном узле РИБ с фильтрами. По умолчанию **Ложь**.
* ● **ЧитаемыеОбъекты** (**Строка**) – список полных имен объектов через запятую, которые обработчик обновления читает при обработке данных.
* ● **ИзменяемыеОбъекты** (**Строка**) – список полных имен объектов через запятую, которые обработчик обновления изменяет при обработке данных.
* ● **Приоритеты** (**ТаблицаЗначений**) – таблица зависимостей между обработчиками, изменяющими или читающими одни и теже объекты:
* ● **Порядок** (**Строка**) – порядок выполнения обработчика относительно другого. Может принимать значения **До**, **После** или **Любой**;
* ● **Идентификатор** (**УникальныйИдентификатор**) – идентификатор процедуры, с которой указывается взаимосвязь;
* ● **Процедура** (**Строка**) – полное имя процедуры, с которой указывается взаимосвязь.  
  Допускается указывать только одно из свойств – **Процедура** или **Идентификатор**.

Например:

Обработчик.ЧитаемыеОбъекты = "Документ.ЗаказПокупателя";

Обработчик.ИзменяемыеОбъекты = "Документ.ЗаказПокупателя";

Обработчик.ПриоритетыВыполнения = ОбновлениеИнформационнойБазы.ПриоритыВыполненияОбработчика();

Приоритет = Обработчик.ПриоритетыВыполнения.Добавить();

Приоритет.Порядок = "До";

Приоритет.Идентификатор = Новый УникальныйИдентификатор("b3be66c5-708d-42c8-a019-818036d09d06");

Приоритет.Процедура = "Документы.ЗаказПокупателя.ОбработатьДанныеДляПереходаНаНовуюВерсию"

### Оперативное обновление на исправительные релизы конфигураций

При разработке исправительных релизов конфигураций рекомендуется сохранять возможность оперативного (динамического) обновления таким образом, чтобы при обновлении на такие версии не требовалось обязательного прекращения работы пользователей и других активных сеансов. Для этого исправительный релиз конфигурации должен удовлетворять следующим условиям:

* ● в него могут быть внесены только «незначительные» изменения, которые позволяют динамически обновлять конфигурацию базы данных (т. е. когда не требуется реструктуризация информационной базы);
* ● обработчики обновления версии ИБ не должны требовать установки монопольного режима для своего выполнения;
* ● изменения в конфигурации должны ограничиваться только теми изменениями, которые не потребуют выполнения обязательных обработчиков обновления, предусмотренных в БСП (свойство **Версия** = «\*»).

Для обязательных обработчиков обновления (свойство **Версия** = «\*») дополнительно предусмотрена возможность программно определять, действительно ли для их выполнения требуется монопольный режим. Для этого следует установить свойство обработчика **МонопольныйРежим**в значение**Истина**. В этом случае:

* ● Такой обработчик вызывается дважды, в него передается параметр **Параметры** типа **Структура**, в котором имеется свойство **МонопольныйРежим** (**Булево**).
* ● При первом вызове в режиме проверки свойство **МонопольныйРежим** содержит значение **Ложь**. При этом код обработчика не должен модифицировать данные ИБ.
* ● Если в ходе выполнения обработчика возникает необходимость внесения изменений в ИБ, обработчик должен установить значение свойства в значение **Истина** и прекратить свое выполнение.
* ● При втором вызове в режиме выполнения свойство **МонопольныйРежим** содержит значение **Истина**. В этом случае код обработчика может модифицировать данные ИБ. Изменение значения свойства в этом случае игнорируется.

Оперативные обработчики обновления должны самостоятельно заботиться о целостности обновляемых данных: если выполняется более одного изменения данных, требуется выполнять действия в транзакции, а если производится чтение данных с последующим изменением, то требуется устанавливать исключительную управляемую блокировку.

Пример обязательного обработчика обновления с проверкой необходимости монопольного режима см. в демонстрационной конфигурации: процедура **ВыполнятьВсегдаПриСменеВерсии** общего модуля **\_ДемоОбновлениеИнформационнойБазыБСП**.

Для гарантированной проверки возможности оперативного обновления на новую версию конфигурации следует выполнить динамическое обновление конфигурации базы данных при активном сеансе «1С:Предприятия» и выполнить запуск нового сеанса администратора. Для этого необходимо:

1. Установить конфигурацию предыдущего релиза.

2. Запустить ИБ предыдущего релиза конфигурации под каким-либо пользователем.

3. Открыть ИБ предыдущего релиза конфигуратором и нажать **F7**.

4. Проверить, что конфигуратор предлагает обновиться динамически. Если не предлагает, см. ниже «Проблема 1».

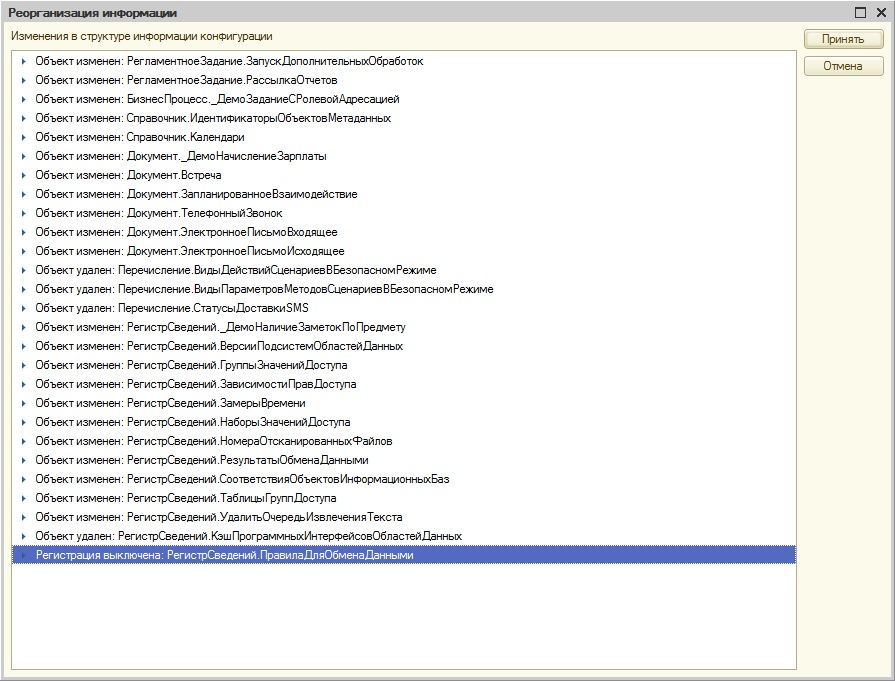
5. Обновить динамически, затем выполнить запуск под администратором по **F5**.

6. Проверить, что запуск прошел без ошибок и выводится окно с описанием изменений в новой версии. Если возникли ошибки, см. ниже «Проблема 2».

Проблема 1. Не проходит динамическое обновление конфигурации.

Для расследования нужно:

* ● завершить сеанс пользователя (запущенный на шаге 2) и нажать **F7** еще раз;
* ● будет выведен список объектов конфигурации, препятствующих динамическому обновлению:



Для дальнейшего расследования причин (разработчиками) нужно сравнить тестируемую конфигурацию с предыдущей версией и выяснить, какие именно изменения блокируют динамическое обновление.

Проблема 2. Ошибки при первом запуске программы администратором.

Помимо «просто» ошибок в обработчиках обновления, может выдаваться ошибка вида:

* ● Невозможно выполнить обновление информационной базы:
* ● невозможно установить монопольный режим,
* ● версия конфигурации не предусматривает обновление без установки монопольного режима.

Она означает:

* ● что один из обработчиков обновления требует монопольного режима для своего выполнения;
* ● либо в конфигурацию были внесены изменения, которые потребовали выполнения обязательных обработчиков обновления, предусмотренных в БСП (для версии «\*»).

Дальнейшее расследование следует выполнять под отладкой либо если была включена константа **Детализировать обновление ИБ в Журнале регистрации**, то информацию о проблемном обработчике можно посмотреть в журнале регистрации.

### Переход с другой конфигурации

Кроме переходов между версиями одной конфигурации подсистема позволяет также выполнять прикладной код при переходе с других конфигураций в режиме обновления. Например, можно предусмотреть переход с базовой версии конфигурации на версию ПРОФ, с ПРОФ на КОРП или другие переходы, при которых меняется не только номер версии, но и имя конфигурации.

При необходимости выполнить прикладной код при таком переходе в процедуре **ПриДобавленииОбработчиковПереходаСДругойПрограммы** общего модуля, имя которого задано в процедуре **ПодсистемыКонфигурацииПереопределяемый.ПриДобавленииПодсистем**, следует определить обработчики перехода в виде:

  Обработчик = Обработчики.Добавить();

  Обработчик.ПредыдущееИмяКонфигурации = "УправлениеТорговлей";

  Обработчик.Процедура = "ОбновлениеУПП.ЗаполнитьУчетнуюПолитику";

Для ветвления логики перехода в зависимости от версии исходной конфигурации внутри обработчика можно воспользоваться функцией **ВерсияИБ** общего модуля **ОбновлениеИнформационнойБазы**.

Пример обработчика перехода см. в общем модуле **\_ДемоОбновлениеИнформационнойБазыБСП**демонстрационной конфигурации.

### Особенности функционирования механизма при работе в модели сервиса

При работе в модели сервиса переход с других конфигураций в режиме обновления не поддерживается (при такой попытке будет вызвано исключение).

Схема выполнения обработчиков обновления версии ИБ в модели сервиса состоит из двух частей:

* ● Выполняется обновление общих (неразделенных) данных. При необходимости выполняется блокировка всей информационной базы (используется монопольный режим). Порядок выполнения обработчиков обновления:
* ● выполняются обработчики с признаком **ОбщиеДанные = Истина**;
* ● устанавливаются блокировки на все области данных;
* ● в очередь заданий добавляются задания на выполнение во всех областях данных обработчиков обновления с признаком **ОбщиеДанные = Ложь**.
* ● Выполняется обновление разделенных данных (во всех областях данных). Выполняется механизмом очереди заданий. При необходимости используется блокировка области данных. Особенности выполнения:
* ● при обновлении существующих областей данных пропускаются обязательные обработчики обновления (**Версия** = «\*»). Чтобы эти обработчики были выполнены, необходимо их повторно зарегистрировать в обработчике обновления общих данных, у которого свойство **УправлениеОбработчиками** равно **Истина**;
* ● при подготовке новых областей данных обязательные обработчики обновления разделенных данных всегда выполняются.

Свойства, специфичные только для обработчиков обновления в модели сервиса:

* ● **ОбщиеДанные** (**Булево**) – если **Истина**, то обработчик обновления будет выполняться из неразделенного сеанса до выполнения любых обработчиков обновления разделенных данных с признаком **ОбщиеДанные = Ложь**. Такой обработчик может обращаться только к неразделенным данным. Допустимо указывать только для обработчиков с режимом выполнения **Монопольно** и **Оперативно**. Если указать значение **Истина** для обработчика с режимом выполнения **Отложенно**, будет выдано исключение. По умолчанию **Ложь**.
* ● **УправлениеОбработчиками** (**Булево**) – если **Истина**, в процедуру обработчика обновления будет передан дополнительный параметр типа **Структура** со свойством **РазделенныеОбработчики**. Свойство содержит дополнительную таблицу обработчиков, в которую нужно добавить обработчики разделенных данных, как и в основную таблицу **Обработчики**. По умолчанию **Ложь**.

Пример обработчика обновления общего (неразделенного) классификатора банков:

Обработчик = Обработчики.Добавить();

Обработчик.Версия = "1.2.3.4";

Обработчик.Процедура = "КлассификаторБанков.ЗаполнитьИНН";

Обработчик.ОбщиеДанные = Истина;

### Подключение обработчиков обновления в доработанных конфигурациях

При доработке типовых конфигураций часто возникает необходимость добавлять свои документы и справочники. В дальнейшем это приводит также к необходимости создавать свои обработчики обновления. Для этих целей следует использовать следующий подход.

Считаем, что исходная типовая конфигурация становится библиотекой со своими собственными номером версии и обработчиками обновления, а доработанная конфигурация выступает в качестве основной конфигурации с собственным именем и своей нумерацией версий, разработанной на базе этой библиотеки.

Например, в типовой конфигурации **БухгалтерияПредприятияКОРП** в общем модуле **ПодсистемыКонфигурацииПереопределяемый** в процедуре **ПриДобавленииПодсистем** предусмотрена ссылка на общий модуль, который содержит описание типовой конфигурации:

МодулиПодсистем.Добавить("ОбновлениеИнформационнойБазыБП");

В общем модуле **ОбновлениеИнформационнойБазыБП** в процедуре **ПриДобавленииПодсистемы** указано имя и версия типовой конфигурации, например:

Описание.Имя = "БухгалтерияПредприятияКОРП";

Описание.Версия = "3.0.38.31";

В этом случае можно смело изменить имя и синоним конфигурации на свои собственные, например, **БухгалтерияПредприятияКОРП\_СRM**, и ввести свою нумерацию версий конфигурации. Затем создать общий модуль вида **ОбновлениеИнформационнойБазы<Сокращение>**, например,**ОбновлениеИнформационнойБазыCRM** и подключить его в конце процедуры **ПриДобавленииПодсистем**общего модуля **ПодсистемыКонфигурацииПереопределяемый**:

МодулиПодсистем.Добавить("ОбновлениеИнформационнойБазыCRM");

В общем модуле **ОбновлениеИнформационнойБазыCRM** в процедуре **ПриДобавленииПодсистемы** указать имя доработанной конфигурации, версию и зависимость от типовой конфигурации, например:

Описание.Имя = "БухгалтерияПредприятияКОРП\_СRM";

Описание.Версия = "1.0.0.1";

Описание.ТребуемыеПодсистемы.Добавить("БухгалтерияПредприятияКОРП");

Обработчики обновления для доработанной конфигурации следует размещать в этом же модуле и привязывать их к новой системе нумерации версий: «1.0.0.2» и т. д.

В дальнейшем, при каждом обновлении доработанной конфигурации на новую версию типовой конфигурации необходимо будет увеличивать номер версии доработанной конфигурации, для того чтобы сработали все обработчики обновления.

Подробнее о настройке модулей **ПодсистемыКонфигурацииПереопределяемый** и **ОбновлениеИнформационнойБазы<Сокращение>** см. раздел «[Подготовка к использованию подсистемы](https://its.1c.ru/db/content/bsp23doc/src/3.29.%20%D0%BE%D0%B1%D0%BD%D0%BE%D0%B2%D0%BB%D0%B5%D0%BD%D0%B8%D0%B5%20%D0%B2%D0%B5%D1%80%D1%81%D0%B8%D0%B8%20%D0%B8%D0%B1.htm?_=1504708101#_подготовка_к_использованию)».

# 3.30. Обновление конфигурации

Подсистема «Обновление конфигурации» предназначена для автоматического обновления конфигурации информационной базы в режиме **1С:Предприятие** «по требованию» или в указанное время в будущем.

Обновление конфигурации не следует использовать, если в программе включена возможность изменения объектов метаданных. В таких случаях рекомендуется выполнять обновление в режиме **Конфигуратор**.

Обновление конфигурации можно установить из основной конфигурации (в этом случае она будет применена к конфигурации базы данных) или из файла (в качестве файла может быть указан cfu-файл обновления конфигурации или cf-файл поставки конфигурации). При работе в файловом режиме работы возможна установка обновления «прямо сейчас» или при завершении работы. При работе в клиент-серверном режиме работы так же возможно запланировать установку обновления на указанное время и выслать на почту отчет о результатах обновления.

При совместном внедрении с инструментарием **1С:Библиотека «Интернет-поддержка пользователей»**(подсистема «ПолучениеОбновленийПрограммы») также появляется возможность поиска и установки обновлений через Интернет.

Процесс установки обновления состоит из следующих шагов:

● Установка блокировки начала сеансов с информационной базой и завершение активных сеансов (в случае планирования обновления на указанное время выполняется непосредственно перед установкой обновления).

● Создание резервной копии (недоступно в клиент-серверном режиме работы).

● Загрузка конфигурации из файла (кроме случая, когда файл обновления уже загружен в основную конфигурацию).

● Обновление конфигурации базы данных.

● Выполнение обработчиков обновления.

● Сжатие таблиц информационной базы (только для файлового режима работы).

● Разрешение подключения новых соединений.

● Перезапуск сеанса (в случае если был выбран вариант установки обновления «прямо сейчас»).

## Настройка

Для использования подсистемы в конфигурации необходимо разместить обработку **УстановкаОбновлений** в командном интерфейсе.

### Настройка прав доступа пользователей

Для настройки прав доступа пользователей к функциям подсистемы «Обновление конфигурации» следует использовать роли, указанные ниже.

Таблица 3.64.

| **№** | **Роли и их назначение** |
| --- | --- |
|  | **АдминистраторСистемы** (из подсистемы «Базовая функциональность»)  Выполнение обновления конфигурации информационной базы |

**Примечание**

Работа с подсистемой недоступна внешним пользователям (подробнее о внешних пользователях см. в разделе «[Пользователи](https://its.1c.ru/db/content/bsp23doc/src/3.37.%20%D0%BF%D0%BE%D0%BB%D1%8C%D0%B7%D0%BE%D0%B2%D0%B0%D1%82%D0%B5%D0%BB%D0%B8.htm?_=1504708101#_пользователи)»).

### Настройка обмена данными

Объекты метаданных подсистемы не следует включать в планы обмена распределенной ИБ (РИБ), а также в планы обмена по правилам конвертации, поскольку для различных узлов в них должны храниться разные данные.

# 3.31. Отправка SMS

Подсистема «Отправка SMS» предназначена для отправки сообщений пользователям посредством SMS.

Для отправки SMS потребуется заключение договора с любым поставщиком услуг, поддерживаемым подсистемой:

● Билайн;

● МТС.

Список поддерживаемых поставщиков услуг может быть расширен при внедрении.

## Настройка

Если в конфигурации не используется подсистема «Настройки программы», то в командном интерфейсе администратора необходимо разместить общую форму **НастройкаОтправкиSMS** (см. пример размещения в форме **Органайзер** обработки **ПанельАдминистрированияБСП** демонстрационной конфигурации).

### Расширение списка поддерживаемых провайдеров SMS

Для расширения списка поддерживаемых провайдеров необходимо:

● Добавить провайдера в перечисление **ПровайдерыSMS**.

● В процедурах **ОтправитьSMS**и **СтатусДоставки**общего модуля **ОтправкаSMSПереопределяемый** и в процедуре **ПриПолученииАдресаПровайдераВИнтернете** общего модуля **ОтправкаSMSКлиентПереопределяемый** написать свою реализацию.

Пример:

Процедура ОтправитьSMS(ПараметрыОтправки, Результат) Экспорт

Если ПараметрыОтправки.Провайдер = Перечисления.ПровайдерыSMS.\_ДемоДругойПровайдер Тогда

// отправка смс через провайдера \_ДемоДругойПровайдер

// ...

КонецЕсли;

КонецПроцедуры

Процедура СтатусДоставки(ИдентификаторСообщения, Провайдер, Логин, Пароль, Результат) Экспорт

Если Провайдер = Перечисления.ПровайдерыSMS.\_ДемоДругойПровайдер Тогда

// проверка статуса отправки смс через провайдера \_ДемоДругойПровайдер

// ...

КонецЕсли;

КонецПроцедуры

Процедура ПриПолученииАдресаПровайдераВИнтернете(Провайдер, АдресВИнтернете) Экспорт

Если Провайдер = ПредопределенноеЗначение("Перечисление.ПровайдерыSMS.\_ДемоДругойПровайдер") Тогда

АдресВИнтернете = "http://yandex.ru/search/?text=рассылка%20смс";

КонецЕсли;

КонецПроцедуры

### Настройка прав доступа пользователей

Таблица 3.65.

| **№** | **Роли и их назначение** |
| --- | --- |
| 1. | **ПолныеПрава** (из подсистемы «Базовая функциональность»)  Отправка SMS, запрос статуса доставки |
| 2. | **ОтправкаSMS**  Отправка SMS, запрос статуса доставки |

## Использование при разработке конфигурации

Программный интерфейс подсистемы доступен через экспортные функции общего модулей **ОтправкаSMS**и **ОтправкаSMSКлиент**.

### Настройка обмена данными

Все объекты метаданных подсистемы, содержащие данные, рекомендуется включить в планы обмена распределенной ИБ (РИБ).

# 3.32. Оценка производительности

Подсистема «Оценка производительности» предназначена для оценки интегральной производительности системы по методике APDEX.

## Настройка

Для настройки подсистемы «Оценка производительности» на сбор данных замеров производительности необходимо принять решение по поводу состава ключевых операций, время выполнения которых требуется замерять и отслеживать.

Ключевые операции – это такие операции, в процессе выполнения которых пользователь непрерывно ожидает отклика от информационной системы. Например: проведение некоторого документа, запись некоторого справочника, загрузка данных и т. п. Ключевые операции не могут включать в себя интерактивные действия пользователя.

Ключевые операции задаются по названию. Настройку приоритета и целевого времени ключевых операций должен будет выполнить администратор информационной базы, после того как в ИБ будут собраны замеры. Настройку приоритета и целевого времени ключевых операций выполнять не обязательно. Данная настройка необходима для последующего анализа замеров.

В коде конфигурации, в местах выполнения ключевых операций, надо вставить вызовы подсистемы «Оценка производительности» для замера времени их выполнения. При этом имя ключевой операции указывается в качестве параметра процедур начала и/или окончания замера времени.

Подсистема «Оценка производительности» поддерживает три сценария работы с замерами времени:

1. Начать замер на клиенте и автоматически завершить замер на клиенте. Кроме того, до момента завершения замера можно программно доопределить имя ключевой операции и комментарий замера времени (если они еще не были известны на момент начала замера).

2. Начать замер на сервере и явно завершить его на сервере (по усмотрению разработчика).

3. Начать замер на клиенте и явно завершить его на клиенте (по усмотрению разработчика). Кроме того, до момента завершения замера можно программно доопределить имя ключевой операции и комментарий замера времени (если они еще не были известны на момент начала замера).

Для выполнения замеров ключевой операции по сценариям № 1 и № 3 необходимо вызвать функцию **НачатьЗамерВремени(Истина, Неопределено)** общего модуля **ОценкаПроизводительностиКлиент**, которая вернет уникальный идентификатор замера. Если параметр функции **АвтоЗавершение = Истина**, то замер времени завершится автоматически. Если же параметр функции **АвтоЗавершение = Ложь**, то для завершения замера времени необходимо вызвать процедуру **ЗавершитьЗамерВремени(УИДЗамера)** общего модуля **ОценкаПроизводительностиКлиент**, где **УИДЗамера** – уникальный идентификатор, который вернула функция **НачатьЗамерВремени**. В случае если **КлючеваяОперация = Неопределено**, необходимо до автозавершения замера установить имя ключевой операции вызовом процедуры **УстановитьКлючевуюОперациюЗамера(УИДЗамера, КлючеваяОперация)** общего модуля **ОценкаПроизводительностиКлиент**. В случае если до завершения автозамера разработчик не установил имя ключевой операции, то данный замер при автозавершении будет удален из клиентского буфера и не будет записан в регистр сведений.

Для выполнения замеров ключевой операции по сценарию № 2 необходимо вызвать функцию **НачатьЗамерВремени** общего модуля **ОценкаПроизводительности**. Затем для завершения замера времени вызывать функцию **ЗакончитьЗамерВремени** общего модуля **ОценкаПроизводительности**.

Пример замера времени проведения документа на клиенте с разделением на замеры корректного проведения и возникновением исключения при проведении (сценарий № 1):

// Модуль формы документа \_ДемоЗаказПокупателя

&НаКлиенте

Перем ИдентификаторЗамераПроведение, ИдентификаторЗамераПроведениеНеНужнаРегистрацияОшибки;

&НаКлиенте

Процедура ПередЗаписью(Отказ, ПараметрыЗаписи)

Если ПараметрыЗаписи.РежимЗаписи = РежимЗаписиДокумента.Проведение Тогда

ИдентификаторЗамераПроведение = ОценкаПроизводительностиКлиент.НачатьЗамерВремени(Истина, "\_ДемоПроведениеДокументаОшибка");

ОценкаПроизводительностиКлиент.УстановитьПризнакОшибкиЗамера(ИдентификаторЗамераПроведение, Истина);

ИдентификаторЗамераПроведениеНеНужнаРегистрацияОшибки = ОценкаПроизводительностиКлиент.НачатьЗамерВремени(Истина);

КонецЕсли;

// Далее следует текст обработчика формы -->>

КонецПроцедуры

&НаКлиенте

Процедура ПослеЗаписи(ПараметрыЗаписи)

ОценкаПроизводительностиКлиент.УстановитьКлючевуюОперациюЗамера(ИдентификаторЗамераПроведение, "\_ДемоПроведениеДокумента");

ОценкаПроизводительностиКлиент.УстановитьПризнакОшибкиЗамера(ИдентификаторЗамераПроведение, Ложь);

ОценкаПроизводительностиКлиент.УстановитьКлючевуюОперациюЗамера(ИдентификаторЗамераПроведениеНеНужнаРегистрацияОшибки, "\_ДемоПроведениеДокументаНеНужнаРегистрацияОшибки");

// Далее следует текст обработчика формы -->>

КонецПроцедуры

Пример замера времени выполнения регламентного задания (сценарий № 2):

// Процедура обработки регламентного задания по выгрузке данных

Процедура ЭкспортОценкиПроизводительности(КаталогиЭкспорта) Экспорт

ДатаНачала = ОценкаПроизводительности.НачатьЗамерВремени();

// Далее следует текст выполнения регламентного задания -->>

ОценкаПроизводительности.ЗакончитьЗамерВремени("ЭкспортОценкиПроизводительности", ДатаНачала);

КонецПроцедуры

Пример замера времени длительной операции на клиенте – формирование отчета в фоновом задании (сценарий № 3):

&НаКлиенте

Перем УникальныйИдентификаторЗамера, УникальныйИдентификаторФоновогоЗадания;

&НаКлиенте

Процедура СформироватьОтчет()

УникальныйИдентификаторЗамера =

ОценкаПроизводительностиКлиент.НачатьЗамерВремени(Ложь, "ОтчетПоПродажам");

// Далее следует текст запуска формирования отчета на сервере в фоновом задании -->>

ПодключитьОбработчикОжидания("ПроверкаФормированияОтчета", 1);

КонецПроцедуры

&НаКлиенте

Процедура ПроверкаФормированияОтчета()

РезультатПроверки = ПроверкаВыполненияФоновогоЗаданияНаСервере(УникальныйИдентификаторФоновогоЗадания);

Если РезультатПроверки.Состояние <> "Активно" Тогда

ОтключитьОбработчикОжидания("ПроверкаФормированияОтчета");

Если РезультатПроверки.Состояние = "Завершено" Тогда

// Далее следует текст вывода отчета на сервере в фоновом задании -->>

ОценкаПроизводительностиКлиент.ЗавершитьЗамерВремени(УникальныйИдентификаторЗамера);

Иначе

ВызватьИсключение(ПодробноеПредставлениеОшибки(ИнформацияОбОшибке()));

КонецЕсли;

КонецЕсли;

КонецПроцедуры

Пример реализации ключевых операций см. в демонстрационной конфигурации.

Необходимо разместить в командном интерфейсе ответственного за оценку производительности константу **ВыполнятьЗамерыПроизводительности** и обработку **ОценкаПроизводительности**. См. пример размещения в демонстрационной конфигурации в группе **Оценка производительности** формы **ПоддержкаИОбслуживание**обработки **ПанельАдминистрированияБСП**.

Для того чтобы система автоматически начала собирать замеры производительности, необходимо константе **ВыполнятьЗамерыПроизводительности** установить значение **Истина** в процедуре – обработчике обновления и первоначального заполнения ИБ.

### Настройка прав доступа пользователей

Роли, используемые для работы с подсистемой:

Таблица 3.66.

| **№** | **Роли и их назначение** |
| --- | --- |
|  | **НастройкаИОценкаПроизводительности**  Анализ результатов замера производительности ключевых операций с помощью обработки **ОценкаПроизводительности** |

## Использование при разработке конфигурации

### Настройка обмена данными

Все объекты подсистемы не должны включаться в планы обмена распределенной ИБ (РИБ), т. к. требования и условия работы в каждой базе могут быть свои. Однако в некоторых задачах, в которых распределенная ИБ (РИБ) рассматривается как единая и однородная система в плане производительности, имеет смысл в планы обмена включить все объекты подсистемы, содержащие данные.

# 3.33. Печать

Подсистема «Печать» предназначена для формирования печатных форм объектов на основе табличных макетов (формат MXL) или макетов офисных документов в форматах Microsoft Word и OpenOffice.org Writer.

Подсистема предоставляет инструменты для размещения команд печати на формах в подменю **Печать**, форму предпросмотра печатных форм, а также различные сервисные возможности по сохранению печатных форм в файлы, отправке по электронной почте и формированию изображений QR-кодов. Кроме того, в подсистему входит рабочее место по редактированию макетов печатных форм.

## Настройка

Необходимо принять решение по поводу состава объектов конфигурации (справочников, документов и т. п.), которые требуется выводить на печать, и в каком виде должны формироваться печатные формы. Затем создать для них команды печати, разработав описательную часть, логику формирования печатной формы, и внести изменения в модули форм, в которых предполагается выводить команды печати.

Описательная часть находится в процедуре **ДобавитьКомандыПечати**, а логика формирования печатной формы зависит от того, в каком виде должны формироваться печатные формы:

* ● Формирование печатной формы в формате табличного документа (с предварительным просмотром или сразу на принтер).
* ● Формирование комплекта табличных документов (с предварительным просмотром или сразу на принтер).
* ● Формирование печатных форм с интерактивным запросом дополнительных параметров у пользователя.
* ● Вывод табличного документа в один из популярных форматов (Microsoft Word, Microsoft Excel, Adobe PDF, OpenOffice.org Writer, HTML, текстовый документ и другие).
* ● Формирование печатной формы в виде офисных документов в форматах Microsoft Word и OpenOffice.org Writer (для тех случаев, когда возможностей табличного макета недостаточно).

### Подготовительные действия

На первом этапе нужно внедрить подсистему в те объекты, в которых предполагается использовать команды печати:

1. В формах каждого объекта, где необходимо разместить команду печати, в процедуре **ПриСозданииНаСервере**добавить следующий код:

&НаСервере

Процедура ПриСозданииНаСервере(Отказ, СтандартнаяОбработка)

// СтандартныеПодсистемы.Печать

УправлениеПечатью.ПриСозданииНаСервере(ЭтотОбъект);

// Конец СтандартныеПодсистемы.Печать

КонецПроцедуры

По умолчанию команды печати располагаются на командной панели формы, но размещение можно переопределить, указав второй параметр в вызове **УправлениеПечатью.ПриСозданииНаСервере**:

УправлениеПечатью.ПриСозданииНаСервере(ЭтотОбъект, Элементы.ГруппаКомандПечати);

Динамическое создание команд печати на форме может замедлять ее открытие, поэтому размещать команды печати рекомендуется в статической группе команд вида **Подменю**.

2. В форме объекта, куда размещается команда печати, добавить код подключаемых обработчиков:

// СтандартныеПодсистемы.Печать

&НаКлиенте

Процедура Подключаемый\_ВыполнитьКомандуПечати(Команда)

УправлениеПечатьюКлиент.ВыполнитьПодключаемуюКомандуПечати(Команда, ЭтотОбъект, Объект);

КонецПроцедуры

&НаКлиенте

Процедура Подключаемый\_ОбновитьКоманды()

УправлениеПечатьюКлиентСервер.ОбновитьКоманды(ЭтотОбъект, Объект);

КонецПроцедуры

// Конец СтандартныеПодсистемы.Печать

3. В форме списка объекта, куда размещается команда печати, добавить код подключаемых обработчиков:

// СтандартныеПодсистемы.Печать

&НаКлиенте

Процедура Подключаемый\_ВыполнитьКомандуПечати(Команда)

УправлениеПечатьюКлиент.ВыполнитьПодключаемуюКомандуПечати(Команда, ЭтотОбъект, Элементы.Список);

КонецПроцедуры

&НаКлиенте

Процедура Подключаемый\_ОбновитьКоманды()

УправлениеПечатьюКлиентСервер.ОбновитьКоманды(ЭтотОбъект, Объект);

КонецПроцедуры

// Конец СтандартныеПодсистемы.Печать

**Важно!**

Поле **Ссылка** динамического списка формы должно быть доступно в обработчике команды. Для этого необходимо у реквизита формы, содержащего динамический список, у поля **Ссылка** поставить флажок **Использовать всегда**.

4. Добавить экспортную процедуру **ДобавитьКомандыПечати** в модуль менеджера:

// Заполняет список команд печати.

//

// Параметры:

//   КомандыПечати – ТаблицаЗначений – состав полей см. в функции УправлениеПечатью.СоздатьКоллекциюКомандПечати

//

Процедура ДобавитьКомандыПечати(КомандыПечати) Экспорт

КонецПроцедуры

5. Добавить выбранные объекты в параметр **СписокОбъектов** в процедуре **ПриОпределенииОбъектовСКомандамиПечати** общего модуля **УправлениеПечатьюПереопределяемый**:

// Определяет объекты, в которых есть процедура ДобавитьКомандыПечати().

//

// Параметры:

//  СписокОбъектов - Массив - список менеджеров объектов.

//

Процедура ПриОпределенииОбъектовСКомандамиПечати(СписокОбъектов) Экспорт

СписокОбъектов.Добавить(Справочники.\_ДемоКонтактныеЛицаПартнеров);

СписокОбъектов.Добавить(Справочники.\_ДемоКонтрагенты);

...

КонецПроцедуры

**Важно!**

Подменю **Печать** в формах журналов документов по умолчанию заполняется командами документов, входящих в журнал. В случаях, когда команды печати необходимо брать непосредственно в модуле менеджера журнала документов, необходимо сделать соответствующую вставку кода в процедуре **ПриПолученииНастроекСпискаКомандПечати** общего модуля **УправлениеПечатьюПереопределяемый**:

// Дополнительные настройки списка команд печати.

//

// Параметры:

//  НастройкиСписка - Структура - модификаторы списка команд печати.

//   \* МенеджерКомандПечати     - МенеджерОбъекта - менеджер объекта, в котором формируется список команд печати;

//   \* АвтоматическоеЗаполнение - Булево - заполнять команды печати из объектов, входящих в состав журнала.

//                                         Значение по умолчанию: Истина.

//

Процедура ПриПолученииНастроекСпискаКомандПечати(НастройкиСписка) Экспорт

Если НастройкиСписка.МенеджерКомандПечати = ЖурналыДокументов.\_ДемоСкладскиеДокументы Тогда

НастройкиСписка.АвтоматическоеЗаполнение = Ложь;

КонецЕсли;

КонецПроцедуры

### Разработка команд печати

Затем необходимо разработать описательную часть, перечислив команды печати для каждого объекта в процедуре **ДобавитьКомандыПечати**, указав представление, идентификатор и другие параметры (подробнее см. таблицу ниже). Например:

// Счет на оплату

КомандаПечати = КомандыПечати.Добавить();

КомандаПечати.МенеджерПечати = "Документ.\_ДемоСчетНаОплатуПокупателю";

КомандаПечати.Идентификатор = "СчетЗаказ";

КомандаПечати.Представление = НСтр("ru = 'Счет на оплату'");

КомандаПечати.ПроверкаПроведенияПередПечатью = Истина;

Параметры команды печати:

Таблица 3.67.

| **Параметр** | **Тип** | **Описание** |
| --- | --- | --- |
| **Идентификатор** (обязательный) | **Строка** или **Массив** | Идентификатор команды печати, по которому менеджер печати определяет печатную форму, которую необходимо сформировать. Пример:  КомандаПечати.Идентификатор = "СчетЗаказ";  Для печати нескольких печатных форм можно указывать одновременно несколько их идентификаторов (строкой, через запятую либо массивом строк), например:  КомандаПечати.Идентификатор = "СчетЗаказ,ГарантийноеПисьмо";  Если необходимо задать количество копий печати для печатной формы, то ее идентификатор нужно продублировать столько раз, сколько копий необходимо сформировать. При этом следует учитывать, что порядок следования печатных форм в комплекте будет соответствовать порядку идентификаторов печатных форм, указанных в этом параметре. Пример (2 счета на оплату + 1 гарантийное письмо):  КомандаПечати.Идентификатор = "СчетЗаказ,СчетЗаказ,ГарантийноеПисьмо";  Идентификатор печатной формы может содержать в себе и альтернативный менеджер печати, если он отличается от указанного в параметре **МенеджерПечати**, например:  КомандаПечати.Идентификатор = "СчетЗаказ,Обработка.\_ДемоПечатнаяФорма.ГарантийноеПисьмо";  В этом примере **ГарантийноеПисьмо** формируется в менеджере печати **Обработка.\_ДемоПечатнаяФорма**, а **СчетЗаказ** – в менеджере печати, указанном в параметре **МенеджерПечати** |
| **Представление** (обязательный) | **Строка** | Представление команды в меню **Печать**. Пример:  КомандаПечати.Представление = НСтр("ru = 'Счет на оплату'"); |
| **МенеджерПечати** (необязательный) | **Строка** | Имя объекта, в модуле менеджера которого располагается процедура **Печать**, формирующая табличные документы для этой команды. Например:  КомандаПечати.МенеджерПечати = "Документ.\_ДемоСчетНаОплатуПокупателю"; |
| **ТипыОбъектовПечати**  (необязательный) | **Массив** | Список типов объектов, для которых предназначена команда печати. Параметр предназначен для команд печати в журналах документов, где требуется проверка передаваемого типа объекта перед вызовом менеджера печати.  Если список не заполнен, то при автоматическом создании списка команд печати в журнале документов он заполняется типом объекта, из которого была импортирована команда печати |
| **Обработчик**  (необязательный) | **Строка** | Клиентский обработчик команды, в который необходимо передать управление вместо стандартного обработчика команды печать. Используется, например, когда печатная форма формируется на клиенте.  Формат "<ИмяОбщегоМодуля>.<ИмяПроцедуры>" используется, когда процедура размещена в общем модуле.  Формат "<ИмяПроцедуры>" используется, когда процедура размещена в модуле основной формы отчета или обработки, указанной в **МенеджерПечати**.  Пример:  КомандаПечати.Обработчик = "\_ДемоСтандартныеПодсистемыКлиент.ПечатьСчетовНаОплатуПокупателю";  Пример обработчика в модуле формы:  // Формирует печатную форму <представление печатной формы>.  //  // Параметры:  //   ПараметрыПечати - Структура - Сведения о печатной форме.  //       \* ОбъектыПечати - Массив - Массив ссылок выбранных объектов.  //       \* Форма - УправляемаяФорма - Форма, из которой вызвана команда печати.  //       \* ДополнительныеПараметры - Структура - Дополнительные параметры печати.  //       Прочие ключи структуры соответствуют колонкам таблицы КомандыПечати,  //       подробнее см. в функции УправлениеПечатью.СоздатьКоллекциюКомандПечати.  //  &НаКлиенте  Функция <ИмяФункции>(ПараметрыПечати) Экспорт  // Обработчик печати.  КонецФункции  Следует иметь в виду, что обработчик вызывается при помощи метода **Вычислить**, поэтому в качестве обработчика может выступать только функция. При этом возвращаемое значение функции никак в дальнейшем не используется подсистемой. |
| **Порядок**  (необязательный) | **Число** | Значение от 1 до 100, указывающее порядок размещения команды по отношению к другим командам. Сортировка команд меню **Печать**осуществляется сначала по полю **Порядок**, затем по представлению.  Значение по умолчанию: 50 |
| **Картинка**  (необязательный) | **Картинка** | Картинка, которая отображается возле команды в меню **Печать**. Пример:  КомандаПечати.Картинка = БиблиотекаКартинок.ФорматPDF; |
| **СписокФорм**  (необязательный) | **Строка** | Имена форм, в которых должна отображаться команда, через запятую. Если параметр не указан, то команда печати будет отображаться во всех формах объекта, где встроена подсистема «Печать». Пример:  КомандаПечати.СписокФорм = "ФормаДокумента,ФормаСписка" |
| **ЗаголовокФормы**  (необязательный) | **Строка** | Произвольная строка, переопределяющая стандартных заголовок формы «Печать документов». Пример:  КомандаПечати.ЗаголовокФормы = НСтр("ru = 'Настраиваемый комплект'"); |
| **ФункциональныеОпции**  (необязательный) | **Строка** | Имена функциональных опций, от которых зависит доступность команды печати, через запятую |
| **ПроверкаПроведенияПередПечатью**  (необязательный) | **Булево** | Признак необходимости проверки проведенности документов перед печатью. Если параметр не указан, то проверка проведенности не выполняется. Пример:  КомандаПечати.ПроверкаПроведенияПередПечатью = Истина; |
| **СразуНаПринтер**  (необязательный) | **Булево** | Признак необходимости печати документов без предварительного просмотра, сразу на принтер. Если параметр не указан, то при выборе команды печати открывается форма предварительного просмотра «Печать документов». Пример:  КомандаПечати.СразуНаПринтер = Истина; |
| **ФорматСохранения**  (необязательный) | **ТипФайлаТабличногоДокумента** | Применяется для быстрого сохранения печатной формы (без дополнительных действий) в различные форматы, отличные от mxl. Если параметр не указан, то формируется обычный mxl. Пример:  КомандаПечати.ФорматСохранения = ТипФайлаТабличногоДокумента.PDF;  При выборе команды печати сразу открывается сформированный в формате pdf документ |
| **ПереопределитьПользовательскиеНастройкиКоличества**  (необязательный) | **Булево** | Признак необходимости отключения в форме **ПечатьДокументов**механизма сохранения/восстановления выбранного пользователем количества экземпляров на печать. Если параметр не указан, то механизм сохранения/восстановления настроек будет работать при открытии формы **ПечатьДокументов**. Пример:  КомандаПечати.ПереопределитьПользовательскиеНастройкиКоличества = Истина; |
| **ДополнитьКомплектВнешнимиПечатнымиФормами**  (необязательный) | **Булево** | Признак необходимости дополнить комплект документов всеми подключенными к объекту внешними печатными формами (подсистема «ДополнительныеОтчетыИОбработки»). Если параметр не указан, внешние печатные формы не добавляются в комплект. Пример:  КомандаПечати.ДополнитьКомплектВнешнимиПечатнымиФормами = Истина; |
| **ФиксированныйКомплект**  (необязательный) | **Булево** | Признак необходимости блокировки от изменения пользователем состава комплекта документов. Если параметр не указан, то пользователь сможет исключать отдельные печатные формы из комплекта в форме **ПечатьДокументов**, а также изменять их количество. Пример:  КомандаПечати.ФиксированныйКомплект = Истина; |
| **ДополнительныеПараметры**  (необязательный) | **Структура** | Произвольные параметры для передачи в менеджер печати |
| **НеВыполнятьЗаписьВФорме**  (необязательный) | **Булево** | Признак необходимости отключения механизма записи объекта перед выполнением команды печати. Используется в исключительных случаях. Если параметр не указан, производится запись объекта, в случае если в форме объекта установлен признак модифицированности. Пример:  КомандаПечати. НеВыполнятьЗаписьВФорме= Истина; |
| **ТребуетсяРасширениеРаботыСФайлами**  (необязательный) | **Булево** | Признак необходимости подключения расширения работы с файлами перед выполнением команды. Если параметр не указан, подключение расширения работы с файлами выполняться не будет |

Также имеется возможность скрывать команду печати, если она не подходит для выбранного пользователем объекта. Для этого необходимо:

* ● при описании команды печати воспользоваться процедурой **ДобавитьУсловиеВидимостиКоманды** общего модуля **УправлениеПечатью**:

Процедура ДобавитьКомандыПечати(КомандыПечати) Экспорт

КомандаПечати = КомандыПечати.Добавить();

КомандаПечати.Идентификатор = "ВыдачаЗарплаты";

КомандаПечати.Представление = НСтр("ru = 'Выдача зарплаты'");

УправлениеПечатью.ДобавитьУсловиеВидимостиКоманды(КомандаПечати, "ХозяйственнаяОперация",

Перечисления.\_ДемоХозяйственныеОперации.ВыдачаЗарплаты, ВидСравнения.Равно);

КонецПроцедуры

* ● в формах, где используются условия видимости команд печати:
* ● разместить процедуру **Подключаемый\_ОбновитьКоманды** (см. «[Подготовительные действия](https://its.1c.ru/db/content/bsp23doc/src/3.33.%20%D0%BF%D0%B5%D1%87%D0%B0%D1%82%D1%8C.htm?_=1504708101#_подготовительные_действия)»);
* ● вставить вызов процедуры **ОбновитьКоманды** общего модуля **УправлениеПечатьюКлиентСервер** в событии **ПриЧтенииНаСервере** в форме объекта/документа;
* ● вставить вызовы процедуры **НачатьОбновлениеКоманд**общего модуля **УправлениеПечатьюКлиент** в событиях **ПриИзменении** реквизитов, участвующих в условиях видимости, а также в событиях **ПриОткрытии** формы объекта/документа и **ПриАктивизацииСтроки** формы списка объектов/документов:

&НаКлиенте

Процедура СписокПриАктивизацииСтроки(Элемент)

УправлениеПечатьюКлиент.НачатьОбновлениеКоманд(ЭтотОбъект);

КонецПроцедуры

* ● у динамических списков для реквизитов, участвующих в условиях видимости команд печати необходимо установить флажок **Использовать всегда**.

Разработка команд печати существенно отличается в зависимости от того, в каком виде должна формироваться печатная форма:

* ● Для формирования печатной формы в формате табличного документа или комплекта табличных документов см. раздел «[Разработка процедуры «Печать»](https://its.1c.ru/db/content/bsp23doc/src/3.33.%20%D0%BF%D0%B5%D1%87%D0%B0%D1%82%D1%8C.htm?_=1504708101#_разработка_процедуры_)».
* ● Для вывода табличного документа в один из популярных форматов (Microsoft Word, Microsoft Excel, Adobe PDF, OpenOffice.org Writer, HTML, текстовый документ и другие) см. раздел «[Разработка процедуры «Печать»](https://its.1c.ru/db/content/bsp23doc/src/3.33.%20%D0%BF%D0%B5%D1%87%D0%B0%D1%82%D1%8C.htm?_=1504708101#_разработка_процедуры_)».
* ● Для формирования печатных форм с интерактивным запросом дополнительных параметров у пользователя см. раздел «[Формирование печатной формы в клиентском контексте](https://its.1c.ru/db/content/bsp23doc/src/3.33.%20%D0%BF%D0%B5%D1%87%D0%B0%D1%82%D1%8C.htm?_=1504708101#_формирование_печатной_формы)».
* ● Для формирования печатной формы в виде офисных документов в форматах Microsoft Word и OpenOffice.org Writer см. раздел «[Использование макетов в формате офисных документов при формировании печатных форм](https://its.1c.ru/db/content/bsp23doc/src/3.33.%20%D0%BF%D0%B5%D1%87%D0%B0%D1%82%D1%8C.htm?_=1504708101#_использование_макетов_в)».

### Разработка процедуры «Печать»

1. В модуль менеджера печати, указанного в параметре **МенеджерПечати**, добавить экспортную процедуру **Печать**:

// Формирует печатные формы.

//

// Параметры:

//  МассивОбъектов – Массив – ссылки на объекты, которые нужно распечатать;

//  ПараметрыПечати – Структура – дополнительные настройки печати;

//  КоллекцияПечатныхФорм – ТаблицаЗначений – сформированные табличные документы (выходной параметр)

//  ОбъектыПечати – СписокЗначений – значение – ссылка на объект;

//                                            представление – имя области, в которой был выведен объект (выходной параметр);

//  ПараметрыВывода – Структура – дополнительные параметры сформированных табличных документов (выходной параметр).

//

Процедура Печать(МассивОбъектов, ПараметрыПечати, КоллекцияПечатныхФорм, ОбъектыПечати, ПараметрыВывода) Экспорт

КонецПроцедуры

2. Разместить в конфигурации макет табличного документа согласно инструкциям раздела «[Размещение макетов печатных форм в конфигурации](https://its.1c.ru/db/content/bsp23doc/src/3.33.%20%D0%BF%D0%B5%D1%87%D0%B0%D1%82%D1%8C.htm?_=1504708101#_размещение_макетов_печатных)».

3. Добавить функцию, формирующую печатную форму (табличный документ), например:

Функция ПечатьСчетаЗаказа(МассивОбъектов, ОбъектыПечати)

. . .

Возврат ТабличныйДокумент;

КонецФункции

4. В процедуре **Печать** разместить код для идентификации требуемой печатной формы и вызова функции по ее формированию (созданную на шаге 3). Например:

НужноПечататьМакет = УправлениеПечатью.НужноПечататьМакет(КоллекцияПечатныхФорм, "СчетЗаказ");

Если НужноПечататьМакет Тогда

УправлениеПечатью.ВывестиТабличныйДокументВКоллекцию(

КоллекцияПечатныхФорм,

"СчетЗаказ",

НСтр("ru = 'Счет на оплату'"),

ПечатьСчетаЗаказа(МассивОбъектов, ОбъектыПечати),

,

"Документ.\_ДемоСчетНаОплатуПокупателю.ПФ\_MXL\_СчетЗаказ");

КонецЕсли;

Параметры процедуры **Печать**:

Таблица 3.68.

| **Параметр** | **Описание** |
| --- | --- |
| **МассивОбъектов** | Массив ссылок на объекты, для которых нужно сформировать табличные документы |
| **ПараметрыПечати** | Произвольные параметры, переданные при вызове команды печати |
| **КоллекцияПечатныхФорм** | Служебная таблица значений. Через нее передаются имена макетов, для которых должны быть сформированы табличные документы. Возвращаются синонимы макетов (представление), сформированные табличные документы с указанием количества экземпляров, картинка для отображения на закладке соответствующего табличного документа в форме печати документов |
| **ОбъектыПечати** | Список значений, в котором обработчик печати устанавливает соответствие между объектами и именами областей печати табличного документа: значение – **Объект**, представление – имя области, в которой был выведен объект |
| **ПараметрыВывода** | Структура с предопределенными ключами, через которую обработчик печати может вернуть параметры:  ● **ПараметрыОтправки** – структура, содержащая информацию для заполнения письма при отправке печатной формы по электронной почте. Содержит следующие поля (описание см. в общем модуле конфигурации **РаботаСПочтовымиСообщениямиКлиент** в процедуре **ОткрытьФормуОтправкиПочтовогоСообщения**):  ● **Получатель**,  ● **Тема**,  **● Текст** |

Процедура **Печать** может формировать один или несколько табличных документов за один вызов. Процедура может быть реализована следующим образом:

* ● При необходимости устанавливаются **ПараметрыВывода**.
* ● Если формируется больше одного табличного документа, то рекомендуется получить необходимые данные до формирования табличных документов.
* ● При формировании табличного документа для нескольких объектов необходимо позаботиться о правильном порядке этих объектов в табличном документе. В процедуру **Печать** в параметр **МассивОбъектов** передается массив объектов в том порядке, в котором их выбрал пользователь, однако «правильный» порядок вывода может быть иным – например, в документах, как правило, упорядочивание выполняется по полю **МоментВремени**.
* ● Если для каждого табличного документа данные получаются разными запросами, то желательно, чтобы документы упорядочивались одинаково.

В связи с тем, что макет может быть отредактирован пользователем в режиме **1С:Предприятие** и параметры в нем могут быть изменены или удалены, для повышения устойчивости кода формирования печатной формы рекомендуется избегать явного присвоения значений параметров в областях печати. Вместо этого следует использовать процедуру **ЗаполнитьЗначенияСвойств**.

Неправильно:

ОбластьПечати.Параметры.Организация = ДанныеПечати.Организация;

ОбластьПечати.Параметры.Контрагент = ДанныеПечати.Контрагент;

Правильно:

ЗаполнитьЗначенияСвойств(ОбластьПечати.Параметры, ДанныеПечати);

Для каждого макета, для которого в обработчике предусмотрено формирование табличного документа, необходимо проверить, нужно ли формировать табличный документ. Проверка осуществляется вызовом функции **УправлениеПечатью.НужноПечататьМакет** с параметрами.

Сформированный табличный документ добавляется в коллекцию вызовом процедуры **УправлениеПечатью.ВывестиТабличныйДокументВКоллекцию** с параметрами.

Пример реализации процедуры **Печать**:

Процедура Печать(МассивОбъектов, ПараметрыПечати, КоллекцияПечатныхФорм, ОбъектыПечати, ПараметрыВывода) Экспорт

// Проверяем, нужно ли для макета СчетЗаказ формировать табличный документ.

Если УправлениеПечатью.НужноПечататьМакет(КоллекцияПечатныхФорм, "СчетЗаказ") Тогда

// Формируем табличный документ и добавляем его в коллекцию печатных форм.

УправлениеПечатью.ВывестиТабличныйДокументВКоллекцию(КоллекцияПечатныхФорм,

"СчетЗаказ", "Счет заказ", ПечатьСчетаЗаказ(МассивОбъектов, ОбъектыПечати));

КонецЕсли;

КонецПроцедуры

Пример реализации функции, формирующей табличный документ:

Функция ПечатьСчетаЗаказ(МассивОбъектов, ОбъектыПечати)

// Создаем табличный документ и устанавливаем имя параметров печати.

ТабличныйДокумент = Новый ТабличныйДокумент;

ТабличныйДокумент.ИмяПараметровПечати = "ПараметрыПечати\_СчетЗаказ";

// Получаем запросом необходимые данные.

Запрос = Новый Запрос();

Запрос.Текст =

"ВЫБРАТЬ

| СчетНаОплатуПокупателю.Ссылка КАК Ссылка,

......................

|ГДЕ

| СчетНаОплатуПокупателю.Ссылка В(&МассивОбъектов)

......................

|";

Запрос.УстановитьПараметр("МассивОбъектов", МассивОбъектов);

Шапка = Запрос.Выполнить().Выбрать();

ПервыйДокумент = Истина;

Пока Шапка.Следующий() Цикл

Если Не ПервыйДокумент Тогда

// Все документы нужно выводить на разных страницах.

ТабличныйДокумент.ВывестиГоризонтальныйРазделительСтраниц();

КонецЕсли;

ПервыйДокумент = Ложь;

// Запомним номер строки, с которой начали выводить текущий документ.

НомерСтрокиНачало = ТабличныйДокумент.ВысотаТаблицы + 1;

......................

// В табличном документе необходимо задать имя области, в которую был

// выведен объект. Нужно для возможности печати комплектов документов.

УправлениеПечатью.ЗадатьОбластьПечатиДокумента(ТабличныйДокумент,

НомерСтрокиНачало, ОбъектыПечати, Шапка.Ссылка);

КонецЦикла;

Возврат ТабличныйДокумент;

КонецФункции

При разработке печатной формы можно предусмотреть ограничение ее вывода (на принтер, в буфер обмена, сохранение в файл). Для этого при формировании табличного документа необходимо установить у него соответствующее значение в свойстве **Вывод**, например:

ТабличныйДокумент.Вывод = ИспользованиеВывода.Запретить

### Формирование печатной формы в клиентском контексте

В отдельных случаях для формирования некоторых печатных форм может потребоваться клиентский контекст. Например, для запроса дополнительных параметров печатной формы у пользователя непосредственно перед печатью. В таких случаях механизм формирования печатной формы необходимо размещать в клиентском модуле, а при описании команды печати в процедуре **ДобавитьКомандыПечати** использовать параметр **Обработчик** для передачи управления в этот модуль.

Принцип создания команд, использующих клиентский контекст, несколько отличается от основного. Процедура **Печать** модуля менеджера объекта не вызывается механизмами подсистемы, поэтому создавать ее для такой команды не требуется.

Процесс создания такой команды выглядит следующим образом:

1. В модуле менеджера объекта в процедуре **ДобавитьКомандыПечати** добавить описание команды (с использованием параметра **Обработчик**), например:

// Счет на оплату в Microsoft Word

КомандаПечати = КомандыПечати.Добавить();

КомандаПечати.МенеджерПечати = "Документ.\_ДемоСчетНаОплатуПокупателю";

КомандаПечати.Идентификатор = "СчетНаОплату(MSWord)";

КомандаПечати.Представление = НСтр("ru = 'Счет на оплату в Microsoft Word'");

КомандаПечати.Картинка = БиблиотекаКартинок.ФорматWord2007;

КомандаПечати.ПроверкаПроведенияПередПечатью = Истина;

КомандаПечати.Обработчик = "\_ДемоСтандартныеПодсистемыКлиент.ПечатьСчетовНаОплатуПокупателю";

**Примечание**

Менеджер печати может быть использован для получения данных на печать (см. раздел «[Использование макетов в формате офисных документов при формировании печатных форм](https://its.1c.ru/db/content/bsp23doc/src/3.33.%20%D0%BF%D0%B5%D1%87%D0%B0%D1%82%D1%8C.htm?_=1504708101#_использование_макетов_в)»).

2. Добавить клиентскую экспортную функцию формирования печатной формы с единственным параметром, в который подсистема «Печать» будет передавать структуру параметров команды. Имя функции может быть произвольным, например:

Функция ПечатьСчетовНаОплатуПокупателю(ПараметрыПечати) Экспорт

.

.

.

КонецФункции

**Примечание 1**

Из этой функции подсистема «Печать» не ожидает никакого результата. Использование функции вместо процедуры обусловлено тем, что ее вызов осуществляется при помощи метода **Вычислить()**.

**Примечание 2**

Для вывода печатных форм на принтер пользователь должен обладать правом «Вывод». При разработке клиентских команд печати следует учитывать наличие этого права.

Если клиентский контекст предполагает только запрос дополнительных параметров, а формирование табличного документа выполняется на сервере, то необходимо также выполнить [инструкцию по разработке процедуры Печать](https://its.1c.ru/db/content/bsp23doc/src/3.33.%20%D0%BF%D0%B5%D1%87%D0%B0%D1%82%D1%8C.htm?_=1504708101#_разработка_процедуры_), а из клиентской функции выполнять передачу управления в процедуру **УправлениеПечатьюКлиент.ВыполнитьКомандуПечати**.

### Вывод в печатную форму изображения QR-кода по заданной текстовой строке

В некоторых случаях в печатной форме документа полезно разместить изображение QR-кода, содержащего закодированные данные. Общий принцип встраивания функционала следующий:

1. В макете печатной формы разместить рисунок типа **Картинка**.

2. В модуле менеджера печати вызвать интерфейсный метод **УправлениеПечатью.ДанныеQRКода**, передав в качестве параметра текстовую строку, которую необходимо закодировать с помощью QR-кода.

3. Полученный результат (двоичные данные) программно вывести в рисунок типа **Картинка**.

Пример реализации см. в демонстрационной конфигурации, в модуле менеджера документа **\_ДемоСчетНаОплатуПокупателю** в процедуре **ПечатьКвитанции** и макете **ПФ\_MXL\_Квитанция**.

### Размещение макетов печатных форм в конфигурации

Макеты печатных форм рекомендуется размещать при объекте, в котором реализована логика формирования печатных форм. К имени макета следует добавить специальный префикс:

* ● макеты табличных документов – **ПФ\_MXL**,
* ● макет Microsoft Word – **ПФ\_DOC**,
* ● макет OpenOffice.org Writer **ПФ\_ODT**.

Например, **\_ДемоПФ\_ODT\_СчетНаОплату**.

В процедурах формирования печатных форм для получения макета следует использовать процедуру **УправлениеПечатью.ПолучитьМакет(<Путь к макету>)**, где путь к макету может быть:

* ● для макетов в документах: **Документ.<Имя документа>.<Имя макетов>**;
* ● для макетов в обработках: **Обработка.<Имя обработки>.<Имя макета>;**
* ● для общих макетов: **ОбщийМакет.<Имя макета>.**

Например, УправлениеПечатью.ПолучитьМакет("Документ.\_ДемоСчетНаОплатуПокупателю.ПФ\_MXL\_СчетЗаказ");

### Разработка печатных форм с использованием макетов в формате офисных документов (Microsoft Word и OpenOffice Writer)

Помимо макетов в формате табличного документа (mxl), можно использовать макеты в форматах Microsoft Word (doc) и OpenOffice Writer (odt). Особенностью использования таких макетов при формировании печатных форм является использование клиентского контекста (подробности в разделе «[Формирование печатной формы в клиентском контексте](https://its.1c.ru/db/content/bsp23doc/src/3.33.%20%D0%BF%D0%B5%D1%87%D0%B0%D1%82%D1%8C.htm?_=1504708101#_формирование_печатной_формы)»).

В общем виде разработка такой печатной формы выглядит следующим образом:

1. Создание макета печатной формы.

2. Создание процедуры подготовки данных для печати.

3. Создание описания команды печати.

4. Создание обработчика команды печати.

#### Создание макета печатной формы

Макет печатной формы может быть подготовлен в одном из форматов: Microsoft Word или OpenOffice.org Writer. Макет представляет собой обычный офисный документ, состоящий из одной или нескольких областей следующих типов:

* ● **СтрокаТаблицы** – используется для вывода множества строк;
* ● **Список** – используется для вывода множества строк;
* ● **Общая** – используется однократно.

Области выделяются при помощи разделителей специального вида:

{v8 Область.<Имя области>}

...

{/v8 Область.<Имя области>}

Каждая область может содержать в себе маркеры параметров для подстановки значений из информационной базы:

{v8 <Имя параметра>}

Макет необходимо разместить в конфигурации (см. «[Размещение макетов печатных форм в конфигурации](https://its.1c.ru/db/content/bsp23doc/src/3.33.%20%D0%BF%D0%B5%D1%87%D0%B0%D1%82%D1%8C.htm?_=1504708101#_размещение_макетов_печатных)»).

#### Создание процедуры подготовки данных для печати

В модуле менеджера объекта необходимо добавить экспортную процедуру **ПолучитьДанныеПечати**

Описание параметров функции **ПолучитьДанныеПечати**:

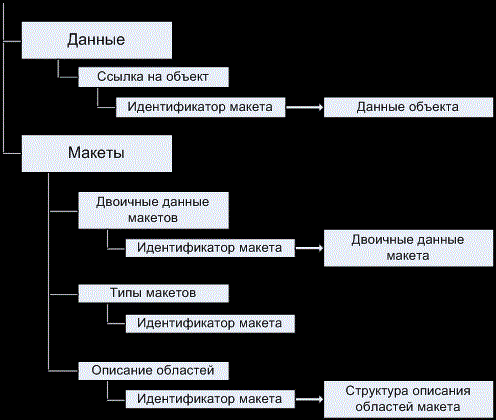
Таблица 3.69.

|  |  |
| --- | --- |
| **Параметр функции** | **Описание** |
| **МассивДокументов** | Массив ссылок на объекты, для которых запрашиваются данные для печати |
| **МассивИменМакетов** | Массив имен макетов, в которые подставляются данные для печати |
| **ДополнительныеПараметры** | Структура с дополнительными параметрами |

Возвращаемое значение функции – соответствие с двумя ключами: **Данные** и **Макеты**. В ключе **Данные**находится соответствие в формате:

* ● ключ – ссылка на объект информационной базы;
* ● значение – структура, где ключ – имя макета, а значение – структура с данными объекта (формат выбирается самостоятельно).

В ключе **Макеты** находится структура с тремя ключами: **ОписаниеОбластей**, **ТипыМакетов** и **ДвоичныеДанныеМакетов**. В ключе **ОписаниеОбластей** содержится структура, в которой в качестве ключа выступает имя макета, а значением является структура, содержащая описание области макета, формируемая с помощью вспомогательной функции **ДобавитьОписаниеОбласти**.



В ключе **ДвоичныеДанныеМакетов** находится соответствие, в котором ключом является имя макета, а значение содержит двоичные данные макета офисного документа.

#### Создание описания команды печати

В модуле менеджера объекта в процедуре **ДобавитьКомандыПечати** надо добавить описание команды (с использованием параметра **Обработчик**), например:

// Счет на оплату в Microsoft Word

КомандаПечати = КомандыПечати.Добавить();

КомандаПечати.МенеджерПечати = "Документ.\_ДемоСчетНаОплатуПокупателю";

КомандаПечати.Идентификатор = "СчетНаОплату(MSWord)";

КомандаПечати.Представление = НСтр("ru = 'Счет на оплату в Microsoft Word'");

КомандаПечати.Картинка = БиблиотекаКартинок.ФорматWord2007;

КомандаПечати.ПроверкаПроведенияПередПечатью = Истина;

КомандаПечати.Обработчик = "\_ДемоСтандартныеПодсистемыКлиент.ПечатьСчетовНаОплатуПокупателю";

Описание параметров процедуры **ДобавитьКомандыПечати** см. в разделе «[Разработка команд печати](https://its.1c.ru/db/content/bsp23doc/src/3.33.%20%D0%BF%D0%B5%D1%87%D0%B0%D1%82%D1%8C.htm?_=1504708101#_разработка_команд_печати)».

#### Создание обработчика команды печати

В параметре **Обработчик** необходимо указать имя функции, которой будет передано управление при выполнении команды печати. В клиентском общем модуле необходимо создать эту функцию и в ней реализовать подготовку печатной формы.

Для заполнения структуры описания областей макета необходимо использовать процедуру **УправлениеПечатью.ДобавитьОписаниеОбласти**. Первым параметром в ней выступает структура с описанием областей, вторым – имя области, как она задана в макете, третьим – тип области.

В макете имеются предопределенные области: **ВерхнийКолонтитул** и **НижнийКолонтитул**, их можно использовать наравне с другими областями.

Пример реализации см. в демонстрационной конфигурации, в модуле менеджера документа **\_ДемоСчетНаОплатуПокупателю** в процедуре **ДобавитьКомандыПечати** (в описании команд **Счет на оплату в Microsoft Word** и **Счет на оплату в OpenOffice.org Writer**), в процедуре **ПолучитьДанныеПечати**, а также в функции **ПечатьСчетовНаОплатуПокупателю** модуля **\_ДемоСтандартныеПодсистемыКлиент**.

**Важно!**

Формирование печатных форм в формате офисных документов недоступно в базовых версиях конфигураций (из-за ограничений платформы «1С:Предприятие», связанных с недоступностью COM-соединений). В базовых версиях конфигураций следует воздержаться от разработки таких команд печати.

### Размещение в командном интерфейсе

1. Для редактирования макетов печатных форм следует разместить в командном интерфейсе ответственного за изменение макетов регистр сведений **ПользовательскиеМакетыПечати**.

2. В персональных настройках пользователя следует разместить:

* ● Команду настройки каталога печати с кодом:

ОткрытьФорму("РегистрСведений.ПользовательскиеМакетыПечати.Форма.НастройкаКаталогаФайловПечати");

* ● Команду для задания действия при выборе макета:

ОткрытьФорму("РегистрСведений.ПользовательскиеМакетыПечати.Форма.ВыбораРежимаОткрытияМакета");

### Настройка прав доступа пользователей

Для настройки прав доступа пользователей к данным подсистемы «Печать» следует использовать роли, указанные ниже.

Таблица 3.70.

| **№** | **Роли и их назначение** |
| --- | --- |
| 1. | **ИзменениеМакетовПечатныхФорм**  Редактирование макетов печатных форм |
| 2. | **ВыводНаПринтерФайлБуферОбмена** (из подсистемы «Базовая функциональность»)  Используется для разрешения вывода печатных форм на принтер и отправки их по электронной почте |

Дополнительно следует создать вспомогательные роли или использовать подходящие роли, существующие в конфигурации, для обеспечения доступа к данным.

Таблица 3.71.

|  |  |
| --- | --- |
| **№** | **Вспомогательные роли и их назначение** |
| 1. | **<ВыводПечатнойФормыНаПринтер>**  Просмотр команды вывода печатных форм на принтер |

Пример настройки прав доступа пользователей:

Таблица 3.72.

| **№** | **Группа пользователей и ее функции** | **Состав ролей** |
| --- | --- | --- |
| 1. | Ответственный за поддержку макетов печатных форм | ● **БазовыеПрава** (из подсистемы «Базовая функциональность»),  ● **ЗапускТонкогоКлиента** (из подсистемы «Базовая функциональность»),  **● ИзменениеМакетовПечатныхФорм** |
| 2. | Пользователь, которому разрешено получать и формировать печатные формы, а также выводить их на печать и пользоваться командами «быстрой печати» на принтер | ● **БазовыеПрава** или **БазовыеПраваВнешнегоПользователя**(из подсистемы «Базовая функциональность»),  ● **ЗапускТонкогоКлиента** (из подсистемы «Базовая функциональность»),  ● **ВыводНаПринтерФайлБуферОбмена**(из подсистемы «Базовая функциональность»),  **● <ВыводПечатнойФормыНаПринтер>** |

## Использование при разработке конфигурации

### Разработка команд печати в расширении конфигурации

Вместо поставки внешних печатных форм в виде внешних обработок рекомендуется вести их разработку с помощью расширений конфигурации. Для поставки команд печати внешних печатных форм в расширении конфигурации необходимо:

1. Добавить в расширение конфигурации обработку и включить ее в состав подсистемы **ПодключаемыеОтчетыИОбработки**.

2. В модуле менеджера обработки определить процедуру **ПриОпределенииНастроек** в соответствии с шаблоном [подключения отчетов и обработок к другим объектам метаданных](https://its.1c.ru/db/content/bsp23doc/src/3.4.%20%D0%B1%D0%B0%D0%B7%D0%BE%D0%B2%D0%B0%D1%8F%20%D1%84%D1%83%D0%BD%D0%BA%D1%86%D0%B8%D0%BE%D0%BD%D0%B0%D0%BB%D1%8C%D0%BD%D0%BE%D1%81%D1%82%D1%8C.htm?_=1504708101#_подключение_отчетов_и) и реализовать команды печати:

#Область ПрограммныйИнтерфейс

// Определяет состав программного интерфейса для интеграции с конфигурацией.

//

// Параметры:

//   Настройки - Структура - Настройки интеграции этого объекта.

//       См. возвращаемое значение функции ПодключаемыеКоманды.НастройкиПодключаемыхОтчетовИОбработок().

//

Процедура ПриОпределенииНастроек(Настройки) Экспорт

Настройки.Размещение.Добавить(Метаданные.Документы.ИмяДокумента);

Настройки.ДобавитьКомандыПечати = Истина;

КонецПроцедуры

// Заполняет список команд печати.

//

// Параметры:

//   КомандыПечати - ТаблицаЗначений - Подробнее см. в УправлениеПечатью.СоздатьКоллекциюКомандПечати().

//

Процедура ДобавитьКомандыПечати(КомандыПечати) Экспорт

// Код по добавлению команд печати.

КонецПроцедуры

#КонецОбласти

// Формирует печатные формы.

//

// Параметры:

//  МассивОбъектов – Массив – ссылки на объекты, которые нужно распечатать;

//  ПараметрыПечати – Структура – дополнительные настройки печати;

//  КоллекцияПечатныхФорм – ТаблицаЗначений – сформированные табличные документы (выходной параметр)

//  ОбъектыПечати – СписокЗначений – значение – ссылка на объект;

//                                            представление – имя области, в которой был выведен объект (выходной параметр);

//  ПараметрыВывода – Структура – дополнительные параметры сформированных табличных документов (выходной параметр).

//

Процедура Печать(МассивОбъектов, ПараметрыПечати, КоллекцияПечатныхФорм, ОбъектыПечати, ПараметрыВывода) Экспорт

КонецПроцедуры

Аналогичным образом в расширение конфигурации можно включить сразу несколько обработок с внешними печатными формами. Кроме того, предусмотрена возможность разрабатывать команды печати в отчетах. Например, это может потребоваться, если для формирования печатной формы используется система компоновки данных (СКД).

См. примеры в демонстрационной конфигурации: обработка **\_ДемоПодключаемыеКомандыПечатьСчетовНаОплатуПокупателю** в составе расширения **\_ДемоПодключаемыеКоманды** и обработка **\_ДемоПечатьКарточкиОрганизации** в составе конфигурации.

При необходимости, разработанные таким образом обработки также можно перенести без каких-либо доработок из расширения конфигурации в состав самой конфигурации.

### Настройка обмена данными

Все объекты метаданных подсистемы, содержащие данные, рекомендуется включать в планы обмена распределенной ИБ (РИБ).

# 3.34. Поиск и удаление дублей

Подсистема «Поиск и удаление дублей» позволяет искать и удалять дубли элементов (справочников и т. п.), заменяя во всех местах использования дублирующие элементы выбранным.

## Настройка

В случае если в конфигурации не используется подсистема «Настройки программы», то в рабочем месте администратора программы необходимо разместить обработку **ПоискИУдалениеДублей**. См. пример в форме **Органайзер** обработки **ПанельАдминистрированияБСП**.

### Объединение дублей и поиск мест использования элементов списков

Принять решение по поводу состава объектов конфигурации, в списках которых востребована возможность объединения дублей и мест использования выбранных элементов. Это могут быть справочники, планы видов характеристик, планы счетов и планы видов расчета, в которых вследствие ручного ввода или синхронизации данных с другими программами часто возникают дубли и необходимо их оперативное устранение.

В формах списков этих объектов необходимо добавить две команды:

● **ОбъединитьВыделенные**, синоним «Объединить выделенные...»;

● **ПоказатьМестаИспользования**, синоним «Места использования».

Затем определить для них обработчики:

// СтандартныеПодсистемы.ПоискИУдалениеДублей

&НаКлиенте

Процедура ОбъединитьВыделенные(Команда)

ПоискИУдалениеДублейКлиент.ОбъединитьВыделенные(Элементы.Список);

КонецПроцедуры

&НаКлиенте

Процедура ПоказатьМестаИспользования(Команда)

ПоискИУдалениеДублейКлиент.ПоказатьМестаИспользования(Элементы.Список);

КонецПроцедуры

// Конец СтандартныеПодсистемы.ПоискИУдалениеДублей

Рекомендуется размещать указанные команды в подменю **Еще** (**Все действия**) и в контекстном меню списка. Пример внедрения см. в справочнике **\_ДемоФизическиеЛица**.

### Ограничения замены ссылок, поиска и удаления дублей

При использовании обработки **ЗаменаСсылок** и процедуры **ЗаменитьСсылки** общего модуля **ОбщегоНазначения** существует возможность контроля допустимости замены одной ссылки на другую. Для этого необходимо добавить в модуль менеджера контролируемого объекта (справочника, плана видов характеристик и т. п.) функцию **ВозможностьЗаменыЭлементов**. В ней реализовать алгоритм, который с прикладной точки зрения анализирует предлагаемые замены и возвращает описание ошибки о недопустимости замены одной ссылки на другую.

Также предусмотрена возможность реализовать собственную логику поиска дублей при вызове функции **НайтиДублиЭлемента** общего модуля **ПоискИУдалениеДублей** и при использовании обработки **ПоискИУдалениеДублей**. Для этого необходимо определить в модуле менеджера интересующего нас объекта две процедуры:

● процедуру **ПараметрыПоискаДублей**, в которой можно переопределить параметры поиска дублей, заданные пользователем или при программном поиске;

● процедуру **ПриПоискеДублей**, в которой реализовать собственную логику определения дублей.

Пример реализации, описания и использования этих функций можно найти в модуле менеджера справочника **\_ДемоНоменклатура**. Пример использования программного интерфейса – в обработчике записи модуля формы этого же справочника.

### Настройка прав доступа пользователей

Для настройки прав доступа пользователей к данным подсистемы нужно использовать следующие роли:

Таблица 3.73.

| **№** | **Роли и их назначение** |
| --- | --- |
| 1. | **ПолныеПрава** (из подсистемы «Базовая функциональность»)  Удаление дублей элементов с заменой во всех местах использования дублирующих элементов выбранным |

## Использование при разработке конфигурации

### Контроль замены ссылок при записи объектов и выполнение связанных действий

При замене ссылок в документах документы записываются без перепроведения, чтобы не нарушать последовательности. Как следствие, не срабатывает событие **ОбработкаПроведения** и вся прикладная логика проведения по регистрам. Подробнее  см. раздел «[Контроль замены ссылок при записи объектов и выполнение связанных действий](https://its.1c.ru/db/content/bsp23doc/src/3.4.%20%D0%B1%D0%B0%D0%B7%D0%BE%D0%B2%D0%B0%D1%8F%20%D1%84%D1%83%D0%BD%D0%BA%D1%86%D0%B8%D0%BE%D0%BD%D0%B0%D0%BB%D1%8C%D0%BD%D0%BE%D1%81%D1%82%D1%8C.htm?_=1504708101#_контроль_замены_ссылок)».

### Настройка обмена данными

Для настройки обмена данными никаких действий не требуется, так как подсистема не предоставляет собственных данных.

# 3.35. Полнотекстовый поиск

Подсистема «Полнотекстовый поиск» предназначена для включения в конфигурации возможности полнотекстового поиска данных. Индексация данных для полнотекстового поиска может проводиться регламентными заданиями **СлияниеИндексаППД** и **ОбновлениеИндексаППД**.

## Настройка

Для использования подсистемы в конфигурации необходимо разместить в командном интерфейсе обработку **ПоискВДанных**.

Пример реализации настроек см. в форме **УправлениеПолнотекстовымПоискомИИзвлечениемТекстов**обработки**ПанельАдминистрированияБСП**.

### Настройка прав доступа пользователей

Для настройки прав доступа пользователей к данным подсистемы «Полнотекстовый поиск» следует использовать роли:

Таблица 3.74.

| **№** | **Роли и их назначение** |
| --- | --- |
| 1. | **БазовыеПрава** (из подсистемы «Базовая функциональность»)  Использование возможностей полнотекстового поиска. Обновление и слияние индекса ППД |
| 2. | **ПолныеПрава** (из подсистемы «Базовая функциональность»)  Включение и отключение полнотекстового поиска. Обновление и слияние индекса ППД |

### Настройка обмена данными

Константу **ИспользоватьПолнотекстовыйПоиск** рекомендуется включать только в состав начального образа подчиненного узла распределенной ИБ. (см. раздел «[Особенности создания начального образа подчиненного узла распределенной ИБ](https://its.1c.ru/db/content/bsp23doc/src/3.28.%20%D0%BE%D0%B1%D0%BC%D0%B5%D0%BD%20%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D0%BC%D0%B8.htm?_=1504708101#_особенности_создания_начального)»).

# 3.36. Получение файлов из Интернета

Подсистема «Получение файлов из Интернета» добавляет в конфигурацию программный интерфейс для получения файлов из сети Интернет по протоколам HTTP, HTTPS и FTP и сохранения полученных файлов на клиенте, сервере или во временном хранилище.

## Настройка

Для использования подсистемы в конфигурации необходимо:

1. В случае если в конфигурации не используется подсистема «Настройки программы», в командном интерфейсе администратора разместить команду для открытия общей формы **ПараметрыПроксиСервера** по настройке прокси-сервера для доступа к Интернету на сервере «1С:Предприятия». Код команды:

ОткрытьФорму("ОбщаяФорма.ПараметрыПроксиСервера");

2. Разместить в форме персональных настроек работы с информационной базой (для пользователя информационной базы) команду настройки прокси-сервера для доступа к Интернету с клиентского места. Код команды:

ОткрытьФорму("ОбщаяФорма.ПараметрыПроксиСервера", Новый Структура("НастройкаПроксиНаКлиенте", Истина));

### Настройка прав доступа пользователей

Для настройки прав доступа пользователей к данным подсистемы «Получение файлов из Интернета» следует использовать роли:

Таблица 3.75.

| **№** | **Роли и их назначение** |
| --- | --- |
| 1. | **БазовыеПрава** (из подсистемы «Базовая функциональность»)  Получение файлов из Интернета, настройка параметров прокси-сервера на стороне клиентского приложения |
| 2. | **АдминистраторСистемы** (из подсистемы «Базовая функциональность»)  Настройка параметров прокси-сервера для сервера «1С:Предприятия» |

Пример настройки прав доступа пользователей:

Таблица 3.76.

| **№** | **Группа пользователей и ее функции** | **Состав ролей** |
| --- | --- | --- |
| 1. | Администратор | ● **АдминистраторСистемы** (из подсистемы «Базовая функциональность»),  ● **ПолныеПрава** (из подсистемы «Базовая функциональность») |
| 2. | Пользователь, которому разрешено получать файлы из Интернета:  ● Загрузка курсов валют | ● **БазовыеПрава** (из подсистемы «Базовая функциональность»),  ● **ЗапускТонкогоКлиента** (из подсистемы «Базовая функциональность»),  **● ПолучениеФайловИзИнтернет** |

## Использование при разработке конфигурации

Программный интерфейс подсистемы доступен через экспортные функции общих модулей:

● **ПолучениеФайловИзИнтернет**,

● **ПолучениеФайловИзИнтернетКлиент**.

### Настройка обмена данными

Все объекты метаданных подсистемы, содержащие данные, рекомендуется включить в планы обмена распределенной ИБ (РИБ).

Исключение из этого правила составляют:

● **Константа.НастройкаПроксиСервера**.

# 3.37. Пользователи

Подсистема «Пользователи» предназначена для просмотра и редактирования списка пользователей и внешних пользователей системы (элементы справочников **Пользователи** и **Внешние пользователи**), который синхронизируется со списком пользователей информационной базы.

Внешние пользователи предназначены для организации доступа к информационной системе тем пользователям, которые представлены в ней одним из объектов (например, **Сотрудники**, **Партнеры**, **Физические лица** и др.). Подсистема предоставляет средства для сопоставления внешних пользователей с объектами информационной базы.

## Настройка

Для использования подсистемы в конфигурации необходимо:

● Если в конфигурации уже имеются объекты, которые идентифицируют внешних пользователей, необходимо принять решение, какие из них будут использоваться и перечислить допустимые типы внешних пользователей:

* ● в определяемых типах **ВнешнийПользователь** (ссылки) и **ВнешнийПользовательОбъект** (объекты);
* ● в свойстве **Тип** **параметра команды** команды **ВнешнийДоступ** справочника **ВнешниеПользователи**;
* ● в определяемом типе **Пользователь** (ссылки). В состав этого типа также входит тип **СправочникСсылка.Пользователи**.

● Для использования возможности указывать значения реквизитов **Физическое лицо** и **Подразделение** в форме элемента справочника **Пользователи** необходимо задать состав определяемых типов **ФизическоеЛицо**и **Подразделение**. Если в конфигурации состав этих определяемых типов не задан, то в форме элемента справочника **Пользователи** эти реквизиты скрываются.

### Размещение в командном интерфейсе

Для использования подсистемы необходимо разместить в командном интерфейсе:

● Справочник **Пользователи**:

* ● для просмотра списка, свойств и выбора пользователей (обычными пользователями);
* ● для просмотра и редактирования состава и свойств пользователей (ответственными за список пользователей и администраторами).

● Справочник **ВнешниеПользователи**:

* ● для просмотра списка, свойств и выбора внешних пользователей (обычными пользователями);
* ● для просмотра и редактирования состава и свойств внешних пользователей (ответственными за список внешних пользователей и администраторами).

Для доступа пользователей к своим персональным настройкам необходимо создать и разместить в форме персональных настроек команду **Сведения о пользователе**.

См. пример размещения в демонстрационной конфигурации, в форме **\_ДемоМоиНастройки** на закладке **Общие**.

В случае если в конфигурации не используется подсистема «Настройки программы», в рабочем месте администратора программы необходимо разместить константы:

● **ИспользоватьВнешнихПользователей** – для включения/выключения возможности использования внешних пользователей;

● **ИспользоватьГруппыПользователей** – для включения/выключения возможности использования групп пользователей и групп внешних пользователей.

Для корректного обновления форм требуется отправлять оповещения об изменении констант.

См. пример в форме **НастройкиПользователейИПрав** обработки **ПанельАдминистрированияБСП**.

Для удобства работы на рабочий стол рекомендуется вынести формы тех подсистем, которые будут доступны внешнему пользователю после успешной аутентификации. Например, это может быть форма доступных анкет респондента или форма заказов покупателя.

### Назначение ролей

Роли отбираются по назначению при выборе в режиме **1С:Предприятие**. Например, роли только для внешних пользователей недоступны для пользователей. При входе пользователя в «1С:Предприятие» выполняется проверка ролей в целях безопасности (в модели сервиса проверяются роли пользователя, в локальном режиме проверяются роли внешнего пользователя). Также назначение ролей проверяются в профилях групп доступа (см. подсистему «[Управление доступом](https://its.1c.ru/db/content/bsp23doc/src/3.53.%20%D1%83%D0%BF%D1%80%D0%B0%D0%B2%D0%BB%D0%B5%D0%BD%D0%B8%D0%B5%20%D0%B4%D0%BE%D1%81%D1%82%D1%83%D0%BF%D0%BE%D0%BC.htm?_=1504708101#_управление_доступом)»).

При разработке ролей необходимо указывать роли специального назначения в процедуре **ПриОпределенииНазначенияРолей** общего модуля **ПользователиПереопределяемый**. Все остальные роли не требуется указывать. Назначение ролей подробно описано в комментарии к процедуре.

// Позволяет указать роли специального назначения. Все остальные роли не требуется указывать -

// это роли, которые предназначены для любых пользователей, кроме внешних пользователей.

//

// Параметры:

//  НазначениеРолей - Структура - со свойствами:

//   \* ТолькоДляАдминистраторовСистемы - Массив - имена ролей, которые при выключенном разделении

//     предназначены для любых пользователей, кроме внешних пользователей, а в разделенном режиме

//     предназначены только для администраторов сервиса, например:

//       Администрирование, ОбновлениеКонфигурацииБазыДанных, АдминистраторСистемы,

//     а также все роли с правами:

//       Администрирование,

//       Администрирование расширений конфигурации,

//       Обновление конфигурации базы данных.

//     Такие роли, как правило, существуют только в БСП и не встречаются в прикладных решениях.

//

//   \* ТолькоДляПользователейСистемы - Массив - имена ролей, которые при выключенном разделении

//     предназначены для любых пользователей, кроме внешних пользователей, а в разделенном режиме

//     предназначены только для неразделенных пользователей (сотрудников технической поддержки сервиса и

//     администраторов сервиса), например:

//       ДобавлениеИзменениеАдресныхСведений, ДобавлениеИзменениеБанков,

//     а также все роли с правами изменения неразделенных данных и следующими правами:

//       Толстый клиент,

//       Внешнее соединение,

//       Automation,

//       Режим все функции,

//       Интерактивное открытие внешних обработок,

//       Интерактивное открытие внешних отчетов.

//     Такие роли в большей части существует в БСП, но могут встречаться и в прикладных решениях.

//

//   \* ТолькоДляВнешнихПользователей - Массив - имена ролей, которые предназначены

//     только для внешних пользователей (роли со специально разработанным набором прав), например:

//       ДобавлениеИзменениеОтветовНаВопросыАнкет, БазовыеПраваВнешнегоПользователя.

//     Такие роли существуют и в БСП, и в прикладных решениях (если используются внешние пользователи).

//

//   \* СовместноДляПользователейИВнешнихПользователей - Массив - имена ролей, которые предназначены

//     для любых пользователей (и внутренних, и внешних, и неразделенных), например:

//       ЧтениеОтветовНаВопросыАнкет, ИспользованиеВариантовОтчетов.

//     Такие роли существуют и в БСП, и в прикладных решениях (если используются внешние пользователи).

//

Процедура ПриОпределенииНазначенияРолей(НазначениеРолей) Экспорт

// ТолькоДляВнешнихПользователей.

НазначениеРолей.ТолькоДляВнешнихПользователей.Добавить(

Метаданные.Роли.\_ДемоОплатаСчетовВнешнимиПользователями.Имя);

НазначениеРолей.ТолькоДляВнешнихПользователей.Добавить(

Метаданные.Роли.\_ДемоЧтениеДанныхОбъектовАвторизации.Имя);

// СовместноДляПользователейИВнешнихПользователей.

НазначениеРолей.СовместноДляПользователейИВнешнихПользователей.Добавить(

Метаданные.Роли.\_ДемоЧтениеДанныхДляОтветовНаВопросыАнкет.Имя);

КонецПроцедуры

### Отображение внешних пользователей в списках

В конфигурациях, в которых предоставляется партнерам или сотрудникам доступ к программе извне, рекомендуется в форме списка справочника добавить колонку показывающую наличие внешнего доступа у этого партнера или сотрудника. Для этого, например, в форме списка справочника **\_ДемоПартнеры**, являющегося объектом авторизации справочника **Внешние пользователи**, необходимо в запросе динамического списка сделать левое соединение со справочником **ВнешниеПользователи** (по реквизиту **ОбъектАвторизации**).

Было:

ВЫБРАТЬ РАЗРЕШЕННЫЕ

Справочник\_ДемоПартнеры.Ссылка КАК Ссылка,

Справочник\_ДемоПартнеры.Наименование КАК Наименование

ИЗ

Справочник.\_ДемоПартнеры КАК Справочник\_ДемоПартнеры

Стало:

ВЫБРАТЬ РАЗРЕШЕННЫЕ

НЕ ВнешниеПользователи.Ссылка ЕСТЬ NULL Как ВнешнийДоступ,

Справочник\_ДемоПартнеры.Ссылка КАК Ссылка,

Справочник\_ДемоПартнеры.Наименование КАК Наименование

ИЗ

Справочник.\_ДемоПартнеры КАК Справочник\_ДемоПартнеры

ЛЕВОЕ СОЕДИНЕНИЕ Справочник.ВнешниеПользователи КАК ВнешниеПользователи

ПО Справочник\_ДемоПартнеры.Ссылка = ВнешниеПользователи.ОбъектАвторизации

И НЕ ВнешниеПользователи.Недействителен

И НЕ ВнешниеПользователи.ПометкаУдаления

● В событие**ПриСозданиеНаСервере** разместить следующий код:

ВнешниеПользователи.НастроитьОтображениеСпискаВнешнихПользователей(ЭтотОбъект);

См. пример в форме **ФормаСписка** справочника **\_ДемоПартнеры в демонстрационной базе**.

### Настройка прав доступа пользователей

Для настройки прав доступа пользователей к данным подсистемы «Пользователи» следует использовать роли, указанные ниже.

Таблица 3.77.

| **№** | **Роли и их назначение** |
| --- | --- |
| 1. | **Базовые права** (из подсистемы «Базовая функциональность»)  Чтение списка пользователей без ограничений |
| 2. | **ДобавлениеИзменениеПользователей** (используется совместно с ролью **БазовыеПрава**).  Добавление и изменение пользователей и групп пользователей без настройки доступа к программе.  Дополнительная роль, предназначенная для секретарей, менеджеров по персоналу: добавление новых пользователей при приеме на работу, исправление ФИО контактной информации и других нормативных свойств пользователя, создание новых групп пользователей.  Сокращает рутинную нагрузку на администратора в крупных компаниях |
| 3. | **ЧтениеВнешнихПользователей** (используется совместно со вспомогательной ролью **<ЧтениеДанныхОбъектовАвторизации>**).  Чтение внешних пользователей и групп внешних пользователей (просмотр списка внешних пользователей без ограничений) |
| 4. | **ДобавлениеИзменениеВнешнихПользователей** (используется совместно со вспомогательной ролью **<ЧтениеДанныхОбъектовАвторизации>**).  Добавление и изменение внешних пользователей и групп внешних пользователей без настройки доступа к программе (просмотр списка внешних пользователей без ограничений).  Дополнительная роль, предназначенная для помощников руководителей менеджеров по продажам и закупкам: добавление новых внешних пользователей при новых контактах, создание новых групп внешних пользователей.  Сокращает рутинную нагрузку на администратора в крупных компаниях |
| 5. | **ПолныеПрава** (из подсистемы «Базовая функциональность»)  Добавление и изменение пользователей, групп пользователей, внешних пользователей, групп внешних пользователей, настроек входа и прав доступа.  Изменение пароля, имени для входа и остальных свойств любого пользователя и внешнего пользователя.  Включение и отключение использования внешних пользователей.  Удаление объектов подсистемы, помеченных на удаление |

Дополнительно следует создать вспомогательные роли или использовать подходящие роли, существующие в конфигурации, для обеспечения доступа к данным, которые не относятся к подсистеме «Пользователи», но требуются для работы с ней.

Таблица 3.78.

| **№** | **Вспомогательные роли и их назначение** |
| --- | --- |
| 1. | **<ЧтениеДанныхОбъектовАвторизации>**  Чтение объектов метаданных, значения которых могут выступать в качестве объектов авторизации |

Пример настройки прав доступа пользователей:

Таблица 3.79.

| **№** | **Группа пользователей и ее функции** | **Состав ролей** |
| --- | --- | --- |
| 1. | Администратор | ● **ПолныеПрава** (из подсистемы «Базовая функциональность») |
| 2. | Ответственный за список пользователей (дополнительно) | **● ДобавлениеИзменениеПользователей** |
| 3. | Ответственный за список внешних пользователей (дополнительно) | ● **ДобавлениеИзменениеВнешнихПользователей**,  **● <ЧтениеДанныхОбъектовАвторизации>** |
| 4. | Пользователь, работающий с внешними пользователями (дополнительно) | **● ЧтениеВнешнихПользователей** |
| 5. | Внешний пользователь | ● **БазовыеПраваВнешнегоПользователя** (из подсистемы «Базовая функциональность»),  **● ЗапускТонкогоКлиента** (из подсистемы «Базовая функциональность») |

### Особые случаи внедрения подсистемы

● [Внедрение подсистемы «Пользователи» без подсистемы «Контактная информация»](https://its.1c.ru/db/content/bsp23doc/src/3.59.%20%D0%BE%D1%81%D0%BE%D0%B1%D1%8B%D0%B5%20%D1%81%D0%BB%D1%83%D1%87%D0%B0%D0%B8%20%D0%B2%D0%BD%D0%B5%D0%B4%D1%80%D0%B5%D0%BD%D0%B8%D1%8F%20%D0%BF%D0%BE%D0%B4%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC.htm?_=1504708101#_внедрение_подсистемы_).

● [Внедрение подсистемы «Пользователи» без подсистемы «Свойства»](https://its.1c.ru/db/content/bsp23doc/src/3.59.%20%D0%BE%D1%81%D0%BE%D0%B1%D1%8B%D0%B5%20%D1%81%D0%BB%D1%83%D1%87%D0%B0%D0%B8%20%D0%B2%D0%BD%D0%B5%D0%B4%D1%80%D0%B5%D0%BD%D0%B8%D1%8F%20%D0%BF%D0%BE%D0%B4%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC.htm?_=1504708101#_внедрение_подсистемы_).

## Использование при разработке конфигурации

При создании новых прикладных объектов метаданных нужно повторно выполнять соответствующую часть процедуры настройки, описанной выше (например, пересматривать состав типов объектов авторизации).

Для получения ссылки на пользователя, который произвел вход в информационную базу, рекомендуется использовать функцию **АвторизованныйПользователь** общего модуля **ПользователиКлиентСервер**, которая возвращает ссылку на элемент справочника **Пользователи** или элемент справочника **ВнешниеПользователи**.

В коде, вызов которого предполагается только в сеансе пользователей, рекомендуется использовать функцию **ТекущийПользователь** общего модуля **ПользователиКлиентСервер**, которая возвращает ссылку на элемент справочника **Пользователи**. При вызове функции в сеансе внешнего пользователя будет вызвано исключение.

В коде, вызов которого предполагается только в сеансе внешних пользователей, рекомендуется использовать функцию **ТекущийВнешнийПользователь**, которая возвращает ссылку на элемент справочника **ВнешниеПользователи**. При вызове функции в сеансе пользователя будет вызвано исключение.

### Недействительные пользователи

В формах пользователя и внешнего пользователя предусмотрен флажок **Недействителен**, связанный с реквизитом **Недействителен** (по умолчанию – **Ложь**). При установке флажка **Недействителен** пользователь скрывается в списке пользователей, списке выбора пользователей, а также при подборе по строке. Для того чтобы увидеть всех пользователей в списке пользователей и списке выбора пользователей, надо установить флажок **Показывать недействительных пользователей**.

При обычном использовании справочников описанное поведение поддерживается автоматически. При открытии списка пользователей или списка выбора пользователей флажок **Показывать недействительных пользователей** снят. При подборе по строке в параметры получения данных выбора автоматически вставляется отбор **Недействителен = Ложь** (если отбор для реквизита **Недействителен** еще не указан).

При обсобых случаях использования следует поддержать описанное поведение самостоятельно:

● При разработке своего списка выбора пользователей. Например, форма, аналогичная общей форме **ВыборИсполнителяБизнесПроцесса**.

● При подготовке отчетов, в которые могут попасть недействительные пользователи. Например, в конфигурации «Документооборот» в отчете по задачам определяется, к какому отделу относится задача, выданная роли. Отдел определяется по пользователям, входящим в роль. Недействительные пользователи будут исключены из этого списка – их отделы не попадут в отчет.

● При обращении к справочнику на языке запросов или из кода.

### Служебные пользователи

В справочнике пользователей предусмотрен реквизит **Служебный**. Этот реквизит применяется в модели сервиса. Если значение реквизита установлено **Истина**, пользователь скрывается в списке пользователей, списке выбора пользователей, а также при подборе по строке.

Как и для флажка **Недействителен**, при обычном использовании справочника описанное поведение поддерживается автоматически, а при особых случаях использования следует поддержать описанное поведение самостоятельно.

### Настройка обмена данными

Для настройки обмена данными следует руководствоваться общими правилами.

В планы обмена распределенной информационной базы (РИБ) рекомендуется включать все объекты метаданных подсистемы, кроме следующих:

● константа**НастройкиВходаПользователей**,

● регистр сведений **СведенияОПользователях**.

В планах обмена распределенной информационной базы (РИБ) рекомендуется отключать регистрацию изменений для следующих объектов метаданных подсистемы (см. также раздел «[Особенности создания начального образа подчиненного узла распределенной ИБ](https://its.1c.ru/db/content/bsp23doc/src/3.28.%20%D0%BE%D0%B1%D0%BC%D0%B5%D0%BD%20%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D0%BC%D0%B8.htm?_=1504708101#_особенности_создания_начального)»):

● константа**ПараметрыРаботыПользователей**.

Из остальных планов обмена (не РИБ) рекомендуется исключать следующие объекты метаданных:

● константа**ПараметрыРаботыПользователей**,

● константа**НастройкиВходаПользователей**,

● регистр сведений **СведенияОПользователях**.

# 3.38. Префиксация объектов

Подсистема «Префиксация объектов» предназначена для автоматического назначения префиксов объектам с учетом настроек программы. Префиксация объектов ведется в разрезах информационных баз и элементов справочника **Организации**.

## Настройка

Настройка подсистемы «Префиксация объектов» различается в зависимости от внедрения совместно с подсистемой «Обмен данными» и наличия в конфигурации справочника **Организации**. Настройка подсистемы сводится к созданию подписок на события для задания номера (или кода) объекта и подписки для очистки номера объекта.

### Создание подписок для установки префикса в номере (коде) объекта

Следует создать необходимое количество подписок на события для задания номера или кода объекта при его записи, если номер или код не заполнены. Значения свойств подписок задать согласно таблице.

Таблица 3.80.

| **Имя свойства** | **Описание** |
| --- | --- |
| **Имя** | Имя следует задавать для удобства визуального восприятия подписки и с учетом ограничений, налагаемых на имена объектов метаданных, например **УстановитьПрефиксИнформационнойБазыИОрганизацииНомеруДокумента** |
| **Источник** | Источниками событий подписки могут быть только объекты типов: **Документы**, **Справочники**,**Планы видов характеристик**, **Бизнес-процессы**. В качестве источников для одной подписки могут быть использованы объекты только одного типа |
| **Событие** | **ПриУстановкеНовогоКода** или **ПриУстановкеНовогоНомера** |
| **Обработчик** | В качестве процедур – обработчиков подписок следует выбирать экспортные процедуры общего модуля **ПрефиксацияОбъектовСобытия** |

**Важно!**

Для объектов **Задача** префикс номеров устанавливается подсистемой «Бизнес-процессы и задачи». Использование подсистемы префиксации для объектов **Задача** будет вызывать ошибки в работе конфигурации.

В общем модуле **ПрефиксацияОбъектовСобытия** предусмотрено три процедуры-обработчика подписок, использование которых различается в зависимости от наличия в конфигурации справочника **Организации** и подсистемы «Обмен данными»:

● **УстановитьПрефиксОрганизации** – следует использовать в том случае, если требуется выполнять префиксацию объектов только в разрезе организаций.

● **УстановитьПрефиксИнформационнойБазы** – следует использовать в том случае, если используется подсистема «Обмен данными», а префиксацию объектов в разрезе организаций выполнять не требуется. К таким типам объектов в основном относятся справочники и планы видов характеристик.

● **УстановитьПрефиксИнформационнойБазыИОрганизации** – следует использовать в том случае, если требуется выполнять префиксацию объектов в разрезе информационных баз и организаций одновременно. К таким типам объектов в основном относятся документы.

Разработчик может выбирать процедуры-обработчики в зависимости от требуемой бизнес-логики.

Если в конфигурации имеется справочник **Организации**, то для возможности префиксации объектов в разрезе организаций следует создать следующие объекты метаданных:

● реквизит **Префикс** (**Строка**, 2) у справочника **Организации**;

● параметр функциональной опции **Организация** с использованием справочника **Организации**;

● функциональная опция **ПрефиксыОрганизаций**с хранением в реквизите справочника **Организации**.

### Создание подписок для переназначения номера (кода) объекта

Следует создать необходимое количество подписок на события для переназначения номера (кода) объекта при изменении даты или значения реквизита **Организация** в шапке объекта. После очистки объекту будет назначен новый код или номер.

Значения свойств подписок надо задать согласно таблице:

Таблица 3.81.

|  |  |
| --- | --- |
| **Имя свойства** | **Описание** |
| **Имя** | Имя следует задавать для удобства визуального восприятия подписки и с учетом ограничений, налагаемых на имена объектов метаданных, например, **ПроверитьНомерДокументаПоДатеИОрганизации** |
| **Источник** | Источниками событий подписки могут быть только объекты типов: **Документы**, **Справочники**,**Планы видов характеристик**, **Бизнес-процессы**. В качестве источников для одной подписки могут быть использованы объекты только одного типа |
| **Событие** | **ПередЗаписью** |
| **Обработчик** | В качестве процедур – обработчиков подписок следует выбирать экспортные процедуры общего модуля **ПрефиксацияОбъектовСобытия** |

В общем модуле **ПрефиксацияОбъектовСобытия** предусмотрено пять процедур – обработчиков подписок, каждую из которых необходимо использовать исходя из типа объектов – источников подписок и наличия реквизита **Организация** в шапке этих объектов:

● **ПроверитьКодСправочникаПоОрганизации** – следует использовать для справочников, у которых нумерация кодов выполняется в разрезе справочника **Организации**.

● **ПроверитьНомерБизнесПроцессаПоДате** – следует использовать для бизнес-процессов, у которых нумерация выполняется только в пределах заданной периодичности.

● **ПроверитьНомерБизнесПроцессаПоДатеИОрганизации** – следует использовать для бизнес-процессов, у которых нумерация выполняется в пределах заданной периодичности и в разрезе справочника **Организации**.

● **ПроверитьНомерДокументаПоДате** – следует использовать для документов, у которых нумерация выполняется только в пределах заданной периодичности.

● **ПроверитьНомерДокументаПоДатеИОрганизации** – следует использовать для документов, у которых нумерация выполняется в пределах заданной периодичности и в разрезе справочника **Организации**.

В большинстве случаев в конфигурации требуется создание только трех подписок: двух для назначения префиксов справочникам и документам и одной для очистки номеров документов. Пример для создания таких подписок представлен в таблице.

Таблица 3.82.

| **Имя подписки** | **Обработчик** |
| --- | --- |
| **УстановитьПрефиксИнформационнойБазыКодуСправочника** | **ПрефиксацияОбъектовСобытия.УстановитьПрефиксИнформационнойБазы** |
| **УстановитьПрефиксИнформационнойБазыИОрганизацииНомеруДокумента** | **ПрефиксацияОбъектовСобытия.УстановитьПрефиксИнформационнойБазыИОрганизации** |
| **ПроверитьНомерДокументаПоДатеИОрганизации** | **ПрефиксацияОбъектовСобытия.ПроверитьНомерДокументаПоДатеИОрганизации** |

Если объекту требуется устанавливать префикс организации, то в составе реквизитов шапки объекта должен присутствовать реквизит **Организация**. Обязательные значения свойств реквизита указаны в таблице.

Таблица 3.83.

| **Свойство** | **Значение** |
| --- | --- |
| **Имя** | **Организация** |
| **Тип** | **СправочникСсылка.Организации** |

### Настройка прав доступа пользователей

Настройка прав доступа пользователей к данным подсистемы «Префиксация объектов» не требуется.

## Использование при разработке конфигурации

### Формат префиксов и логика их установки объектам

Назначение префикса номеру или коду объекта происходит в момент первой записи объекта, если код или номер не были назначены вручную. Назначение префикса также происходит, если номер или код объекта были очищены перед его записью.

Префикс имеет фиксированную длину 3 или 5 символов. В префиксе нельзя использовать символ «-» (дефис); он используется для разделения префикса c номером и добавляется автоматически. Соответственно префикс формируется по одному из двух шаблонов: «**ОРИБ-**» или «**ИБ-**», где ОР – 2 символа префикса справочника**Организации**; ИБ – 2 символа префикса информационной базы; «-» (дефис) – разделитель префикса.

Для конфигураций, в которых используется подсистема префиксации, длину номеров документов и кодов справочников рекомендуется устанавливать не менее 11 символов.

Если префикс информационной базы или префикс организации не указан, то считается, что такой префикс отсутствует, он заменяется двумя нулями – «00».

Для предопределенных элементов ссылочных типов данных, создаваемых в режиме конфигуратора, рекомендуется назначать префикс кода элемента в момент его создания. Как правило, для таких элементов требуется назначить только префикс информационной базы. Например, для преопределенного элемента справочника в конфигураторе может быть назначен такой код: «УТ-00000001». Если этого не выполнить, то, если настроен обмен данными между разными конфигурациями, для справочников с предопределенными элементами может быть нарушена уникальность кодов этих элементов.

### Программный интерфейс

Программный интерфейс подсистемы доступен через экспортные процедуры-обработчики общих модулей **ПрефиксацияОбъектовСобытия** и **ПрефиксацияОбъектовКлиентСерверПереопределяемый**.

Для взаимодействия с подсистемой печати предусмотрена экспортная функция общего модуля. Функция позволяет сформировать номер документа для вывода в печатной форме: **ПрефиксацияОбъектовСобытия.ПолучитьНомерНаПечать**.

Функция отбрасывает префикс организации, отбрасывает префикс информационной базы (опционально), отбрасывает пользовательские префиксы (опционально), удаляет лидирующие нули в номере объекта.

**Пример**

Документ имеет номер ФР0Г-А001234. После отбрасывания префикса организации и незначащих нулей номера для печати получим значение: Г-А1234.

В подсистеме предусмотрен обработчик события «При получении номера на печать». Обработчик срабатывает перед стандартной обработкой и предназначен для формирования номера объекта нестандартным способом. Например, когда формат номера документа отличается от стандартного. Для задания кода обработчика используется процедура **ПрефиксацияОбъектовКлиентСерверПереопределяемый.ПриПолученииНомераНаПечать**.

### Дополнительные возможности по настройке формата префиксов

Подсистема предоставляет возможность добавлять к префиксу произвольную последовательность символов, например, дополнительный префикс «А» для документов **Счет-фактура на аванс**. Для этого в модуле объекта, в обработчике **ПриУстановкеНовогоНомера** следует задать значение переменной **Префикс**.

Пример номера документа с установленным префиксом: ФР0Г-А001234 – документ № 1234 с пользовательским префиксом «А», заведенный на организацию «ФР», созданный в информационной базе «Г».

### Переопределение имени реквизита, который содержит ссылку на организацию

В подсистеме предусмотрена стандартная обработка установки префикса организации. Стандартная обработка подразумевает, что значение ссылки на организации выбирается из реквизита шапки объекта с именем **Организация**. Однако в некоторых объектах реквизит может именоваться иначе (например, **Владелец** или **ГоловнаяОрганизация**). Если ссылка на организацию в объекте располагается в реквизите с именем, отличным от стандартного имени **Организация**, то необходимо воспользоваться обработчиком **ПолучитьПрефиксообразующиеРеквизиты** в переопределяемом модуле подсистемы **ПрефиксацияОбъектовПереопределяемый**.

### Настройка обмена данными

Для настройки обмена данными никаких действий не требуется, так как подсистема не предоставляет собственных данных.

# 3.39. Присоединенные файлы

Подсистема «Присоединенные файлы» предоставляет средства работы с файлами-вложениями, присоединенными к произвольным объектам информационной базы ссылочного типа. Хранение файлов может быть организовано как в информационной базе, так и внешним образом – в томах (сетевых ресурсах).

Поддерживаются такие возможности работы с файлами, как добавление файлов в базу с диска или сканера, электронная подпись и шифрование, поддержка группового изменения и полнотекстового поиска по присоединенным файлам.

## Настройка

Принять решение по поводу состава объектов конфигурации, которые должны содержать присоединенные файлы («объекты с файлами»). Такими объектами, как правило, являются справочники или документы, при которых могут содержаться файлы-вложения. Примеры: справочник **Товары** с присоединенными к нему файлами-спецификациями, сертификатами и т. п.; справочник **Должности** с приложенными к нему текстами должностных обязанностей, требованиями к соискателю и т. п.

Выполнить общую настройку:

1. Выполнить настройку общей файловой функциональности – см. раздел «[Файловые функции](https://its.1c.ru/db/content/bsp23doc/src/3.55.%20%D1%84%D0%B0%D0%B9%D0%BB%D0%BE%D0%B2%D1%8B%D0%B5%20%D1%84%D1%83%D0%BD%D0%BA%D1%86%D0%B8%D0%B8.htm?_=1504708101#_файловые_функции)».

2. Создать подписки:

Таблица 3.84.

|  |  |
| --- | --- |
| **ПереопределитьПолучаемуюФормуПрисоединенногоФайла** | |
| Событие | **ОбработкаПолученияФормы** |
| Обработчик | **ПрисоединенныеФайлыКлиентСервер.**  **ПереопределитьПолучаемуюФормуПрисоединенногоФайла** |
| **УстановитьПометкуУдаленияПрисоединенныхФайловДокументов** | |
| Событие | **ПередЗаписью** |
| Обработчик | **ПрисоединенныеФайлы.**  **УстановитьПометкуУдаленияПрисоединенныхФайловДокументов** |

Для каждого «объекта с файлами» выполнить настройку:

1. Создать справочник для хранения присоединенных файлов. Для этого в качестве заготовки скопировать в конфигурацию справочник **\_ДемоПроектыПрисоединенныеФайлы** из демонстрационной конфигурации и задать ему имя по шаблону:

**<Префикс>ПрисоединенныеФайлы**,

где **<Префикс>** – имя объекта метаданных, для которого настраиваются присоединенные файлы. Например, для справочника **Номенклатура** справочник с файлами должен называться **НоменклатураПрисоединенныеФайлы**. Задать синоним, например: **Присоединенные файлы** (**Номенклатура**).

2. У реквизита **ВладелецФайла** установить тип – «объект с файлами». Например, **СправочникСсылка.Номенклатура**.

3. Включить в состав определяемых типов **ПрисоединенныйФайл** (ссылки) и **ПрисоединенныйФайлОбъект**(объекты) справочник, созданный на шаге 1. Например, **СправочникСсылка.НоменклатураПрисоединенныеФайлы**.

4. Расширить состав определяемых типов **ВладелецПрисоединенныхФайлов** (ссылки) и **ВладелецПрисоединенныхФайловОбъект** (объекты, кроме документов), добавив в него тип «объект с файлами». Например **СправочникСсылка.Номенклатура**.

5. Расширить состав типов свойства **Источник** у подписки **ПереопределитьПолучаемуюФормуПрисоединенногоФайла**, включив в него тип – справочник с файлами, созданный на шаге 1. Например, **СправочникМенеджер.НоменклатураПрисоединенныеФайлы**.

6. Расширить состав типов свойства **Источник** подписки **УстановитьПометкуУдаленияПрисоединенныхФайловДокументов,** включив в него тип – «объект с файлами» (только документы). Например, **ДокументОбъект.СчетНаОплатуПокупателю.**

7. Если при интерактивном копировании объекта, содержащего присоединенные файлы, требуется их автоматическое копирование в новый объект, то в форме объекта необходимо:

* ● в параметры формы добавить ключевой параметр **ЗначениеКопирования** такого же типа, как и сам объект;
* ● в модуле формы в процедуру **ПриЗаписиНаСервере** вставить следующий код:

ПрисоединенныеФайлы.ПриЗаписиНаСервере(Отказ, ТекущийОбъект, ПараметрыЗаписи, Параметры);

### Настройка прав доступа пользователей

Использование присоединенных файлов в общем случае предполагает наличие двух ролей: для просмотра (открытия) и редактирования файлов. Предполагается, что роли настраиваются по владельцу файлов. Ограничения доступа на уровне записей к присоединенным файлам устанавливаются по владельцу.

Таблица 3.85.

| **№** | **Роли и их назначение** |
| --- | --- |
| 1. | **ПолныеПрава** (из подсистемы «Базовая функциональность»)  Удаление помеченных на удаление присоединенных файлов |

Дополнительно следует создать вспомогательные роли или использовать подходящие роли, существующие в конфигурации, для добавления/изменения и чтения «объектов с файлами». В этом случае настройка доступа к справочнику присоединенных файлов устанавливается аналогично «объекту с файлами».

Таблица 3.86.

| **№** | **Вспомогательные роли и их назначение** |
| --- | --- |
| 1. | **<ДобавлениеИзменениеНоменклатуры>**  Добавление/изменение «объектов с файлами». Добавление/изменение присоединенных файлов, редактирование, подписание, шифрование |
| 2. | **<ЧтениеНоменклатуры>**  Чтение «объектов с файлами». Чтение присоединенных файлов, открытие карточки присоединенного файла, открытие файла во внешней программе |

Пример настройки прав доступа пользователей приведен ниже.

Таблица 3.87.

| **№** | **Группа пользователей и ее функции** | **Состав ролей** |
| --- | --- | --- |
| 1. | Администратор  Общее администрирование системы | ● **ПолныеПрава** (из подсистемы «Базовая функциональность») |
| 2. | Менеджер  Добавление, изменение «объектов с файлами» и их файлов. Редактирование файлов, подписание файлов, шифрование, обновление данных файла из другого файла | ● **БазовыеПрава** (из подсистемы «Базовая функциональность»),  ● **ЗапускТонкогоКлиента** (из подсистемы «Базовая функциональность»),  ●**<ДобавлениеИзменениеНоменклатуры>**(из прикладной конфигурации) |
| 3. | Пользователь  Просмотр списка файлов, открытие карточки файла, открытие файла для просмотра | ● **БазовыеПрава** (из подсистемы «Базовая функциональность»),  ● **ЗапускТонкогоКлиента** (из подсистемы «Базовая функциональность»),  ● **<ЧтениеНоменклатуры>** (из прикладной конфигурации) |

### Особые случаи внедрения подсистемы

● [Внедрение подсистемы «Работа с файлами» вместе с подсистемой «Присоединенные файлы»](https://its.1c.ru/db/content/bsp23doc/src/3.59.%20%D0%BE%D1%81%D0%BE%D0%B1%D1%8B%D0%B5%20%D1%81%D0%BB%D1%83%D1%87%D0%B0%D0%B8%20%D0%B2%D0%BD%D0%B5%D0%B4%D1%80%D0%B5%D0%BD%D0%B8%D1%8F%20%D0%BF%D0%BE%D0%B4%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC.htm?_=1504708101#_внедрение_подсистемы_).

## Использование при разработке конфигурации

При создании новых прикладных объектов метаданных нужно повторно выполнять соответствующую часть процедуры настройки, описанной выше.

В случае если на форме того или иного объекта требуется разместить поле для выбора присоединяемого файла, необходимо переопределить стандартное поведение кнопок открытия и выбора объекта (файла).

Для открытия формы присоединенного файла необходимо использовать процедуру **ПрисоединенныеФайлыКлиент.ОткрытьФормуПрисоединенногоФайла()**.

Пример обработчика события **Открытие** элемента формы из демонстрационной конфигурации:

&НаКлиенте

Процедура РегламентПроектаОткрытие(Элемент, СтандартнаяОбработка)

ПрисоединенныеФайлыКлиент.ОткрытьФормуПрисоединенногоФайла(Объект.РегламентПроекта, СтандартнаяОбработка);

КонецПроцедуры

Для открытия формы выбора файлов по владельцу необходимо использовать процедуру **ПрисоединенныеФайлыКлиент.ОткрытьФормуВыбораФайлов**.

Пример обработчика события **Начало выбора** элемента формы из демонстрационной конфигурации:

&НаКлиенте

Процедура РегламентПроектаНачалоВыбора(Элемент, ДанныеВыбора, СтандартнаяОбработка)

ПрисоединенныеФайлыКлиент.ОткрытьФормуВыбораФайлов(Объект.Ссылка, Элемент, СтандартнаяОбработка);

КонецПроцедуры

Если в списке «объектов с файлами» требуется вывести признак наличия присоединенных к объекту файлов, то необходимо в запросе динамического списка сделать левое соединение с регистром сведений **НаличиеПрисоединенныхФайлов** (по измерению **ОбъектСФайлами**).

Пример запроса из демонстрационной конфигурации (форма списка справочника **\_ДемоПроекты**):

ВЫБРАТЬ

Справочник\_ДемоПроекты.Ссылка,

Справочник\_ДемоПроекты.ПометкаУдаления,

Справочник\_ДемоПроекты.Предопределенный,

Справочник\_ДемоПроекты.Код,

Справочник\_ДемоПроекты.Наименование,

Справочник\_ДемоПроекты.РеквизитДопУпорядочивания,

ВЫБОР

КОГДА НаличиеПрисоединенныхФайлов.ЕстьФайлы ЕСТЬ NULL ТОГДА

0

КОГДА НаличиеПрисоединенныхФайлов.ЕстьФайлы ТОГДА

1

ИНАЧЕ

0

КОНЕЦ КАК ЕстьФайлы

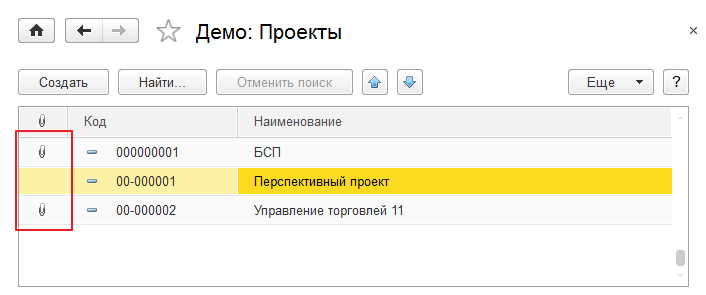
ИЗ

Справочник.\_ДемоПроекты КАК Справочник\_ДемоПроекты

ЛЕВОЕ СОЕДИНЕНИЕ РегистрСведений.НаличиеПрисоединенныхФайлов КАК НаличиеПрисоединенныхФайлов

ПО Справочник\_ДемоПроекты.Ссылка = НаличиеПрисоединенныхФайлов.ОбъектСФайлами

Пример списка «со скрепкой»:



### Настройка обмена данными

Для настройки обмена данными следует руководствоваться общими правилами. В план обмена необходимо включить все объекты метаданных, содержащие данные.

Регистр сведений **ПрисоединенныеФайлы** нужно использовать только на момент создания начального образа подчиненного узла в распределенной ИБ (см. раздел [«Особенности создания начального образа подчиненного узла распределенной ИБ»](https://its.1c.ru/db/content/bsp23doc/src/3.28.%20%D0%BE%D0%B1%D0%BC%D0%B5%D0%BD%20%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D0%BC%D0%B8.htm?_=1504708101#_особенности_создания_начального)).

# 3.40. Проверка легальности получения обновлений

Подсистема «Проверка легальности получения обновлений» позволяет запросить у пользователя подтверждение, что файлы обновления были получены легальным способом. Может использоваться как перед обновлением информационной базы (после того как обновление уже было применено к базе, но до первого запуска) или непосредственно перед обновлением конфигурации средствами подсистемы «Обновление конфигурации».

## Настройка

Для настройки подсистемы в конфигурации дополнительных действий не требуется.

### Настройка прав доступа пользователей

Для использования подсистемы достаточно одной роли:

Таблица 3.88.

| **№** | **Роли и их назначение** |
| --- | --- |
| 1. | **ПолныеПрава** (из подсистемы «Базовая функциональность»)  Открытие формы для подтверждения легальности получения обновления |

## Использование при разработке конфигурации

Программный интерфейс подсистемы представлен общим модулем**ПроверкаЛегальностиПолученияОбновленияКлиент**.

### Настройка обмена данными

Для настройки обмена данными никаких действий не требуется, так как подсистема не предоставляет собственных данных.

# 3.41. Работа в модели сервиса

Подсистема «Работа в модели» содержит базовый функционал, обязательный для всех прикладных решений, рассчитанных на работу в модели сервиса, а также ряд подсистем, которые расширяют другие подсистемы для работы в модели сервиса (например, «Валюты в модели сервиса» и пр.).

Перед внедрением данной подсистемы рекомендуется ознакомиться с методикой разработки «1С:Технология разработки решений 1cFresh» и разделом «[Механизм разделения данных](https://its.1c.ru/db/v8doc)» в книге «Руководство разработчика» из состава документации к платформе «1С:Предприятие».

**Важно!**

В составе подсистемы поставляются два общих реквизита-разделителя: **ОбластьДанныхОсновныеДанные**и **ОбластьДанныхВспомогательныеДанные**. Корректное функционирование подсистем библиотеки при добавлении других разделителей в общем случае не поддерживается.

**Примечание**

Непосредственно в саму подсистему «Работа в модели сервиса» объекты библиотеки не включены, они относятся к дочерним подсистемам в ветке **Работа в модели сервиса**.

## Настройка

### Разработка форм настройки системы для администратора информационной базы и для администратора области данных

В отличие от локального режима работы, при работе в модели сервиса функции администрирования системы разделены на два уровня:

● Администратор информационной базы (роли **ПолныеПрава** и **АдминистраторСистемы**) выполняет настройку и обслуживание системы в целом, для всех областей данных. Например, настройку доступа к внешним ресурсам на сервере «1С:Предприятия» (прокси-сервер, почта, тома хранения файлов и т. п.), каких-либо общих параметров системы, которые должны распространяться на пользователей всех областей данных.

● Администратор области данных (роль **ПолныеПрава**) выполняет настройку параметров системы для конкретной области данных. Например: включение/отключение функциональных опций, ведение списка пользователей, удаление помеченных объектов и т. п.

Таким образом, при разработке форм настроек следует учитывать, что настройки информационной базы должны быть видны только администратору информационной базы (роль **АдминистраторСистемы**), а настройки области данных – всем администраторам (роль **ПолныеПрава**).

Если используется подсистема «Настройки программы», то прикладные формы настроек рекомендуется разрабатывать в том же стиле – в виде панелей настроек, которые «подстраиваются» под права пользователя. Также необходимо отразить в документации тот факт, что для настройки общих параметров администратору информационной базы необходимо войти в любую область данных. Подробнее см. раздел документации «[Настройки программы](https://its.1c.ru/db/content/bsp23doc/src/3.27.%20%D0%BD%D0%B0%D1%81%D1%82%D1%80%D0%BE%D0%B9%D0%BA%D0%B8%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D1%8B.htm?_=1504708101#_настройки_программы)».

В противном случае требуется разработать отдельные формы настроек для администратора информационной базы и для администратора области данных. Формы настроек должны хорошо подходить как для сценариев настройки двумя разными людьми при работе в модели сервиса, так и для сценария настройки одним человеком при работе в локальном режиме:

● Формы настроек, предназначенные для администратора информационной базы, должны быть доступны только роли **АдминистраторСистемы** и скрываться при работе в локальном режиме. Для этого форму следует включить в состав функциональной опции **ИспользоватьРазделениеПоОбластямДанных**.

● Формы настроек, предназначенные для администратора области данных, должны содержать все настройки (и настройки информационной базы и настройки области данных), быть доступны для роли **ПолныеПрава** и рассчитаны на работу без роли **АдминистраторСистемы**.

### Настройка общих реквизитов-разделителей и объектов метаданных

При работе в модели сервиса в конфигурации должен быть включен механизм разделения данных, который позволяет разделить все хранимые данные, а также работу прикладного решения на отдельные части. Для разделения данных в составе библиотеки поставляются два общих реквизита-разделителя:

● общий реквизит **ОбластьДанныхОсновныеДанные** – тип **Число(7)**, использование разделяемых данных –**Независимо**. Значение этого разделителя должно быть установлено для всех пользователей информационной базы (кроме администратора информационной базы);

● общий реквизит **ОбластьДанныхВспомогательныеДанные** – тип **Число(7)**, использование разделяемых данных – **Независимо и совместно**. Данный разделитель не используется для разделения пользователей и аутентификации.

При внедрении подсистемы «Работа в модели сервиса» в состав конфигурации важно сохранять исходный порядок сортировки общих реквизитов-разделителей в дереве метаданных.

При работе в модели сервиса все данные, хранящиеся в информационной базе (и объекты метаданных, соответствующие этим данным), могут быть разделены на следующие виды:

● **Общие (неразделенные) данные** – данные, которые являются общими для всех областей данных и не разделяются общими реквизитами-разделителями. Из сеансов, в которых не установлены значения разделителей, общие данные доступны только для чтения.

● **Разделенные данные** – данные, относящиеся к области данных, разделяются общими реквизитами-разделителями, поставляемыми в составе подсистемы «Работа в модели сервиса»:

* ● **Основные данные** – данные области, обладающие прикладным характером. Объекты метаданных, соответствующие общим данным, разделяются разделителем **ОбластьДанныхОсновныеДанные**;
* ● **Вспомогательные данные** – данные, логически являющиеся частью данных области, но доступные из неразделенных сеансов, разделяются разделителем **ОбластьДанныхВспомогательныеДанные**.

По умолчанию большинство объектов метаданных должны быть включены в состав общего реквизита **ОбластьДанныхОсновныеДанные**, т. е. должны быть разделенными. Поскольку у общего реквизита **ОбластьДанныхОсновныеДанные** свойство **Автоиспользование** установлено в значение **Использовать**, то для вновь создаваемых объектов конфигурации ничего предпринимать не требуется.

Из состава разделителя **ОбластьДанныхОсновныеДанные** должны быть удалены следующие объекты:

● Все регламентные задания.

● Объекты метаданных, предназначенные для хранения настроек информационной базы (общих для всех областей данных). В основном это константы, например: **МаксимальноеКоличествоИсполняющихФоновыхЗаданий**.

● Различные классификаторы, общие для всех областей данных, такие как классификатор валют, адресный классификатор и т. п.

В состав разделителя **ОбластьДанныхВспомогательныеДанные** должны включаться только служебные данные (не имеющие ценности для абонента). Для общего реквизита свойство **Автоиспользование** установлено в значение **Не использовать** таким образом, вновь создаваемые объекты метаданных не будут включаться в состав данного разделителя.

Все объекты метаданных, относящиеся к общим (неразделенным) данным, должны быть включены в состав любой из подписок на события, для которой в качестве обработчика назначена одна из следующих процедур (в зависимости от типа объекта метаданных):

● **РаботаВМоделиСервиса.КонтрольНеразделенныхОбъектовПриЗаписи**,

● **РаботаВМоделиСервиса.КонтрольНеразделенныхНаборовЗаписейПриЗаписи**.

Например, это могут быть следующие подписки на события, поставляемые непосредственно в составе подсистемы **БазоваяФункциональностьВМоделиСервиса**:

● **КонтрольНеразделенныхОбъектовПриЗаписи**,

● **КонтрольНеразделенныхНаборовЗаписейПриЗаписи**.

### Настройка функциональных опций «ИспользоватьРазделениеПоОбластямДанных» и «НеИспользоватьРазделениеПоОбластямДанных»

В случае если конфигурация рассчитана на работу в обоих режимах: в локальном режиме и в модели сервиса, – необходимо принять решение по поводу состава объектов конфигурации, которые должны быть размещены в пользовательском интерфейсе:

● только при работе в модели сервиса (и не должны при работе в локальном режиме);

● только при работе в локальном режиме (и не должны при работе в модели сервиса);

● в обоих режимах работы.

Объекты первой группы нужно включить в состав функциональной опции **ИспользоватьРазделениеПоОбластямДанных**, а второй – в состав функциональной опции **НеИспользоватьРазделениеПоОбластямДанных**.

## Использование при разработке конфигурации

После включения разделения данных (константа **ИспользоватьРазделениеПоОбластямДанных**) не следует входить и работать в области данных со значением разделителя 0. Кроме того, обратное выключение разделения не поддерживается в полном объеме (т. е. не рекомендуется переводить информационную базу из модели сервиса в локальный режим).

### Работа при условно выключенном разделении

Независимо от того, рассчитана ли конфигурация на работу в локальном режиме, ее код должен быть работоспособен при условно выключенном разделении (т. е. в «обычном» режиме, как будто никакого разделения нет). Поэтому если в конфигурации есть какой-то код, специфичный для модели сервиса, по его выполнение необходимо предварять проверкой того, что разделение включено.

Для проверки того, что разделение данных включено, следует использовать функцию **РазделениеВключено**общего модуля **ОбщегоНазначенияПовтИсп**.

### Установка монопольного режима

Для установки (снятия) монопольного доступа к информационной базе (области данных) следует использовать процедуры **ЗаблокироватьИБ** и **РазблокироватьИБ** общего модуля **ОбщегоНазначения**. При этом в сеансе с неустановленным использованием разделителя **ОбластьДанныхОсновныеДанные** эти процедуры устанавливают (снимают) монопольный режим ИБ (в точности так, как это происходит в локальном режиме работы).

### Последовательное выполнение регламентных и фоновых заданий с помощью очереди заданий

Для снижения нагрузки на кластер серверов «1С:Предприятия» при работе прикладных решений в модели сервиса не следует использовать разделенные регламентные задания. Вместо этого необходимо использовать очередь заданий:

● исключить регламентные задания из состава разделителя **ОбластьДанныхОсновныеДанные**;

● добавить регламентные задания в список, формируемый процедурой **ЗаполнитьСписокРазделенныхРегламентныхЗаданий** общего модуля **ОчередьЗаданийПереопределяемый**.

Очередь заданий последовательно выполняет задания в указанных областях данных с помощью исполняющих фоновых заданий. Задания могут быть помещены в очередь на выполнение через программный интерфейс или созданы автоматически на основе регламентных заданий. Программный интерфейс очереди заданий представлен экспортными процедурами и функциями общего модуля **ОчередьЗаданий**.

При необходимости выполнить какую-либо произвольную процедуру во всех или нескольких областях данных следует также использовать очередь заданий (чтобы предотвратить одновременный запуск большого количества фоновых заданий в каждой области данных). Для этого необходимо:

● добавить задание в очередь с помощью процедуры **ДобавитьЗадание** общего модуля **ОчередьЗаданий**;

● добавить имя процедуры (обработчика задания) в список, формируемый процедурой **ПолучитьРазрешенныеМетодыОчередиЗаданий** общего модуля **ОчередьЗаданийПереопределяемый**.

**Примечание**

При разработке обработчиков заданий, выполняемых механизмом очереди заданий, следует учитывать следующую особенность: непосредственный запуск фоновых заданий из таких процедур невозможен. Для запуска фоновых заданий следует временно отключить использование разделения сеанса, запустить фоновое задание и включить разделение обратно:

ОбщегоНазначения.УстановитьРазделениеСеанса(Ложь);

ФоновыеЗадания.Выполнить("имя процедуры-обработчика");

ОбщегоНазначения.УстановитьРазделениеСеанса(Истина);

Это требование обусловлено тем, что выполнение заданий очереди производится в исполняющих фоновых заданиях, которые запускаются в сеансах без указания разделителей и переключают разделение сеансов для исполняемых заданий.

### Роли разделенных пользователей

При разработке ролей, которые предполагается использовать для работы разделенных пользователей, следует иметь в виду следующее:

● Для всех объектов метаданных, не входящих в состав разделителей **ОбластьДанныхОсновныеДанные** и **ОбластьДанныхВспомогательныеДанные**, должны быть сняты права «Добавление», «Изменение», «Удаление».

● А также должны быть сняты права на конфигурацию:

* ● администрирование,
* ● обновление конфигурации базы данных,
* ● монопольный режим,
* ● толстый клиент,
* ● внешнее соединение,
* ● аutomation,
* ● интерактивное открытие внешних обработок,
* ● интерактивное открытие внешних отчетов,
* ● режим «Все функции».

При работе в разделенном режиме роли, которые не удовлетворяют вышеприведенным требованиям, будут автоматически удалены из состава профилей групп доступа и будут недоступны для назначения пользователям. Кроме того, при попытке входа в систему пользователя с такими ролями будет выдана ошибка и работа будет завершена.

### Настройка прав доступа пользователей

Для настройки прав доступа пользователей следует использовать роли, приведенные ниже.

Таблица 3.89.

| **№** | **Роли и их назначение** |
| --- | --- |
| 1. | **ПолныеПрава** (из подсистемы «Базовая функциональность»)  Функции администратора области данных: настройка параметров системы для конкретной области данных и неограниченный доступ ко всем разделенным данным |

Примеры настройки прав доступа пользователей приведены ниже.

Таблица 3.90.

| **№** | **Группа пользователей и ее функции** | **Состав ролей** |
| --- | --- | --- |
| 1. | Администратор области данных | ● **ПолныеПрава** (из подсистемы «Базовая функциональность») |
| 2. | Ответственный за нормативно-справочную информацию (в модели сервиса) | ● **ЗапускТонкогоКлиента** (из подсистемы «Базовая функциональность»),  ● **БазовыеПрава** (из подсистемы «Базовая функциональность»),  ●**ДобавлениеИзменениеДополнительныхРеквизитовИСведений**,  ● **ДобавлениеИзменениеКурсовВалют**,  ● **ДобавлениеИзменениеГрафиковРаботы**,  **● ДобавлениеИзменениеВидовКонтактнойИнформации** |

### Настройка обмена данными

Для настройки обмена данными следует руководствоваться общими правилами.

В планы обмена распределенной информационной базы (РИБ) рекомендуется включать все объекты метаданных подсистемы, кроме следующих:

● константа **ВнутреннийАдресМенеджераСервиса**,

● константа **ВыполнитьРезервноеКопированиеОбластиДанных**,

● константа **ДатаПоследнегоСтартаКлиентскогоСеанса**,

● константа **ИспользованиеКаталогаДополнительныхОтчетовИОбработокВМоделиСервиса**,

● константа **КаталогОбменаФайламиВМоделиСервиса**,

● константа **КаталогОбменаФайламиВМоделиСервиса Linux**,

● константа **КлючОбластиДанных**,

● константа **КонечнаяТочкаМенеджераСервиса**,

● константа **КопироватьОбластиДанныхИзЭталонной**,

● константа **МаксимальнаяДлительностьВыполненияИсполняющегоФоновогоЗадания**,

● константа **МаксимальноеКоличествоИсполняющихФоновыхЗаданий**,

● константа **МинимальныйИнтервалРегламентныхЗаданийДополнительныхОтчетовИОбработокВМоделиСервиса**,

● константа **НезависимоеИспользованиеДополнительныхОтчетовИОбработокВМоделиСервиса**,

● константа **ПоддержкаРезервногоКопирования**,

● константа **ПредставлениеОбластиДанных**,

● константа **ПрефиксОбластиДанных**,

● константа **РазмерБлокаПередачиФайла**,

● константа **РазрешитьВыполнениеДополнительныхОтчетовИОбработокРегламентнымиЗаданиямиВМоделиСервиса**,

● константа **РежимИспользованияИнформационнойБазы**,

● константа **СообщениеБлокировкиПриОбновленииКонфигурации**,

● константа **УдалитьИмяСлужебногоПользователяМенеджераСервиса**,

● константа **УдалитьПарольСлужебногоПользователяМенеджераСервиса**,

● константа **ЧасовойПоясОбластиДанных**,

● справочник **ОчередьЗаданий**,

● справочник **ОчередьЗаданийОбластейДанных**,

● справочник **ПоставляемыеДанные**,

● справочник **ПоставляемыеДополнительныеОтчетыИОбработки**,

● справочник **СообщенияОбластейДанных**,

● справочник **ШаблоныЗаданийОчереди**,

● регистр сведений **ВерсииПодсистемОбластейДанных**,

● регистр сведений **ИспользованиеПоставляемыхДополнительныхОтчетовИОбработокВОбластяхДанных**,

● регистр сведений **ИспользованиеДополнительныхОтчетовИОбработокСервисаВАвтономномРабочемМесте**,

● регистр сведений **ОбластиДанных**,

● регистр сведений **ОбластиТребующиеОбработкиПоставляемыхДанных**,

● регистр сведений **ОчередьИзвлеченияТекста**,

● регистр сведений **ОчередьИнсталляцииПоставляемыхДополнительныхОтчетовИОбработокВОбластиДанных**,

● регистр сведений **ПоставляемыеДанныеТребующиеОбработки**,

● регистр сведений **РейтингАктивностиОбластейДанных**,

● регистр сведений **УдалитьОбластиДанных**,

● регистр сведений **УдалитьОбластиКРезервномуКопированию**,

● регистр сведений **УдалитьОчередьЗаданий**,

● регистр сведений **УдалитьОчередьИзвлеченияТекста**,

● регистр сведений **УдалитьРейтингАктивностиОбластейДанных**.

### Получение списка пользователей для отображения

Для отображения списка пользователей в большинстве случаев следует использовать формы справочника **Пользователи**. В тех случаях, когда это невозможно, следует убирать из списка неразделенных пользователей при включенном разделении данных. В качестве примера реализации можно использовать форму списка справочника **Пользователи**.

## Обмен сообщениями

Подсистема «Обмен сообщениями» (в группе подсистем «Работа в модели сервиса») предназначена для интеграции разнородных информационных систем на платформе «1С:Предприятие» посредством обмена сообщениями. Подсистему следует использовать, когда канал передачи данных ненадежный. Подсистема не является заменой технологии обмена данными, технологии веб-сервисов и внешнего соединения.

|  |
| --- |
|  |
|  | https://its.1c.ru/db/content/bsp23doc/src/_img/bb4930f70ba50b6391a4d4b0238cf1d5/image028.gif?_=1504708101 |

**Отправитель** – это приложение, которое размещает сообщение (пакет данных) в канале, которое затем считывается другими приложениями (получателями).

**Получатель** – это приложение, которое считывает сообщение из канала.

**Отправители** и**Получатели** также называются конечными точками.

**Канал сообщений** – логический маршрут для транспортировки сообщений. Удобно применять иерархию, например «УправлениеПроектами\СозданиеПроекта\...».

Сообщение состоит из тела («прикладной» пакет данных) и заголовка (метаданные, информация, которая используется только самой подсистемой обмена сообщениями). Приложения должны согласовать канал и формат сообщения. Взаимодействие осуществляется в асинхронном режиме.

Работа подсистемы осуществляется в несколько этапов:

1. **Создание**. Отправитель создает сообщение с полезной (прикладной) информацией.

2. **Отправка**. Отправитель помещает сообщение в канал.

3. **Доставка**. Подсистема обмена сообщениями доставляет сообщение с машины-отправителя на машину-получатель.

4. **Получение**. Получатель извлекает сообщение из канала.

5. **Обработка**. Получатель считывает полезную информацию из сообщения.

Существует два вида каналов сообщений: широковещательные каналы сообщений (publish-subscribe channel) и адресные каналы сообщений (point-to-point channel). Широковещательные каналы характеризуются тем, что отправитель не «знает» получателя. Сообщение доставляется всем конечным точкам, которые «сообщили» о себе, что они подписаны на широковещательный канал. Адресные каналы характеризуются тем, что отправитель явно указывает получателя. Сообщение доставляется всем конечным точкам, которые на стороне отправителя определены как получатели для текущего канала сообщений. Либо есть возможность указать получателя явно в коде в момент отправки сообщения в канал.

### Настройка

Для использования подсистемы в конфигурации необходимо:

1. Разместить в интерфейсе администратора команду **ОбменСообщениями**.

2. Реализовать обработчики каналов сообщений.

3. Подключить обработчики каналов сообщений к подсистеме.

4. Определить получателей адресных сообщений при необходимости.

5. Опубликовать веб-сервисы **MessageExchange** конфигураций, для которых настраивается интеграция.

#### Обработчики каналов сообщений

В качестве обработчиков каналов сообщений могут выступать общие модули или модули менеджеров объектов конфигурации. В конфигурации необходимо создать общий модуль (один или несколько), который будет обработчиком каналов сообщений. В этом модуле следует объявить процедуру **ОбработатьСообщение**. В процедуре необходимо реализовать логику обработки в зависимости от имени канала сообщений.

**Важно!**

Канал сообщений однозначно идентифицирует тип сообщения. Поэтому для диспетчеризации сообщений достаточно только имени канала. Использовать информацию тела сообщений для диспетчеризации сообщений не рекомендуется.

#### Подключение обработчиков каналов сообщений к подсистеме

Подключение обработчиков каналов сообщений выполняется в общем модуле **ОбменСообщениямиПереопределяемый** в процедуре **ПолучитьОбработчикиКаналовСообщений**. Для подключения одного обработчика канала сообщений в процедуре необходимо написать код:

Обработчик = Обработчики.Добавить();

Обработчик.Канал = "<Имя канала>";

Обработчик.Обработчик = <Ссылка на общий модуль>;

В качестве значения **<Имя канала>** следует использовать имя канала сообщений, который необходимо подключить к подсистеме, например **УправлениеПроектами\СозданиеПроекта**. В качестве обработчика канала сообщений следует указать ссылку на общий модуль, например, **СообщенияУправлениеПроектами**.

Количество подключаемых каналов не ограничено.

Пример настройки подсистемы можно посмотреть в демонстрационной конфигурации.

#### Получатели адресных сообщений

При необходимости в подсистеме можно задать программным образом получателей адресных сообщений. Для этого следует использовать процедуру **ПолучателиСообщения** переопределяемого модуля подсистемы.

### Использование при разработке конфигурации

Перед использованием подсистемы необходимо:

● Решить, с какими конфигурациями должна уметь интегрироваться текущая конфигурация.

● Определить, какие сообщения и по каким каналам сообщений принимают конфигурации.

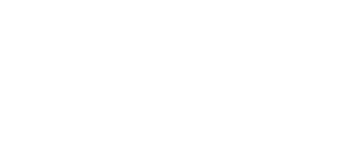
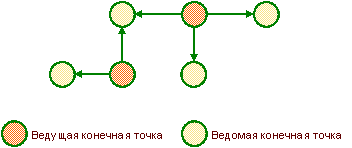
● Определить, какие каналы сообщений должна «прослушивать» текущая конфигурация:

* ● для приема ответных сообщений о других конфигураций,
* ● для возможности интеграции с текущей конфигурацией «извне».

● Определить топологию обмена сообщениями (ведущие и ведомые конечные точки).

#### Топология обмена сообщениями

Конечные точки подразделяются на ведущие и ведомые. Такое разделение сделано для оптимизации нагрузки на серверы и обеспечения синхронной отправки и доставки сообщений между информационными базами – корреспондентами. Ведущая конечная точка инициирует процесс отправки и получения сообщений. Ведомые конечные точки участвуют в процессе отправки и получения сообщений только по требованию ведущей конечной точки. Подключение конечных точек необходимо начинать со стороны ведущей конечной точки из соображений балансировки нагрузки на серверы и топологии обмена сообщениями. Чем больше конечных точек связано с ведущей, тем эффективнее обмен сообщениями. В любой момент времени ведомую конечную точку можно сделать ведущей (поменять их ролями).



Пример топологии обмена сообщениями

#### Подключение конечной точки

Выполнить подключение конечной точки к информационной базе можно интерактивно и программно.

Для интерактивного подключения конечной точки необходимо в форме настройки подсистемы выполнить команду **Подключить конечную точку**. В открывшейся форме необходимо задать настройки подключения текущей информационной базы к конечной точке и настройки подключения конечной точки к текущей информационной базе. Затем нажать **Подключить**.

Для программного подключения конечной точки необходимо в прикладном коде конфигурации выполнить вызов процедуры **ОбменСообщениями.ПодключитьКонечнуюТочку** и передать методу необходимые параметры. При успешном подключении конечной точки метод вернет в параметр **КонечнаяТочка** ссылку на подключенную конечную точку. Для программного обновления параметров подключения конечной точки следует использовать процедуру **ОбменСообщениями.ОбновитьНастройкиПодключенияКонечнойТочки**.

#### Отправка сообщений

Подсистема предоставляет возможность отправки обычных сообщений и быстрых сообщений. Разница между обычными и быстрыми сообщениями заключается только в скорости доставки. Быстрые сообщения доставляются максимально быстро до получателя. Обычные сообщения доставляются с заданным интервалом, обычно один раз в минуту. Для отправки сообщений необходимо использовать программный интерфейс подсистемы, представленный экспортными процедурами общего модуля **ОбменСообщениями**.

После отправки быстрых сообщений необходимо вызвать метод **ДоставитьСообщения** для немедленной доставки всех быстрых сообщений получателю.

**Важно!**

Методы отправки сообщений следует выполнять только в активной транзакции. Метод доставки быстрых сообщений **ДоставитьСообщения** нетранзакционный.

#### Обработчики событий отправки и получения сообщений

Подсистема предоставляет обработчики событий отправки и получения сообщений. Обработчики событий представлены экспортными процедурами **ПриОтправкеСообщения** и **ПриПолученииСообщения** общего модуля **ОбменСообщениямиПереопределяемый**.

#### Настройка прав доступа пользователей

Для настройки прав доступа пользователей к данным подсистемы «Обмен сообщениями» следует использовать роли, приведенные ниже.

Таблица 3.91.

| **№** | **Роли и их назначение** |
| --- | --- |
| 1. | **АдминистраторСистемы** (из подсистемы «Базовая функциональность»)  Подключение конечных точек. Администрирование и монитор сообщений системы |
| 2. | **УдаленныйДоступОбменСообщениями**  Удаленный доступ к подсистеме обмена сообщениями через Интернет |

Примеры настройки прав доступа пользователей приведены ниже.

Таблица 3.92.

| **№** | **Группа пользователей и ее функции** | **Состав ролей** |
| --- | --- | --- |
| 1. | Администратор  Подключение конечных точек. Администрирование и монитор сообщений системы | **АдминистраторСистемы** (из подсистемы «Базовая функциональность») |
| 2. | Удаленный доступ  Пользователь, от имени которого выполняется подключение к конечной точке через Интернет. Обеспечивает функционирование подсистемы обмена сообщениями на стороне конечной точки-корреспондента (получение сообщений, обработка очереди сообщений) | **УдаленныйДоступОбменСообщениями**  **ВыполнениеОбменовДанными** (из подсистемы «Обмен данными») |
| 3. | Пользователь  Отправка сообщений. В процессе своей работы пользователь выполняет действия, которые могут приводить к отправке сообщений | **<Любые роли>** |

#### Настройка обмена данными

Все объекты метаданных подсистемы, содержащие данные, рекомендуется включить в планы обмена распределенной ИБ (РИБ).

Исключение из этого правила составляют:

● константа **БлокировкаОтправкиБыстрыхСообщений**,

● справочник**СообщенияСистемы**,

● регистр сведений**НастройкиОтправителя**,

● регистр сведений**ПодпискиПолучателей**.

# 3.42. Работа с контрагентами

В соответствии с законом № 134-ФЗ от 28.06.2014 с I квартала 2015 года все плательщики НДС обязаны представлять в электронном виде декларации по НДС, в состав которых включены сведения о счетах-фактурах из книг покупок, книг продаж, журналов счетов-фактур. Отчеты с некорректными контрагентами не будут приняты ФНС. Для сокращения ошибок в данных была разработана подсистема «Работа с контрагентами».

Подсистема «Работа с контрагентами» позволяет реализовать проверку контрагентов по данным ФНС в справочнике контрагентов, документах, отчетах. Подсистема также включает механизм для автоматического заполнения реквизитов контрагентов по данным ЕГРЮЛ и ЕГРИП.

## Настройка

В определяемых типах **Контрагент** (ссылки) и **КонтрагентОбъект** (объекты) указать тип справочника контрагентов.

В случае если планируется переопределять интерфейсные решения рекомендуется снять с поддержки и доработать общий макет с инструкцией по проверке контрагентов **ИнструкцияПоПроверкеКонтрагентов**.

Для определения объекта и ссылки на контрагента или документ по переданной форме необходимо заполнить процедуру **ПолучитьОбъектИСсылкуПоФорме** общего модуля **РаботаСКонтрагентамиКлиентСерверПереопределяемый**. Процедура обязательна для заполнения.

Для переопределения стандартных цветовых решений по выводу результатов проверки контрагентов необходимо заполнить процедуру **ОпределитьЦветаРезультатовПроверки** модуля **РаботаСКонтрагентамиПереопределяемый**.

Иностранные контрагенты не подлежат проверке. Для определения того, является ли контрагент иностранным, необходимо заполнить процедуру **ОпределитьЯвляетсяЛиКонтрагентИностранным** модуля **РаботаСКонтрагентамиПереопределяемый**.

Для доступа к настройкам проверки контрагентов необходимо использовать обработку **ПанельАдминистрированияБСП** подсистемы «Настройка программы» либо команду **НастройкиПроверкиКонтрагентов** для доступа к отдельной общей форме настроек **НастройкиПроверкиКонтрагентов**.

Чтобы после срабатывания проверки в справочнике контрагентов поля ввода не очищались, необходимо полям ввода установить свойство **ОбновлениеТекстаРедактирования** в значение **При изменении значения**.

Если требуется отказаться от отображения стандартной блокирующей формы с предложением включить сервис проверки контрагентов, то необходимо переопределить стандартное поведение в процедуре **ПредложитьВключитьПроверкуКонтрагентов** общего модуля **РаботаСКонтрагентамиКлиентПереопределяемый**.

## Проверка в справочнике контрагентов

Для сохранения результатов проверки в форму карточки контрагента нужно добавить реквизит формы **РеквизитыПроверкиКонтрагентов** произвольного типа.

В обработчик события **ПриСозданииНаСервере** необходимо добавить вызов:

// СтандартныеПодсистемы.РаботаСКонтрагентами

ПроверкаКонтрагентов.ПриСозданииНаСервереКонтрагент(ЭтотОбъект, Параметры);

// Конец СтандартныеПодсистемы.РаботаСКонтрагентами

В обработчик события **ПередЗаписьюНаСервере** необходимо добавить вызов:

// СтандартныеПодсистемы.РаботаСКонтрагентами

ПроверкаКонтрагентов.ПередЗаписьюНаСервереКонтрагент(ЭтотОбъект, ТекущийОбъект);

// Конец СтандартныеПодсистемы.РаботаСКонтрагентами

В обработчик события **ПриОткрытии** необходимо добавить вызов:

// СтандартныеПодсистемы.РаботаСКонтрагентами

ПроверкаКонтрагентовКлиент.ПриОткрытииКонтрагент(ЭтотОбъект);

// Конец СтандартныеПодсистемы.РаботаСКонтрагентами

В обработчики событий изменения реквизитов ИНН, КПП, вида контрагента необходимо добавить запуск проверки контрагентов:

// СтандартныеПодсистемы.РаботаСКонтрагентами

ПроверкаКонтрагентовКлиент.ЗапуститьПроверкуКонтрагентаВСправочнике(ЭтотОбъект);

// Конец СтандартныеПодсистемы.РаботаСКонтрагентами

В модуль формы карточки контрагента нужно добавить следующие функции:

// СтандартныеПодсистемы.РаботаСКонтрагентами

&НаКлиенте

Процедура Подключаемый\_ПоказатьПредложениеИспользоватьПроверкуКонтрагентов()

ПроверкаКонтрагентовКлиент.ПредложитьВключитьПроверкуКонтрагентов(ЭтотОбъект);

КонецПроцедуры

// Конец СтандартныеПодсистемы.РаботаСКонтрагентами

// СтандартныеПодсистемы.РаботаСКонтрагентами

&НаКлиенте

Процедура Подключаемый\_ОбработатьРезультатПроверкиКонтрагентов()

ПроверкаКонтрагентовКлиент.ОбработатьРезультатПроверкиКонтрагентовВСправочнике(ЭтотОбъект);

КонецПроцедуры

// Конец СтандартныеПодсистемы.РаботаСКонтрагентами

В форме списка и в форме выбора справочника контрагентов необходимо указать произвольный запрос динамического списка. В запросе добавить соединение с регистром сведений **СостоянияКонтрагентов** и добавить поля **ПроверкаКонтрагентовКонтрагентНеСуществует** и **ПроверкаКонтрагентовСостояние**. После доработки запрос будет выглядеть следующим образом:

ВЫБРАТЬ

<Поля контрагента>,

ВЫБОР

КОГДА СостоянияКонтрагентов.Состояние В (&Состояния)

ТОГДА ИСТИНА

ИНАЧЕ ЛОЖЬ

КОНЕЦ КАК ПроверкаКонтрагентовКонтрагентНеСуществует,

СостоянияКонтрагентов.Состояние КАК ПроверкаКонтрагентовСостояние

ИЗ

Справочник.\_ДемоКонтрагенты КАК Справочник\_ДемоКонтрагенты

ЛЕВОЕ СОЕДИНЕНИЕ РегистрСведений.СостоянияКонтрагентов КАК СостоянияКонтрагентов

ПО Справочник\_ДемоКонтрагенты.Ссылка = СостоянияКонтрагентов.Контрагент

И &ИспользованиеПроверкиВозможно

В обработчик события **ПриСозданииНаСервере** необходимо добавить следующий вызов:

// СтандартныеПодсистемы.РаботаСКонтрагентами

ПроверкаКонтрагентов.ПриСозданииНаСервереСписокКонтрагентов(Список);

// Конец СтандартныеПодсистемы.РаботаСКонтрагентами

См. пример реализации справочника контрагентов – **\_ДемоКонтрагенты**.

Для определения имени справочника контрагентов, а также реквизитов, в которых хранятся ИНН и КПП, необходимо переопределить процедуру **ОпределитьСвойстваСправочникаКонтрагенты** общего модуля **РаботаСКонтрагентамиКлиентСерверПереопределяемый**. Процедура обязательна для заполнения.

Для вывода результата проверки контрагента в карточке контрагента необходимо переопределить процедуру **ОтобразитьРезультатПроверкиКонтрагентаВСправочнике** общего модуля **РаботаСКонтрагентамиКлиентСерверПереопределяемый**. Процедура обязательна для заполнения.

Для переопределения стандартного условного оформления в форме списка и выбора справочника контрагентов необходимо заполнить процедуру **УстановитьУсловноеОформлениеСпискаКонтрагентов** модуля **РаботаСКонтрагентамиПереопределяемый**.

Для дополнения стандартной структуры реквизита формы **РеквизитыПроверкиКонтрагентов** своими параметрами необходимо реализовать процедуру **ДополнитьРеквизитыФормыКонтрагент** модуля **РаботаСКонтрагентамиПереопределяемый**.

Если требуется переопределить стандартный механизм определения, кем является контрагент: физическим или юридическим лицом, – то необходимо реализовать процедуру **ОпределитьЮрИлиФизЛицоКонтрагент** модуля **РаботаСКонтрагентамиПереопределяемый**.

Если требуется дополнить параметры запуска фонового задания при проверке справочника, то необходимо реализовать процедуру **ДополнитьПараметрыЗапускаФоновогоЗадания** модуля **РаботаСКонтрагентамиКлиентСерверПереопределяемый**.

Для справочника контрагента реализована возможность выполнить предварительную проверку ИНН, КПП и даты «на клиенте» до отправки данных на проверку на веб-сервис ФНС.

Если необходимо отключить режим предварительной проверки «на клиенте», то в процедуре **ДополнитьРеквизитыФормыКонтрагент** общего модуля **РаботаСКонтрагентамиПереопределяемый необходимо**в состав входящего параметра-структуры **НовыеРеквизиты** надо добавить элемент **ВыполнятьПредварительнуюПроверкуКонтрагента**, равный **Ложь**.

В подсистеме «Работа с контрагентами» есть специальный режим, позволяющий проверять контрагентов даже при отключенной автоматической проверке. В карточке контрагента в подменю **Еще** необходимо добавить кнопку и назвать ее, например, **Проверить контрагента**. По этой кнопке вызывать процедуру **ПроверитьКонтрагентаПоКнопке** общего модуля **ПроверкаКонтрагентовКлиент**. В форме контрагента необходимо разместить экспортную переменную:

// СтандартныеПодсистемы.РаботаСКонтрагентами

&НаКлиенте

Перем ФормаДлительнойОперации Экспорт;

// Конец СтандартныеПодсистемы.РаботаСКонтрагентами

## Проверка контрагентов в документах

При реализации проверки контрагентов в документах необходимо в форме документа добавить реквизит формы **РеквизитыПроверкиКонтрагентов** произвольного типа.

Проверка контрагентов в документе выполняется в фоновом режиме. Для этого в разделе объявления переменных необходимо добавить объявление клиентской переменной **ПроверкаКонтрагентовПараметрыОбработчикаОжидания**:

// СтандартныеПодсистемы.РаботаСКонтрагентами

&НаКлиенте

Перем ПроверкаКонтрагентовПараметрыОбработчикаОжидания Экспорт;

// Конец СтандартныеПодсистемы.РаботаСКонтрагентами

Если в документе есть табличные части, содержащие контрагентов, которых нужно проверять, то необходимо разместить колонку реквизита с именем **Состояние** и **ЭтоНекорректныйКонтрагент**:

* ● Добавить колонку реквизита с именем **Состояние** (тип **ПеречислениеСсылка.СостоянияСуществованияКонтрагента**).
* ● Добавить колонку реквизита с именем **ЭтоНекорректныйКонтрагент** (тип **Булево**). Данную колонку необходимо:
* ● вынести на форму;
* ● присвоить ей имя **Имя таблицы + ЭтоНекорректныйКонтрагент**;
* ● сгруппировать с колонкой контрагента;
* ● в группе разместить второй колонкой (за контрагентом;
* ● установить свойства:
* ● **Вид = Поле картинки**;
* ● **Заголовок = "Есть ошибка по данным ФНС"**;
* ● **Ширина** = 2;
* ● **РастягиватьПоГоризонтали = Ложь**;
* ● **ГиперссылкаЯчейки = Истина**;
* ● **ОтображатьВШапке = Ложь**;
* ● **ПоложениеЗаголовка = Нет**;
* ● **КартинкаЗначений = НекорректныйКонтрагент.**
* ● У группы колонок, объединяющих контрагента и признак некорректности необходимо установить свойства:
* ● **Группировка = Горизонтальная**;
* ● **ОтображатьВШапке = Истина**;
* ● **ОтображатьЗаголовок = Истина**;
* ● **Группировка = Горизонтальная**;
* ● **ФиксацияВТаблице** как у колонки с контрагентом;
* ● **РастягиватьПоГоризонтали = Ложь**;
* ● **Ширина** = 17.
* ● У колонки с контрагентом необходимо установить свойства:
* ● **ФиксацияВТаблице** = Нет;
* ● **ОтображатьВШапке** = Нет;
* ● **РастягиватьПоГоризонтали = Ложь**;
* ● **Заголовок**.

См. пример размещения колонок в форме **ФормаДокумента** документа **\_ДемоСчетФактураПолученный**(табличная часть **Агентские услуги**).

Указанные выше пункты по добавлению колонок можно не выполнять, если не критична скорость открытия формы. В этом случае данные действия будут выполнены программно.

В обработчики событий **ПриСозданииНаСервере** необходимо добавить вызов:

// СтандартныеПодсистемы.РаботаСКонтрагентами

ПроверкаКонтрагентов.ПриСозданииНаСервереДокумент(ЭтотОбъект, Параметры);

// Конец СтандартныеПодсистемы.РаботаСКонтрагентами

В обработчики событий **ПриОткрытии** необходимо добавить вызов:

// СтандартныеПодсистемы.РаботаСКонтрагентами

ПроверкаКонтрагентовКлиент.ПриОткрытииДокумент(ЭтотОбъект);

// Конец СтандартныеПодсистемы.РаботаСКонтрагентами

В обработчики событий **ПередЗаписьюНаСервере** необходимо добавить вызов:

// СтандартныеПодсистемы.РаботаСКонтрагентами

ПроверкаКонтрагентов.ПередЗаписьюНаСервереДокумент(ЭтотОбъект, ТекущийОбъект);

// Конец СтандартныеПодсистемы.РаботаСКонтрагентами

В обработчики событий **ОбработкаОповещения** необходимо добавить вызов:

  // СтандартныеПодсистемы.РаботаСКонтрагентами

ПроверкаКонтрагентовКлиент.ОбработкаОповещения(ЭтотОбъект, ИмяСобытия, Параметр, Источник);

// Конец СтандартныеПодсистемы.РаботаСКонтрагентами

В обработчики событий **ПослеЗаписиНаСервере** необходимо добавить вызов:

// СтандартныеПодсистемы.РаботаСКонтрагентами

ПроверкаКонтрагентов.ПослеЗаписиНаСервере(ЭтотОбъект, ТекущийОбъект);

// Конец СтандартныеПодсистемы.РаботаСКонтрагентами

В обработчики событий **ПриЗаписиНаСервере** необходимо добавить вызов:

// СтандартныеПодсистемы.РаботаСКонтрагентами

ПроверкаКонтрагентов.ПриЗаписиНаСервере(ЭтотОбъект, ТекущийОбъект);

// Конец СтандартныеПодсистемы.РаботаСКонтрагентами

В обработчики событий изменения реквизитов, требующих запуска проверки (изменение даты, контрагента и т. д.) необходимо добавить вызов:

// СтандартныеПодсистемы.РаботаСКонтрагентами

ПроверкаКонтрагентовКлиент.ЗапуститьПроверкуКонтрагентовВДокументе(ЭтотОбъект, Объект.Дата);

// Конец СтандартныеПодсистемы.РаботаСКонтрагентами

При этом второй параметр функции **ЗапуститьПроверкуКонтрагентовВДокументе** общего модуля **ПроверкаКонтрагентовКлиент** необходимо указывать следующим образом:

* ● дату, если произошло изменение даты документа;
* ● таблицу формы, если произошло изменение контрагента или даты в таблице формы;
* ● элемент управления, в котором произошло изменение, требующее запуска проверки контрагента.

В обработчик события выбора таблицы формы, в которой находятся проверяемые контрагенты, необходимо добавить вызов:

// СтандартныеПодсистемы.РаботаСКонтрагентами

ПроверкаКонтрагентовКлиент.ТаблицаФормыВыбор(ЭтотОбъект, Элемент, Поле);

// Конец СтандартныеПодсистемы.РаботаСКонтрагентами

В модуль формы документа нужно добавить следующие функции:

// СтандартныеПодсистемы.РаботаСКонтрагентами

&НаКлиенте

Процедура Подключаемый\_ПоказатьПредложениеИспользоватьПроверкуКонтрагентов()

ПроверкаКонтрагентовКлиент.ПредложитьВключитьПроверкуКонтрагентов(ЭтотОбъект);

КонецПроцедуры

// Конец СтандартныеПодсистемы.РаботаСКонтрагентами

// СтандартныеПодсистемы.РаботаСКонтрагентами

&НаКлиенте

Процедура Подключаемый\_ОбработатьРезультатПроверкиКонтрагентов()

ПроверкаКонтрагентовКлиент.ОбработатьРезультатПроверкиКонтрагентовВДокументе(ЭтотОбъект);

КонецПроцедуры

// Конец СтандартныеПодсистемы.РаботаСКонтрагентами

// СтандартныеПодсистемы.РаботаСКонтрагентами

&НаСервере

Процедура ОтобразитьРезультатПроверкиКонтрагента() Экспорт

ПроверкаКонтрагентов.ОтобразитьРезультатПроверкиКонтрагентаВДокументе(ЭтотОбъект);

КонецПроцедуры

// Конец СтандартныеПодсистемы.РаботаСКонтрагентами

// СтандартныеПодсистемы.РаботаСКонтрагентами

&НаСервере

Процедура ПроверитьКонтрагентовФоновоеЗадание(ПараметрыФоновогоЗадания) Экспорт

ПроверкаКонтрагентов.ПроверитьКонтрагентовВДокументеФоновоеЗадание(ЭтотОбъект, ПараметрыФоновогоЗадания);

КонецПроцедуры

// Конец СтандартныеПодсистемы.РаботаСКонтрагентами

В форме списка и в форме документов необходимо указать произвольный запрос динамического списка. В запросе добавить соединение с регистром сведений **ДокументыСОшибкамиПроверкиКонтрагентов** и добавить поле **СодержитНекорректныхКонтрагентов**. После доработки запрос будет выглядеть следующим образом:

ВЫБРАТЬ

<Поля документа>,

ВЫБОР

КОГДА ДокументыСОшибкамиПроверкиКонтрагентов.Документ ЕСТЬ NULL

ТОГДА ЛОЖЬ

ИНАЧЕ ИСТИНА

КОНЕЦ КАК СодержитНекорректныхКонтрагентов

ИЗ

Документ.\_ДемоПоступлениеТоваров КАК Документ\_ДемоПоступлениеТоваров

ЛЕВОЕ СОЕДИНЕНИЕ РегистрСведений.ДокументыСОшибкамиПроверкиКонтрагентов КАК ДокументыСОшибкамиПроверкиКонтрагентов

ПО Документ\_ДемоПоступлениеТоваров.Ссылка = ДокументыСОшибкамиПроверкиКонтрагентов.Документ

И &ИспользованиеПроверкиВозможно

В обработчик события **ПриСозданииНаСервере** необходимо добавить следующий вызов:

// СтандартныеПодсистемы.РаботаСКонтрагентами

ПроверкаКонтрагентов.ПриСозданииНаСервереСписокДокументов(Список);

// Конец СтандартныеПодсистемы.РаботаСКонтрагентами

См. пример реализации документов **\_ДемоПоступлениеТоваров**, **\_ДемоСчетФактураПолученный**.

Условно документы разделены на два вида:

* ● Документы (документ может быть одновременно нескольких видов):
* ● документы с контрагентом в шапке;
* ● документы с контрагентом в табличных частях;
* ● документы со счетом-фактурой в подвале.
* ● Счета-фактуры.

Для определения того, к какому виду относится документ, необходимо заполнить процедуру **ОпределитьВидДокумента** общего модуля **РаботаСКонтрагентамиКлиентСерверПереопределяемый**. Обязательно к заполнению.

Для определения ссылки на счет-фактуру по форме документа-основания для выписки счета-фактуры необходимо заполнить процедуру **ПолучитьСчетФактуру** общего модуля **РаботаСКонтрагентамиКлиентСерверПереопределяемый**. Обязательна к заполнению.

Если документ является документом со счетом-фактурой в подвале формы, то необходимо заполнить процедуру **СформироватьОписаниеДокументовСоСчетомФактуройВПодвале** общего модуля **РаботаСКонтрагентамиПереопределяемый**. Обязательна к заполнению.

Если документ является документом с контрагентом в шапке, то необходимо заполнить процедуру **СформироватьОписаниеДокументовСКонтрагентомВШапке** общего модуля **РаботаСКонтрагентамиПереопределяемый**. Обязательна к заполнению.

Если документ является документом с контрагентом в табличных частях, то необходимо заполнить процедуру **СформироватьОписаниеДокументовСТабличнымиЧастями** общего модуля **РаботаСКонтрагентамиПереопределяемый**. Обязательна к заполнению.

Если документ является счетом-фактурой, то необходимо заполнить процедуру **СформироватьОписаниеСчетовФактур** общего модуля **РаботаСКонтрагентамиПереопределяемый**. Обязательна к заполнению.

Если документ является счетом-фактурой, то необходимо заполнить процедуру **ПолучитьИменаРеквизитовКонтрагентовВСчетеФактуре** общего модуля **РаботаСКонтрагентамиПереопределяемый**. Обязательна к заполнению.

Для определения даты в документах необходимо заполнить процедуру **ОпределитьДатуВДокументе** общего модуля **РаботаСКонтрагентамиПереопределяемый**. Обязательна к заполнению.

Для определения даты в счетах-фактах необходимо заполнить процедуру **ОпределитьДатуВСчетеФактуре**общего модуля **РаботаСКонтрагентамиПереопределяемый**. Обязательна к заполнению.

Для переопределения стандартной обработки результата фонового задания по проверке контрагентов в документе необходимо заполнить процедуру **ОтобразитьРезультатПроверкиКонтрагентаВДокументе** общего модуля **РаботаСКонтрагентамиПереопределяемый**.

Для переопределения стандартного отображения результата проверки контрагентов в документе необходимо заполнить процедуру **ПрорисоватьСостоянияКонтрагентовВДокументе** общего модуля**РаботаСКонтрагентамиПереопределяемый**.

Для переопределения стандартного условного оформления в списке документов необходимо заполнить процедуру **УстановитьУсловноеОформлениеСпискаДокументов** общего модуля **РаботаСКонтрагентамиПереопределяемый**.

Для переопределения стандартного условного оформления в табличной части документа необходимо заполнить процедуру **УстановитьУсловноеОформлениеТабличнойЧастиДокумента** общего модуля **РаботаСКонтрагентамиПереопределяемый**.

Для переопределения необходимости вывода колонки с результатом проверки в табличной части необходимо заполнить процедуру **ДобавитьКолонкиТаблицыФормыСРезультатамиПроверкиВТабличныеЧасти** общего модуля **РаботаСКонтрагентамиПереопределяемый**.

Для определения необходимости проверки контрагентов в документе после срабатывания события, перехваченного в обработке оповещения, необходимо заполнить процедуру **ОпределитьНеобходимостьПроверкиКонтрагентовВОбработкеОповещения** в переопределяемом общем модуле **РаботаСКонтрагентамиКлиентПереопределяемый**.

Для переопределения сформированной подсистемой подсказки о результате проверки контрагента в документе необходимо заполнить процедуру **ПослеФормированияПодсказкиВДокументе** в переопределяемом общем модуле **РаботаСКонтрагентамиКлиентСерверПереопределяемый**.

Для дополнения стандартной структуры реквизита формы **РеквизитыПроверкиКонтрагентов** своими параметрами необходимо реализовать процедуру **ДополнитьРеквизитыФормыДокумент** модуля **РаботаСКонтрагентамиПереопределяемый**.

Если требуется переопределить стандартный механизм заполнения данных о контрагентах счетов-фактур, то необходимо реализовать процедуру **ЗаполнитьДанныеКонтрагентовВСчетеФактуре** модуля **РаботаСКонтрагентамиПереопределяемый**.

Если требуется переопределить стандартный механизм определения сохраняемых данных о результатах проверки контрагентов счетов-фактур, то необходимо реализовать процедуру **СохраняемыеРезультатыПроверкиСчетФактурыВДокументе** модуля **РаботаСКонтрагентамиПереопределяемый**.

Если требуется переопределить стандартный механизм сохранения данных о результатах проверки счетов-фактур, выписанных на основании документа, то необходимо реализовать процедуру **ЗапомнитьРезультатПроверкиКонтрагентовСчетФактурыДокументе** модуля **РаботаСКонтрагентамиПереопределяемый**.

В карточке контрагента в подменю **Еще** необходимо добавить кнопку и назвать ее, например, **Проверить контрагентов**. По этой кнопке вызывать процедуру **ПроверитьКонтрагентовВДокументеПоКнопке** общего модуля **ПроверкаКонтрагентовКлиент**. В форме документа необходимо разместить экспортную переменную:

// СтандартныеПодсистемы.РаботаСКонтрагентами

&НаКлиенте

Перем ФормаДлительнойОперации Экспорт;

// Конец СтандартныеПодсистемы.РаботаСКонтрагентами

## Проверка контрагентов в отчетах

В общем случае алгоритм проверки контрагентов в отчете следующий. Сначала формируется отчет без проверки контрагента, но результат формирования отчета запоминается (данные, полученный табличный документ, сформированный список всех контрагентов). После окончания формирования отчета в фоновом задании запускается проверка контрагентов. Для проверки контрагентов используются данные, полученные при первом формировании отчета. При отработке фонового здания вместо исходного отчета выводится отчет, в котором отображены результаты проверки контрагентов. Проверка контрагентов в отчете поддерживает возможность проверки отчета, выводимого с делением на разделы.

При реализации проверки контрагентов в отчетах необходимо в форме отчета добавить реквизит формы **РеквизитыПроверкиКонтрагентов** произвольного типа.

Проверка контрагентов в отчете выполняется в фоновом режиме. Для этого необходимо в разделе объявления переменных необходимо добавить объявление клиентской переменной **ПроверкаКонтрагентовПараметрыОбработчикаОжидания**:

// СтандартныеПодсистемы.РаботаСКонтрагентами

&НаКлиенте

Перем ПроверкаКонтрагентовПараметрыОбработчикаОжидания Экспорт;

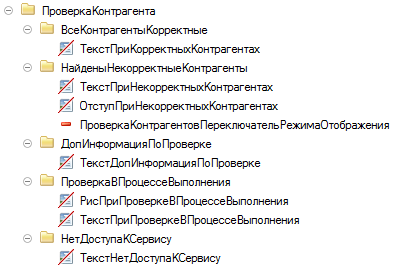
// Конец СтандартныеПодсистемы.РаботаСКонтрагентами

Для хранения перечня некорректных контрагентов необходимо добавить реквизит формы с типом таблица значений **ПроверкаКонтрагентовНедействующиеКонтрагенты**. Данный реквизит необходим только для отчетов, выводимых без деления на блоки. Список колонок и их типы:



Пример см. в форме **ФормаОтчета** отчета **\_ДемоКнигаПокупок**.

Для отображения результата проверки контрагентов в отчете необходимо разместить панель проверки контрагентов, состоящую из следующих элементов управления, а также реквизит формы **ПроверкаКонтрагентовПереключательРежимаОтображения** с обработчиком события **ПриИзменении**.



Пример см. в форме **ФормаОтчета** отчета **\_ДемоКнигаПокупок**.

В обработчики событий **ПриСозданииНаСервере** необходимо добавить вызов:

// СтандартныеПодсистемы.РаботаСКонтрагентами

ПроверкаКонтрагентов.ПриСозданииНаСервереОтчет(ЭтотОбъект, ОтчетСРазделами);

// Конец СтандартныеПодсистемы.РаботаСКонтрагентами

где **ОтчетСРазделами** – это режим формирования отчета. Если отчет будет формироваться блоками (по разделам), то **ОтчетСРазделами** должен равняться **Истина**. Если отчет будет формироваться и выводиться одним целым, то **ОтчетСРазделами** должен равняться **Ложь.**

В обработчики событий **ПриОткрытии** необходимо добавить вызов:

// СтандартныеПодсистемы.РаботаСКонтрагентами

ПроверкаКонтрагентовКлиент.ОтчетПриОткрытии(ЭтотОбъект);

// Конец СтандартныеПодсистемы.РаботаСКонтрагентами

Для очистки результатов предыдущих проверок контрагентов в отчете перед формированием отчета необходимо добавить вызов:

// СтандартныеПодсистемы.РаботаСКонтрагентами

ПроверкаКонтрагентов.ПередФормированиемОтчета(ЭтотОбъект);

// Конец СтандартныеПодсистемы.РаботаСКонтрагентами

При подготовке параметров формирования отчета необходимо добавить параметры, необходимые для последующего запуска проверки контрагентов. Для этого нужно добавить вызов:

// СтандартныеПодсистемы.РаботаСКонтрагентами

ПроверкаКонтрагентов.ДобавитьПараметрыДляПроверкиКонтрагентов(ЭтотОбъект, ПараметрыОтчета, ЭтоПервоеФормированиеОтчета, РезультатФормированияОтчета);

// Конец СтандартныеПодсистемы.РаботаСКонтрагентами

Параметр **РезультатФормированияОтчета** указывается только для случая формирования отчета без деления по разделам.

После формирования отчета для сохранения результатов формирования отчета и подготовки к запуску проверки необходимо добавить вызов:

// СтандартныеПодсистемы.РаботаСКонтрагентами

ПроверкаКонтрагентов.ЗапомнитьРезультатФормированияОтчета(ЭтотОбъект, РезультатВыполнения, АдресХранилища);

// Конец СтандартныеПодсистемы.РаботаСКонтрагентами

где **АдресХранилища** – адрес временного хранилища с результатом заполнения отчета. Во временном хранилище хранится структура с обязательным ключом **ДанныеДляПроверкиКонтрагентов**.

Для возможности вывода результата формирования отчета без деления на разделы после отработки фонового задания по проверке контрагентов вместо стандартного вывода отчета необходимо добавить вызов:

// СтандартныеПодсистемы.РаботаСКонтрагентами

ПроверкаКонтрагентов.ВывестиОтчет(ЭтотОбъект, Результат, РезультатФормированияОтчета);

// Конец СтандартныеПодсистемы.РаботаСКонтрагентами

Для возможности вывода результата формирования отчета с делением на разделы после отработки фонового задания по проверке контрагентов вместо стандартного вывода отчета необходимо добавить вызов:

// СтандартныеПодсистемы.РаботаСКонтрагентами

ПроверкаКонтрагентов.ВывестиРазделОтчета(ЭтотОбъект, Результат, ТекущийНомерРаздела);

// Конец СтандартныеПодсистемы.РаботаСКонтрагентами

После формирования отчета для запуска фонового задания по проверке контрагентов необходимо добавить вызов:

// СтандартныеПодсистемы.РаботаСКонтрагентами

ПроверкаКонтрагентовКлиент.ЗапуститьПроверкуКонтрагентовВОтчете(ЭтотОбъект);

// Конец СтандартныеПодсистемы.РаботаСКонтрагентами

При изменении настроек отчета, влияющих на формирования отчета и результат проверки, например периода формирования отчета, необходимо сбрасывать результат предыдущих проверок контрагента при помощи вызова:

// СтандартныеПодсистемы.РаботаСКонтрагентами

ПроверкаКонтрагентовКлиентСервер.СброситьАктуальностьОтчета(ЭтотОбъект);

// Конец СтандартныеПодсистемы.РаботаСКонтрагентами

При формировании данных отчета их необходимо запомнить для последующей проверки контрагентов:

// СтандартныеПодсистемы.РаботаСКонтрагентами

ПроверкаКонтрагентов.ЗапомнитьДанныеОтчета(СтруктураПараметров, Выборка);

// Конец СтандартныеПодсистемы.РаботаСКонтрагентами

При завершении формирования отчета необходимо возвратить результат формирования отчета (табличный документ и список контрагентов), чтобы затем использовать при проверке контрагентов. Для этого необходимо добавить вызов:

// СтандартныеПодсистемы.РаботаСКонтрагентами

ПроверкаКонтрагентов.ДополнитьРезультатФормированияОтчетаДаннымиПоПроверкеКонтрагентов(РезультатФормирования, СтруктураПараметров);

// Конец СтандартныеПодсистемы.РаботаСКонтрагентами

В момент вывода строк отчета определяется, нужно ли их выводить. Для этого используется функция:

// СтандартныеПодсистемы.РаботаСКонтрагентами

Если ПроверкаКонтрагентов.ВыводитьСтрокуОтчета(СтруктураПараметров, Контрагент, ИНН, КПП, Дата) Тогда

// Конец СтандартныеПодсистемы.РаботаСКонтрагентами

При формировании отчета выполняется формирования полного списка контрагентов. Для этого при выводе строк отчета необходимо использовать:

// СтандартныеПодсистемы.РаботаСКонтрагентами

ПроверкаКонтрагентов.ДобавитьКонтрагентаНаПроверку(СтруктураПараметров, ОбластьТабличногоДокумента.Верх, Контрагент, ИНН, КПП, Дата);

// Конец СтандартныеПодсистемы.РаботаСКонтрагентами

При формировании отчета, выводимого по разделам, список табличных документов разделов формируется следующим вызовом:

// СтандартныеПодсистемы.РаботаСКонтрагентами

ПроверкаКонтрагентов.ДобавитьРазделОтчета(СтруктураПараметров, ЗаголовокЛиста, СчетчикСегментов, ТабличныйДокумент);

//Конец СтандартныеПодсистемы.РаботаСКонтрагентами

В модуль формы отчета нужно добавить следующие функции:

СтандартныеПодсистемы.РаботаСКонтрагентамиботаСКонтрагентами

&НаКлиенте

Процедура Подключаемый\_ПоказатьПредложениеИспользоватьПроверкуКонтрагентов()

ПроверкаКонтрагентовКлиент.ПредложитьВключитьПроверкуКонтрагентов(ЭтотОбъект);

КонецПроцедуры

// Конец СтандартныеПодсистемы.РаботаСКонтрагентами

// СтандартныеПодсистемы.РаботаСКонтрагентами

&НаСервере

Функция ПроверитьКонтрагентов() Экспорт

ПараметрыОтчета = ПодготовитьПараметрыОтчета(Ложь);

ПроверкаКонтрагентов.ПроверитьКонтрагентовВОтчете(ЭтотОбъект, ПараметрыОтчета);

КонецФункции

// Конец СтандартныеПодсистемы.РаботаСКонтрагентами

// СтандартныеПодсистемы.РаботаСКонтрагентами

&НаКлиенте

Процедура ПереключательРежимаОтображенияПриИзменении(Элемент)

ПереключитьРежимОтображенияОтчета();

КонецПроцедуры

// Конец СтандартныеПодсистемы.РаботаСКонтрагентами

Следующие процедуры также необходимо добавить в модуль формы отчета в виде, который зависит от того, формируется отчет по разделам или нет.

Процедура **ОтобразитьРезультатПроверкиКонтрагента** для случая формирования без деления по разделам:

// СтандартныеПодсистемы.РаботаСКонтрагентами

&НаСервере

Процедура ОтобразитьРезультатПроверкиКонтрагента() Экспорт

ПроверкаКонтрагентов.ОтобразитьРезультатПроверкиКонтрагентаВОтчете(ЭтотОбъект, Результат, РезультатФормированияОтчета);

КонецПроцедуры

// Конец СтандартныеПодсистемы.РаботаСКонтрагентами

Процедура **ОтобразитьРезультатПроверкиКонтрагента** для случая формирования c делением по разделам:

// СтандартныеПодсистемы.РаботаСКонтрагентами

&НаСервере

Процедура ОтобразитьРезультатПроверкиКонтрагента() Экспорт

ПроверкаКонтрагентов.ОтобразитьРезультатПроверкиКонтрагентаВОтчете(ЭтотОбъект, Результат, Неопределено, ТекущийНомерРаздела);

КонецПроцедуры

// Конец СтандартныеПодсистемы.РаботаСКонтрагентами

Процедура **ПереключитьРежимОтображенияОтчета** для случая формирования без деления по разделам:

&НаСервере

Процедура ПереключитьРежимОтображенияОтчета()

// СтандартныеПодсистемы.РаботаСКонтрагентами

ПроверкаКонтрагентов.ПереключитьРежимОтображенияОтчета(ЭтотОбъект, Результат, РезультатФормированияОтчета);

// Конец СтандартныеПодсистемы.РаботаСКонтрагентами

КонецПроцедуры

Процедура **ПереключитьРежимОтображенияОтчета** для случая формирования с делением по разделам:

&НаСервере

Процедура ПереключитьРежимОтображенияОтчета()

// СтандартныеПодсистемы.РаботаСКонтрагентами

ПроверкаКонтрагентов.ПереключитьРежимОтображенияРазделаОтчета(ЭтотОбъект, Результат, ТекущийНомерРаздела);

// Конец СтандартныеПодсистемы.РаботаСКонтрагентами

КонецПроцедуры

В модуль формы отчета нужно добавить функцию по проверке контрагентов. В данной функции необходимо добавить вызов функции по подготовке параметров отчета (индивидуальна для каждого отчета), а только затем запустить проверку контрагентов:

&НаСервере

Функция ПроверитьКонтрагентов() Экспорт

ПараметрыОтчета = ПодготовитьПараметрыОтчета(Ложь);

// СтандартныеПодсистемы.РаботаСКонтрагентами

ПроверкаКонтрагентов.ПроверитьКонтрагентовВОтчете(ЭтотОбъект, ПараметрыОтчета);

// Конец СтандартныеПодсистемы.РаботаСКонтрагентами

КонецФункции

Пример см. в форме **ФормаОтчета** отчета **\_ДемоКнигаПокупок**.

Для переопределения стандартного отображения результата проверки в отчете необходимо переопределить процедуру **УстановитьВидПанелиПроверкиКонтрагентовВОтчете** общего модуля **РаботаСКонтрагентамиКлиентСерверПереопределяемый**.

Для подсветки в отчете контрагента с ошибкой необходимо переопределить процедуру **ВыделитьКонтрагентаСОшибкойВСтрокеОтчета** общего модуля **РаботаСКонтрагентамиПереопределяемый**. Обязательна к заполнению.

Для формирования отчета, в котором используется проверка контрагентов, из общего модуля необходимо переопределить процедуру **СформироватьОтчет** общего модуля **РаботаСКонтрагентамиПереопределяемый**. Обязательна к заполнению.

Для вывода результата проверки контрагента в поле табличного документа необходимо переопределить процедуру **ВывестиОтчет** общего модуля **РаботаСКонтрагентамиПереопределяемый**. Обязательна к заполнению.

Для определения того, нужно ли выводить строку отчета в отчете по некорректным контрагентам, необходимо переопределить процедуру **ОпределитьНужноЛиВыводитьСтрокуОтчета** общего модуля **РаботаСКонтрагентамиПереопределяемый**. Обязательна к заполнению.

Для переопределения стандартного отображения результата проверки в отчете необходимо переопределить процедуру **ОтобразитьРезультатПроверкиКонтрагентаВОтчете** общего модуля **РаботаСКонтрагентамиПереопределяемый**.

### Настройка прав доступа пользователей

Для настройки прав доступа пользователей к данным подсистемы «Работа с контрагентами» следует использовать роли, указанные ниже.

Таблица 3.93.

| **№** | **Роли и их назначение** |
| --- | --- |
| 1. | **АдминистраторСистемы** (из подсистемы «Базовая функциональность»)  ● Включение и отключение использования проверки контрагентов,  ● Настройка регламентного задания по проверке контрагентов в справочнике |
| 2. | **БазовыеПрава**  Для проверки контрагентов в справочнике, документах, отчете |
| 3. | **ИспользованиеДосьеКонтрагента**  Использование сервиса**Досье контрагента** |

## Заполнение реквизитов контрагентов

Для реализации автоматического заполнения реквизитов контрагента по данным единых государственных реестров юридических лиц и предпринимателей (ЕРГЮЛ и ЕГРИП) следует использовать функции **РеквизитыЮридическогоЛицаПоИНН** и **РеквизитыПредпринимателяПоИНН** общего модуля **ДанныеЕдиныхГосРеестров**.

Для автоматического заполнения реквизитов контрагента по наименованию используется общая форма **ЗаполнениеРеквизитовКонтрагента**. При заполнении табличной части со списком подходящих контрагентов используется функция **РеквизитыЮридическихЛицПоНаименованию** общего модуля **ДанныеЕдиныхГосРеестров**. В общей форме также есть возможность указать ИНН контрагента для заполнения его реквизитов.

Пример автоматического заполнения реквизитов реализован в форме элемента справочника **\_ДемоКонтрагенты**демонстрационной конфигурации.

## Досье контрагента

Отчет **ДосьеКонтрагента** выводит расширенную информацию из ЕГРЮЛ и ЕГРИП о контрагенте, его видах деятельности, лицензиях, руководителе, учредителях, связях с другими контрагентами.

В получаемых от веб-сервиса данных ЕГРЮЛ и ЕГРИП о видах деятельности контрагента есть только информация о кодах по ОКВЭД. Для вывода в отчет информации о наименовании видов деятельности по ОКВЭД, соответствующих этим кодам, необходимо переопределить процедуру **ЗаполнитьНаименованияВидовДеятельностиКонтрагента**общего модуля **РаботаСКонтрагентамиПереопределяемый**. Без переопределения в отчет выводится только информация о кодах по ОКВЭД, без наименований.

Роль **ИспользованиеДосьеКонтрагента**, дающую доступ к формированию отчета, необходимо назначить профилям сотрудников, ответственных за работу с контрагнетами. В демонстрационной конфигурации эта роль включена в профили «Ответственный за нормативно-справочную информацию», «Руководитель» и «Бухгалтер».

Отчет может использоваться как для показа информации о существующем в информационной базе контрагенте, так и о контрагенте, отсутствующем в базе. Для поиска контрагентами в ЕГРЮЛ или ЕГРИП указывается его ИНН или наименование, при этом для поиска используется тот же механизм, что и для заполнения реквизитов контрагента.

Кроме информации из ЕГРЮЛ и ЕГРИП в отчет может выводиться информация о контрагенте по данным информационной базы. Может быть выведена произвольная информация, определяемая разработчиком конфигурации.

Для вывода информации о контрагенте из информационной базы необходимо переопределить процедуры **ЗаполнитьОписаниеДанныхПрограммы**и **ДополнитьИнформациюДаннымиПрограммы**общего модуля **РаботаСКонтрагентамиПереопределяемый**. Без переопределения никакая информация о контрагенте по данным информационной базы в этом режиме открытия отчета не выводится.

## Использование при разработке конфигурации

Программный интерфейс подсистемы представлен экспортными функциями общих модулей:

* ● **РаботаСКонтрагентамиКлиентПереопределяемый**;
* ● **РаботаСКонтрагентамиКлиентСерверПереопределяемый**;
* ● **РаботаСКонтрагентамиПереопределяемый**;
* ● **ПроверкаКонтрагентов**;
* ● **ПроверкаКонтрагентовВызовСервера**;
* ● **ПроверкаКонтрагентовВызовСервераПовтИсп**;
* ● **ПроверкаКонтрагентовКлиент**;
* ● **ПроверкаКонтрагентовКлиентСервер**;
* ● **ПроверкаКонтрагентовКлиентСерверПовтИсп**;
* ● **РегламентированныеДанныеКлиентСервер**;
* ● **ДанныеЕдиныхГосРеестров**.

### Настройка обмена данными

Константу **ИспользоватьПроверкуКонтрагентов** рекомендуется включить в планы обмена распределенной ИБ (РИБ). Все остальные объекты метаданных подсистемы не следует включать в планы обмена РИБ.

# 3.43. Работа с почтовыми сообщениями

Подсистема «Работа с почтовыми сообщениями» добавляет в конфигурацию программный интерфейс по отправке сообщений электронной почты, а также пользовательский интерфейс для поддержки учетных записей электронной почты.

## Настройка

В случае если в конфигурации не используется подсистема «Настройки программы», в рабочем месте администратора программы необходимо разместить справочник **УчетныеЗаписиЭлектроннойПочты** и общую команду **НастройкаСистемнойУчетнойЗаписиЭлектроннойПочты**. См. пример в форме **Органайзер** обработки **ПанельАдминистрированияБСП**.

### Особые случаи внедрения подсистемы

● [Внедрение подсистем «Работа с почтовыми сообщениями» и «Управление доступом»](https://its.1c.ru/db/content/bsp23doc/src/3.59.%20%D0%BE%D1%81%D0%BE%D0%B1%D1%8B%D0%B5%20%D1%81%D0%BB%D1%83%D1%87%D0%B0%D0%B8%20%D0%B2%D0%BD%D0%B5%D0%B4%D1%80%D0%B5%D0%BD%D0%B8%D1%8F%20%D0%BF%D0%BE%D0%B4%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC.htm?_=1504708101#_внедрение_подсистем_).

### Настройка прав доступа пользователей

Для настройки прав доступа пользователей к данным подсистемы «Работа с почтовыми сообщениями» следует использовать роли:

Таблица 3.94.

| **№** | **Роли и их назначение** |
| --- | --- |
| 1. | **ЧтениеУчетныхЗаписейЭлектроннойПочты**  Чтение учетных записей электронной почты, возможность компоновать и отправлять сообщения электронной почты.  При совместном использовании с подсистемой «Управление доступом» для ограничения доступа к отдельным учетным записям электронной почты используется вид доступа **УчетныеЗаписиЭлектроннойПочты** |
| 2. | **ПолныеПрава** (из подсистемы «Базовая функциональность»)  Добавление и изменение учетных записей электронной почты.  Удаление помеченных на удаление объектов подсистемы |

Пример настройки прав доступа пользователей:

Таблица 3.95.

| **№** | **Группа пользователей и ее функции** | **Состав ролей** |
| --- | --- | --- |
| 1. | Пользователь:  ● Чтение учетных записей электронной почты, отправка электронного сообщения, получение электронного сообщения по учетной записи | ● **БазовыеПрава** (из подсистемы «Базовая функциональность»),  ● **ЗапускТонкогоКлиента** (из подсистемы «Базовая функциональность»),  **●ЧтениеУчетныхЗаписейЭлектроннойПочты** |
| 2. | Администратор:  ● Добавление новых и изменение существующих учетных записей электронной почты.  ● Управление доступом пользователей к учетным записям | ● **Полные права** (из подсистемы «Базовая функциональность») |

## Использование при разработке конфигурации

Программный интерфейс подсистемы доступен через экспортные функции общих модулей **РаботаСПочтовымиСообщениями** и **РаботаСПочтовымиСообщениямиКлиент**.

### Настройка обмена данными

Все объекты метаданных подсистемы, содержащие данные, рекомендуется включить в планы обмена распределенной ИБ (РИБ).

# 3.44. Работа с файлами

Подсистема «Работа с файлами» предназначена для коллективного редактирования файлов произвольного формата в иерархической структуре папок. Хранение файлов может быть организовано как непосредственно в информационной базе, так и внешним образом – в томах (сетевых ресурсах). Вместе с файлом может быть сохранена и история его изменений (версии файла).

Поддерживаются такие возможности работы с файлами, как добавление файлов в базу из файловой системы, создание файлов по шаблону, электронная подпись и шифрование, поддержка группового изменения и полнотекстового поиска по файлам.

Кроме того, имеется возможность присоединять файлы к произвольным объектам конфигурации ссылочного типа.

## Настройка

Выполнить настройку общей файловой функциональности – см. раздел «[Файловые функции](https://its.1c.ru/db/content/bsp23doc/src/3.55.%20%D1%84%D0%B0%D0%B9%D0%BB%D0%BE%D0%B2%D1%8B%D0%B5%20%D1%84%D1%83%D0%BD%D0%BA%D1%86%D0%B8%D0%B8.htm?_=1504708101#_файловые_функции)».

### Настройка присоединения файлов к объектам

Необходимо принять решение по поводу состава объектов конфигурации («объекты с файлами»), к которым необходимо присоединять файлы (вложения). Такими объектами, как правило, являются справочники, при которых могут содержаться файлы-вложения. Примеры:

● справочник **Товары** с присоединенными к нему файлами: спецификациями, сертификатами и т. п.;

● справочник **Должности** с приложенными к нему текстами должностных обязанностей, требованиями к соискателю и т. п.

**Важно!**

Состав «объектов с файлами» следует определять сдержанно, так как использование этой функциональности приводит к дополнительным затратам на запись и чтение файлов и самих объектов с файлами в случае использования ограничений доступа (см. раздел «[Управление доступом](https://its.1c.ru/db/content/bsp23doc/src/3.53.%20%D1%83%D0%BF%D1%80%D0%B0%D0%B2%D0%BB%D0%B5%D0%BD%D0%B8%D0%B5%20%D0%B4%D0%BE%D1%81%D1%82%D1%83%D0%BF%D0%BE%D0%BC.htm?_=1504708101#_управление_доступом)»).

Выполнить настройку таких «объектов с файлами»:

1. Задать тип определяемых типов **ВладелецФайлов** (ссылки) и **ВладелецФайловОбъект** (объекты, кроме документов), поместив туда ссылки на объекты с файлами.

2. Обеспечить удаление присоединенных файлов документов с помощью подписки на событие **ПередЗаписью**(например, см. подписку на событие **\_** и **\_ДемоУстановитьПометкуУдаленияФайловДокументов**).

Если перед закрытием формы объекта-владельца требуется проверить наличие занятых текущим пользователем файлов, то необходимо:

● Добавить реквизит формы **МожноЗакрытьФормуСФайлами** типа **Булево**.

● В обработчике формы **ПередЗакрытием** вставить вызов процедуры **ПоказатьПодтверждениеЗакрытияФормыСФайлами** общего модуля **РаботаСФайламиКлиент** по шаблону:

&НаКлиенте

Процедура ПередЗакрытием(Отказ, СтандартнаяОбработка)

РаботаСФайламиКлиент.ПоказатьПодтверждениеЗакрытияФормыСФайлами(ЭтотОбъект, Отказ, Объект.Ссылка);

КонецПроцедуры

● Другой код, который необходимо выполнять в обработчике **ПередЗакрытием**, следует размещать после вызова процедуры и проверки параметра **Отказ**. Например:

&НаКлиенте

Процедура ПередЗакрытием(Отказ, СтандартнаяОбработка)

РаботаСФайламиКлиент.ПоказатьПодтверждениеЗакрытияФормыСФайлами(ЭтотОбъект, Отказ, Объект.Ссылка);

Если Отказ Тогда

Возврат;

КонецЕсли;

<другой код...>

КонецПроцедуры

### Размещение в командном интерфейсе

Для использования подсистемы необходимо разместить в командном интерфейсе администратора команды справочника **Файлы**:

● Команда **Файлы** (имя **ПапкиФайлов**) – открывает форму дерева папок с отображением файлов в выбранной папке.

● Команда **Все файлы**– отображает все файлы, которые содержатся в справочнике.

● Команда **Редактируемые файлы** (имя **ЗанятыеФайлы**) – отображает все редактируемые файлы, которые содержатся в справочнике.

См. пример в демонстрационной конфигурации в подразделе «Работа с файлами» раздела «Органайзер» (подсистема **\_ДемоРаботаСФайлами** подсистемы **\_ДемоОрганайзер**).

Для редактирования персональных настроек работы с файлами следует предусмотреть в конфигурации форму настройки и разместить на ней следующие элементы:

● Действие при выборе файла.

● Выбор режима открытия при выборе файла.

● Показывать подсказки при редактировании файлов (только веб-клиент).

● Показывать занятые файлы при завершении работы.

● Показывать колонку **Размер** в списках файлов.

● Сравнивать версии при помощи.

● Настройка основного рабочего каталога.

● Настройка сканирования.

● Установить расширение работы с криптографией.

Пример реализации персональных настроек работы с файлами можно найти на закладке **Работа с файлами**общей формы **\_ДемоМоиНастройки**.

### Настройка прав доступа пользователей

Для настройки прав доступа пользователей к данным подсистемы «Работа с файлами» следует использовать роли, приведенные ниже.

Таблица 3.96.

| **№** | **Роли и их назначение** |
| --- | --- |
| 1. | **ПолныеПрава** (из подсистемы «Базовая функциональность»)  Включение и отключение использования подсистемы «Работа с файлами». Удаление помеченных на удаление объектов подсистемы.  Полный доступ к файлам без каких-либо ограничений доступа к данным |
| 2. | **РаботаСПапкамиФайлов** (из подсистемы «Работа с файлами»  Просмотр, добавление, редактирование файлов в папках. Права на каждую папку файлов настраиваются отдельно в режиме **1С:Предприятие** |

### Особые случаи внедрения подсистемы

● [Внедрение подсистемы «Работа с файлами» вместе с подсистемой «Присоединенные файлы](https://its.1c.ru/db/content/bsp23doc/src/3.59.%20%D0%BE%D1%81%D0%BE%D0%B1%D1%8B%D0%B5%20%D1%81%D0%BB%D1%83%D1%87%D0%B0%D0%B8%20%D0%B2%D0%BD%D0%B5%D0%B4%D1%80%D0%B5%D0%BD%D0%B8%D1%8F%20%D0%BF%D0%BE%D0%B4%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC.htm?_=1504708101#_внедрение_подсистемы_).

● [Внедрение подсистем «Работа с файлами» и «Управление доступом»](https://its.1c.ru/db/content/bsp23doc/src/3.59.%20%D0%BE%D1%81%D0%BE%D0%B1%D1%8B%D0%B5%20%D1%81%D0%BB%D1%83%D1%87%D0%B0%D0%B8%20%D0%B2%D0%BD%D0%B5%D0%B4%D1%80%D0%B5%D0%BD%D0%B8%D1%8F%20%D0%BF%D0%BE%D0%B4%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC.htm?_=1504708101#_внедрение_подсистем_).

● [Внедрение подсистемы «Работа с файлами» без подсистемы «Электронная подпись»](https://its.1c.ru/db/content/bsp23doc/src/3.59.%20%D0%BE%D1%81%D0%BE%D0%B1%D1%8B%D0%B5%20%D1%81%D0%BB%D1%83%D1%87%D0%B0%D0%B8%20%D0%B2%D0%BD%D0%B5%D0%B4%D1%80%D0%B5%D0%BD%D0%B8%D1%8F%20%D0%BF%D0%BE%D0%B4%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC.htm?_=1504708101#_внедрение_подсистемы_).

● [Внедрение подсистемы «Работа с файлами» без подсистемы «Свойства»](https://its.1c.ru/db/content/bsp23doc/src/3.59.%20%D0%BE%D1%81%D0%BE%D0%B1%D1%8B%D0%B5%20%D1%81%D0%BB%D1%83%D1%87%D0%B0%D0%B8%20%D0%B2%D0%BD%D0%B5%D0%B4%D1%80%D0%B5%D0%BD%D0%B8%D1%8F%20%D0%BF%D0%BE%D0%B4%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC.htm?_=1504708101#_внедрение_подсистем_).

## Использование при разработке конфигурации

Кроме возможности прикрепления файлов к тем или иным объектам ссылочного типа подсистема предоставляет следующие дополнительные возможности.

### Работа с одним присоединенным файлом

При необходимости можно добавить к произвольному объекту ссылочного типа реквизит типа **СправочникСсылка.Файлы**. Например, реквизит **ПрисоединенныйФайл** или **Фотография** в справочнике физических лиц. Использование этого реквизита в пользовательском интерфейсе определяется по месту и может быть, например, таким:

● Поле ввода и кнопки рядом с ним (**Открыть**, **Занять** и пр.).

● Поле картинки в режиме гиперссылки, при нажатии на которую открывается карточка файла.

● Гиперссылка в форме объекта, при нажатии на которую открывается карточка файла или сам файл.

После добавления реквизита типа **СправочникСсылка.Файлы** к объекту метаданных (объекту с файлом) необходимо также:

● Добавить ссылку на объект с файлом к составному типу реквизита **ВладелецФайла** справочника **Файлы**.

● Добавить ссылку на объект с файлом к составному типу параметра **ВладелецФайла** формы элемента справочника **Файлы**.

### Интерфейс работы с компонентом сканирования (TWAIN)

Методы:

1. **Версия()** – вернет строку с описанием версии.

2. **ВыбратьУстройство()** – откроет диалог (средствами драйвера TWAIN) и вернет строку с идентификатором сканера или другого устройства.

3. **НачатьСканирование**(**ОткрыватьДиалог**, **ИмяУстройства**(**Строка**), **ФорматКартинки**(**Строка**), **Разрешение**, **Цветность**, **Поворот**, **РазмерБумаги**, **Сжатие**) – выполнит сканирование и сохранит файл на диск во временный каталог Windows, вернет **Истина**, если не было ошибок. Если **ОткрыватьДиалог = Истина** то откроет диалог сканирования (поставляемый драйвером сканера) с возможностью настроек разрешения и пр.:

* ● **ОткрыватьДиалог**(**Булево**);
* ● **ИмяУстройства**(**Строка**);
* ● **ФорматКартинки**(**Строка**) – формат сохранения картинки. По умолчанию PNG;
* ● **Разрешение**(**Число**) – разрешение DPI, например 200;
* ● **Цветность** (**Число**) – принимает значения 0 (черно-белое), 1 (оттенки серого), 2 (цветное);
* ● **Поворот** (**Число**) – угол поворота в градусах, допустимы значения 0, 90, 180, 270. Не все сканеры поддерживают эту настройку;
* ● **РазмерБумаги** (**Число**). Не все сканеры поддерживают эту настройку. Принимает значения:
* ● 0 (не задано),
* ● 11 (А3),
* ● 1 (А4),
* ● 5 (А5),
* ● 6 (B4),
* ● 2 (B5),
* ● 7 (B6),
* ● 14 (C4),
* ● 15 (C5),
* ● 16 (C6),
* ● 3 (USLetter),
* ● 4 (USLegal),
* ● 10 (USExecutive.
* ● **Сжатие** (**Число**) – для формата JPG сюда передается качество (от 1 до 100, 100 – максимальное качество и максимальный размер), для формата TIF – сжатие. Значения:
* ● 2 – LZW,
* ● 3 – CCITT3,
* ● 4 – CCITT4,
* ● 5 – RLE,
* ● 6 – без сжатия.
* ● **ДвустороннееСканирование** (**Булево**).

4. **ЕстьУстройства** – вернет **Истина**, если есть хоть одно TWAIN-устройство.

5. **ПолучитьУстройства** – вернет список доступных устройств в виде текстовых строк, разделенных разделителем (переводом строк).

6. **Тест** – то же, что **НачатьСканирование,** но не обращается к TWAIN-драйверу, а сразу генерирует событие. Используется для тестирования.

7. **ПолучитьНастройку**(**ИмяУстройства**, **ИмяНастройки**) – возвращает значение указанной настройки (**Число**). Возможные имена настроек:

* ● XRESOLUTION (разрешение – DPI);
* ● PIXELTYPE (цветность);
* ● ROTATION (поворот);
* ● SUPPORTEDSIZES (размер бумаги);
* ● DUPLEX (двусторонне сканирование).

8. **ОбъединитьВМногостраничныйФайл** (**СтрокаВсехПутей**, **ФайлРезультата**,**ПутьКПрограммеКонвертации**) – объединяет несколько изображений в один многостраничный файл (TIF или PDF). Для объединения в PDF требуется установка программы ImageMagick и Ghostscript:

* ● **СтрокаВсехПутей** – пути ко все файлам изображений, разделенные символом «**\***»;
* ● **ФайлРезультата** – путь к результирующему файлу – TIF или PDF;
* ● **ПутьКПрограммеКонвертации** – путь к программе конвертации (ImageMagick). Нужен, только если **ФайлРезультата** имеет расширение PDF.

События:

1. Внешнее событие. **Источник = TWAIN** **Событие = ImageAcquired**, в параметре **Данные** передается полный путь к файлу картинки.

2. Внешнее событие. **Источник = TWAIN** **Событие = EndBatch** – приходит при окончании сканирования пачки изображений (например, с потокового сканера).

3. Внешнее событие. **Источник = TWAIN** **Событие = UserPressedCancel** – приходит, когда пользователь нажал **Отмена** в стандартном диалоге сканирования (поставляемом производителем сканера).

### Настройка обмена данными

Для настройки обмена данными следует руководствоваться общими правилами.

Регистр сведений **ХранимыеФайлыВерсий** надо использовать только на момент создания начального образа подчиненного узла в распределенной ИБ. (см. раздел «[Особенности создания начального образа подчиненного узла распределенной ИБ](https://its.1c.ru/db/content/bsp23doc/src/3.28.%20%D0%BE%D0%B1%D0%BC%D0%B5%D0%BD%20%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D0%BC%D0%B8.htm?_=1504708101#_особенности_создания_начального)»).

В планы обмена рекомендуется включать все объекты метаданных подсистемы, кроме:

● регистра сведений **РабочиеКаталогиФайлов**,

● регистра сведений **ФайлыВРабочемКаталоге**.

# 3.45. Рассылка отчетов

Подсистема «Рассылка отчетов» позволяет настраивать рассылки вариантов отчетов и отчетов подсистемы «Дополнительные отчеты и обработки». Рассылки могут выполняться как по расписанию, так и по требованию.

## Настройка

После завершения переноса объектов подсистемы необходимо выполнить ее настройку:

● Настроить типы получателей рассылки.

● Расширить состав поддерживаемых форматов файлов.

● Разместить в командном интерфейсе.

### Настроить типы получателей рассылки

В качестве получателей рассылки могут выступать различные объекты базы данных с одним общим правилом – для этих объектов должна вестись контактная информация в разрезе еmail-адресов. В процессе настройки рассылки пользователь выбирает тип получателей и вид контактной информации.

Использование справочников **Пользователи** и **Группы пользователей** в качестве получателей **Пользователи**предопределено для подсистемы.

Для подключения дополнительных типов получателей необходимо в справочнике **РассылкиОтчетов**:

1. Перечислить типы получателей в свойстве **Тип** определяемого типа **ПолучательРассылки**.

2. В модуле **РассылкаОтчетовПереопределяемый**:

* ● (Опционально) Задать настройки каждого типа через процедуру **ПереопределитьТаблицуТиповПолучателей**: представление, основной вид контактной информации типа E-mail, путь к форме выбора и дополнительный тип, дополняющий элементы основного типа. Если не задать настройки, то будут использованы настройки типа по умолчанию.
* ● (Опционально) Написать код (или изменить запрос) формирования списка получателей в процедуре **ПередФормированиемСпискаПолучателейРассылки**. Эта процедура вызывается непосредственно перед выполнением запроса по получателям. Может потребоваться только в случае использования нестандартной иерархии (например, как в связке справочников **Пользователи** и **ГруппыПользователей**).

### Расширить состав поддерживаемых форматов файлов

Если требуется расширить состав форматов, поддерживаемых подсистемой **РассылкаОтчетов**, то необходимо:

1. Добавить новые форматы в перечисление **ФорматыСохраненияОтчетов**.

2. В модуле **РассылкаОтчетовПереопределяемый**:

* ● Определить параметры формата в процедуре **ПереопределитьПараметрыФорматов**.
* ● Указать обработчик сохранения в новые форматы в процедуре **ПередСохранениемТабличногоДокументаВФормат**.

### Разместить в командном интерфейсе

Для использования подсистемы необходимо разместить командном интерфейсе ответственных за рассылки объект метаданных:

● Справочник **РассылкиОтчетов** – для настройки и интерактивного выполнения рассылок.

В случае если в конфигурации не используется подсистема «Настройки программы», команду открытия списка справочника **РассылкиОтчетов** следует разместить еще и в рабочем месте администратора программы.

### Подключить нестандартные отчеты

Для участия в рассылках отчеты должны использовать типовой формат хранения и вывода настроек (только средствами СКД) и формирование при помощи метода **СкомпоноватьРезультат** (см. также описание события **ПриКомпоновкеРезультата** в синтакс-помощнике).

Следует найти все отчеты с собственной основной формой и проанализировать, как они формируются и как задаются их настройки.

Если отчет не использует типовой формат хранения и вывода настроек, то следует провести анализ возможности перевода на штатные средства:

● формирование в событии **ПриКомпоновкеРезультата** методом **СкомпоноватьРезультат**,

● настройка при помощи СКД (см. также «[Подключить форму отчета](https://its.1c.ru/db/content/bsp23doc/src/3.8.%20%D0%B2%D0%B0%D1%80%D0%B8%D0%B0%D0%BD%D1%82%D1%8B%20%D0%BE%D1%82%D1%87%D0%B5%D1%82%D0%BE%D0%B2.htm?_=1504708101#_подключить_форму_отчета)»).

Дополнительно рассылка отчетов позволяет переопределять поведение при выборе отдельных параметров СКД. Для этого в модуле **РассылкаОтчетовКлиентПереопределяемый** определены следующие процедуры:

● **ПриАктивизацииСтрокиНастройки** – позволяет запретить непосредственное редактирование значения настройки, например, если в настройке хранится только представление, а фактически, значение хранится в дополнительных свойствах пользовательских настроек СКД.

● **ПриНачалеВыбораНастройки** – позволяет открыть собственную форму для редактирования значения настройки.

● **ПриОчисткеНастройки** – позволяет определить поведение в том случае, если для настройки доступна кнопка очистки.

Если перевод на штатные средства невозможен, то отчет следует добавить в процедуру **ОпределитьИсключаемыеОтчеты** общего модуля **РассылкаОтчетовПереопределяемый**.

### Настройка прав доступа пользователей

Для настройки прав доступа пользователей к данным подсистемы «Рассылка отчетов» следует использовать роли, указанные ниже.

Таблица 3.97.

| **№** | **Роли и их назначение** |
| --- | --- |
| 1. | **ДобавлениеИзменениеРассылокОтчетов**  Настройка и выполнение рассылок отчетов.  Включает в себя право «Вывод».  Используется совместно с ролью **ЧтениеУчетныхЗаписейЭлектроннойПочты**(из подсистемы «Работа с почтовыми сообщениями») |
| 2. | **ПолныеПрава** (из подсистемы «Базовая функциональность»)  Смена автора рассылки.  Настройка и выполнение рассылок отчетов |

Примеры настройки прав доступа пользователей приведены ниже.

Таблица 3.98.

| **№** | **Группа пользователей и ее функции** | **Состав ролей** |
| --- | --- | --- |
| 1. | Администратор  ● Настройка и выполнение рассылок отчетов  ● Смена ответственных за рассылки отчетов | **ПолныеПрава** (из подсистемы «Базовая функциональность») |
| 2. | Ответственный за рассылки отчетов  ● Настройка и выполнение рассылок отчетов | **БазовыеПрава** (из подсистемы «Базовая функциональность»),  **ЧтениеУчетныхЗаписейЭлектроннойПочты**(из подсистемы «Работа с почтовыми сообщениями»),  **ДобавлениеИзменениеРассылокОтчетов** |

## Использование при разработке конфигурации

### Переопределение флажка «ОтчетПустой»

Если для формирования используется собственная логика (в **ПриКомпоновкеРезультата** свойство **СтандартнаяОбработка** устанавливается в **Ложь**), то в обработчике **ПриКомпоновкеРезультата** через дополнительные свойства пользовательских настроек следует установить признак **ОтчетПустой**:

    КомпоновщикНастроек.ПользовательскиеНастройки.ДополнительныеСвойства.Вставить("ОтчетПустой", <Истина, если отчет не содержит данных>);

Это нужно чтобы у пользователей работал флажок **Отправлять пустой**, который позволяет отключить отправку пустых отчетов. Для определения, что отчет не пустой, рассылка отчетов использует стандартные средства СКД, которые в некоторых случаях могут не работать или работать недостоверно.

Также этот признак можно использовать и в других случаях, когда требуется переопределить типовую логику или когда рассылка не может достоверно определить что отчет пустой. Например, если в отчете есть диаграммы, то всегда считается что отчет не пустой.

## Настройка обмена данными

Объекты метаданных подсистемы, содержащие данные, не рекомендуется включать в планы обмена распределенной ИБ. Рекомендуемая модель – независимая настройка рассылок для каждого узла РИБ.

# 3.46. Регламентные задания

Подсистема «Регламентные задания» позволяет редактировать состав и расписание регламентных заданий, просматривать историю выполнения регламентных и фоновых заданий, а также анализировать ошибки при их выполнении.

## Настройка

Подсистема не требует специальной настройки.

### Размещение в командном интерфейсе

Для использования подсистемы необходимо разместить командном интерфейсе администратора объект метаданных:

● **Обработка.РегламентныеИФоновыеЗадания** – для просмотра фоновых и регламентных заданий, интерактивного выполнения регламентных заданий (настройки обработки регламентных заданий в файловой ИБ).

### Настройка зависимостей регламентных заданий от функциональных опций

В тех случаях, когда выполнение предопределенного регламентного задания зависит от включенных одной или нескольких функциональных опций, необходимо программно управлять признаком **Использование** в зависимости от установленных функциональных опций. Если этого не сделать, регламентное задание будет приводить к запуску сеанса, занимая вычислительные ресурсы сервера «1С:Предприятия». Для настройки зависимостей регламентного задания от функциональных опций необходимо:

1. В состав определяемого типа **МестоХраненияФункциональныхОпций** добавить константы, соответствующие функциональным опциям, используемым для управления регламентными заданиями.

2. Добавить вставку в процедуре **ПриОпределенииЗависимостейРегламентныхЗаданий** общего модуля **РегламентныеЗаданияПереопределяемый**. Например:

Зависимость = Зависимости.Добавить();

Зависимость.РегламентноеЗадание = Метаданные.РегламентныеЗадания.ОбновлениеСтатусовДоставкиSMS;

Зависимость.ФункциональнаяОпция = Метаданные.ФункциональныеОпции.ИспользоватьПочтовыйКлиент;

Также можно настроить доступность регламентного задания в подчиненном узле РИБ, модели сервиса и необходимость отключения задания при перемещении информационной базы (подробнее см. комментарий к процедуре).

3. Дополнительно следует обезопасить выполнение регламентного задания, включенного через консоль или другим способом, минуя включение функциональной опции, вставив в начало процедуры обработки регламентного задания следующий код:

ОбщегоНазначения.ПриНачалеВыполненияРегламентногоЗадания();

### Настройка прав доступа пользователей

Для настройки прав доступа пользователей к данным подсистемы «Регламентные задания» следует использовать роль, указанную ниже.

Таблица 3.99.

| **№** | **Роли и их назначение** |
| --- | --- |
| 1. | **ПолныеПрава** (из подсистемы «Базовая функциональность»)  Добавление и изменение регламентных заданий. Настройка обработки регламентных заданий. Просмотр фоновых заданий |

## Использование при разработке конфигурации

При разработке конфигурации никаких дополнительных действий не требуется.

### Настройка обмена данными

Подсистема не предоставляет собственных данных, кроме констант **НастройкиВыполненияРегламентныхЗаданий** и **ПараметрыБлокировкиРаботыСВнешнимиРесурсами**, которые следует исключить из всех планов обмена.

# 3.47. Резервное копирование ИБ

Подсистема «Резервное копирование ИБ» позволяет проводить резервное копирование информационной базы из режима **1С:Предприятие** «по требованию» либо в соответствии с настроенным расписанием. Также с помощью данной подсистемы можно проводить восстановление информационной базы из резервной копии.

## Настройка

Для использования подсистемы в конфигурации необходимо разместить в командном интерфейсе администратора обработки **Резервное копирование ИБ** и **Настройка резервного копирования ИБ**.

### Настройка прав доступа пользователей

Для настройки прав доступа пользователей к функциям подсистемы «Резервное копирование ИБ» следует использовать роли, указанные ниже.

Таблица 3.100.

| **№** | **Роли и их назначение** |
| --- | --- |
| 1. | **АдминистраторСистемы** (из подсистемы «Базовая функциональность»)  Выполнение настройки, резервного копирования и восстановления информационной базы |

## Использование при разработке конфигурации

При разработке конфигурации никаких дополнительных действий не требуется.

### Настройка обмена данными

Объекты метаданных подсистемы не следует включать в планы обмена распределенной ИБ (РИБ), а также в планы обмена по правилам конвертации, поскольку для различных узлов в них должны храниться разные данные:

● константа **ПараметрыРезервногоКопирования**.

# 3.48. Свойства

Подсистема «Свойства» позволяет создавать и редактировать в режиме **1С:Предприятие** дополнительные свойства произвольных объектов конфигурации (не внося изменения в саму конфигурацию). При этом дополнительные свойства можно выводить как в списках, так и в отчетах программы, как и обычные реквизиты объектов. Объектами со свойствами могут быть любые объекты ссылочного типа..

Дополнительные свойства объектов могут быть двух видов:

● дополнительные реквизиты - это свойства, которые хранятся в самом объекте (в специализированной табличной части). Дополнительные реквизиты являются неотъемлемой частью объекта, вводятся при его редактировании и доступны для редактирования тем же пользователям, которым доступен и сам объект со свойствами. Например, реквизиты вес, размеры, цвет справочника товары.

● дополнительные сведения - не являются неотъемлемыми свойствами объекта. Они, как правило, доступны для редактирования тем пользователям, которые не имеют доступа к самому объекту. При внедрении подсистемы совместно с подсистемой «Управление доступом» имеется возможность ограничивать доступ пользователей к отдельным дополнительным сведениям. Например, дата ближайшей поставки товара.

## Настройка

Перед внедрением подсистемы в конфигурацию необходимо принять решение о составе объектов конфигурации, которые пользователи смогут расширять дополнительными свойствами.В этот список следует включить те объекты, свойства которых с большой вероятностью потребуется определять пользователю. Такими объектами, как правило, являются справочники, которые имеют слишком универсальный характер. Пример – справочник товаров, для которого только при внедрении будет понятна востребованность таких свойств, как вес, размеры, цвет и т. д.

После чего выбрать, какие виды дополнительных свойств необходимы в выбранных объектах:

● только дополнительные реквизиты,

● только дополнительные сведения,

● дополнительные свойства обоих типов.

Затем необходимо принять решение по поводу доступных (для выбора пользователем) типов значений дополнительных свойств. При работе пользователь сможет создавать дополнительные свойства таких типов для всех объектов, в которых будет внедрена подсистема «Свойства». Как правило, в этот список включаются некоторые справочники и перечисления конфигурации. Не рекомендуется включать в состав типов дополнительных свойств такие объекты, как документы, задачи, бизнес-процессы и пр. Также не рекомендуется избыточно расширять состав типов значений дополнительных свойств, т. к. это приведет к усложнению настройки системы пользователем.

### Настройка объектов с дополнительными реквизитами

1. Создать табличную часть **ДополнительныеРеквизиты** с реквизитами:

Таблица 3.101.

| **Имя** | **Тип** | **Подсказка** |
| --- | --- | --- |
| **Свойство** | **ПланВидовХарактеристикСсылка.ДополнительныеРеквизитыИСведения** | Дополнительный реквизит |
| **Значение** | **Характеристика.ДополнительныеРеквизитыИСведения** | Значение дополнительного реквизита |
| **ТекстоваяСтрока** | **Строка неограниченной длины** | Полный текст строкового дополнительного реквизита |

Реквизиту **Значение** в свойстве **Связи параметров выбора** установить связь: **Отбор.Владелец**(**ДополнительныеРеквизиты.Свойство**).

2. В форме объекта рекомендуется создать специальную группу полей или страницу с наименованием **ГруппаДополнительныеРеквизиты** для размещения элементов управления, редактирующих свойства объекта.

3. В обработчике события **ПриСозданииНаСервере** формы объекта необходимо выполнить вызов:

// СтандартныеПодсистемы.Свойства

ДополнительныеПараметры = Новый Структура;

ДополнительныеПараметры.Вставить("ИмяЭлементаДляРазмещения", "ГруппаДополнительныеРеквизиты");

УправлениеСвойствами.ПриСозданииНаСервере(ЭтотОбъект, ДополнительныеПараметры);

// Конец СтандартныеПодсистемы.Свойства

где **ГруппаДополнительныеРеквизиты** – имя группы формы, созданной на шаге 2, в которой будут располагаться поля формы, предназначенные для редактирования дополнительных реквизитов. Если этот параметр не указывать, то элементы управления для редактирования свойств будут размещаться в нижней части формы.

При использования формы для работы с разными объектами следует передать параметр **ПроизвольныйОбъект** со значением **Истина** и в дальнейшем вызывать процедуру **ОбновитьЭлементыДополнительныхРеквизитов** для разных объектов.

4. В модуле формы каждого объекта со свойствами необходимо добавить процедуру:

#Область ОбработчикиКомандФормы

// ОБРАБОТЧИКИ КОМАНД ФОРМЫ

...

// СтандартныеПодсистемы.Свойства

&НаКлиенте

Процедура Подключаемый\_РедактироватьСоставСвойств()

УправлениеСвойствамиКлиент.РедактироватьСоставСвойств(ЭтотОбъект, Объект.Ссылка);

КонецПроцедуры

// Конец СтандартныеПодсистемы.Свойства

#КонецОбласти

5. В обработчике события **ПриОткрытии** добавить код:

// СтандартныеПодсистемы.Свойства

УправлениеСвойствамиКлиент.ПослеЗагрузкиДополнительныхРеквизитов(ЭтотОбъект);

// Конец СтандартныеПодсистемы.Свойства

6. В обработчике события **ОбработкаОповещения** добавить код:

// СтандартныеПодсистемы.Свойства

Если УправлениеСвойствамиКлиент.ОбрабатыватьОповещения(ЭтотОбъект, ИмяСобытия, Параметр) Тогда

ОбновитьЭлементыДополнительныхРеквизитов();

УправлениеСвойствамиКлиент.ПослеЗагрузкиДополнительныхРеквизитов(ЭтотОбъект);

КонецЕсли;

// Конец СтандартныеПодсистемы.Свойства

7. Добавить вспомогательные процедуры:

#Область СлужебныеПроцедурыИФункции

// СтандартныеПодсистемы.Свойства

&НаСервере

Процедура ОбновитьЭлементыДополнительныхРеквизитов()

УправлениеСвойствами.ОбновитьЭлементыДополнительныхРеквизитов(ЭтотОбъект);

КонецПроцедуры

&НаКлиенте

Процедура ОбновитьЗависимостиДополнительныхРеквизитов()

УправлениеСвойствамиКлиент.ОбновитьЗависимостиДополнительныхРеквизитов(ЭтотОбъект);

КонецПроцедуры

&НаКлиенте

Процедура Подключаемый\_ПриИзмененииДополнительногоРеквизита(Элемент)

УправлениеСвойствамиКлиент.ОбновитьЗависимостиДополнительныхРеквизитов(ЭтотОбъект);

КонецПроцедуры

// Конец СтандартныеПодсистемы.Свойства

#КонецОбласти

8. В обработчике события **ПриЧтенииНаСервере** добавить код:

// СтандартныеПодсистемы.Свойства

УправлениеСвойствами.ПриЧтенииНаСервере(ЭтотОбъект, ТекущийОбъект);

// Конец СтандартныеПодсистемы.Свойства

9. В обработчике события **ОбработкаПроверкиЗаполненияНаСервере** добавить код:

// СтандартныеПодсистемы.Свойства

УправлениеСвойствами.ОбработкаПроверкиЗаполнения(ЭтотОбъект, Отказ, ПроверяемыеРеквизиты);

// Конец СтандартныеПодсистемы.Свойства

10. В обработчике события **ПередЗаписьюНаСервере** добавить код:

// СтандартныеПодсистемы.Свойства

УправлениеСвойствами.ПередЗаписьюНаСервере(ЭтотОбъект, ТекущийОбъект);

// Конец СтандартныеПодсистемы.Свойства

### Настройка отложенной инициализации дополнительных реквизитов

Дополнительные реквизиты, расположенные на отдельной странице (закладке) формы, необходимо загружать отложенно. Это позволит существенно ускорить открытие форм.

**Важно!**

При отложенной инициализации будет невозможно выполнить перемещение дополнительных реквизитов при помощи команды **Изменить форму** из группы, в которой они расположены.

1. В обработчике события **ПриСозданииНаСервере** формы объекта, в блоке кода:

// СтандартныеПодсистемы.Свойства

ДополнительныеПараметры = Новый Структура;

ДополнительныеПараметры.Вставить("ИмяЭлементаДляРазмещения", "ГруппаДополнительныеРеквизиты");

УправлениеСвойствами.ПриСозданииНаСервере(ЭтотОбъект, ДополнительныеПараметры);

// Конец СтандартныеПодсистемы.Свойства

дополнить структуру новым параметром **ОтложеннаяИнициализация**:

ДополнительныеПараметры.Вставить("ОтложеннаяИнициализация", Истина);

2. В обработчике **ПриСменеСтраницы** группы страниц, в которой выводятся дополнительные реквизиты, добавить код:

// СтандартныеПодсистемы.Свойства

Если ТекущаяСтраница.Имя = "СтраницаДополнительно"

И Не ЭтотОбъект.ПараметрыСвойств.ВыполненаОтложеннаяИнициализация Тогда

СвойстваВыполнитьОтложеннуюИнициализацию();

УправлениеСвойствамиКлиент.ПослеЗагрузкиДополнительныхРеквизитов(ЭтотОбъект);

КонецЕсли;

// Конец СтандартныеПодсистемы.Свойства

где имя страницы **СтраницаДополнительно** – имя страницы, на которой расположена группа **ГруппаДополнительныеРеквизиты**. Если **ГруппаДополнительныеРеквизиты**являетсястраницей, то используется ее имя.

3. Добавить вспомогательную процедуру:

#Область СлужебныеПроцедурыИФункции

&НаСервере

Процедура СвойстваВыполнитьОтложеннуюИнициализацию()

УправлениеСвойствами.ЗаполнитьДополнительныеРеквизитыВФорме(ЭтотОбъект);

КонецПроцедуры

#КонецОбласти

### Настройка объектов с дополнительными сведениями

Перечислить типы объектов с дополнительными сведениями в составе определяемого типа **ВладелецДополнительныхСведений**. Поддерживаемые типы объектов – документы, справочники, планы видов характеристик, планы счетов, планы видов расчета, бизнес-процессы и задачи.

### Настройка динамических списков

Для возможности отображения дополнительных реквизитов и сведений в формах списков объектов со свойствами необходимо добавить поле **Ссылка** из запроса динамического списка в таблицу формы, связанную с этим динамическим списком. При этом добавленное поле **Ссылка** следует по умолчанию скрыть от пользователей, сняв флаг **Видимость** в свойстве **Пользовательская видимость.**

### Настройка наборов свойств объектов

В большинстве случаев достаточно одного набора свойств для объекта метаданных, например, у всех элементов справочника товаров должны быть такие свойства, как вес, размеры, цвет и т.д. Для этого необходимо:

1. В справочнике **НаборыДополнительныхРеквизитовИСведений** создать предопределенный элемент с именем **Справочник\_<ИмяОбъекта>**, если объект – справочник; **Документ\_<ИмяОбъекта>**, если объект – документ, и т. д. Например: **Справочник\_Сотрудники**, **Документ\_АвансовыйОтчет**, **БизнесПроцесс\_Продажа**. Наименование элемента при этом заполнять не требуется (оно будет заполнено автоматически на основании представления списка того объекта, к которому относится данный элемент).

2. Добавить в обработчике обновления информационной базы вызов процедуры **УстановитьПараметрыНабораСвойств** общего модуля **УправлениеСвойствами** для заполнения настроек добавленного набора свойств.

### Особые случаи настройки наборов свойств

#### Настройка объектов с разными наборами свойств у разных групп объектов

В отдельных случаях может быть необходима более гибкая настройка. Например, в зависимости от вида товара у него могут быть уникальные характеристики: у картриджей для принтеров требуется свойство - **Тип принтера**, а у мебели - **Материал.**

При этом подходе набор свойств, применяемый для конкретного экземпляра объекта, определяется по группе объектов, содержащей ссылку на набор свойств. В демонстрационной конфигурации этот вариант применения свойств показан на справочнике **\_ДемоНоменклатура**. В каждом элементе справочника применяется тот или иной набор свойств в зависимости от вида номенклатуры (группы объектов). Справочник **\_ДемоНоменклатура**имеет реквизит **ВидНоменклатуры**, и в зависимости от его значения элемент справочника получает тот или иной набор дополнительных свойств, ссылка на который извлекается в объекте справочника **\_ДемоВидыНоменклатура**. Для настройки требуется следующее:

1. Создать справочник, значения которого соответствуют разным наборам свойств объекта со свойствами. Например, для справочника **\_ДемоНоменклатура** в демонстрационной конфигурации это справочник **\_ДемоВидыНоменклатуры**.

2. В этом справочнике добавить реквизит **НаборСвойств** типа **СправочникСсылка.НаборыДополнительныхРеквизитовИСведений**. Включить реквизит в состав функциональной опции (если нет, создать) **ИспользоватьДополнительныеРеквизитыИСведения<Имя конфигурации>**. Если справочник группирует объекты более чем одного типа, допустимо добавить два и более реквизитов с подходящими именами, например: **НаборСвойствОшибок**, **НаборСвойствЗадач**.

3. В обработчике **ПередЗаписью** модуля объекта этого справочника выполнить вызов **УправлениеСвойствами**.**ПриЗаписиВидаОбъекта** и передать туда в качестве параметров записываемый объект и имя набора свойств, например:

// СтандартныеПодсистемы.Свойства

УправлениеСвойствами.ПередЗаписьюВидаОбъекта(ЭтотОбъект, "Справочник\_ДемоНоменклатура");

// Конец СтандартныеПодсистемы.Свойства

Если справочник группирует объекты более чем одного типа, следует сделать два или более вызова, указав имена используемых реквизитов, например:

// СтандартныеПодсистемы.Свойства

УправлениеСвойствами.ПередЗаписьюВидаОбъекта(ЭтотОбъект, "Справочник\_ДемоОшибки","НаборСвойствОшибок");

УправлениеСвойствами.ПередЗаписьюВидаОбъекта(ЭтотОбъект, "Справочник\_ДемоЗадачи","НаборСвойствЗадач");

// Конец СтандартныеПодсистемы.Свойства

4. В объекте – владельце свойств создать реквизит, значения которого определяют применение того или иного набора свойств для конкретного экземпляра объекта. Для справочника **\_ДемоНоменклатура** это реквизит **ВидНоменклатуры** типа **СправочникСсылка.\_ДемоВидыНоменклатуры**.

5. В модуле формы объекта реализовать обработчик **ПриИзменении** для элемента формы, редактирующего этот реквизит. Для справочника **\_ДемоНоменклатура** это обработчик:

&НаКлиенте

Процедура ВидНоменклатурыПриИзменении(Элемент)

ОбновитьЭлементыДополнительныхРеквизитов();

КонецПроцедуры

6. В процедуре **ЗаполнитьНаборыСвойствОбъекта** общего модуля **УправлениеСвойствамиПереопределяемый** вставить условие, заполняющее набор свойств для конкретного объекта. Например, для **\_ДемоНоменклатура** это следующее условие:

Процедура ЗаполнитьНаборыСвойствОбъекта(Объект, ТипСсылки, НаборыСвойств, СтандартнаяОбработка, КлючНазначения) Экспорт

…

Если ТипСсылки = Тип("СправочникСсылка.\_ДемоНоменклатура") Тогда

ЗаполнитьНаборСвойствПоВидуНоменклатуры(

Объект, ТипСсылки, НаборыСвойств);

КонецЕсли;

…

КонецПроцедуры

Процедура ЗаполнитьНаборСвойствПоВидуНоменклатуры(Номенклатура, ТипСсылки, НаборыСвойств)

Если ТипЗнч(Номенклатура) = ТипСсылки Тогда

Номенклатура = ОбщегоНазначения.ЗначенияРеквизитовОбъекта(

Номенклатура, "ЭтоГруппа, ВидНоменклатуры");

КонецЕсли;

Если Номенклатура.ЭтоГруппа = Ложь Тогда

Строка = НаборыСвойств.Добавить();

Строка.Набор = ОбщегоНазначения.ЗначениеРеквизитаОбъекта(

Номенклатура.ВидНоменклатуры, "НаборСвойств");

КонецЕсли;

КонецПроцедуры

Если справочник группирует объекты более чем одного типа, следует вписать в условие несколько типов, например:

Если ТипСсылки = Тип("СправочникСсылка.\_ДемоОшибки")

 ИЛИ ТипСсылки = Тип("СправочникСсылка.\_ДемоЗадачи") Тогда

ЗаполнитьНаборСвойствПоПроекту(

Объект, ТипСсылки, НаборыСвойств);

КонецЕсли;

7. В справочнике **НаборыДополнительныхРеквизитовИСведений** создать предопределенную группу с именем **Справочник\_<ИмяОбъекта>**, если объект – справочник: **Документ\_<ИмяОбъекта>**, если объект – документ, и т. д. Например, для справочника **\_ДемоНоменклатура** создана предопределенная группа **Справочник\_\_ДемоНоменклатура**. Наименование элемента при этом заполнять не требуется (оно будет заполнено автоматически на основании представления списка того объекта, к которому относится данный элемент).

8. Добавить в обработчике обновления информационной базы вызов процедуры **УстановитьПараметрыНабораСвойств** общего модуля **УправлениеСвойствами** для заполнения настроек добавленного набора свойств.

#### Настройка объектов с несколькими объединяющимися наборами свойств

Наборы свойств при этом должны объединяться в зависимости от тех или иных реквизитов объекта. Например, в справочнике партнеры могут быть введены клиенты, поставщики, конкуренты и партнеров, с которыми связывают прочие отношения. При этом один и тот же партнер может одновременно быть и тем, и другим, и третьим, а его дополнительные свойства должны динамически выводиться в зависимости от того, каких он типов.

В демонстрационной конфигурации этот вариант показан на справочнике **\_ДемоПартнеры**. В нем введены реквизиты **Клиент**, **Поставщик**, **Конкурент**, **ПрочиеОтношения** типа **Булево**, которые и определяют применение одного или сразу нескольких наборов свойств.

При этом подходе справочник имеет один предопределенный набор свойств для всех экземпляров объектов, а также несколько предопределенных наборов свойств, которые могут применяться к экземплярам данного объекта по тем или иным условиям.

Последовательность настройки такого объекта:

1. Принять решение по поводу наборов свойств объекта. Например, для справочника **\_ДемоПартнеры**демонстрационной конфигурации это четыре набора свойств: реквизиты **Клиент**, **Поставщик**, **Конкурент**, **ПрочиеОтношения**.

2. Выявить реквизиты объекта, значения которых определяют применение того или иного набора свойств для конкретного экземпляра объекта. Например, для справочника **\_ДемоПартнеры** демонстрационной конфигурации это реквизиты **Клиент**, **Поставщик**, **Конкурент**, **ПрочиеОтношения**.

3. В справочнике **НаборыДополнительныхРеквизитовИСведений** создать предопределенную группу с именем **Справочник\_<ИмяОбъекта>**, если объект – справочник; **Документ\_<ИмяОбъекта>**, если объект – документ, и т. д. Например, **Справочник\_Контрагенты**. Наименование элемента при этом заполнять не требуется (оно будет заполнено автоматически на основании представления списка того объекта, к которому относится данный элемент).

4. В этой предопределенной группе для каждого из наборов свойств создать предопределенный элемент с именами **Справочник\_<ИмяОбъекта>\_<ИмяНабораСвойств>**, например **Справочник\_Партнеры\_Клиенты**. Прим этом наименование элемента следует задавать краткое, например **Клиенты**; оно не должно содержать имя и тип объекта метаданных.

5. В этой предопределенной группе создать предопределенный элемент с именем **Справочник\_<ИмяОбъекта>\_Общие** и наименованием **Общие**, который будет содержать общие свойства для всех объектов.

6. В общем модуле **УправлениеСвойствамиПереопределяемый** в процедуре **ЗаполнитьНаборыСвойствОбъекта** следует вписать условие для данного типа объектов. Для справочника **\_ДемоПартнеры** это код вида:

Процедура ЗаполнитьНаборыСвойствОбъекта(Объект, ТипСсылки, НаборыСвойств, СтандартнаяОбработка) Экспорт

…

Если ТипСсылки = Тип("СправочникСсылка.\_ДемоПартнеры") Тогда

ЗаполнитьНаборыСвойствПартнера(Объект, ТипСсылки, НаборыСвойств);

КонецЕсли;

…

КонецПроцедуры

Процедура ЗаполнитьНаборыСвойствПартнера(Партнер, ТипСсылки, НаборыСвойств)

Если ТипЗнч(Партнер) = ТипСсылки Тогда

Объект = ОбщегоНазначения.ЗначенияРеквизитовОбъекта(

Партнер, "Клиент, Конкурент, Поставщик, ПрочиеОтношения, ЭтоГруппа");

Иначе

Объект = Партнер;

КонецЕсли;

Если Объект.ЭтоГруппа = Ложь Тогда

Строка = НаборыСвойств.Добавить();

Строка.Набор = Справочники.НаборыДополнительныхРеквизитовИСведений.Справочник\_Партнеры\_Общие;

Строка.Отображение = ОтображениеОбычнойГруппы.Линия;

Строка.ОтображатьЗаголовок = Истина;

Строка.Заголовок = НСтр("ru = 'Для всех партнеров'");

Если Объект.Клиент Тогда

Строка = НаборыСвойств.Добавить();

Строка.Набор = Справочники.НаборыДополнительныхРеквизитовИСведений.Справочник\_Партнеры\_Клиенты;

Строка.Отображение = ОтображениеОбычнойГруппы.Линия;

Строка.ОтображатьЗаголовок = Истина;

Строка.Заголовок = НСтр("ru = 'Для клиентов'");

КонецЕсли;

…

КонецПроцедуры

В приведенном примере проверяются значения реквизитов справочника **Клиент**, **Поставщик**, **Конкурент**, **ПрочиеОтношения** и наборы свойств добавляются в таблицу **НаборыСвойств**. В поле **Набор** устанавливается ссылка на набор свойств, а в остальные поля могут быть установлены значения оформления набора в форме.

7. В форме объекта необходимо реализовать обработчики **ПриИзменении** для тех реквизитов, которые определят состав наборов свойств, применяемых для данного экземпляра объекта. Например, для формы справочника **\_ДемоПартнеры** это обработчики вида:

&НаКлиенте

Процедура КлиентПриИзменении(Элемент)

// СтандартныеПодсистемы.Свойства

ОбновитьЭлементыДополнительныхРеквизитов();

// Конец СтандартныеПодсистемы.Свойства

КонецПроцедуры

### Настройка дополнительных характеристик объектов метаданных

Для каждого объекта конфигурации со свойствами необходимо открыть и заполнить диалог редактирования дополнительных характеристик.

Если для объекта настроены дополнительные реквизиты:

Таблица 3.102.

|  |  |  |
| --- | --- | --- |
| **Поле ключа** | **Поле отбора видов** | **Значение отбора видов** |
| **Виды характеристик** | | |
| **Справочник.НаборыДополнительныхРеквизитовИСведений.ТабличнаяЧасть.ДополнительныеРеквизиты** | | |
| Свойство | Ссылка | **<Предопределенный элемент справочника НаборыДополнительныхРеквизитовИСведений>** |
| **Значения характеристик** | | |
| **Справочник.<Имя объекта со свойствами>.ТабличнаяЧасть.ДополнительныеРеквизиты** | | |
| Ссылка | Свойство | **Значение** |

Если для объекта настроены дополнительные сведения:

Таблица 3.103.

|  |  |  |
| --- | --- | --- |
| **Поле ключа** | **Поле отбора видов** | **Значение отбора видов** |
| **Виды характеристик** | | |
| **Справочник.НаборыДополнительныхРеквизитовИСведений.ТабличнаяЧасть.ДополнительныеСведения** | | |
| Свойство | Ссылка | **<Предопределенный элемент справочника НаборыДополнительныхРеквизитовИСведений>** |

Таблица 3.104.

|  |  |  |
| --- | --- | --- |
| **Значения характеристик** | | |
| **Поле объекта** | **Поле вида** | **Поле значения** |
| **РегистрСведений.ДополнительныеСведения** | | |
| Объект | Свойство | **Значение** |

где **<Предопределенный элемент (группа) справочника НаборыДополнительныхРеквизитовИСведений>** – предопределенный элемент (группа), созданный на предыдущем этапе.

### Отключение неиспользуемых наборов свойств

В случае если объект, использующий набор свойств, отключен по функциональной опции, имеется возможность отключить неиспользуемый набор свойств. Для этого можно воспользоваться свойством **Используется** набора свойств. При установке значения свойства **Используется** в значение **Ложь** набор свойств перестает отображаться в форме списка, также свойства перестают отображаться в форме объекта-владельца. Для отключения неиспользуемого набора свойств необходимо:

● Написать обработчик обновления, отключающий неиспользуемые наборы свойств при переходе на новую версию. Пример кода обработчика:

ПараметрыНабора = УправлениеСвойствами.СтруктураПараметровНабораСвойств();

ПараметрыНабора.Используется = ПолучитьФункциональнуюОпцию("ИспользоватьВнешнихПользователей");

УправлениеСвойствами.УстановитьПараметрыНабораСвойств("Справочник\_ВнешниеПользователи", ПараметрыНабора);

● Добавить аналогичный код в событие при изменении значения функциональной опции. См. пример в подписке \_**ДемоОбновитьИспользованиеВнешнихПользователей**.

### Размещение в командном интерфейсе

Для использования подсистемы необходимо разместить в командном интерфейсе объект метаданных:

● Справочник **НаборыДополнительныхРеквизитовИСведений** – для работы по просмотру и редактированию состава свойств (редактором состава свойств и администратором).

В случае если в конфигурации не используется подсистема «Настройки программы», в рабочем месте администратора программы необходимо разместить константы:

● **ИспользоватьДополнительныеРеквизитыИСведения** – для включения и выключения подсистемы;

● **ИспользоватьОбщиеДополнительныеЗначения** – для включения и выключения команды добавления свойств с общим списком значений (по образцу);

● **ИспользоватьОбщиеДополнительныеРеквизитыИСведения** – для включения и выключения команды добавления общих свойств.

См. пример в форме **ОбщиеНастройки** обработки **ПанельАдминистрированияБСП**.

### Настройка прав доступа пользователей

Для настройки прав доступа пользователей к данным подсистемы «Свойства» следует использовать роли:

Таблица 3.105.

| **№** | **Роли и их назначение** |
| --- | --- |
| 1. | **БазовыеПрава**  Чтение наборов свойств и дополнительных свойств. Просмотр значений свойств объектов |
| 2. | **ЧтениеДополнительныхСведений**  Чтение дополнительных сведений.  При совместном использовании с подсистемой «Управление доступом» имеется возможность ограничивать доступ пользователей к отдельным дополнительным сведениям. См. раздел «[Особые случаи внедрения подсистем](https://its.1c.ru/db/content/bsp23doc/src/3.59.%20%D0%BE%D1%81%D0%BE%D0%B1%D1%8B%D0%B5%20%D1%81%D0%BB%D1%83%D1%87%D0%B0%D0%B8%20%D0%B2%D0%BD%D0%B5%D0%B4%D1%80%D0%B5%D0%BD%D0%B8%D1%8F%20%D0%BF%D0%BE%D0%B4%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC.htm?_=1504708101#_особые_случаи_внедрения)» |
| 3. | **ИзменениеДополнительныхСведений**  Добавление и изменение дополнительных сведений.  При совместном использовании с подсистемой «Управление доступом» имеется возможность ограничивать доступ пользователей к отдельным дополнительным сведениям. См. раздел «[Особые случаи внедрения подсистемы](https://its.1c.ru/db/content/bsp23doc/src/3.59.%20%D0%BE%D1%81%D0%BE%D0%B1%D1%8B%D0%B5%20%D1%81%D0%BB%D1%83%D1%87%D0%B0%D0%B8%20%D0%B2%D0%BD%D0%B5%D0%B4%D1%80%D0%B5%D0%BD%D0%B8%D1%8F%20%D0%BF%D0%BE%D0%B4%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC.htm?_=1504708101#_особые_случаи_внедрения)» |
| 4. | **ДобавлениеИзменениеДополнительныхРеквизитовИСведений**  Добавление и изменение свойств, наборов свойств, значений свойств объектов |
| 5. | **ПолныеПрава** (из подсистемы «Базовая функциональность»)  Включение и отключение использования свойств. Удаление помеченных на удаление объектов подсистемы |

Пример настройки прав доступа пользователей:

Таблица 3.106.

| **№** | **Группа пользователей и ее функции** | **Состав ролей** |
| --- | --- | --- |
| 1. | Ответственный за настройку состава дополнительных свойств | ● **БазовыеПрава** (из подсистемы «Базовая функциональность»),  ●**ДобавлениеИзменениеДополнительныхРеквизитовИСведений**,  ● **ЗапускТонкогоКлиента** (из подсистемы «Базовая функциональность») |
| 2. | Просмотр значений дополнительных сведений объектов информационной базы | ● **БазовыеПрава** (из подсистемы «Базовая функциональность»),  ● **ЧтениеДополнительныхСведений**,  ● **ЗапускТонкогоКлиента** (из подсистемы «Базовая функциональность») |
| 3. | Редактирование значений дополнительных сведений объектов информационной базы | ● **БазовыеПрава** (из подсистемы «Базовая функциональность»),  ● **ИзменениеДополнительныхСведений**,  ● **ЗапускТонкогоКлиента** (из подсистемы «Базовая функциональность») |

## Особые случаи внедрения подсистемы

● [Внедрение подсистемы «Свойства» без подсистемы «Взаимодействия»](https://its.1c.ru/db/content/bsp23doc/src/3.59.%20%D0%BE%D1%81%D0%BE%D0%B1%D1%8B%D0%B5%20%D1%81%D0%BB%D1%83%D1%87%D0%B0%D0%B8%20%D0%B2%D0%BD%D0%B5%D0%B4%D1%80%D0%B5%D0%BD%D0%B8%D1%8F%20%D0%BF%D0%BE%D0%B4%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC.htm?_=1504708101#_внедрение_подсистемы_).

● [Внедрение подсистемы «Свойства» без подсистемы «Работа с файлами»](https://its.1c.ru/db/content/bsp23doc/src/3.59.%20%D0%BE%D1%81%D0%BE%D0%B1%D1%8B%D0%B5%20%D1%81%D0%BB%D1%83%D1%87%D0%B0%D0%B8%20%D0%B2%D0%BD%D0%B5%D0%B4%D1%80%D0%B5%D0%BD%D0%B8%D1%8F%20%D0%BF%D0%BE%D0%B4%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC.htm?_=1504708101#_внедрение_подсистемы_).

● [Внедрение подсистем «Свойства» и «Управление доступом»](https://its.1c.ru/db/content/bsp23doc/src/3.59.%20%D0%BE%D1%81%D0%BE%D0%B1%D1%8B%D0%B5%20%D1%81%D0%BB%D1%83%D1%87%D0%B0%D0%B8%20%D0%B2%D0%BD%D0%B5%D0%B4%D1%80%D0%B5%D0%BD%D0%B8%D1%8F%20%D0%BF%D0%BE%D0%B4%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC.htm?_=1504708101#_внедрение_подсистем_).

## Использование при разработке конфигурации

При создании новых прикладных объектов метаданных нужно повторно выполнять соответствующую часть процедуры настройки, описанной выше.

### Переименование предопределенных элементов справочника «Наборы дополнительных реквизитов и сведений»

При изменении наименования предопределенного элемента справочника **НаборыДополнительныхРеквизитовИСведений** требуется добавить оперативный обработчик обновления, в котором вызвать процедуру **ОбновитьНаименованияНаборовИСвойств** модуля **УправлениеСвойствами**.

При изменении представления объекта метаданных, который соответствует предопределенному элементу, обработчик обновления добавлять не требуется.

### Настройка обмена данными

Для настройки обмена данными следует руководствоваться общими правилами.

В планы обмена распределенной информационной базы (РИБ) рекомендуется включать все объекты метаданных подсистемы.

В планах обмена распределенной информационной базы (РИБ) рекомендуется отключать регистрацию изменений для следующих объектов метаданных подсистемы (см. также раздел «[Особенности создания начального образа подчиненного узла распределенной ИБ](https://its.1c.ru/db/content/bsp23doc/src/3.28.%20%D0%BE%D0%B1%D0%BC%D0%B5%D0%BD%20%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D0%BC%D0%B8.htm?_=1504708101#_особенности_создания_начального)»):

● константа **ПараметрыДополнительныхРеквизитовИСведений**.

# 3.49. Склонение представлений объектов

Подсистема «Склонение представлений объектов» предназначена для автоматического склонения представлений объектов с возможностью ручной корректировки пользователем.

## Настройка

Принять решение по поводу объектов метаданных конфигурации ссылочного типа, представления которых следует склонять.

Если в конфигурации не используется подсистема «Настройки программы», тогда на рабочем месте администратора программы (администратора сервиса, если конфигурация работает в модели сервиса) необходимо разместить константы:

● **ИспользоватьСервисСклоненияMorpher**,

● **ЛогинДоступаКСервисуMorpher**.

Реквизит **ПарольДоступаКСервисуMorpher** нужно разместить на форме по правилам безопасного хранения паролей (см. документацию к подсистеме «Базовая функциональность»).

См. пример в форме **ОбщиеНастройки** обработки **ПанельАдминистрированияБСП**.

Для каждого объекта метаданных, который определен для встраивания подсистемы склонения объектов, необходимо:

● Все склоняемые объекты перечислить в свойстве **Тип** определяемого типа **ОбъектСклонения**.

● Во всех формах элементов, для которых встраивается склонение, в обработчике **ПриСозданииНаСервере**добавить фрагмент кода:

// СтандартныеПодсистемы.СклонениеПредставленийОбъектов

СклонениеПредставленийОбъектов.ПриСозданииНаСервере(ЭтотОбъект, Представление);

// Конец СтандартныеПодсистемы.СклонениеПредставленийОбъектов

В параметре **Представление** указывается реквизит, который должен склоняться.

● Во всех формах элементов, для которых встраивается склонение, в обработчике **ПриЗаписиНаСервере**добавить фрагмент кода:

* ● Если склоняемое представление – **ФИО**:

// СтандартныеПодсистемы.СклонениеПредставленийОбъектов

СклонениеПредставленийОбъектов.ПриЗаписиНаСервере(ЭтотОбъект, Представление, ТекущийОбъект.Ссылка, Истина, Пол);

// Конец СтандартныеПодсистемы.СклонениеПредставленийОбъектов

В параметре **Представление** указывается реквизит, который должен склоняться.

В 5-м параметре передается пол физического лица (1 – мужской, 2 – женский, **Неопределено** в случае, если пол не указан).

* ● Если склоняемое представление – **Не ФИО**:

// СтандартныеПодсистемы.СклонениеПредставленийОбъектов

СклонениеПредставленийОбъектов.ПриЗаписиНаСервере(ЭтотОбъект, Представление, ТекущийОбъект.Ссылка);

// Конец СтандартныеПодсистемы.СклонениеПредставленийОбъектов

В параметре **Представление** указывается реквизит, который должен склоняться.

● Во всех формах элементов, для которых встраивается склонение, рядом с реквизитом, который необходимо просклонять, добавить команду в виде гиперссылки с названием **Склонения**. Обработчик команды:

* ● Если склоняемое представление – **ФИО**:

// СтандартныеПодсистемы.СклонениеПредставленийОбъектов

СклонениеПредставленийОбъектовКлиент.ОбработатьКомандуСклонения(ЭтотОбъект, Представление, Истина, Пол);

// Конец СтандартныеПодсистемы.СклонениеПредставленийОбъектов

В параметре **Представление** указывается реквизит, который должен склоняться.

В 4-м параметре передается пол физического лица (1 – мужской, 2 – женский, **Неопределено** в случае, если пол не указан).

* ● Если склоняемое представление – **Не ФИО:**

// СтандартныеПодсистемы.СклонениеПредставленийОбъектов

СклонениеПредставленийОбъектовКлиент.ОбработатьКомандуСклонения(ЭтотОбъект, Представление);

// Конец СтандартныеПодсистемы.СклонениеПредставленийОбъектов

В параметре **Представление** указывается реквизит, который должен склоняться.

● Во всех формах элементов, для которых встраивается склонение, в обработчике **ПриИзменении** реквизита, который должен склоняться добавить фрагмент кода:

// СтандартныеПодсистемы.СклонениеПредставленийОбъектов

СклонениеПредставленийОбъектовКлиент.ПриИзмененииПредставления(ЭтотОбъект);

// Конец СтандартныеПодсистемы.СклонениеПредставленийОбъектов

● Во всех формах элементов, для которых встраивается склонение, добавить в модуль формы следующий фрагмент кода:

* ● Если склоняемое представление – **ФИО**:

// СтандартныеПодсистемы.СклонениеПредставленийОбъектов

&НаКлиенте

Процедура Подключаемый\_ПросклонятьПредставлениеПоВсемПадежам() Экспорт

СклонениеПредставленийОбъектовКлиент.ПросклонятьПредставлениеПоВсемПадежам(ЭтотОбъект, Представление, Истина, Пол);

КонецПроцедуры

// Конец СтандартныеПодсистемы.СклонениеПредставленийОбъектов

В параметре **Представление** указывается реквизит, который должен склоняться.

В 4-м параметре передается пол физического лица (1 – мужской, 2 – женский, **Неопределено** в случае, если пол не указан).

* ● Если склоняемое представление – **Не ФИО**:

// СтандартныеПодсистемы.СклонениеПредставленийОбъектов

&НаКлиенте

Процедура Подключаемый\_ПросклонятьПредставлениеПоВсемПадежам() Экспорт

СклонениеПредставленийОбъектовКлиент.ПросклонятьПредставлениеПоВсемПадежам(ЭтотОбъект, Представление);

КонецПроцедуры

// Конец СтандартныеПодсистемы.СклонениеПредставленийОбъектов

В параметре **Представление** указывается реквизит, который должен склоняться.

## Настройка прав доступа пользователей

Для настройки прав доступа пользователей к данным подсистемы «Склонение представлений объектов» следует использовать роли, приведенные ниже.

Таблица 3.107.

| **№** | **Роли и их назначение** |
| --- | --- |
| 1. | **ПолныеПрава** (из подсистемы «Базовая функциональность»)  Включение и отключение использования веб-сервиса склонения **Morpher.ru**, настройка доступа к сервису. Полный доступ к регистру сведений **Склонения представлений объектов** |

## Использование при разработке конфигурации

Программный интерфейс подсистемы представлен экспортными функциями общих модулей:

● **СклонениеПредставленийОбъектов**;

● **СклонениеПредставленийОбъектовКлиент**.

Склонения представлений объектов хранятся в регистре сведений **Склонения представлений объектов**.

## Настройка обмена данными

Для настройки обмена данными следует руководствоваться общими правилами.

В планы обмена распределенной информационной базы (РИБ) рекомендуется включать все объекты метаданных подсистемы.

# 3.50. Структура подчиненности

Подсистема «Структура подчиненности» предоставляет возможность вывода из формы документа, справочника и ПВХ отчета о родительских и дочерних документах, справочниках и ПВХ текущего объекта, а также об их взаимосвязях.

## Настройка

Прежде всего необходимо принять решение по поводу списка документов, справочников и ПВХ для которых требуется выводить отчет **Структура подчиненности**. Затем задать список типов выбранных объектов в типах параметра общей команды **СтруктураПодчиненности**.

Для настройки связи между родительскими и дочерними документами, справочниками и ПВХ, выводимыми в отчет **Структура подчиненности**, необходимо настроить критерий отбора **СвязанныеДокументы**. В свойстве **Тип**требуется указать возможные типы родительских объектов, а в свойстве **Состав** – указать реквизиты подчиненных документов, справочников и ПВХ, в которых будет происходить поиск родительских объектов.

### Настройка переопределяемых модулей

При необходимости вписать реализацию в функции переопределяемого модуля подсистемы **СтруктураПодчиненностиПереопределяемый**.

### Настройка прав доступа пользователей

Для настройки прав доступа пользователей к данным подсистемы «Структура подчиненности» следует использовать следующую роль:

Таблица 3.108.

| **№** | **Роли и их назначение** |
| --- | --- |
| 1. | **ПросмотрСвязанныеДокументы**  Дает пользователю право на команду и общую форму «Структура подчиненности» |

## Использование при разработке конфигурации

### Настройка обмена данными

Для настройки обмена данными никаких действий не требуется, так как подсистема не предоставляет собственных данных.

# 3.51. Текущие дела

Подсистема «Текущие дела» позволяет выводить список дел пользователя на рабочем столе (новые письма, задачи, заявки, несогласованные заказы и т. п.). Список дел определяется разработчиком конфигурации при внедрении подсистемы и может включать в себя важные и обычные дела, однострочные и многострочные, с количественными показателями и без. Список дел сгруппирован по разделам командного интерфейса, предусмотрена возможность настраивать видимость и порядок отображения, автоматическое обновление через заданный интервал.

## Настройка

Разместить основную форму обработки **ТекущиеДела** в правой колонке рабочей области начальной страницы.

Принять решение по поводу состава дел, выводимых на рабочем столе. Например, это могут быть уведомления о новых письмах, задачах, заявках, несогласованных заказах и других делах, с которыми сталкиваются пользователи по своим должностным обязанностям. С каждым делом необходимо связать форму списка или рабочего места, которое должно открываться по нажатию на гиперссылку дела.

Рекомендуется группировать дела по разделам командного интерфейса, в которых размещены команды открытия рабочих мест по разбору этих дел. Например, если работа с письмами и задачами предусмотрена в разделе **Органайзер**, то соответствующие им дела также следует размещать в этом разделе панели **Текущие дела**.

При этом с помощью процедуры **ПриОпределенииПорядкаРазделовКомандногоИнтерфейса** общего модуля **ТекущиеДелаПереопределяемый** можно указать порядок следования разделов панели, который в точности соответствует порядку в интерфейсе программы.

### Добавление нового дела

Для добавления нового дела на панель **Текущие дела** необходимо в процедуре **ПриОпределенииОбработчиковТекущихДел** общего модуля **ТекущиеДелаПереопределяемый** объявить общий модуль или модуль менеджера объекта, который будет предоставлять сведения об этом деле. Например:

Процедура ПриОпределенииОбработчиковТекущихДел(Обработчики) Экспорт

Обработчики.Добавить(Документы.\_ДемоЗаказПокупателя);

КонецПроцедуры

После этого необходимо в указанном модуле определить экспортную процедуру **ПриЗаполненииСпискаТекущихДел** по шаблону:

// Заполняет список текущих дел пользователя.

//

// Параметры:

//  ТекущиеДела – ТаблицаЗначений – таблица с колонками:

//   \* Идентификатор           – Строка – внутренний идентификатор дела, используемый механизмом "Текущие дела".

//   \* ЕстьДела                – Булево – если Истина, дело выводится в списке текущих дел пользователя.

//   \* Важное                  – Булево – если Истина, дело будет выделено красным цветом.

//   \* Представление           – Строка – представление дела, выводимое пользователю.

//   \* Количество              – Число  – количественный показатель дела, выводится в строке заголовка дела.

//   \* Форма                   – Строка – полный путь к форме, которую необходимо открыть при нажатии на гиперссылку

//                                        дела на панели "Текущие дела".

//   \* ПараметрыФормы          – Структура – параметры, с которыми нужно открывать форму показателя.

//   \* Владелец                – Строка, объект метаданных – строковый идентификатор дела, которое будет владельцем для текущего

//                               или объект метаданных подсистема.

//   \* Подсказка               – Строка – текст подсказки.

//

Процедура ПриЗаполненииСпискаТекущихДел(ТекущиеДела) Экспорт

КонецПроцедуры

И вписать в нее код по добавлению дел в таблицу **ТекущиеДела**.

**Важно!**

При добавлении дела необходимо проверять право **Редактирование** текущего пользователя на объект метаданных, предоставляющий это дело, а также значения функциональных опций (если они предусмотрены).

Пример заполнения дела по документам **\_ДемоЗаказПокупателя**:

Процедура ПриЗаполненииСпискаТекущихДел(ТекущиеДела) Экспорт

Если Не ПравоДоступа("Редактирование", Метаданные.Документы.\_ДемоЗаказПокупателя) Тогда

Возврат;

КонецЕсли;

КоличествоЗаказовПокупателя = КоличествоЗаказовПокупателя();

СписокОтбора = Новый СписокЗначений;

СписокОтбора.Добавить(Перечисления.\_ДемоСтатусыЗаказовПокупателей.НеСогласован);

СписокОтбора.Добавить(Перечисления.\_ДемоСтатусыЗаказовПокупателей.Согласован);

ИдентификаторЗаказыПокупателя = "ЗаказыПокупателя";

Дело = ТекущиеДела.Добавить();

Дело.Идентификатор  = ИдентификаторЗаказыПокупателя;

Дело.ЕстьДела       = КоличествоЗаказовПокупателя.Всего > 0;

Дело.Представление  = НСтр("ru = 'Заказы покупателя'");

Дело.Количество     = КоличествоЗаказовПокупателя.Всего;

Дело.Форма          = "Документ.\_ДемоЗаказПокупателя.Форма.ФормаСписка";

Дело.ПараметрыФормы = Новый Структура("Отбор", Новый Структура("СтатусЗаказа", СписокОтбора));

Дело.Владелец       = Метаданные.Подсистемы.\_ДемоОрганайзер;

Дело = ТекущиеДела.Добавить();

Дело.Идентификатор  = "ЗаказыПокупателяНеСогласовано";

Дело.ЕстьДела       = КоличествоЗаказовПокупателя.НеСогласовано > 0;

Дело.Важное         = Истина;

Дело.Представление  = НСтр("ru = 'Не согласовано'");

Дело.Количество     = КоличествоЗаказовПокупателя.НеСогласовано;

Дело.Владелец       = ИдентификаторЗаказыПокупателя;

Дело = ТекущиеДела.Добавить();

Дело.Идентификатор  = "ЗаказыПокупателяСогласовано";

Дело.ЕстьДела       = КоличествоЗаказовПокупателя.Согласовано > 0;

Дело.Представление  = НСтр("ru = 'Согласовано'");

Дело.Количество     = КоличествоЗаказовПокупателя.Согласовано;

Дело.Владелец       = ИдентификаторЗаказыПокупателя;

КонецПроцедуры

### Настройка прав доступа пользователей

Для настройки прав доступа пользователей к функциям подсистемы «Текущие дела» нужно использовать следующую роль:

Таблица 3.109.

| **№** | **Роли и их назначение** |
| --- | --- |
| 1. | **ИспользованиеТекущихДел**  Предоставляет доступ для работы со списком текущих дел |

## Использование при разработке конфигурации

### Настройка обмена данными

Для настройки обмена данными никаких действий не требуется, так как подсистема не предоставляет собственных данных.

# 3.52. Удаление помеченных объектов

Подсистема «Удаление помеченных объектов» предназначена для безвозвратного удаления объектов информационной базы.

## Настройка

В случае если в конфигурации не используется подсистема «Настройки программы», в рабочем месте администратора программы необходимо разместить обработку **Удаление помеченных объектов.**

См. пример в форме **Органайзер** обработки **ПанельАдминистрированияБСП**.

### Очистка мест использования при удалении помеченных

При удалении помеченных объектов производится контроль ссылочной целостности базы путем поиска ссылок на удаляемый объект. Если на удаляемый объект ссылаются другие объекты программы, то объект не удаляется, а пользователю выводятся результаты поиска ссылок на удаляемый объект.

Из результатов поиска автоматически исключаются измерения регистров сведений с признаком **Ведущее**, а также исключения поиска ссылок.

Для исключений поиска следует обеспечить очистку ссылок удаляемого объекта из данных объектов-исключений в обработчике **ПередУдалением** удаляемого объекта (или с помощью подписки на одноименное событие). Например, см. подписку на событие **ВариантыОтчетовПередУдалениемИдентификатораОбъектаМетаданных** в демонстрационной базе. Подробнее про исключения поиска ссылок см. в разделе «[Базовая функциональность](https://its.1c.ru/db/content/bsp23doc/src/3.4.%20%D0%B1%D0%B0%D0%B7%D0%BE%D0%B2%D0%B0%D1%8F%20%D1%84%D1%83%D0%BD%D0%BA%D1%86%D0%B8%D0%BE%D0%BD%D0%B0%D0%BB%D1%8C%D0%BD%D0%BE%D1%81%D1%82%D1%8C.htm?_=1504708101#_настройка_исключений_поиска)».

Для измерений регистров сведений с признаком «Ведущее» специальная обработка не требуется, поскольку при удалении ссылки автоматически удаляются все записи по этому измерению.

### Настройка прав доступа пользователей

Для настройки прав доступа пользователей к данным подсистемы следует использовать следующие роли:

Таблица 3.110.

| **№** | **Роли и их назначение** |
| --- | --- |
| 1. | **ПолныеПрава** (из подсистемы «Базовая функциональность»)  Безвозвратное удаление объектов информационной базы |

## Использование при разработке конфигурации

### Настройка обмена данными

Для настройки обмена данными никаких действий не требуется, так как подсистема не предоставляет собственных данных.

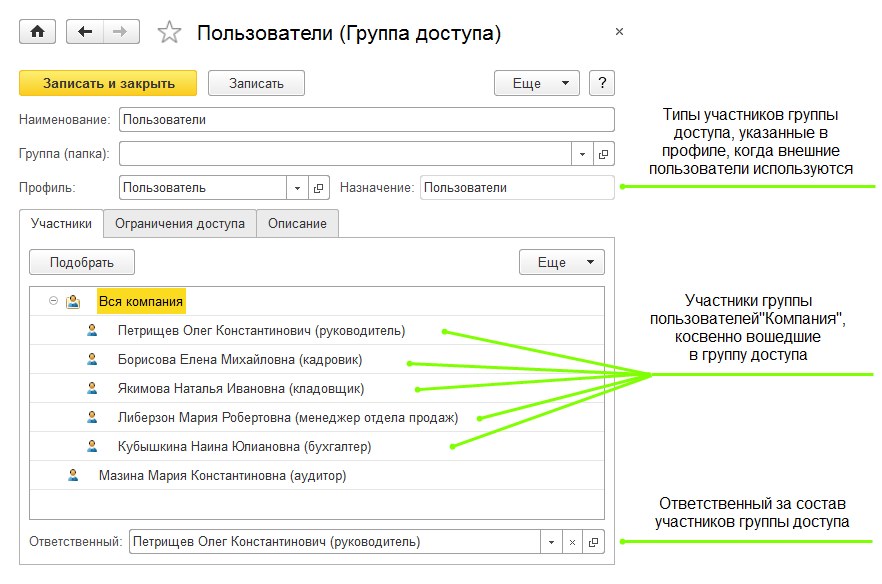
# 3.53. Управление доступом

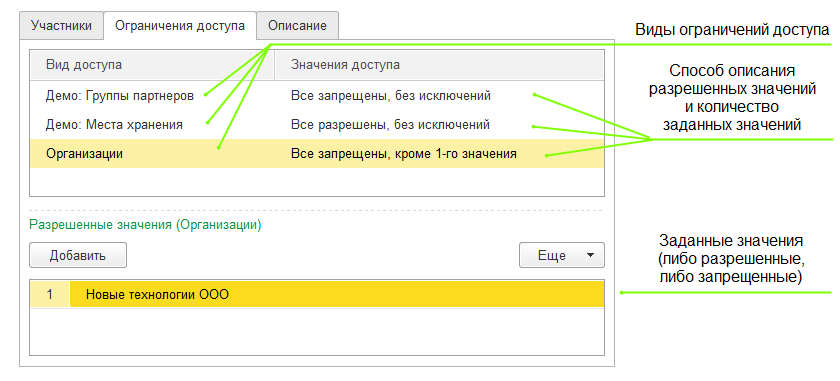
Подсистема «Управление доступом» позволяет настраивать права пользователей для произвольных элементов данных информационной базы (элементов справочников, документов, записей регистров, бизнес-процессов, задач и т. д.). Возможно ограничение прав как для отдельных типов объектов метаданных, так и на уровне записей одного типа объекта. Кроме того, для отдельных объектов информационной базы возможна индивидуальная настройка прав доступа подобно папкам файлов операционной системы.

Для своей работы подсистема использует возможности подсистемы «Пользователи».

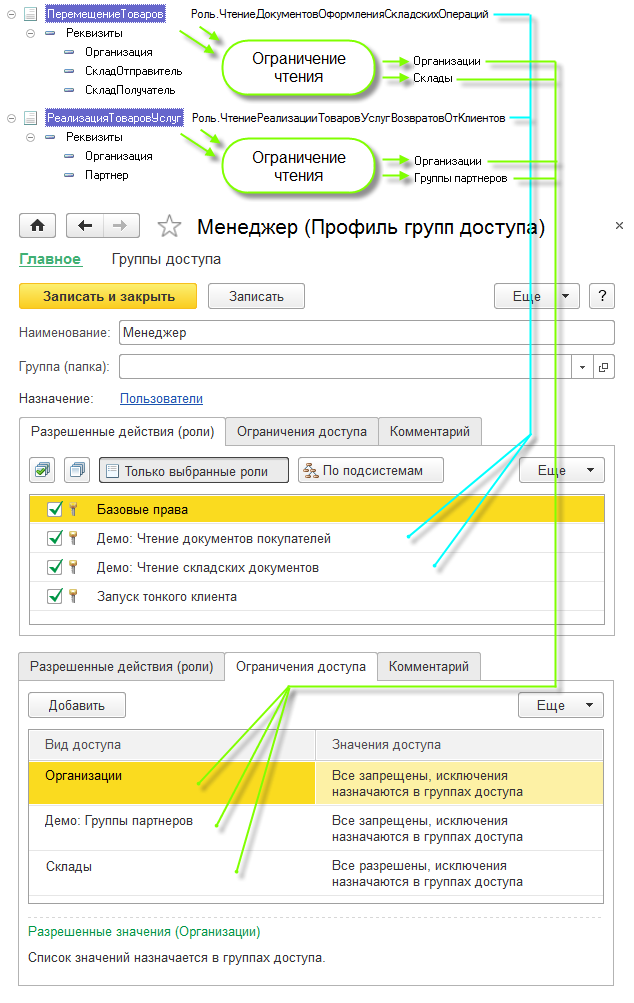
## Группы доступа и их профили

Для настройки прав и ограничений прав пользователей и групп пользователей предназначен справочник **Группы доступа**. Если для некоторых объектов информационной базы разрешен доступ с помощью внешних пользователей, то в группы доступа можно также помещать внешних пользователей и группы внешних пользователей.





Состав и логика ограничения прав доступа определяются профилем, заданным в группе доступа.



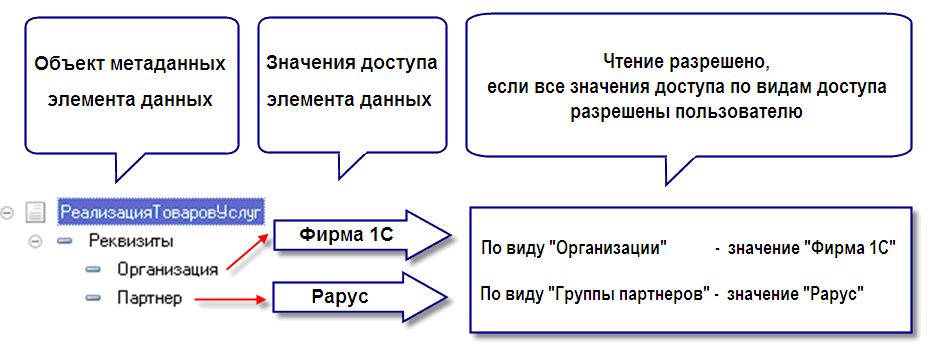
**Профиль групп доступа** представляет собой совокупность ролей (разрешенных действий) и видов доступа (в разрезе которых ограничивается доступ к данным информационной базы).

**Вид доступа** – это тип (или несколько типов) объектов информационной базы, в разрезе которых ограничиваются права доступа к данным информационной базы. Например, право чтения документов информационной базы может быть ограничено в разрезе определенных организаций, разрешенных для пользователя. В простейшем случае, если право чтения ограничено по виду доступа **Организации**, то будут прочитаны только те элементы данных (документы), у которых в поле **Организация** указана организация из списка разрешенных (или незапрещенных) организаций. Если же в группе доступа ограничение по виду доступа **Организации** не задано, то могут быть прочитаны все документы (как будто задан пустой список запрещенных организаций).

В самой группе доступа настраиваются списки разрешенных или запрещенных значений по видам доступа, например: списки организаций, групп партнеров, складов, групп номенклатуры и т. д., которые используются в логике ограничения прав для объектов информационной базы.

Таким образом, можно сказать, что у отдельного пользователя есть право доступа к объекту информационной базы, если пользователю разрешены связанные с этим объектом значения видов доступа.

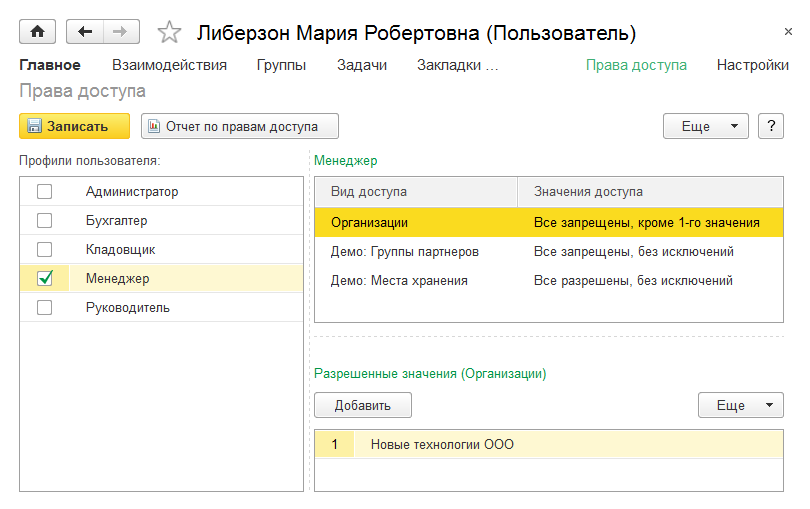
См. также раздел «[Разработка ролей и профилей групп доступа](https://its.1c.ru/db/content/bsp23doc/src/3.53.%20%D1%83%D0%BF%D1%80%D0%B0%D0%B2%D0%BB%D0%B5%D0%BD%D0%B8%D0%B5%20%D0%B4%D0%BE%D1%81%D1%82%D1%83%D0%BF%D0%BE%D0%BC.htm?_=1504708101#_разработка_ролей_и)».



### Упрощенный интерфейс настройки прав доступа

Для простых конфигураций предусмотрен режим упрощенного интерфейса (включается в переопределяемом модуле).

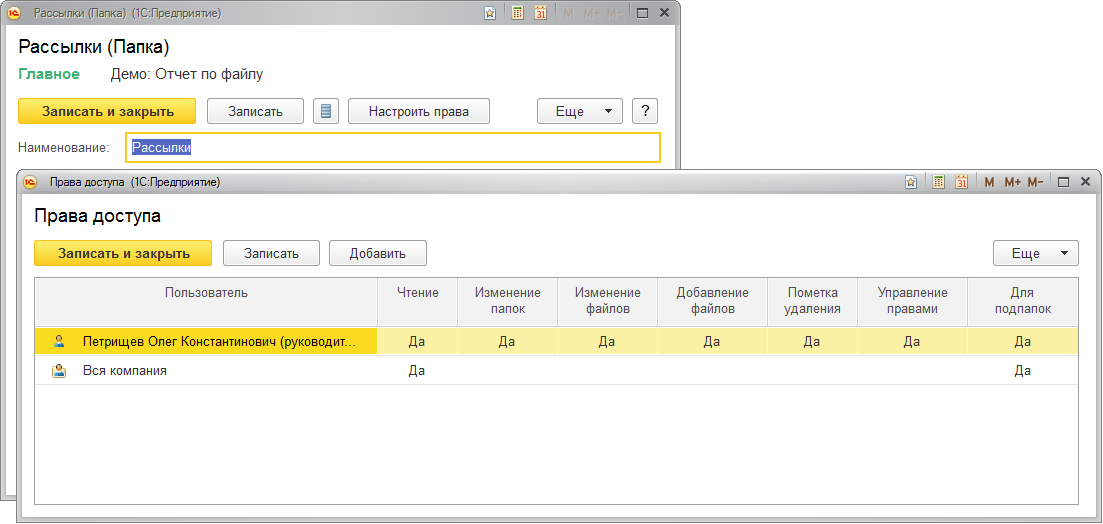
В этом режиме пользователю ставится в соответствие один или несколько профилей, по каждому из которых можно настроить предусмотренные ограничения прав доступа.



### Настройка прав доступа к отдельным объектам информационной базы

Для отдельных объектов информационной базы возможна индивидуальная настройка прав доступа подобно папкам файлов операционной системы.

Например, для объектов справочника **Папки файлов** можно настроить доступ пользователей к отдельным папкам, а также содержащимся в них дочерним папкам и элементам справочника **Файлы**. При этом индивидуальная настройка прав доступа к папкам возможна только для тех пользователей, которые входят в группу доступа пользователей со специальной ролью **Работа с папками файлов**.



Пример настройки групп доступа пользователей:

1. Профиль **Работа с папками файлов (дополнительно)** содержит роль **Работа с папками файлов**, которая предоставляет доступ к справочникам **Папки файлов** и **Файлы**.

2. Группа доступа **Работа с папками файлов (дополнительно)** содержит профиль **Работа с папками файлов (дополнительно)**.

Рекомендуется использовать одну группу доступа для всех пользователей. Тогда сочетание с другими группами доступа будет просто представить, т. к. в группе будут все права на папки, файлы и другие списки, доступ к которым зависит от настроек прав на папки файлов.

## Настройка

Включает в себя последовательность действий:

1. [Выбор режима обычного или упрощенного интерфейса](https://its.1c.ru/db/content/bsp23doc/src/3.53.%20%D1%83%D0%BF%D1%80%D0%B0%D0%B2%D0%BB%D0%B5%D0%BD%D0%B8%D0%B5%20%D0%B4%D0%BE%D1%81%D1%82%D1%83%D0%BF%D0%BE%D0%BC.htm?_=1504708101#_выбор_режима_обычного).

2. [Определить состав ролей и профилей групп доступа в конфигурации](https://its.1c.ru/db/content/bsp23doc/src/3.53.%20%D1%83%D0%BF%D1%80%D0%B0%D0%B2%D0%BB%D0%B5%D0%BD%D0%B8%D0%B5%20%D0%B4%D0%BE%D1%81%D1%82%D1%83%D0%BF%D0%BE%D0%BC.htm?_=1504708101#_определить_состав_ролей).

3. [Описать логику ограничений прав и составить список требуемых видов доступа](https://its.1c.ru/db/content/bsp23doc/src/3.53.%20%D1%83%D0%BF%D1%80%D0%B0%D0%B2%D0%BB%D0%B5%D0%BD%D0%B8%D0%B5%20%D0%B4%D0%BE%D1%81%D1%82%D1%83%D0%BF%D0%BE%D0%BC.htm?_=1504708101#_описать_логику_ограничений).

4. [Разработка видов доступа. Добавление дополнительных метаданных и размещение некоторых из них на формах интерфейса пользователя](https://its.1c.ru/db/content/bsp23doc/src/3.53.%20%D1%83%D0%BF%D1%80%D0%B0%D0%B2%D0%BB%D0%B5%D0%BD%D0%B8%D0%B5%20%D0%B4%D0%BE%D1%81%D1%82%D1%83%D0%BF%D0%BE%D0%BC.htm?_=1504708101#_разработка_видов_доступа.).

5. [Разработка прав для настройки прав отдельных объектов](https://its.1c.ru/db/content/bsp23doc/src/3.53.%20%D1%83%D0%BF%D1%80%D0%B0%D0%B2%D0%BB%D0%B5%D0%BD%D0%B8%D0%B5%20%D0%B4%D0%BE%D1%81%D1%82%D1%83%D0%BF%D0%BE%D0%BC.htm?_=1504708101#_разработка_прав_для_1).

6. [Настройка вычисления использования видов доступа в зависимости от настроек программы](https://its.1c.ru/db/content/bsp23doc/src/3.53.%20%D1%83%D0%BF%D1%80%D0%B0%D0%B2%D0%BB%D0%B5%D0%BD%D0%B8%D0%B5%20%D0%B4%D0%BE%D1%81%D1%82%D1%83%D0%BF%D0%BE%D0%BC.htm?_=1504708101#_настройка_вычисления_использования).

7. [Разработка ролей профилей групп доступа](https://its.1c.ru/db/content/bsp23doc/src/3.53.%20%D1%83%D0%BF%D1%80%D0%B0%D0%B2%D0%BB%D0%B5%D0%BD%D0%B8%D0%B5%20%D0%B4%D0%BE%D1%81%D1%82%D1%83%D0%BF%D0%BE%D0%BC.htm?_=1504708101#_разработка_ролей_и).

8. [Разработка ограничений прав доступа](https://its.1c.ru/db/content/bsp23doc/src/3.53.%20%D1%83%D0%BF%D1%80%D0%B0%D0%B2%D0%BB%D0%B5%D0%BD%D0%B8%D0%B5%20%D0%B4%D0%BE%D1%81%D1%82%D1%83%D0%BF%D0%BE%D0%BC.htm?_=1504708101#_разработка_ограничений_прав).

9. [Разработка процедур «Заполнить наборы значений доступа»](https://its.1c.ru/db/content/bsp23doc/src/3.53.%20%D1%83%D0%BF%D1%80%D0%B0%D0%B2%D0%BB%D0%B5%D0%BD%D0%B8%D0%B5%20%D0%B4%D0%BE%D1%81%D1%82%D1%83%D0%BF%D0%BE%D0%BC.htm?_=1504708101#_разработка_процедур_).

10. [Настройка свойств объектов метаданных](https://its.1c.ru/db/content/bsp23doc/src/3.53.%20%D1%83%D0%BF%D1%80%D0%B0%D0%B2%D0%BB%D0%B5%D0%BD%D0%B8%D0%B5%20%D0%B4%D0%BE%D1%81%D1%82%D1%83%D0%BF%D0%BE%D0%BC.htm?_=1504708101#_настройка_свойств_объектов).

11. [Описание видов доступа, используемых в ограничениях объектов для отчета «Права доступа»](https://its.1c.ru/db/content/bsp23doc/src/3.53.%20%D1%83%D0%BF%D1%80%D0%B0%D0%B2%D0%BB%D0%B5%D0%BD%D0%B8%D0%B5%20%D0%B4%D0%BE%D1%81%D1%82%D1%83%D0%BF%D0%BE%D0%BC.htm?_=1504708101#_описание_видов_доступа,).

12. [Создание описаний поставляемых профилей групп доступа для начального заполнения и обновления информационной базы](https://its.1c.ru/db/content/bsp23doc/src/3.53.%20%D1%83%D0%BF%D1%80%D0%B0%D0%B2%D0%BB%D0%B5%D0%BD%D0%B8%D0%B5%20%D0%B4%D0%BE%D1%81%D1%82%D1%83%D0%BF%D0%BE%D0%BC.htm?_=1504708101#_создание_описаний_поставляемых).

13. [Размещение в командном интерфейсе](https://its.1c.ru/db/content/bsp23doc/src/3.53.%20%D1%83%D0%BF%D1%80%D0%B0%D0%B2%D0%BB%D0%B5%D0%BD%D0%B8%D0%B5%20%D0%B4%D0%BE%D1%81%D1%82%D1%83%D0%BF%D0%BE%D0%BC.htm?_=1504708101#_размещение_в_командном).

14. [Настройка прав доступа пользователей](https://its.1c.ru/db/content/bsp23doc/src/3.53.%20%D1%83%D0%BF%D1%80%D0%B0%D0%B2%D0%BB%D0%B5%D0%BD%D0%B8%D0%B5%20%D0%B4%D0%BE%D1%81%D1%82%D1%83%D0%BF%D0%BE%D0%BC.htm?_=1504708101#_настройка_прав_доступа).

### Выбор режима обычного или упрощенного интерфейса

Для простых конфигураций с небольшим числом пользователей рекомендуется использовать упрощенный интерфейс. В режиме упрощенного интерфейса блокируется изменение поставляемых профилей (создавать новые допустимо).

Для включения упрощенного интерфейса следует, во-первых, изменить возвращаемое значение в переопределяемом модуле.

Процедура ПриОпределенииИнтерфейсаНастройкиДоступа(УпрощенныйИнтерфейс) Экспорт

УпрощенныйИнтерфейс = Истина;

КонецПроцедуры

Во-вторых, следует изменить в общей команде **ПраваДоступа** свойство **Группа** с **Панель навигации формы.Перейти** на **Командная панель формы.Важное**.

Не следует размещать списки профилей и групп доступа в интерфейс администратора (доступ к этим спискам только для нестандартных случаев через «Все функции»).

### Определить состав ролей и профилей групп доступа в конфигурации

Методические рекомендации по проектированию состава ролей конфигурации изложены в разделе «[Базовая функциональность](https://its.1c.ru/db/content/bsp23doc/src/3.4.%20%D0%B1%D0%B0%D0%B7%D0%BE%D0%B2%D0%B0%D1%8F%20%D1%84%D1%83%D0%BD%D0%BA%D1%86%D0%B8%D0%BE%D0%BD%D0%B0%D0%BB%D1%8C%D0%BD%D0%BE%D1%81%D1%82%D1%8C.htm?_=1504708101#_базовая_функциональность)».

Состав профилей может быть определен, например, исходя из должностных обязанностей сотрудников, работающих в системе.

### Описать логику ограничений прав и составить список требуемых видов доступа

При проектировании логики ограничений доступа рекомендуется придерживаться следующего подхода.

Для каждого объекта метаданных предлагается сформулировать правила ограничения доступа.

* ● описать правила ограничения чтения;
* ● описать правила ограничения добавления и изменения.

При описании правил следует использовать определенные реквизиты объекта, содержащие ссылки на объекты (организации, контрагенты: далее – значения доступа), которые будут использоваться для отбора разрешенных объектов (например, документов). Следует уточнять, как именно должны работать отборы (далее – логика ограничения): требуется ли проверять, что, например, организация и контрагент разрешены для пользователя одновременно, или же достаточно, что пользователю разрешена либо организация, либо контрагент.

Например, опишем логику ограничения доступа к документу **Перемещение товаров**, ответив на вопрос «Как нужно ограничить доступ к документу?»:

1. Разрешить чтение документа пользователю, если ему разрешены указанная в документе организация и склад-отправитель или склад-получатель.

2. Разрешить добавление и изменение документа пользователю, если ему разрешены организация и склад-отправитель и склад-получатель.

3. Удаление всегда запрещено, кроме пользователя с ролью **ПолныеПрава**.

Отсюда следует, что доступ к документу должен ограничиваться в разрезе элементов справочников **Организации**и **Склады**. И, следовательно, в список видов доступа надо включить виды доступа **Организации** и **Склады**.

В результате мы составим полный список видов доступа для всех объектов метаданных конфигурации.

В то же время по сформулированным выше правилам в дальнейшем будут разработаны тексты ограничения доступа для ролей конфигурации.

При этом следует иметь в виду, что логика ограничения доступа к объектам метаданных всегда описывается независимо от состава ролей конфигурации, которые предоставляют доступ к объектам метаданных. Это требование продиктовано соображением производительности работы системы: при попытке доступа к объекту от лица пользователя, которому назначено несколько таких ролей, платформа **1С:Предприятие**проверяет совпадение текстов ограничений у прав доступа в этих ролях и, если совпадения нет, выполняет отдельно 2 запроса, каждый со своим текстом ограничения, а результаты запросов объединяет, что сильно снижает производительность системы. Для того чтобы исключить снижение производительности, при наличии в конфигурации нескольких ролей, предоставляющих доступ к одному объекту метаданных, следует указывать в таких ролях одинаковые тексты ограничений у соответствующих прав доступа.

### Разработка видов доступа. Добавление дополнительных метаданных и размещение некоторых из них на формах интерфейса пользователя

Для добавления в конфигурацию нового вида доступа необходимо в модуле **УправлениеДоступомПереопределяемый** в процедуру **ПриЗаполненииВидовДоступа** добавить описания видов доступа, а также расширить состав определяемых типов.

Например, для вида доступа **МестаХранения** указать тип справочник **МестаХранения**. Если доступ планируется ограничивать не по отдельным значениям вида доступа, а по их группам, то дополнительно следует включить типы объектов-групп. Например, для вида доступа **ГруппыНоменклатуры** указать тип значений справочник **Номенклатура** и тип групп значений справочник **ГруппаДоступаНоменклатуры**.

В некоторых случаях у вида доступа может быть несколько типов значений, тогда для их добавления следует использовать процедуру **ДобавитьДополнительныеТипыВидаДоступа** модуля **УправлениеДоступом**.

Может понадобиться для одного значения (например, номенклатуры) указывать не одну группу значений (группу доступа номенклатуры), а несколько. В этом случае следует установить свойство вида доступа **НесколькоГруппЗначений** в **Истина**.

Например, код процедуры может выглядеть следующим образом:

// Заполняет виды доступа, используемые в ограничениях прав доступа.

// Виды доступа Пользователи и ВнешниеПользователи уже заполнены.

// Их можно удалить, если они не требуются для ограничения прав доступа.

//

// Параметры:

//  ВидыДоступа - ТаблицаЗначений - с колонками:

//   \* Имя                    - Строка – имя, используемое в описании поставляемых

//                                       профилей групп доступа и текстах ОДД.

//   \* Представление          - Строка - представляет вид доступа в профилях и группах доступа.

//   \* ТипЗначений            - Тип    - тип ссылки значений доступа.       Например,

//                                       Тип("СправочникСсылка.Номенклатура").

//   \* ТипГруппЗначений       - Тип    - тип ссылки групп значений доступа. Например,

//                                       Тип("СправочникСсылка.ГруппыДоступаНоменклатуры").

//   \* НесколькоГруппЗначений - Булево - Истина указывает, что для значения доступа (Номенклатуры) можно

//                                       выбрать несколько групп значений (Групп доступа номенклатуры).

//

Процедура ПриЗаполненииВидовДоступа(ВидыДоступа) Экспорт

ВидДоступа = ВидыДоступа.Добавить();

ВидДоступа.Имя = "МестаХранения";

ВидДоступа.Представление = НСтр("ru = 'Места хранения'");

ВидДоступа.ТипЗначений   = Тип("СправочникСсылка.\_ДемоМестаХранения");

ВидДоступа = ВидыДоступа.Добавить();

ВидДоступа.Имя = "ГруппыНоменклатуры";

ВидДоступа.Представление    = НСтр("ru = 'Группы номенклатуры'");

ВидДоступа.ТипЗначений      = Тип("СправочникСсылка.\_ДемоНоменклатура");

ВидДоступа.ТипГруппЗначений = Тип("СправочникСсылка.\_ДемоГруппыДоступаНоменклатуры");

УправлениеДоступом.ДобавитьДополнительныеТипыВидаДоступа(ВидДоступа,

Тип("СправочникСсылка.\_ДемоВидыНоменклатуры"),

Тип("СправочникСсылка.\_ДемоГруппыДоступаНоменклатуры"));

…

КонецПроцедуры

Если для вида доступа требуется указать тип групп значений, то нужно создать дополнительный справочник (например, **ГруппыДоступаНоменклатура**). При этом у значения доступа (номенклатуры) нужно создать дополнительный реквизит **ГруппаДоступа** с типом групп значений (например, **ГруппыДоступаНоменклатуры**). Если вдобавок свойство **НесколькоГруппЗначений** установлено **Истина**, тогда вместо реквизита **ГруппаДоступа** нужно создать табличную часть **ГруппыДоступа** с реквизитом **ГруппаДоступа** того же типа.

Добавленные реквизит **ГруппаДоступа** или табличную часть **ГруппыДоступа** с реквизитом **ГруппаДоступа**нужно разместить в форме элемента значения доступа (например, справочника **Номенклатура**).

В определяемый тип **ЗначениеДоступа** нужно добавить типы значений доступа и типы групп значений доступа. В определяемый тип **ЗначениеДоступаСГруппамиЗначенийДоступаОбъект** нужно добавить типы значений доступа, для которых предусмотрены типы групп значений доступа.

Чтобы изменения вступили в силу, требуется обновить вспомогательные данные. См. раздел «[Обновление вспомогательных данных во время разработки](https://its.1c.ru/db/content/bsp23doc/src/3.4.%20%D0%B1%D0%B0%D0%B7%D0%BE%D0%B2%D0%B0%D1%8F%20%D1%84%D1%83%D0%BD%D0%BA%D1%86%D0%B8%D0%BE%D0%BD%D0%B0%D0%BB%D1%8C%D0%BD%D0%BE%D1%81%D1%82%D1%8C.htm?_=1504708101#_обновление_вспомогательных_данных)».

#### Обязательные виды доступа

К обязательным видам доступа, которые входят в состав подсистемы, относятся: **Пользователи**, **ВнешниеПользователи**.

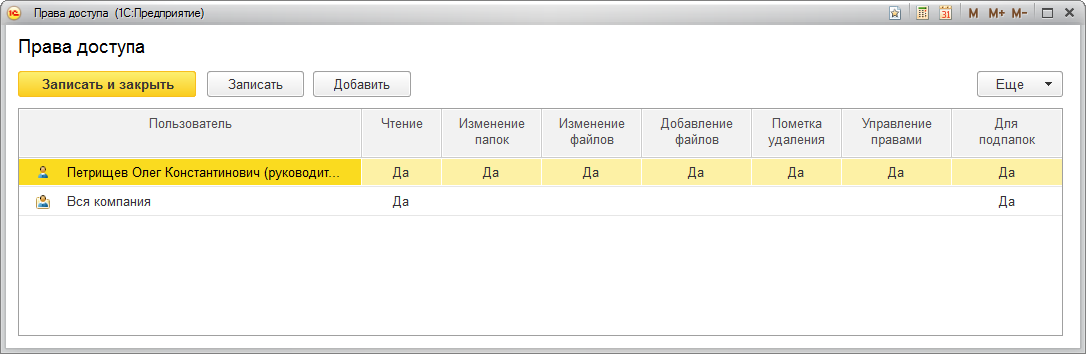
В состав разрешенных пользователей вида доступа **Пользователи** автоматически включается текущий пользователь: аналогично в состав разрешенных внешних пользователей вида доступа **ВнешниеПользователи**автоматически включается текущий внешний пользователь.

#### Настройка функциональной опции «ОграничиватьДоступНаУровнеЗаписей<ИмяПрикладногоРешения>»

Для реквизитов, созданных выше, необходимо добавить функциональную опцию **ОграничиватьДоступНаУровнеЗаписей<ИмяПрикладногоРешения>**, например **ОграничиватьДоступНаУровнеЗаписейУправлениеТоровлей**. В свойство **Хранение** задать константу **ОграничиватьДоступНаУровнеЗаписей**.

### Разработка прав для настройки прав отдельных объектов

В случае необходимости настройки прав для отдельных объектов информационной базы, следует определить список типов этих объектов и в модуле **УправлениеДоступомПереопределяемый** и для каждого типа вписать права в процедуру **ПриЗаполненииВозможныхПравДляНастройкиПравОбъектов**, которые используются для автоматического создания формы настройки этих прав.



Например, для получения состава прав (колонок), как на рисунке, требуется следующий код:

Процедура ПриЗаполненииВозможныхПравДляНастройкиПравОбъектов(ВозможныеПрава) Экспорт

////////////////////////////////////////////////////////////

// Справочник.ПапкиФайлов

// Право "Чтение папок и файлов".

Право = ВозможныеПрава.Добавить();

Право.ВладелецПрав  = Метаданные.Справочники.ПапкиФайлов.ПолноеИмя();

Право.Имя           = "Чтение";

Право.Заголовок     = НСтр("ru = 'Чтение'");

Право.Подсказка     = НСтр("ru = 'Чтение папок и файлов'");

Право.НачальноеЗначение = Истина;

// Права для стандартных шаблонов ограничений доступа.

Право.ЧтениеВТаблицах.Добавить("\*");

// Право "Изменение папок"

Право = ВозможныеПрава.Добавить();

Право.ВладелецПрав  = Метаданные.Справочники.ПапкиФайлов.ПолноеИмя();

Право.Имя           = "ИзменениеПапок";

Право.Заголовок     = НСтр("ru = 'Изменение

                                 |папок'");

Право.Подсказка     = НСтр("ru = 'Добавление, изменение и

                                 |пометка удаления папок файлов'");

// Права, требуемые для этого права.

Право.ТребуемыеПрава.Добавить("Чтение");

// Права для стандартных шаблонов ограничений доступа.

Право.ИзменениеВТаблицах.Добавить(Метаданные.Справочники.ПапкиФайлов.ПолноеИмя());

// Право "Изменение файлов"

Право = ВозможныеПрава.Добавить();

Право.ВладелецПрав  = Метаданные.Справочники.ПапкиФайлов.ПолноеИмя();

Право.Имя           = "ИзменениеФайлов";

Право.Заголовок     = НСтр("ru = 'Изменение

                                 |файлов'");

Право.Подсказка     = НСтр("ru = 'Изменение файлов в папке'");

// Права, требуемые для этого права.

Право.ТребуемыеПрава.Добавить("Чтение");

// Права для стандартных шаблонов ограничений доступа.

Право.ИзменениеВТаблицах.Добавить("\*");

// Право "Добавление файлов"

Право = ВозможныеПрава.Добавить();

Право.ВладелецПрав  = Метаданные.Справочники.ПапкиФайлов.ПолноеИмя();

Право.Имя           = "ДобавлениеФайлов";

Право.Заголовок     = НСтр("ru = 'Добавление

                                 |файлов'");

Право.Подсказка     = НСтр("ru = 'Добавление файлов в папку'");

// Права, требуемые для этого права.

Право.ТребуемыеПрава.Добавить("ИзменениеФайлов");

// Право "Пометка удаления файлов".

Право = ВозможныеПрава.Добавить();

Право.ВладелецПрав  = Метаданные.Справочники.ПапкиФайлов.ПолноеИмя();

Право.Имя           = "ПометкаУдаленияФайлов";

Право.Заголовок     = НСтр("ru = 'Пометка

                                 |удаления'");

Право.Подсказка     = НСтр("ru = 'Пометка удаления файлов в папке'");

// Права, требуемые для этого права.

Право.ТребуемыеПрава.Добавить("ИзменениеФайлов");

Право = ВозможныеПрава.Добавить();

Право.ВладелецПрав  = Метаданные.Справочники.ПапкиФайлов.ПолноеИмя();

Право.Имя           = "УправлениеПравами";

Право.Заголовок     = НСтр("ru = 'Управление

                                 |правами'");

Право.Подсказка     = НСтр("ru = 'Управление правами папки'");

// Права, требуемые для этого права.

Право.ТребуемыеПрава.Добавить("Чтение");

КонецПроцедуры

* ● Указать состав типов как состав владельцев прав в процедуре **ПриЗаполненииВозможныхПравДляНастройкиПравОбъектов**:
* ● в определяемом типе **ВладелецНастроекПрав**,
* ● в определяемом типе **ВладелецНастроекПравОбъект**,
* ● в общей команде **НастроитьПрава**.
* ● Использовать в модуле **УправлениеДоступом** функцию **ЕстьПраво** для получения пользователем разрешения на права.
* ● В ограничениях доступа к данным, которые реализуются с использованием стандартных шаблонов ОДД (см. ниже), следует использовать специальный вид доступа **НастройкиПрав**.

Чтобы изменения вступили в силу, требуется обновить вспомогательные данные. См. раздел «[Обновление вспомогательных данных во время разработки](https://its.1c.ru/db/content/bsp23doc/src/3.4.%20%D0%B1%D0%B0%D0%B7%D0%BE%D0%B2%D0%B0%D1%8F%20%D1%84%D1%83%D0%BD%D0%BA%D1%86%D0%B8%D0%BE%D0%BD%D0%B0%D0%BB%D1%8C%D0%BD%D0%BE%D1%81%D1%82%D1%8C.htm?_=1504708101#_обновление_вспомогательных_данных)».

### Настройка вычисления использования видов доступа в зависимости от настроек программы

В процедуре **ПриЗаполненииИспользованияВидаДоступа** общего модуля **УправлениеДоступомПереопределяемый** можно установить код вычисления использования видов доступа. Например, код процедуры может выглядеть следующим образом:

// Заполняет использование видов доступа в зависимости от функциональных опций конфигурации,

// например "ИспользоватьГруппыДоступаНоменклатуры".

//

// Параметры:

//  ВидДоступа    - Строка - имя вида доступа заданное в процедуре ПриЗаполненииВидовДоступа.

//  Использование - Булево - начальное значение Истина.

//

Процедура ПриЗаполненииИспользованияВидаДоступа(ИмяВидаДоступа, Использование) Экспорт

Если ИмяВидаДоступа = "\_ДемоГруппыНоменклатуры" Тогда

Использование = Константы.\_ДемоОграничиватьДоступПоНоменклатуре.Получить();

ИначеЕсли ИмяВидаДоступа = "\_ДемоГруппыПартнеров" Тогда

Использование = Константы.\_ДемоОграничиватьДоступПоПартнерам.Получить();

ИначеЕсли ИмяВидаДоступа = "\_ДемоФизическиеЛица" Тогда

Использование = Константы.\_ДемоОграничиватьДоступПоФизическимЛицам.Получить();

КонецЕсли;

КонецПроцедуры

При изменении использования вида доступа требуется вызвать обновление разрешенных значений – например, из модуля менеджера значения константы:

Процедура ПриЗаписи(Отказ)

Если ОбменДанными.Загрузка Тогда

Возврат;

КонецЕсли;

УправлениеДоступом.ОбновитьРазрешенныеЗначенияПриИзмененииИспользованияВидовДоступа();

КонецПроцедуры

Для изменения действующего значения в ограничениях доступа к объектам требуется перезапуск сеанса.

### Разработка ролей и профилей групп доступа

Для каждой спроектированной роли:

* ● добавить в конфигурацию объект метаданных «роль», установить имя и синоним; снять флажок **Устанавливать права для новых объектов**; снять флажок **Независимые права подчиненных объектов**;
* ● установить флажки напротив требуемых прав по каждому объекту метаданных конфигурации;
* ● у объектов метаданных, на которые установлены права, снять лишние флажки напротив требуемых прав по полям (реквизитам, табличным частям, измерениям, ресурсам м т. д.);
* ● для каждого права каждого объекта метаданных задать тексты ограничений доступа с использованием стандартных шаблонов. Применяемые в роли стандартные шаблоны следует предварительно скопировать из поставляемой роли **ИзменениеУчастниковГруппДоступа**.

При необходимости переименовать роль следует учитывать методику раздела «[Использование идентификаторов объектов метаданных](https://its.1c.ru/db/content/bsp23doc/src/3.4.%20%D0%B1%D0%B0%D0%B7%D0%BE%D0%B2%D0%B0%D1%8F%20%D1%84%D1%83%D0%BD%D0%BA%D1%86%D0%B8%D0%BE%D0%BD%D0%B0%D0%BB%D1%8C%D0%BD%D0%BE%D1%81%D1%82%D1%8C.htm?_=1504708101#_использование_идентификаторов_объек)».

Затем в режиме **1С:Предприятие** необходимо создать требуемое количество профилей групп доступа пользователей, заполнив их ролями и видами доступа, созданными на предыдущем шаге.

**Важно!**

В профиль следует включать только те виды доступа, которые задействованы в ограничениях прав на списки, настроенные в ролях, выбранных для профиля. Настройка видов доступа в профиле и группе доступа не влияет на ограничение прав на те списки, в ограничении которых виды доступа не задействованы.

В профиле возможно использовать предустановленные виды доступа. Вариант «предустановленный» обозначает, что вид доступа не настраивается в группе доступа, а настраивается только в профиле для всех групп доступа. Среди видов доступа могут быть те, которые имеют смысл больше как предустановленные, нежели настраиваемые в группе доступа. Предустанавливать в профиле можно любой вид доступа, если это удобно (или требуется) с точки зрения администрирования групп доступа этого профиля. Так, виды доступа, которые предназначены, например, для «разрезания» одной таблицы по признаку **ХозяйственнаяОперация**, логично использовать предустановленными. Для кассира – свой список хозопераций, для подотчетного лица – свой, а для бухгалтера – всех хозопераций по отношению, например, к документам **Расходный кассовый ордер**.

Затем надо вписать в процедуру **ПриЗаполненииПоставляемыхПрофилейГруппДоступа** общего модуля **УправлениеДоступомПереопределяемый** код по созданию и обновлению поставляемых профилей групп доступа при начальном заполнении и обновлении информационной базы. Для автоматического формирования текста этой процедуры следует применять обработку **ИнструментыРазработчикаУправлениеДоступом.epf** из демонстрационной конфигурации, которая формирует код на основании профилей групп доступа, существующих в информационной базе.

### Разработка ограничений прав доступа

Для установки текстов ограничений прав доступа используется один из четырех стандартных шаблонов:

#ПоЗначениям(…)

#ПоНаборамЗначений(…)

#ПоЗначениямРасширенный(…)

#ПоЗначениямИНаборамРасширенный(…)

**Примечание**

Подробное описание назначения и параметров шаблонов приведено в комментариях в начале текстов самих шаблонов. Рекомендуется изучить это описание перед прочтением дальнейшего материала. Стандартные шаблоны поставляются в роли **ИзменениеУчастниковГруппДоступа**.

Наличие сразу нескольких шаблонов объясняется необходимостью решать различные задачи. Так, шаблоны **#ПоЗначениям** и **#ПоНаборамЗначений** позволяют решить большую часть простых задач.

Для шаблонов **#ПоЗначениямИНаборамРасширенный** (только для вида доступа **Объект**) и **#ПоНаборамЗначений** можно использовать модификатор **РасширенноеИЛИ**, который меняет расчет ограничений с несколькими проверками, результаты которых объединяются по **ИЛИ**. Например, электронное письмо доступно, если разрешена **Учетная запись** или **Организация**. Без модификатора, если ограничение по учетным записям или организациям не используется, все письма будут разрешены. Случаи неиспользования: выключена константа использования вида доступа, нет вида доступа в профиле.

Шаблоны с модификатором **РасширенноеИЛИ** работают медленнее, поэтому не следует использовать модификатор там, где нет логики ограничения по **ИЛИ**. Если в ограничении используется вид доступа **Объект**, а типы объектов и их логика ограничения заранее неизвестны, модификатор нужно указать, когда требуется расширенный расчет **ИЛИ**.

#### Шаблон «#ПоЗначениям»

Шаблон **#ПоЗначениям** позволяет реализовать проверку разрешения значений доступа в полях таблицы (указывать виды доступа нужно, чтобы текст ограничения автоматически оптимизировался, когда вид доступа не используется). Например, текст ограничения доступа вида:

#ПоЗначениям("Документ.ОприходованиеТоваров", "", "",

"Организации", "Организация",

"Склады", "Склад", …).

ограничивает чтение таким образом, что пользователь увидит в списке документов **Оприходование товаров**только те документы, у которых в реквизите **Организация** есть значение, разрешенное пользователю по виду доступа **Организации**, и в реквизите **Склад** есть значение, разрешенное по виду доступа **Склады**. Причем значения должны быть разрешены в группе доступа пользователей одновременно, а профиль групп доступа пользователей должен включать роль с правом чтения документов **Оприходование товаров**.

В шаблоне можно выполнить максимум 16 проверок.

Этот и другие шаблоны поддерживают передачу условия на языке запросов. Для этого вместо имени вида доступа нужно указать имя **Условие**, а вместо имени поля – текст на языке запросов. Например:

#ПоЗначениям("Документ.ОприходованиеТоваров", "", "",

"Организации", "Организация",

"Склады", "Склад",

"Условие", "Т.ХозяйственнаяОперация = Значение(Перечисление.ХозяйвственныеОперации.Инветаризация) ", …),

В этом примере добавляется условие: документ доступен только при хозяйственной операции **Инвентаризация**. Из этого следует, что при другой хозяйственной операции, например **Прочее**, документ не будет доступен ни одному пользователю, кроме пользователей с ролью **Полные права**.

#### Шаблон «#ПоНаборамЗначений»

В первую очередь шаблон **#ПоНаборамЗначений** предназначен для ограничения чтения журналов документов, у которых документы имеют различные ограничения чтения, например по составу реквизитов. Обычно для журнала приходится делать анализ типа документа, а затем применять те же проверки, что и для документа соответствующего типа, т. е. повторять логику ограничений всех документов в каждом журнале, в который входят документы. Как правило, этот способ очень «дорогой» в плане производительности и трудоемок с точки зрения разработки и сопровождения. Поэтому целесообразно предварительно записывать значения доступа объектных элементов данных, необходимые для работы логики ограничения доступа, в специальный регистр – **НаборыЗначенийДоступа**. Такие данные называются наборами значений доступа.

Например, если для документа **Оприходование товаров** в регистре **НаборыЗначенийДоступа** имеются записи:

<Документ1>, , "Организации", "Организация1"

<Документ1>, , "Склады", "Склад1";

то для их использования текст ограничения права чтения журнала, содержащего этот документ, должен быть таким:

#ПоНаборамЗначений("ЖурналДокументов.СкладскиеДокументы", "", "", "Ссылка"),

где **Ссылка** – имя стандартного реквизита журнала.

В этом случае логика работы шаблона **#ПоНаборамЗначений** такова: по переданному значению стандартного реквизита **Ссылка** строки журнала в регистре **Наборы значений доступа** будут найдены все строки, содержащие **Значение доступа**, и выполнена проверка, что все значения доступа одновременно разрешены пользователю в одной из его групп доступа.

Помимо журналов документов шаблон **#ПоНаборамЗначений** может применяться для реализации логики ограничения доступа и к другим объектам метаданных. При этом использование заранее подготовленных наборов значений позволяет избежать тех ограничений, которые накладываются языком запросом. Применение шаблона **#ПоНаборамЗначений** требует дополнительной настройки объекта метаданных. К объекту метаданных необходимо добавить табличную часть **Наборы значений доступа** с реквизитами, совпадающими с измерениями и ресурсами регистра сведений **Наборы значений доступа**. В тексте ограничения доступа в качестве четвертого параметра шаблона **#ПоНаборамЗначений** указывается пустая строка. Также требуется указать типы этих объектов в подписке на событие **ЗаполнитьНаборыЗначенийДоступаТабличныхЧастей** (для справочников) через определяемый тип **ВладелецСОграничениемПоНаборамЗначенийДоступаОбъект**или в дополнительно созданной подписке на события, имя которой начинается, как имя поставляемой подписки, а обработчик совпадает (только для документов).

#### Шаблон «#ПоЗначениямРасширенный»

В некоторых случаях возможностей шаблонов **#ПоЗначеним** и **#ПоНаборамЗначений** бывает недостаточно. Например, требуется ограничить документ **Перемещение товаров** так, чтобы чтение документа было доступно пользователю исходя из разрешенности **Организации** и **СкладаПолучателя** или **СкладаОтправителя**, а запись документа была доступна только при разрешенности всех трех: **Организации**, **СкладаПолучателя** и **СкладаОтправителя**. В этом случае право **Запись** можно ограничить, применив шаблон **#ПоЗначениям**, а право **Чтение** – нет.

Для ограничений, предполагающих логику с **ИЛИ**, предусмотрены расширенные шаблоны, в частности **#ПоЗначениямРасширенный**. Например, ограничение для права **Чтение** выглядит следующим образом:

#ПоЗначениямРасширенный("Документ.ПеремещениеТоваров", "", "",

"",

"",

"Организации", "Т.Организация", "И(",

"Склады", "Т.СкладОтправитель", "ИЛИ",

"Склады", "Т.СкладПолучатель", ")", …)

Требуется явно указывать псевдоним текущей таблицы «Т.» перед полем.

Кроме реализации логики **ИЛИ** расширенные шаблоны могут использоваться для проверки реквизитов табличных частей – через присоединяемые таблицы. Для того чтобы к основной таблице присоединить табличную часть, в четвертый параметр шаблона нужно передать текст на языке запросов, например:

"Внутреннее Соединение Документ.ПеремещениеТоваров.Номенклатура

КАК Т2 по Т.Ссылка = Т2.Ссылка",

Для шаблона основная таблица расширяется полями табличной части, а за счет явного применения псевдонимов их можно указывать, подобно полям основной таблицы. Т. е. в шаблон можно добавить проверку реквизита табличной части, например:

"ГруппыНоменклатуры", "Т2.Номенклатура", ""

В результате получится такое ограничение:

#ПоЗначениямРасширенный("Документ.ПеремещениеТоваров", "", "",

"Внутреннее Соединение Документ.ПеремещениеТоваров.Номенклатура КАК Т2 по Т.Ссылка = Т2.Ссылка",

"",

"Организации", "Т.Организация", "И(",

"Склады", "Т.СкладОтправитель", "ИЛИ",

"Склады", "Т.СкладПолучатель", ")И",

"ГруппыНоменклатуры", "Т2.Номенклатура", "", …)

Когда основная таблица расширяется полями присоединенной таблицы, происходит умножение строк. Например, табличная часть содержит 2 строки c разной номенклатурой. В этом случае документ будет доступен, если выполнились условия проверки основной таблицы («шапки» документа) и доступна хотя бы одна из двух номенклатур табличной части.

#### Шаблон «#ПоЗначениямИНаборамРасширенный»

Для решения задачи ограничения доступа к объектам метаданных в зависимости от ограничений к другим объекта может использоваться шаблон **#ПоЗначениямИНаборамРасширенный**.

Например, файл (элемент справочника **Файлы**), прикрепленный к документу **Поступление товаров и услуг**, должен быть доступен пользователю при тех же условиях, что и сам документ.

Такая задача могла бы быть решена «повтором» логики ограничения документа **Поступление товаров и услуг** в ограничении справочника **Файлы**. Однако на практике это привело бы к необходимости синхронно изменять логику ограничения документов и файлов. Кроме того, поскольку элементы справочника **Файлы** могут «прикрепляться» к произвольному числу объектов метаданных (например, к папкам файлов, документам **Заказ**, **Расходная накладная** и т. п.), то задача реализации логики ограничения доступа к файлам сравнима по сложности с ограничением доступа к журналу документов, который содержит все виды документов конфигурации.

Поэтому для решения такой задачи подсистема «Управление доступом» предоставляет специальный вид доступа **по объекту доступа**. При использовании этого вида доступа в шаблоне следует указывать в качестве имени вида доступа **Объект**. Вид доступа **Объект** предполагает, что в поле находится значение, которое является ссылкой на объект данных (например, документ). А в регистре **Наборы значений доступа**, как и для шаблона **#ПоНаборамЗначений**, записаны наборы значений для этого объекта, которые следует проверить. Исключение составляют владельцы настроек прав – для них наборы значений доступа записывать не требуется, но процедура заполнения может понадобиться, если они используются для заполнения наборов значений доступа зависимых объектов.

Так, используя шаблон **#ПоЗначениямИНаборамРасширенный**, можно проверить не одно значение доступа, а набор значений, подготовленный при записи документа, ссылка на который находится в реквизите, например, **Владелец** элемента справочника **Файлы**.

#ПоЗначениямИНаборамРасширенный("Справочник.Файлы", "", "",

"",

"",

"Объект", "Т.Владелец" , "", …)

Наборы значений элементов справочника **Файлы** требуют связанного обновления при изменении, например, наборов значений доступа документа – владельца файлов **Поступление товаров и услуг**.

При использовании вида доступа **Объект** кроме проверки наборов выполняется еще проверка прав на объект. В ограничении права чтения таблицы выполняется проверка права чтения объекта и наборов ограничения чтения объекта, а в ограничении прав добавления, изменения, удаления таблицы выполняется проверка права изменения объекта и наборов ограничения изменения объекта.

Права объекта, которые проверяются в ограничении по виду доступа **Объект**, можно переопределить в процедуре **ПриЗаполненииЗависимостейПравДоступа** модуля **УправлениеДоступомПереопределяемый**.

Структура регистра **Наборы значений доступа** поддерживает логику **ИЛИ** для значений доступа и различные наборы по правам доступа. Для реализации логики **ИЛИ** добавлено измерение **Номер набора** типа **Число**, а для наборов по правам добавлены два ресурса: **Чтение**, **Изменение** – типа **Булево**.

Сложную логику **ИЛИ** записать в регистр «напрямую» не получится. Номер набора позволяет сделать для одного ссылочного элемента данных (например, документа) несколько наборов значений, по которым проверяется доступность документа. Причем достаточно, чтобы значения хотя бы одного набора значений были доступны одновременно. Точнее, результаты проверки значений доступа по видам доступа в пределах набора с одним номером объединяются по **И**, а результаты проверки целых наборов объединяются по **ИЛИ**. Например, для документа **Перемещение товаров** регистр **Наборы значений доступа** будет содержать записи для логики права **Чтение**:

<Документ2>, 0, "Организации", "Организация1"

<Документ2>, 0, "Склады", "Склад1";

<Документ2>, 1, "Организации", "Организация1"

<Документ2>, 1, "Склады", "Склад2";

Видно, что, хотя значений доступа три: **Организация1**, **Склад1**, **Склад2**, – записей в регистре четыре. Это связано с раскрытием скобок в логическом выражении **О1** и (**С1** или **С2**), что приводит к логическому выражению суммы произведений **О1** и **С1** или **О1** и **С2**.

Действие раскрытия скобок первичного логического выражения, объединяющего результаты проверки значений доступа, необходимо для заполнения наборов записей регистра сведений **Наборы значений доступа**.

Записи для логики права **Изменение**:

<Документ2>, 0, "Организации", "Организация1"

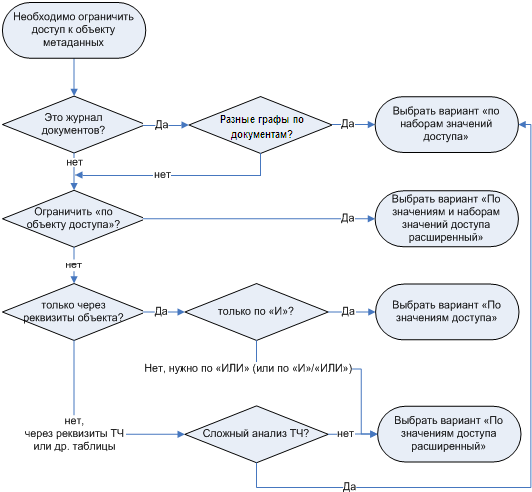
<Документ2>, 0, "Склады", "Склад1";

<Документ2>, 0, "Склады", "Склад2";

Пример процедуры заполнения приведен ниже.

Шаблон **#ПоЗначениямИНаборамРасширенный** отличается от шаблона **#ПоЗначениямРасширенный** только поддержкой специального вида доступа **Объект**.

Для быстрого выбора шаблона можно использовать схему-алгоритм.



### Разработка процедур «Заполнить наборы значений доступа»

При использовании стандартных шаблонов **#ПоНаборамЗначений** или **#ПоЗначениямИНаборамЗначений** с видом доступа **Объект** потребуется записать в регистр **Наборы значений доступа** сведения, требуемые для их работы.

Запись наборов значений доступа выполняется при записи объектов – например, документов. Перед записью вызывается пользовательская процедура заполнения наборов значений доступа.

Процедура заполнения помещается в модуль объекта, и его тип включается в подписку на события **Записать наборы значений доступа** и в состав типов измерения **Объект** регистра сведений **Наборы значений доступа**.

При заполнении можно использовать только следующие свойства таблицы:

* ● **Номер набора** (если набор один, то заполнять не нужно);
* ● **Вид доступа** (только для специальных видов доступа **ПравоЧтения**, **ПравоИзменения**);
* ● **Значение доступа** (назначить явно);
* ● **Чтение**, **Изменение** (если набор для всех прав, то заполнять не нужно).

Количество заполненных записей должно быть больше нуля, иначе будет вызвано исключение. Это важно, т. к. процедура заполнения наборов значений доступа после включения режима **Ограничивать доступ на уровне записей** вызывается из регламентного задания **Заполнение данных для ограничения доступа** для тех объектов, которые имеют пустой набор записей в регистре **Наборы значений доступа**. При выключении режима **Ограничивать доступ на уровне записей** производится запись пустого набора. Следовательно, при включении режима выполнится заполнение очищенных наборов (или полное заполнение). Регламентное задание вызывается многократно и отрабатывает порцию заполнения. Очищение наборов происходит «по ходу» перезаписи объектов.

Технология порционного заполнения/очистки использует пустой набор записей как флажок состояния. Когда нужно записать заведомо «запрещенный набор», записывается не пустой набор, а следующий:

Строка = Таблица.Добавить();

Строка.ЗначениеДоступа = Перечисления.ДополнительныеЗначенияДоступа.ДоступЗапрещен;

Когда нужно записать заведомо «разрешенный набор», записывается не пустой набор, а следующий:

Строка = Таблица.Добавить();

Строка.ЗначениеДоступа = Перечисления.ДополнительныеЗначенияДоступа.ДоступРазрешен;

Пример процедуры:

* ● **ЗаполнитьНаборыЗначенийДоступа** в модуле объекта (тип объекта включается в подписку на событие **ЗаписатьНаборыЗначенийДоступа** (через определяемый тип **ВладелецНаборовЗначенийДоступаОбъект**) или в дополнительно созданную подписку на события, имя которой начинается, как имя поставляемой подписки, а обработчик совпадает).

Процедура ЗаполнитьНаборыЗначенийДоступа(Таблица) Экспорт

Строка = Таблица.Добавить();

Строка.Изменение       = Истина;

Строка.ЗначениеДоступа = Организация1;

Строка = Таблица.Добавить();

Строка.ЗначениеДоступа = Склад1;

Строка = Таблица.Добавить();

Строка.ЗначениеДоступа = Склад2;

Строка = Таблица.Добавить();

Строка.НомерНабора     = 1;

Строка.Чтение          = Истина;

Строка.ЗначениеДоступа = Организация1;

Строка = Таблица.Добавить();

Строка.НомерНабора     = 1;

Строка.ЗначениеДоступа = Склад1;

Строка = Таблица.Добавить();

Строка.НомерНабора     = 2;

Строка.Чтение          = Истина;

Строка.ЗначениеДоступа = Организация1;

Строка = Таблица.Добавить();

Строка.НомерНабора     = 2;

Строка.ЗначениеДоступа = Склад2;

КонецПроцедуры

Если в процедуре заполнения не задан номер набора, то считается, что добавлен один набор с номером 1. Если в пределах набора с одним номером не указано ни одно из прав **Чтение** или **Изменение**, то считается, что набор относится ко всем правам.

В наборе допускается только одна запись одним и тем же значением доступа, дубли строк и наборов удаляются автоматически в процедуре **ДобавитьНаборыЗначенийДоступа**, которая вызывается для этих целей перед записью наборов значений доступа в регистр сведений **НаборыЗначенийДоступа** со сжатием наборов (лишние наборы удаляются по правилам упрощения логических функций).

Чтобы изменения вступили в силу, требуется обновить вспомогательные данные. См. раздел «[Обновление вспомогательных данных во время разработки](https://its.1c.ru/db/content/bsp23doc/src/3.4.%20%D0%B1%D0%B0%D0%B7%D0%BE%D0%B2%D0%B0%D1%8F%20%D1%84%D1%83%D0%BD%D0%BA%D1%86%D0%B8%D0%BE%D0%BD%D0%B0%D0%BB%D1%8C%D0%BD%D0%BE%D1%81%D1%82%D1%8C.htm?_=1504708101#_обновление_вспомогательных_данных)».

#### Зависимые наборы значений доступа

В случае, когда наборы значений доступа нужно сформировать с учетом доступа к другому объекту (к примеру, доступ к бизнес-процессу **\_ДемоЗаданиеСРолевойАдресацией** определяется по доступу к его объекту-основанию, например по элементу справочника **Файлы**), появляется необходимость включать в состав одних наборов другие наборы.

Для реализации включения одних наборов в другие разработана процедура **ДобавитьНаборыЗначенийДоступа**общего модуля **УправлениеДоступом**, которая добавляет в таблицу-приемник **Приемник** наборы из таблицы-источника **Источник** либо логическим сложением, либо логическим умножением. Например, наборы задачи:

Процедура ЗаполнитьНаборыЗначенийДоступа(Таблица) Экспорт

Строка = Таблица.Добавить();

Строка.ЗначениеДоступа = Исполнитель;

// Создание пустой таблицы.

ТаблицаБизнесПроцесса = УправлениеДоступом.ТаблицаНаборыЗначенийДоступа();

// Заполнение наборов бизнес-процесса.

УправлениеДоступом.ЗаполнитьНаборыЗначенийДоступа(БизнесПроцесс, ТаблицаБизнесПроцесса);

// Логическое сложение наборов в таблицах (по «ИЛИ»).

УправлениеДоступом.ДобавитьНаборыЗначенийДоступа(Таблица, ТаблицаБизнесПроцесса);

КонецПроцедуры

Следует разработать процедуры **ЗаполнитьНаборыЗначенийДоступа** для всех объектов, которые могут быть объектом-основанием или задействованы при формировании наборов значений доступа.

Могут быть объекты, у которых наборы значений доступа не записываются в регистр сведений **НаборыЗначенийДоступа**, а используются только при формировании других записываемых наборов значений доступа. Типы таких объектов нужно включить в подписку на события **ЗаписатьЗависимыеНаборыЗначенийДоступа** (через определяемый тип **ВладелецВнешнихЗначенийВНаборахЗначенийДоступаОбъект**) или в дополнительно созданную подписку на события, имя которой начинается, как имя поставляемой подписки, а обработчик совпадает.

Для реализации зависимостей необходимо разработать логику вызова перезаписи наборов значений доступа зависимых объектов в обработчике **ПриИзмененииНаборовЗначенийДоступа** общего модуля **УправлениеДоступомПереопределяемый**, например:

Процедура ПриИзмененииНаборовЗначенийДоступа(Знач Ссылка, СсылкиНаЗависимыеОбъекты) Экспорт

ПолноеИмя = Ссылка.Метаданные().ПолноеИмя();

Если ПолноеИмя = "Задача.ЗадачаИсполнителя" Тогда

// Зависимые типы объектов:

//  БизнесПроцесс.Задание

Запрос = Новый Запрос(

"ВЫБРАТЬ

| Задание.Ссылка

|ИЗ

| БизнесПроцесс.Задание КАК Задание

|ГДЕ

| Задание.Предмет = &Предмет");

Запрос.УстановитьПараметр("Предмет", Ссылка);

СсылкиНаЗависимыеОбъекты = Запрос.Выполнить().Выгрузить().ВыгрузитьКолонку("Ссылка");

ИначеЕсли ПолноеИмя = "БизнесПроцесс.Задание" Тогда

// Зависимые типы объектов:

//  БизнесПроцесс.Задание

// через Задача.ЗадачаИсполнителя в основании.

Запрос = Новый Запрос(

"ВЫБРАТЬ РАЗЛИЧНЫЕ

| Задание.Ссылка

|ИЗ

| БизнесПроцесс.Задание КАК Задание

| ВНУТРЕННЕЕ СОЕДИНЕНИЕ Задача.ЗадачаИсполнителя КАК ЗадачаИсполнителя

| ПО Задание.Предмет = ЗадачаИсполнителя.Ссылка

| И (ЗадачаИсполнителя.БизнесПроцесс = &БизнесПроцесс)");

Запрос.УстановитьПараметр("БизнесПроцесс", Ссылка);

СсылкиНаЗависимыеОбъекты = Запрос.Выполнить().Выгрузить().ВыгрузитьКолонку("Ссылка");

КонецЕсли;

КонецПроцедуры

Специальные имена видов доступа **ПравоЧтения**, **ПравоИзменения** требуются, когда нужно разработать ограничение с зависимостью по правам «подчиненного» объекта от «ведущего» объекта. В шаблонах ОДД в проверку по виду доступа **Объект** встроена логика проверки зависимых прав, как и в этих видах доступа. В связи с этим сочетать данные виды доступа с видом доступа **Объект** не требуется. Также не требуется использовать эти виды доступа при заполнении зависимых наборов значений доступа. Для этих целей предусмотрен третий параметр в процедуре **ЗаполнитьНаборыЗначенийДоступа**, при указании которого соответствующие проверки добавляются автоматически.

Чтобы изменения вступили в силу, требуется обновить вспомогательные данные. См. раздел «[Обновление вспомогательных данных во время разработки](https://its.1c.ru/db/content/bsp23doc/src/3.4.%20%D0%B1%D0%B0%D0%B7%D0%BE%D0%B2%D0%B0%D1%8F%20%D1%84%D1%83%D0%BD%D0%BA%D1%86%D0%B8%D0%BE%D0%BD%D0%B0%D0%BB%D1%8C%D0%BD%D0%BE%D1%81%D1%82%D1%8C.htm?_=1504708101#_обновление_вспомогательных_данных)».

### Настройка свойств объектов метаданных

Таблица 3.111.

|  |  |
| --- | --- |
| **Свойство объекта метаданных** | **Настройка** |
| **ОпределяемыйТип.ЗначениеДоступа** | ● Обязательные типы:  ● **СправочникСсылка.Пользователи**,  ● **СправочникСсылка.ГруппыПользователей**,  ● **СправочникСсылка.ВнешниеПользователи**,  ● **СправочникСсылка.ГруппыВнешнихПользователей**.  ● Типы значений доступа, например **СправочникСсылка.Организации**.  ● Типы групп значений доступа, например **СправочникСсылка.\_ДемоГруппыДоступаКонтрагентов** |
| **ОпределяемыйТип.ВладелецНастроекПрав** | ● Типы владельцев настроек прав, например **СправочникСсылка.ПапкиФайлов** |
| **ОпределяемыйТип.ВладелецНастроекПравОбъект** | ● Типы владельцев настроек прав, например **СправочникОбъект.ПапкиФайлов** |
| **ОпределяемыйТип.ЗначениеДоступаСГруппамиЗначенийДоступаОбъект** | ● Типы значений доступа, для которых указан тип групп значений (например, группы доступа контрагентов **СправочникОбъект.\_ДемоКонтрагенты**) |
| **ОпределяемыйТип.ВладелецНаборовЗначенийДоступаОбъект**(совместно с типами, указанными в создаваемых подписках на события **ЗаписатьНаборыЗначенийДоступа<Окончание>**) | ● Типы объектов, для которых записываются наборы значений доступа используемые, как значения в проверках по виду **Объект** через шаблон **#ПоЗначениямИНаборамРасширенный**.  ● Типы объектов с ограничением, настроенным с помощью шаблона **#ПоНаборамЗначений** |
| **ОпределяемыйТип.ВладелецВнешнихЗначенийВНаборахЗначенийДоступаОбъект**(совместно с типами, указанными в создаваемых подписках на события **ЗаписатьЗависимыеНаборыЗначенийДоступа<Окончание>**) | Типы объектов, которые используются в процедурах **ЗаполнитьНаборыЗначенийДоступа**для формирования наборов, но не входят в состав типов владельцев наборов значений доступа, т. е. отсутствуют в типе **ОпределяемыйТип.ВладелецНаборовЗначенийДоступаОбъект**(а также среди типов, указанных в соответствующих создаваемых подписках на события). Например, **СправочникОбъект.Файлы**, используемый для формирования наборов **БизнесПроцессОбъект.\_ДемоЗаданиеСРолевойАдресацией** (см. также обработчик **УправлениеДоступомПереопределяемый.ПриИзмененииНаборовЗначенийДоступа(…)**) |
| **ОпределяемыйТип.ВладелецСОграничениемПоНаборамЗначенийДоступаОбъект**(совместно с типами, указанными в создаваемых подписках на события **ЗаполнитьНаборыЗначенийДоступаТабличныхЧастей<Окончание>**) | Используется для тех объектов (кроме документов), которые используют наборы значений доступа в шаблоне **#ПоНаборамЗначений** для ограничения объекта. Для документов необходимо создавать отдельную подписку |
| **ТабличнаяЧасть.НаборыЗначенийДоступа.(НомерНабора, ЗначениеДоступа, Уточнение, Чтение, Изменение)** | Табличная часть создается в объектах метаданных прикладной конфигурации, которые используют наборы значений доступа в шаблоне **#ПоНаборамЗначений** для ограничения доступа к ним.  Состав полей табличной части одинаков для всех таких объектов. Состав типов, как у тех же полей в регистре сведений **НаборыЗначенийДоступа** |

### Описание видов доступа, используемых в ограничениях объектов для отчета «Права доступа»

Для использования отчета **Права доступа** требуется задать список видов доступа, используемых в ограничении прав объектов метаданных. Список задается в переопределяемом модуле в виде многострочной строки формата **<Полное имя таблиц>.<Право>.<Вид доступа>**. Для автоматического формирования кода процедуры следует применить инструменты разработчика. Пример заполнения процедуры:

Процедура ПриЗаполненииВидовОграниченийПравОбъектовМетаданных(Описание) Экспорт

Описание =

"

|БизнесПроцесс.Задание.Добавление.Пользователи

|БизнесПроцесс.Задание.Изменение.Пользователи

|БизнесПроцесс.Задание.Чтение.Пользователи

|…

|…

|…

|Справочник.ПапкиФайлов.Изменение.\_ДемоПапкиФайлов

|Справочник.ПапкиФайлов.Чтение.\_ДемоПапкиФайлов

|Справочник.УчетныеЗаписиЭлектроннойПочты.Чтение.УчетныеЗаписиЭлектроннойПочты

|";

КонецПроцедуры

### Создание описаний поставляемых профилей групп доступа для начального заполнения и обновления информационной базы

Для создания описаний профилей групп доступа, поставляемых в конфигурации, используемых при начальном заполнении и обновлении информационной базы, а также при интерактивном восстановлении поставляемых профилей, измененных администратором системы, используется процедура **ЗаполнитьПоставляемыеПрофилиГруппДоступа** общего модуля **УправлениеДоступомПереопределяемый**.

Для создания кода описания на встроенном языке рекомендуется применять обработку в демонстрационной конфигурации **ИнструментыРазработчикаУправлениеДоступом**.

Процедура ЗаполнитьПоставляемыеПрофилиГруппДоступа(ОписанияПрофилей, ПараметрыОбновления) Экспорт

// Профиль "Пользователь".

ЗаполнитьПрофильПользователь(ОписанияПрофилей);

// Профиль "Бухгалтер".

ЗаполнитьПрофильБухгалтер(ОписанияПрофилей);

...

// Дополнительные профили, которые не используются самостоятельно при настройке

// прав пользователя, а дополняют основные профили, перечисленные выше.

// Профиль "Ответственный за составы участников групп доступа (дополнительно)".

ЗаполнитьПрофильОтветственныйЗаСоставыУчастниковГруппДоступа(ОписанияПрофилей);

// Профиль "Ответственный за ведение номенклатуры (дополнительно)".

ЗаполнитьПрофильОтветственныйЗаВедениеНоменклатуры(ОписанияПрофилей);

...

КонецПроцедуры

Процедура ЗаполнитьПрофильПользователь(Знач ОписанияПрофилей)

ОписаниеПрофиля = УправлениеДоступом.НовоеОписаниеПрофиляГруппДоступа();

ОписаниеПрофиля.Имя           = "\_ДемоПользователь";

ОписаниеПрофиля.Идентификатор = "09e56dbf-90a0-11de-862c-001d600d9ad2";

ОписаниеПрофиля.Наименование  = НСтр("ru = 'Пользователь'");

ОписаниеПрофиля.Описание =

НСтр("ru = 'Общие разрешенные действия для большинства пользователей.

|Как правило, это права на просмотр данных информационной системы.'");

// Использование 1С:Предприятия.

ОписаниеПрофиля.Роли.Добавить("ЗапускВебКлиента");

ОписаниеПрофиля.Роли.Добавить("ЗапускТолстогоКлиента");

ОписаниеПрофиля.Роли.Добавить("ЗапускТонкогоКлиента");

ОписаниеПрофиля.Роли.Добавить("ВыводНаПринтерФайлБуферОбмена");

ОписаниеПрофиля.Роли.Добавить("СохранениеДанныхПользователя");

// Использование программы.

ОписаниеПрофиля.Роли.Добавить("БазовыеПрава");

ОписаниеПрофиля.Роли.Добавить("ПросмотрОписанияИзмененийПрограммы");

...

ОписанияПрофилей.Добавить(ОписаниеПрофиля);

КонецПроцедуры

...

Для того чтобы изменения вступили в силу, требуется обновить вспомогательные данные. См. раздел «[Обновление вспомогательных данных во время разработки](https://its.1c.ru/db/content/bsp23doc/src/3.4.%20%D0%B1%D0%B0%D0%B7%D0%BE%D0%B2%D0%B0%D1%8F%20%D1%84%D1%83%D0%BD%D0%BA%D1%86%D0%B8%D0%BE%D0%BD%D0%B0%D0%BB%D1%8C%D0%BD%D0%BE%D1%81%D1%82%D1%8C.htm?_=1504708101#_обновление_вспомогательных_данных)».

### Размещение в командном интерфейсе

Для использования подсистемы (кроме случая упрощенного интерфейса) необходимо в командном интерфейсе разместить объекты метаданных:

* ● **Справочник.ГруппыДоступа**:
* ● для частой работы администратора по просмотру и редактированию групп доступа (например, в подразделе **Важное** рядом со справочниками **Пользователи** и **ВнешниеПользователи**);
* ● для частой работы администратора группы доступа по просмотру и редактированию участников групп доступа (например, в подразделе **Важное** рядом со справочником **Пользователи**).
* ● **Справочник.ПрофилиГруппДоступа** – для редкой работы администратора по просмотру списка и содержания профилей групп доступа (например, в подразделе **См. также**).

В случае если в конфигурации не используется подсистема «Настройки программы», в рабочем месте администратора программы необходимо разместить константы:

* ● **ОграничиватьДоступНаУровнеЗаписей** – для включения/выключения ограничения доступа на уровне записей (если в ограничениях прав используются наборы значений доступа, то при включении константы необходимо выдавать предупреждение о необходимости заполнения данных для ограничения доступа).

См. пример в форме **НастройкиПользователейИПрав** обработки **ПанельАдминистрированияБСП**.

### Настройка прав доступа пользователей

Для настройки прав доступа пользователей к данным подсистемы «Управление доступом» следует использовать следующие роли:

Таблица 3.112.

| **№** | **Роли и их назначение** |
| --- | --- |
| 1. | **БазовыеПрава** (из подсистемы «Базовая функциональность»)  Просмотр списка пользователей.  Просмотр списка своих групп доступа в форме **Права доступа**.  Формирование и просмотр отчета о своих правах доступа |
| 2. | **ИзменениеУчастниковГруппДоступа**  Изменение состава участников и просмотр тех групп доступа, у которых пользователь установлен как администратор группы доступа |
| 3. | **ПолныеПрава** (из подсистемы «Базовая функциональность»)  Просмотр и редактирование групп доступа (в том числе назначение администраторов групп доступа).  Удаление объектов подсистемы, помеченных на удаление.  Ручной запуск регламентного задания **Заполнение данных для ограничения доступа**.  Включение и отключение ограничений прав доступа на уровне записей.  Разработка профилей групп доступа |

Пример настройки прав доступа пользователей:

| **№** | **Группа пользователей и ее функции** | **Состав ролей** |
| --- | --- | --- |
| 1. | Ответственный за составы участников групп доступа (дополнительно):  ● Нужен для работы пользователя, которого назначили ответственным за состав участников в группе доступа (например начальник подразделения, руководитель проектной группы и др.) | **ИзменениеУчастниковГруппДоступа** |

### Особые случаи внедрения подсистемы

* ● [Внедрение подсистем «Бизнес-процессы и задачи» и «Управление доступом»](https://its.1c.ru/db/content/bsp23doc/src/3.59.%20%D0%BE%D1%81%D0%BE%D0%B1%D1%8B%D0%B5%20%D1%81%D0%BB%D1%83%D1%87%D0%B0%D0%B8%20%D0%B2%D0%BD%D0%B5%D0%B4%D1%80%D0%B5%D0%BD%D0%B8%D1%8F%20%D0%BF%D0%BE%D0%B4%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC.htm?_=1504708101#_внедрение_подсистем_).
* ● [Внедрение подсистем «Взаимодействия» и «Управление доступом»](https://its.1c.ru/db/content/bsp23doc/src/3.59.%20%D0%BE%D1%81%D0%BE%D0%B1%D1%8B%D0%B5%20%D1%81%D0%BB%D1%83%D1%87%D0%B0%D0%B8%20%D0%B2%D0%BD%D0%B5%D0%B4%D1%80%D0%B5%D0%BD%D0%B8%D1%8F%20%D0%BF%D0%BE%D0%B4%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC.htm?_=1504708101#_внедрение_подсистем_).
* ● [Внедрение подсистем «Работа с почтовыми сообщениями» и «Управление доступом»](https://its.1c.ru/db/content/bsp23doc/src/3.59.%20%D0%BE%D1%81%D0%BE%D0%B1%D1%8B%D0%B5%20%D1%81%D0%BB%D1%83%D1%87%D0%B0%D0%B8%20%D0%B2%D0%BD%D0%B5%D0%B4%D1%80%D0%B5%D0%BD%D0%B8%D1%8F%20%D0%BF%D0%BE%D0%B4%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC.htm?_=1504708101#_внедрение_подсистем_).
* ● [Внедрение подсистем «Работа с файлами» и «Управление доступом»](https://its.1c.ru/db/content/bsp23doc/src/3.59.%20%D0%BE%D1%81%D0%BE%D0%B1%D1%8B%D0%B5%20%D1%81%D0%BB%D1%83%D1%87%D0%B0%D0%B8%20%D0%B2%D0%BD%D0%B5%D0%B4%D1%80%D0%B5%D0%BD%D0%B8%D1%8F%20%D0%BF%D0%BE%D0%B4%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC.htm?_=1504708101#_внедрение_подсистем_).
* ● [Внедрение подсистем «Свойства» и «Управление доступом»](https://its.1c.ru/db/content/bsp23doc/src/3.59.%20%D0%BE%D1%81%D0%BE%D0%B1%D1%8B%D0%B5%20%D1%81%D0%BB%D1%83%D1%87%D0%B0%D0%B8%20%D0%B2%D0%BD%D0%B5%D0%B4%D1%80%D0%B5%D0%BD%D0%B8%D1%8F%20%D0%BF%D0%BE%D0%B4%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC.htm?_=1504708101#_внедрение_подсистем_).

## Использование при разработке конфигурации

При создании новых прикладных объектов метаданных нужно повторно выполнять соответствующую часть процедуры настройки, описанной выше.

Программный интерфейс подсистемы доступен через экспортные функции общего модуля **УправлениеДоступом**.

### Проверка работы логики ограничений прав

Ошибка доступа к данным связана либо с недостаточным, либо избыточным доступом к данным информационной базы. В общем виде подобные нештатные ситуации могут возникать по причинам, перечисленным ниже.

* ● Возможная ошибка при настройке прав доступа:
* ● некорректная настройка состава пользователей или групп пользователей группы доступа – необходимо включить пользователя или группу пользователей в группы доступа или исключить его из них;
* ● некорректный профиль группы доступа – выбрать другой профиль групп доступа, создать дополнительный профиль или скорректировать имеющийся профиль;
* ● некорректная настройка ограничений доступа в группе доступа (списков разрешенных значений, например: организаций, групп партнеров, складов) – необходимо добавить или исключить значений доступа из группы доступа.
* ● Возможная ошибка при разработке или доработке конфигурации на внедрении (в компетенции разработчика конфигурации):
* ● некорректная настройка состава ролей в профиле;
* ● некорректная настройка состава видов доступа в профиле.
* ● Ошибки разработчика конфигурации:
* ● ошибка в параметрах стандартного шаблона;
* ● ошибка в составе типов подписок на события;
* ● ошибка в процедуре заполнения наборов значений доступа;
* ● ошибка в описании видов доступа.
* ● Ошибки разработчика этой подсистемы:
* ● ошибка в стандартном шаблоне;
* ● ошибка особенностей работы СУБД;
* ● ошибка в общих модулях подсистемы.

Для диагностики ошибок доступа к данным можно воспользоваться следующими средствами:

* ● Убедиться, что содержимое регистра сведений **Наборы значений доступа** содержит наборы значений доступа, которые записаны таким образом, что наборы с одинаковыми номерами для одного объекта соответствуют логике ограничения доступа по условию **И**, а затем наборы с разными номерами «складываются» по условию **ИЛИ**:
* ● если по интересующему нас объекту в регистре сведений нет записей, то объект не будет доступен для всех пользователей;
* ● если по интересующему нас объекту нет ни одного набора, относящегося к конкретному праву доступа, значит, для всех пользователей нет этого права на объект.
* ● Нужно иметь в виду, что если в профиль не добавлена роль для чтения каких-либо данных, но предусмотрены роли для работы с другими связанными объектами, то возможна ситуация, когда в полях этих объектов будут отображаться значения вида **Ссылка не найдена (… : …)** вместо самих данных.

### Разработка обработчиков обновления ИБ

В процессе разработки конфигурации в нее могут быть внесены изменения, которые потребуют обновления данных для ограничения доступа. Ниже перечислены типовые ситуации, когда обновление необходимо.

1. При каждом обновлении информационной базы выполняются обработчики обновления в модуле **УправлениеДоступомСлужебный**:

* ● Для обновления неразделенных вспомогательных данных (кэшей свойств встраивания), которые сохраняются в константе **ПараметрыОграниченияДоступа** вызывается процедура **ОбновитьПараметрыОграниченияДоступа**. Процедура обновляет общие параметры, права ролей, зависимости прав доступа, описания поставляемых профилей, состав предопределенных профилей, описание возможных прав для настройки прав объектов, например:
* ● при добавлении объектов метаданных (доступ к которым может ограничиваться) и добавлении/изменении ролей обновляется регистр сведений **ПраваРолей** и регистрируется состав объектов метаданных, для которых внесены изменения;
* ● при изменении описаний поставляемых профилей групп доступа регистрируются их изменения;
* ● при изменении зависимостей прав доступа между объектами метаданных регистрируются их изменения.
* ● Для обновления разделенных вспомогательных данных (разных для каждой области данных) вызывается процедура **ОбновитьВспомогательныеДанныеПоИзменениямКонфигурации**.

Соответственно, в процессе разработки необходимо учитывать, что для «вступления в силу» перечисленных изменений необходим запуск обработчиков (перед проверкой текущей разработки).

Обычно для этого требуется изменить версию конфигурации на следующую. Для повторных запусков обработчиков следует изменять версию в регистре сведений **ВерсииПодсистем** на предыдущую.

Для разработки рекомендуется использовать обработку **ИнструментыРазработчикаОбновлениеВспомогательныхДанных.epf**, которая входит в состав дистрибутива и в демонстрационную конфигурацию.

При закладке в хранилище результатов разработки, требующих обновления вспомогательных данных, рекомендуется поместить в хранилище новую версию конфигурации, чтобы обновление информационной базы выполнилось у всех участков разработки.

2. **Обновление поставляемых профилей групп доступа**. Если в конфигурации было изменено описание поставляемых профилей групп доступа в процедуре **ПриЗаполненииПоставляемыхПрофилейГруппДоступа**общего модуля **УправлениеДоступомПереопределяемый**, то никаких действий не требуется. При обновлении информационной базы обновление поставляемых профилей будет выполнено автоматически. Для управления обновлением предусмотрены свойства в параметре процедуры **ПараметрыОбновления**. Их значения можно переопределить (подробнее см. комментарий к процедуре).

3. **Начальное заполнение наборов значений доступа** необходимо выполнить при включении ограничений на уровне записей, если в ролях конфигурации имеются ограничения прав, которые используют наборы значений доступа.

При этом, если среди видов доступа применяются группы значений доступа, также требуется выполнить заполнение групп значений доступа.

Для выполнения начального заполнения следует воспользоваться одним из двух способов:

* ● Разрешить выполнение регламентного задания **ЗаполнениеДанныхДляОграниченияДоступа**; этот способ позволяет выполнить начальное заполнение наборов значений доступа порциями в фоне, не мешая работе администратора. Однако для того, чтобы изменения вступили в силу и с базой смогли начать работу другие пользователи, необходимо дождаться завершения отработки регламентным заданием всех порций.
* ● Вызвать процедуру **ЗаполнениеДанныхДляОграниченияДоступа** общего модуля **УправлениеДоступом** с параметром **КоличествоДанных** = 0\*.

4. **Обновление наборов значений доступа** необходимо выполнять в случаях:

* ● Если в конфигурации были добавлены виды доступа, использующие группы значений доступа, или изменены существующие виды доступа так, что они стали использовать группы значений доступа, то необходимо выполнить обновление наборов значений доступа. Для этого следует воспользоваться методикой, описанной в пункте 3.
* ● Если в конфигурации у определенных объектов метаданных был изменен алгоритм формирования наборов значений доступа или были изменены зависимости по правам. Для решения этой задачи следует выбрать один из двух вариантов:
* ● перезаписать наборы значений доступа с помощью процедуры **ОбновитьНаборыЗначенийДоступа**общего модуля **УправлениеДоступом**;
* ● очистить наборы значений доступа для соответствующих объектов и заполнить их способом, который описан в пункте 3.

\* Операция обновления наборов значений доступа соизмерима с перезаписью всех документов, доступ к которым ограничивается на уровне записей, поэтому процесс обновления на большой базе может занять достаточно много времени. Поэтому лучше использовать предусмотренное регламентное задание. Для этого достаточно включить его использование (отключение произойдет автоматически).

### Настройка обмена данными

Для настройки обмена данными следует руководствоваться общими правилами.

В планы обмена распределенной информационной базы (РИБ) рекомендуется включать все объекты метаданных подсистемы.

В планах обмена распределенной информационной базы (РИБ) рекомендуется отключать регистрацию изменений для следующих объектов метаданных подсистемы (см. также раздел «[Особенности создания начального образа подчиненного узла распределенной ИБ](https://its.1c.ru/db/content/bsp23doc/src/3.28.%20%D0%BE%D0%B1%D0%BC%D0%B5%D0%BD%20%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D0%BC%D0%B8.htm?_=1504708101#_особенности_создания_начального)»):

* ● константа **ПараметрыОграниченияДоступа**,
* ● регистр сведений **ПраваРолей**,
* ● регистр сведений **ЗависимостиПравДоступа**.

Из остальных планов обмена (не РИБ) рекомендуется исключать следующие объекты метаданных:

* ● константа **ПараметрыОграниченияДоступа**,
* ● регистр сведений **ПраваРолей**,
* ● регистр сведений **ЗависимостиПравДоступа**.

# 3.54. Управление итогами и агрегатами

Подсистема «Управление итогами и агрегатами» предоставляет средства для администрирования итогов и агрегатов регистров информационной базы. Подсистема позволяет как выполнять типовые административные операции, так и дает доступ к следующим возможностям:

● включение/отключение использования итогов и агрегатов,

● разделение итогов,

● установка периода и пересчет итогов,

● перестроение и обновление агрегатов,

● расчет оптимальных агрегатов.

## Настройка

После переноса объектов подсистемы в конфигурацию необходимо установить расписание регламентных заданий, а также флажок **Использование**:

● **ОбновлениеАгрегатов**,

● **ПерестроениеАгрегатов**,

● **УстановкаПериодаРасчитанныхИтогов**.

### Размещение в командном интерфейсе

Для использования подсистемы необходимо разместить в командном интерфейсе администратора объект метаданных – обработку **УправлениеИтогамиИАгрегатами** для использования всех возможностей подсистемы.

### Настройка прав доступа пользователей

Для настройки прав доступа пользователей к данным подсистемы «Управление итогами и агрегатами» следует использовать роли:

Таблица 3.113.

| **№** | **Роли и их назначение** |
| --- | --- |
| 1. | **ПолныеПрава** (из подсистемы «Базовая функциональность»)  Доступ к администрированию итогов и агрегатов (доступ к обработке) |

## Использование при разработке конфигурации

При разработке конфигурации никаких дополнительных действий не требуется.

### Настройка обмена данными

Для настройки обмена данными никаких действий не требуется, так как подсистема не предоставляет собственных данных.

### Настройка обмена данными

Константу **ПараметрыИтоговИАгрегатов** не рекомендуется включать в планы обмена распределенной ИБ.

# 3.55. Файловые функции

Подсистема «Файловые функции» не имеет самостоятельной прикладной ценности. Она включает в себя общую файловую функциональность (функции, общие формы и др.), необходимую для работы других подсистем, таких как: «Работа с файлами», «Присоединенные файлы», «Поставляемые данные».

## Настройка

### Размещение в командном интерфейсе

В случае если в конфигурации не используется подсистема «Настройки программы», на рабочем месте администратора программы необходимо разместить константы:

● **ЗапрещатьЗагрузкуФайловПоРасширению**,

● **ИзвлекатьТекстыФайловНаСервере**,

● **МаксимальныйРазмерФайла**,

● **СписокЗапрещенныхРасширений**,

● **СписокРасширенийТекстовыхФайлов**,

● **СписокРасширенийФайловOpenDocument**,

● **ХранитьФайлыВТомахНаДиске**.

См. пример в демонстрационной конфигурации в форме **НастройкиРаботыСФайлами** обработки **ПанельАдминистрированияБСП**.

Затем разместить в командном интерфейсе администратора программы следующие объекты метаданных:

● справочник **ТомаХраненияФайлов**,

● обработка **АвтоматическоеИзвлечениеТекстов**.

### Настройка прав доступа пользователей

Для настройки прав доступа пользователей к данным подсистемы «Файловые функции» следует использовать следующие роли:

Таблица 3.114.

| **№** | **Роли и их назначение** |
| --- | --- |
| 1. | **Полные права**  Настройка подсистемы, создание томов для хранения файлов на диске |
| 2. | **Базовые права**  Право просмотра общих форм |

## Использование при разработке конфигурации

При разработке конфигурации никаких дополнительных действий не требуется.

### Настройка обмена данными

Если подсистема используется совместно с подсистемой «Обмен данными», то в модуле менеджера плана обмена распределенной информационной базы (РИБ) необходимо вписать реализацию в экспортную функцию **ИмяФормыСозданияНачальногоОбраза**:

Функция ИмяФормыСозданияНачальногоОбраза() Экспорт

Возврат "ОбщаяФорма.СозданиеНачальногоОбразаСФайлами";

КонецФункции

А также указать этот план обмена в качестве типа параметра общей команды **СозданиеНачальногоОбразаСФайлами**. Пример плана обмена с поддержкой миграции файлов в томах на диске см. в демонстрационной конфигурации – **\_ДемоОбменВРаспределеннойИнформационнойБазеСВнешнимиФайлами**.

В планы обмена рекомендуется включать все объекты метаданных подсистемы, кроме:

● регистра сведений **НомераОтсканированныхФайлов**,

● регистра сведений **СтатусыСинхронизацииФайловСОблачнымСервисом**,

● регистра сведений **НастройкиСинхронизацииФайлов**,

● константы **ИзвлекатьТекстыФайловНаСервере**,

● константы **ХранитьФайлыВТомахНаДиске**,

● константы **СинхронизироватьФайлы**,

● справочника **УчетныеЗаписиСинхронизацииФайлов**,

● справочника **ТомаХраненияФайлов**.

# 3.56. Центр мониторинга

Подсистема «Центр мониторинга» предназначена для сбора **обезличенной статистики** использования конфигурации, настроек конфигурации. В базовом варианте подсистема обеспечивает сбор сведений о технологических дампах платформы, количестве записей во всех таблицах информационной базы, а также значениях функциональных опций. Однако список собираемых показателей может быть расширен при внедрении конфигурации.

## Настройка

В базовом виде центр мониторинга позволяет собирать следующую информацию:

● информацию о системе;

● статистику использования конфигурации;

● при наличии подсистемы «Оценка производительности» – также агрегированную информацию по ключевым операциям.

Информация о системе:

**ИмяКомпьютера**: "F27001CA71718F89FC74E2AA42ABD5DF",

**ВерсияОС**: "version 6.1 Service Pack 1 (Build 7601)",

**ВерсияПриложения**: "8.3.6.1848",

**ИдентификаторКлиента**: "1d77d7a0-3025-400f-9be0-3b5655bf2edf",

**ОперативнаяПамять**: "7 876",

**Процессор**: "GenuineIntel Intel64 Family 6 Model 58 Stepping 9 3300 MHz",

**ТипПлатформы**: "Windows x86-64",

**ТекущийЯзык**: "Русский",

**ТекущийКодЛокализации**: "ru",

**ТекущийЯзыкСистемы**: "ru",

**ТекущийРежимЗапуска**: "Управляемое приложение",

**ЧасовойПоясСеанса**: "Europe/Moscow"

Базовая статистика по использованию конфигурации содержит количество записей в таблицах объектов метаданных:

● **План обмена**,

● **Справочник**,

● **Документ**,

● **Журнал документов**,

● **Перечисление**,

● **План видов характеристик**,

● **План счетов**,

● **План видов расчетов**,

● **Регистр сведений**,

● Регистр накопления,

● **Регистр бухгалтерии**,

● **Регистр расчета**,

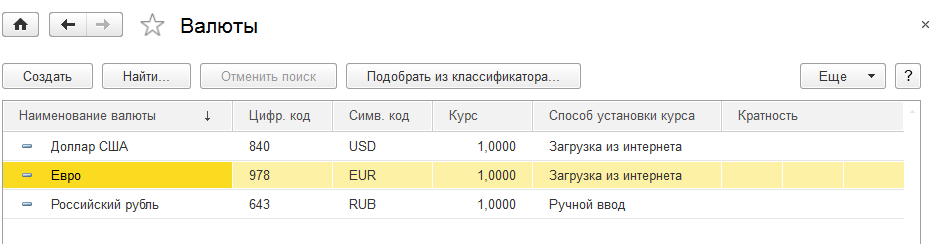
● **Бизнес-процесс**,

● **Задача**.

Базовая статистика по настройкам конфигурации содержит значение функциональных опций с хранением значения в константе с типом **Булево**.

Если этой статистики недостаточно и необходимо, например, проанализировать заполнение данных по некоторому признаку, то надо сформировать свой текст запроса для сбора статистики в процедуре **ПриСбореПоказателейСтатистикиКонфигурации** общего модуля **ЦентрМониторингаПереопределяемый**.

Возьмем справочник **Валюты**.



Процедура ПриСбореПоказателейСтатистикиКонфигурации() Экспорт

// Собрать статистику по способу установки курса справочника Валюты

// В регистр сведений <СтатистикаКонфигураций> будут внесена записи следующего вида

//

// Справочник.Валюты.СпособУстановкиКурса1 Количество1

// Справочник.Валюты.СпособУстановкиКурса2 Количество2

// ...

// Справочник.Валюты.СпособУстановкиКурсаN КоличествоN

СоответствиеИменМетаданных = Новый Соответствие;

ТекстЗапроса = "ВЫБРАТЬ

| СпособУстановкиКурса КАК СпособУстановкиКурса,

| КОЛИЧЕСТВО(\*) КАК Количество

|ИЗ

| Справочник.Валюты КАК Валюты

         |СГРУППИРОВАТЬ ПО

         | СпособУстановкиКурса

   |";

СоответствиеИменМетаданных.Вставить("Справочник.Валюты", ТекстЗапроса);

ЦентрМониторинга.ЗаписатьСтатистикуКонфигурации(СоответствиеИменМетаданных);

КонецПроцедуры

**Важно!**

В целях защиты конфиденциальной информации запрещено собирать и передавать любые данные из информационной базы и сведения о ней, за исключением следующих:

● Хеш имени компьютера,

● Версия операционной системы,

● Версия приложения,

● Идентификатор клиента,

● Объем оперативной памяти,

● Тип центрального процессора,

● Тип платформы,

● Языковые настройки,

● Текущий режим запуска,

● Часовой пояс,

● Количество записей в таблице информационной базы,

● Количество записей в таблице информационной базы с группировкой по аналитике (группировкой аналитики могут быть только признаки, содержащие значение **Булево**, значение **Перечисления**, значение предопределенных данных).

В случае необходимости анализировать статистику информационной базы с группировкой по аналитике, содержащей конфиденциальную информацию, в качестве признака разрешается использовать хеш представления аналитики.

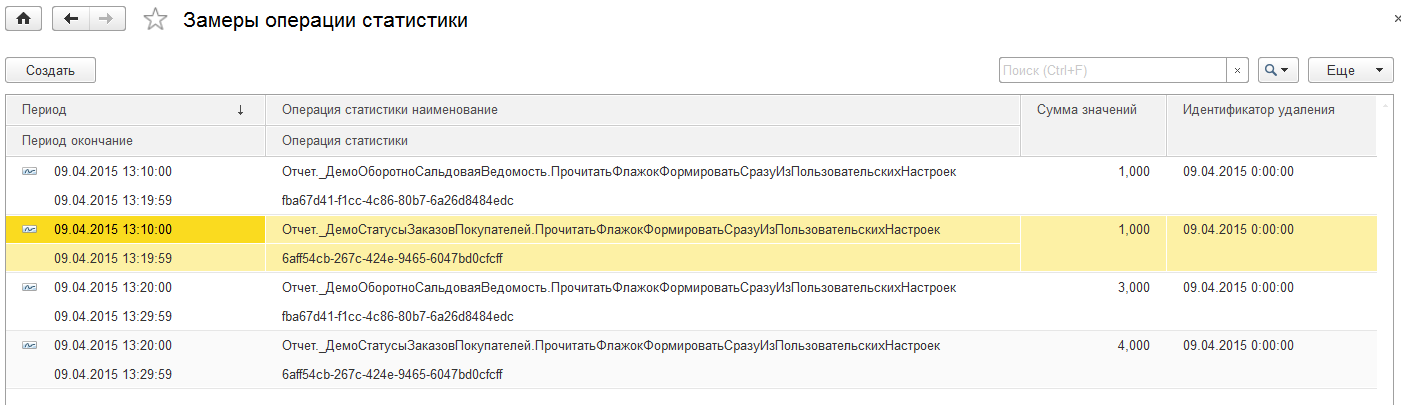
Также **Центр мониторинга** может собирать **статистику**использования функционала конфигурации, так называемую **бизнес-статитстику**. Например, необходимо анализировать информацию о том, какие отчеты и как часто используются с параметром **ФормироватьСразуИзПользовательскихНастроек**. Для регистрации данной бизнес-статистики необходимо найти место в конфигурации на сервере, где выполняется данная проверка. Необходимое место вставки программного кода – процедура **ПриСозданииНаСервере** общей формы **ФормаОтчета**.

Если НастройкиОтчета.ПрочитатьФлажокФормироватьСразуИзПользовательскихНастроек Тогда

ЦентрМониторинга.ЗаписатьОперациюБизнесСтатистики(НастройкиОтчета.ПолноеИмя + ".ПрочитатьФлажокФормироватьСразуИзПользовательскихНастроек", 1);

КонецЕсли;

После агрегации бизнес-статистики мы получим следующий результат



По полученным данным можно сделать следующий вывод:

За период с **13:10:00 09.04.2015** по **13:29:59 09.04.2015** отчет **\_ДемоОборотноСальдоваяВедомость**открывался с параметром Ф**ормироватьСразуИзПользовательскихНастроек** четыре раза, а отчет **\_ДемоСтатусыЗаказовПокупателей** открывался с параметром **ФормироватьСразуИзПользовательскихНастроек** пять раз.

Главное отличие регистрации бизнес-статистики от статистики использования конфигурации заключается в том, что бизнес-статистика агрегирует операции. Сбор статистики использования конфигурации выполняет срез информации, предыдущие данные удаляются.

**Центр мониторинга** при наличии подсистемы **Оценка производительности** может анализировать и передавать TOP N худших операций производительности, влияющих на общую производительность системы (по умолчанию 10). Но в данном случае необходимо быть внимательным. Замеры оценки производительности передаются в агрегированном виде за период сбора статистики, сами замеры – не передаются. Количеством худших операций производительности управляет сервис приема информации, при необходимости собираются все агрегированные замеры за период.

### Настройка прав доступа пользователей

Роли, используемые для работы с подсистемой:

Таблица 3.115.

| **№** | **Роли и их назначение** |
| --- | --- |
|  | **Администратор системы**  Принятие решения по поводу согласия/несогласия на сбор и отправку обезличенной статистики конфигурации. Настройка адреса сервиса примема данных |

## Использование при разработке конфигурации

Программный интерфейс подсистемы представлен экспортными процедурами и функциями общих модулей **ЦентрМониторинга** и **ЦентрМониторингаПереопределяемый** в стандартных областях **ПрограммныйИнтерфейс**.

### Настройка обмена данными

Объекты метаданных подсистемы не следует включать в планы обмена распределенной ИБ (РИБ), а также в планы обмена по правилам конвертации, поскольку для различных узлов мониторинг системы может работать по своим правилам.

# 3.57. Шаблоны сообщений

Подсистема «Шаблоны сообщений» позволяет экономить время, создавая письма и сообщения SMS по заранее подготовленным шаблоном. В тексте шаблонов сообщений можно определить реквизиты, которые будут заполняться данными из документов или справочников программы. Кроме того, в шаблоне можно указать, что к письму нужно прикреплять вложения и печатные формы.

Подсистема «Шаблоны сообщений» работает совместно с подсистемами «[Работа с почтовыми сообщениями](https://its.1c.ru/db/content/bsp23doc/src/3.43.%20%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%B0%20%D1%81%20%D0%BF%D0%BE%D1%87%D1%82%D0%BE%D0%B2%D1%8B%D0%BC%D0%B8%20%D1%81%D0%BE%D0%BE%D0%B1%D1%89%D0%B5%D0%BD%D0%B8%D1%8F%D0%BC%D0%B8.htm?_=1504708101#_работа_с_почтовыми)», «[Отправка SMS](https://its.1c.ru/db/content/bsp23doc/src/3.31.%20%D0%BE%D1%82%D0%BF%D1%80%D0%B0%D0%B2%D0%BA%D0%B0%20sms.htm?_=1504708101#_отправка_sms_1)», а также «[Взаимодействия](https://its.1c.ru/db/content/bsp23doc/src/3.10.%20%D0%B2%D0%B7%D0%B0%D0%B8%D0%BC%D0%BE%D0%B4%D0%B5%D0%B9%D1%81%D1%82%D0%B2%D0%B8%D1%8F.htm?_=1504708101#_взаимодействия)». Если какой-либо из перечисленных подсистем нет в составе конфигурации, то соответствующая функциональность шаблонов писем или SMS автоматически скрывается из интерфейса.

## Настройка

Для того чтобы пользователи имели возможность создавать шаблоны сообщение на основании документов или справочников, необходимо принять решение по поводу состава типов таких объектов (определить предметы шаблонов сообщений). Например, это могут быть документы счетов на оплату, справочники контрагентов, сотрудников и т.п.

● Затем выбранные объекты перечислить в свойстве **Тип** определяемого типа **ПредметШаблонаСообщения**;

во всех модулях менеджеров этих объектов добавить следующие процедуры и при необходимости, вписать их реализацию:

// СтандартныеПодсистемы.ШаблоныСообщений

// Вызывается при подготовке шаблонов сообщений и позволяет переопределить список реквизитов и вложений.

//

// Параметры:

//  Реквизиты               - ДеревоЗначений - список реквизитов шаблона.

//         \*\* Имя            - Строка - Уникальное имя общего реквизита.

//         \*\* Представление  - Строка - Представление общего реквизита.

//         \*\* Тип            - Тип    - Тип реквизита. По умолчанию строка.

//         \*\* Формат         - Строка - Формат вывода значения для чисел, дат, строк и булевых значений.

//  Вложения                - ТаблицаЗначений - печатные формы и вложения

//         \*\* Имя            - Строка - Уникальное имя вложения.

//         \*\* Представление  - Строка - Представление варианта.

//         \*\* ТипФайла       - Строка - Тип вложения, который соответствует расширению файла: "pdf", "png", "jpg", mxl" и др.

//  ДополнительныеПараметры - Структура - дополнительные сведения о шаблоне сообщений.

//

Процедура ПриПодготовкеШаблонаСообщения(Реквизиты, Вложения, ДополнительныеПараметры) Экспорт

КонецПроцедуры

// Вызывается в момент создания сообщений по шаблону для заполнения значений реквизитов и вложений.

//

// Параметры:

//  Сообщение - Структура - структура с ключами:

//    \* ЗначенияРеквизитов - Соответствие - список используемых в шаблоне реквизитов.

//      \*\* Ключ     - Строка - имя реквизита в шаблоне;

//      \*\* Значение - Строка - значение заполнения в шаблоне.

//    \* ЗначенияОбщихРеквизитов - Соответствие - список используемых в шаблоне общих реквизитов.

//      \*\* Ключ     - Строка - имя реквизита в шаблоне;

//      \*\* Значение - Строка - значение заполнения в шаблоне.

//    \* Вложения - Соответствие - значения реквизитов

//      \*\* Ключ     - Строка - имя вложения в шаблоне;

//      \*\* Значение - ДвоичныеДанные, Строка - двоичные данные или адрес во временном хранилище вложения.

//  ПредметСообщения - ЛюбаяСсылка - ссылка на объект являющийся источником данных.

//  ДополнительныеПараметры - Структура -  Дополнительная информация о шаблоне сообщения.

//

Процедура ПриФормированииСообщения(Сообщение, ПредметСообщения, ДополнительныеПараметры) Экспорт

КонецПроцедуры

// Заполняет список получателей SMS при отправке сообщения сформированного по шаблону.

//

// Параметры:

//   ПолучателиSMS - ТаблицаЗначений - список получается SMS.

//     \* НомерТелефона - Строка - номер телефона, куда будет отправлено сообщение SMS.

//     \* Представление - Строка - представление получателя сообщения SMS.

//     \* Контакт       - СправочникСсылка - контакт, которому принадлежит номер телефона.

//  ПредметСообщения - ЛюбаяСсылка - ссылка на объект являющийся источником данных.

//

Процедура ПриЗаполненииТелефоновПолучателейВСообщении(ПолучателиSMS, ПредметСообщения) Экспорт

КонецПроцедуры

// Заполняет список получателей письма при отправки сообщения сформированного по шаблону.

//

// Параметры:

//   ПолучателиПисьма - ТаблицаЗначений - список получается письма.

//     \* Адрес           - Строка - адрес электронной почты получателя.

//     \* Представление   - Строка - представление получается письма.

//  ПредметСообщения - ЛюбаяСсылка - ссылка на объект являющийся источником данных.

//

Процедура ПриЗаполненииПочтыПолучателейВСообщении(ПолучателиПисьма, ПредметСообщения) Экспорт

КонецПроцедуры

// Конец СтандартныеПодсистемы.ШаблоныСообщений

● Если код процедур оставить незаполненными, то в шаблонах сообщений будут доступны реквизиты предмета  и определенные в нем печатные формы. При наличии у предмета контактной информации список адресатов будет заполнен, в противном случае он будет пустой.

● Пример реализации этих процедур см. в демонстрационной конфигурации в справочнике **\_ДемоФизическиеЛица**.

### Размещение в командном интерфейсе

Для использования подсистемы необходимо разместить в командном интерфейсе:

● Справочник **ШаблоныСообщений** – для ведения списка шаблонов сообщений (выполняется ответственным за список шаблонов сообщений или администраторам).

В случае если в конфигурации не используется подсистема «[Настройки программы](https://its.1c.ru/db/content/bsp23doc/src/3.27.%20%D0%BD%D0%B0%D1%81%D1%82%D1%80%D0%BE%D0%B9%D0%BA%D0%B8%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D1%8B.htm?_=1504708101#_настройки_программы)», в рабочем месте администратора программы необходимо разместить константу:

● **ИспользоватьШаблоныСообщений** – для включения и выключения подсистемы;

См. пример в форме **Органайзер** обработки **ПанельАдминистрированияБСП**.

### Настройка прав доступа пользователей

Для настройки прав доступа пользователей к данным подсистемы «Шаблоны сообщений» следует использовать роли, приведенные ниже.

Таблица 3.116.

| **№** | **Роли и их назначение** |
| --- | --- |
|  | **БазовыеПрава**  Добавление и изменение персональных шаблонов сообщений  (доступных только автору).  Просмотр и использование общих шаблонов сообщений (доступных всем пользователям). |
|  | **ДобавлениеИзменениеШаблоновСообщений**  Подготовка общих шаблонов сообщений. Добавление и изменение шаблонов сообщений.  Администрирование списка шаблонов сообщений. |
|  | **ПолныеПрава** (из подсистемы «Базовая функциональность»).  Включение и отключение использования подсистемы «Шаблонов сообщений». Удаление помеченных на удаление объектов подсистемы. |

Пример настройки прав доступа пользователей:

Таблица 3.117.

| **№** | **Группа пользователей и ее функции** | **Состав ролей** |
| --- | --- | --- |
|  | Ответственный за список шаблонов сообщений | ● **БазовыеПрава** (из подсистемы «Базовая функциональность»),  ● **ДобавлениеИзменениеШаблоновСообщений** |

### Особые случаи внедрения подсистемы

● [Внедрение подсистемы «Шаблоны сообщений» без подсистемы «Управление доступом».](https://its.1c.ru/db/content/bsp23doc/src/3.59.%20%D0%BE%D1%81%D0%BE%D0%B1%D1%8B%D0%B5%20%D1%81%D0%BB%D1%83%D1%87%D0%B0%D0%B8%20%D0%B2%D0%BD%D0%B5%D0%B4%D1%80%D0%B5%D0%BD%D0%B8%D1%8F%20%D0%BF%D0%BE%D0%B4%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC.htm?_=1504708101#_внедрение_подсистемы_)

● [Внедрение подсистемы «Шаблоны сообщений» без подсистемы «Присоединенные файлы».](https://its.1c.ru/db/content/bsp23doc/src/3.59.%20%D0%BE%D1%81%D0%BE%D0%B1%D1%8B%D0%B5%20%D1%81%D0%BB%D1%83%D1%87%D0%B0%D0%B8%20%D0%B2%D0%BD%D0%B5%D0%B4%D1%80%D0%B5%D0%BD%D0%B8%D1%8F%20%D0%BF%D0%BE%D0%B4%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC.htm?_=1504708101#_внедрение_подсистемы_)

## Использование при разработке конфигурации

По умолчанию в шаблонах сообщений доступны все реквизиты предмета сообщения, за исключением реквизитов с типом **ХранилищеЗначения** и некоторых стандартных реквизитов (**ПометкаУдаления**, **ЭтоГруппа** и др.). Также доступны поля контактной информации и доп. реквизиты и сведения. В качестве вложений к письмам доступы определенные в конфигурации печатные формы, внешние печатные формы и печатные формы из расширений. При наличии контактной информации у предмета будет заполонен список адресатов: для почтовых сообщений из полей с видом контактной информации электронная почта, а для сообщений SMS с видом контактной информации телефон.

В случаях, когда при создании шаблонов сообщений недостаточно стандартных средств, можно предусмотреть собственные реквизиты, печатные формы и вложения

1.  В процедуре **ПриПодготовкеШаблонаСообщения**определить состав используемых реквизитов и вложений:

НовыйРеквизит = Реквизиты.Добавить();

НовыйРеквизит.Имя = "СостояниеКонтрагента";

НовыйРеквизит.Представление = НСтр("ru = 'Состояние контрагента'");

НовыйРеквизит.Тип = Новый ОписаниеТипов("Строка");

Карта = Вложения.Добавить();

КартинкаКнопки.Имя = "КартаРасположенияОфиса'";

КартинкаКнопки.Представление = НСтр("ru = Карта расположения офиса'");

КартинкаКнопки.ТипФайла = "jpg";

2. В процедуре **ПриФормированииСообщения**заполнить ранее определенные реквизиты и вложения:

Сообщение.ЗначенияРеквизитов["СостояниеКонтрагента"] = ПроверкаКонтрагентов.СостояниеКонтрагента(ПредметСообщения.ИНН, ПредметСообщения.КПП);

АдресКартинкиВоВременноеХранилище = КартаРасположенияОфис();

Сообщение.Вложения["КартаРасположенияОфиса"] = АдресКартинкиВоВременноеХранилище;

3. Если у предмета нет контактной информации или требуется особенный список адресатов, то список получателей письма или сообщений SMS  должен быть заполен по контактной информации реквизита предмета или используя собственный алгоритм. Для этого в модуле менеджера объекта, являющимся предметом шаблонов сообщений, требуется. В этом случае в процедурах **ПриЗаполненииТелефоновПолучателейВСообщении**и **ПриЗаполненииПочтыПолучателейВСообщении** заполнить получателей сообщения:

Процедура ПриЗаполненииПочтыПолучателейВСообщении(ПолучателиПисьма, ПредметСообщения) Экспорт

ШаблоныСообщений.ЗаполнитьПолучателей(ПолучателиПисьма, ПредметСообщения, "Контрагент");

Получатель = ПолучателиПисьма.Добавить();

Получатель.Адрес = ПредметСообщения.ЭлектроннаяПочтаСтрокой;

Получатель.Представление = ПредметСообщения.ЭлектроннаяПочтаПредставление;

КонецПроцедуры

Процедура ПриЗаполненииТелефоновПолучателейВСообщении(ПолучателиSMS, ПредметСообщения) Экспорт

ШаблоныСообщений.ЗаполнитьПолучателей(ПолучателиSMS, ПредметСообщения, "Контрагент", Перечисления.ТипыКонтактнойИнформации.Телефон);

КонецПроцедуры

Пример использования см. в документе **\_ДемоСчетНаОплатуПокупателю** и справочнике **\_ДемоКонтрагенты**демонстрационной конфигурации.

В тех случаях, когда требуется определить список реквизитов, вложений или список получателей сразу для всех или нескольких предметов шаблонов сообщений, то необходимо размещать код в одноименных процедурах общего модуля **ШаблоныСообщенийПереопределяемый**.

### Разработка предметов шаблонов сообщений, не привязанных к объектам конфигурации.

В случаях, когда необходимо создать назначение для шаблонов сообщений, которое не является объектом конфигурации, то в процедуре **ПриОпределенииНастроек** общего модуля **ШаблоныСообщенийПереопределяемый** необходимо добавить код, например:

Предмет = Настройки.ПредметыШаблонов.Добавить();

Предмет.Имя = "ОповещениеКлиентаИзменениеЗаказа";

Предмет.Представление = НСтр("ru = 'Оповещение клиента ""Изменение заказа""'");

Пример использования таких назначений см. в документе **\_ДемоЗаказПокупателя** демонстрационной конфигурации.

### Добавление общих реквизитов

Помимо реквизитов, специфичных для конкретного предмета шаблона сообщений, предусмотрена возможность определять реквизиты, общие для всех предметов. Например, для того чтобы сделать доступным для выбора реквизит **Основная организация** во всех шаблонах сообщений, необходимо описать общий реквизит в общем модуле **ШаблоныСообщенийПереопределяемый**в процедуре **ПриОпределенииНастроек**:

НовыйРеквизит = Настройки.ОбщиеРеквизиты.Строки.Добавить();

НовыйРеквизит.Имя = "ОсновнаяОрганизация";

НовыйРеквизит.Представление = НСтр("ru = 'Основная организация'");

НовыйРеквизит.Тип = Тип("СправочникСсылка.\_ДемоОрганизации");

Заполнение общих реквизитов производится в общем модуле **ШаблоныСообщенийПереопределяемый**в процедуре **ПриФормированииСообщения аналогично другим реквизитам.**

### Настройка обмена данными

Все объекты метаданных подсистемы, содержащие данные, рекомендуется включать в планы обмена распределенной ИБ (РИБ).

# 3.58. Электронная подпись

Подсистема «Электронная подпись» предоставляет программный и пользовательский интерфейс для работы со средствами криптографии. Поддерживаются электронная подпись, проверка электронной подписи, проверка сертификатов, шифрование и расшифровка.

Кроме того, подсистема позволяет оформить заявление на выпуск нового сертификата квалифицированной электронной подписи в удостоверяющем центре ООО «НПЦ «1С» ([http://ca.1c.ru](http://ca.1c.ru/)) с помощью удобного помощника.

## Настройка

Принять решение по поводу объектов конфигурации ссылочного типа (справочников, документов и т. п.), в которых должны быть реализованы команды по подписанию и проверке электронной подписи. Также необходимо определиться с местом размещения этих команд в формах объектов.

В выбранных объектах метаданных добавить реквизиты согласно таблице 3.116.

Таблица 3.118.

| **Реквизит** | **Тип** | **Подсказка** |
| --- | --- | --- |
| **ПодписанЭП** | **Булево** | Признак того, что файл подписан электронно |
| Табличная часть **ЭлектронныеПодписи** | | |
| **ДатаПодписи** | **Дата** (Дата и время) | Дата установки подписи |
| **ИмяФайлаПодписи** | **Строка** (длиной 260) | Имя файла подписи (служебное поле) |
| **Комментарий** | **Строка** (неограниченной длины) | Комментарий к подписи |
| **КомуВыданСертификат** | **Строка** (неограниченной длины) | Владелец сертификата (фамилия имя, организация, должность) |
| **Отпечаток** | **Строка** (длиной 28) | Отпечаток сертификата (служебное поле) |
| **Подпись** | **ХранилищеЗначения** | Двоичные данные подписи (служебное поле) |
| **УстановившийПодпись** | **СправочникСсылка.Пользователи** | Пользователь, установивший подпись |
| **Сертификат** | **ХранилищеЗначения** | Двоичные данные сертификата (служебное поле) |
| **ДатаПроверкиПодписи** | **Дата** (Дата и время) | Дата последней проверки подписи |
| **ПодписьВерна** | **Булево** | Результат последней проверки подписи |

Пример настройки реквизитов можно посмотреть в демонстрационной конфигурации в справочнике **ВерсииФайлов**.

Добавить функциональную опцию **ИспользоватьЭлектронныеПодписи<ИмяБиблиотеки>**, включив в нее реквизиты и табличные части объектов метаданных, описанные выше (пример можно посмотреть в функциональной опции **\_ДемоИспользоватьЭлектронныеПодписиБСП**).

Реализовать команды по подписанию и проверке электронной подписи с использованием экспортных функций общего модуля **ЭлектроннаяПодписьКлиент**. Пример реализации команд можно посмотреть в демонстрационной конфигурации – команды **Подписать** и **Проверить** в форме элемента справочника **Файлы**.

Принять решение по поводу объектов конфигурации ссылочного типа (справочников, документов и т. п.), в которых должны быть реализованы команды по шифрованию и расшифровке. Также определиться с местом размещения этих команд в формах объектов.

В выбранных объектах метаданных добавить реквизиты согласно таблице 3.117.

Таблица 3.119.

| **Реквизит** | **Тип** | **Подсказка** |
| --- | --- | --- |
| **Зашифрован** | **Булево** | Признак того, что файл зашифрован |
| Табличная часть **СертификатыШифрования** | | |
| **Отпечаток** | Строка длиной 28 | Отпечаток сертификата (служебное поле) |
| **Представление** | Строка неограниченной длины | Владелец сертификата (фамилия имя, организация, должность) |
| **Сертификат** | **ХранилищеЗначения** | Двоичные данные сертификата (служебное поле) |

Пример настройки реквизитов можно посмотреть в демонстрационной конфигурации в справочнике **Файлы**.

Добавить функциональную опцию **ИспользоватьШифрование<ИмяБиблиотеки>**, включив в нее реквизиты и табличные части объектов метаданных, описанные выше (пример можно посмотреть в функциональной опции **\_ДемоИспользоватьШифрованиеБСП**).

Реализовать команды по шифрованию и расшифровке с использованием экспортных функций общего модуля **ЭлектроннаяПодписьКлиент**. Пример реализации команд можно посмотреть в демонстрационной конфигурации – команды **Зашифровать** и **Расшифровать** в форме элемента справочника **Файлы**.

В интерфейсе персональных настроек пользователя разместить команду открытия общей формы **НастройкиЭлектроннойПодписиИШифрования** с помощью процедуры **ОткрытьНастройкиЭлектроннойПодписиИШифрования** общего модуля **ЭлектроннаяПодписьКлиент**. См. пример в демонстрационной конфигурации в общей форме **\_ДемоМоиНастройки**.

В случае если в конфигурации не используется подсистема «Настройки программы», в рабочем месте администратора программы необходимо разместить константы **ИспользоватьЭлектронныеПодписи**,**ИспользоватьШифрование** и команду открытия общей формы **НастройкиЭлектроннойПодписиИШифрования** с помощью процедуры **ОткрытьНастройкиЭлектроннойПодписиИШифрования** общего модуля **ЭлектроннаяПодписьКлиент**. См. пример в демонстрационной конфигурации в форме **ОбщиеНастройки** обработки **ПанельАдминистрированияБСП**.

Для работы помощника, позволяющего оформить заявление на выпуск сертификата, требуется внедрить подсистемы «Контактная информация», «Адресный классификатор», «Печать» и «Работа с контрагентами». Если этого не сделать, помощник будет скрыт. Также можно внедрить подсистему «Организации». Для эффективного использования помощника необходимо в общий модуль **ЭлектроннаяПодписьПереопределяемый** вписать код в процедуры **ПриЗаполненииРеквизитовОрганизацииВЗаявленииНаСертификат**,**ПриЗаполненииРеквизитовВладельцаВЗаявленииНаСертификат**,**ПриЗаполненииРеквизитовРуководителяВЗаявленииНаСертификат**,**ПриЗаполненииРеквизитовПартнераВЗаявленииНаСертификат**. Пример можно посмотреть в демонстрационной конфигурации.

### Настройка прав доступа пользователей

Для настройки прав доступа пользователей к данным подсистемы «Электронная подпись» следует использовать следующие роли:

Таблица 3.120.

| **№** | **Роли и их назначение** |
| --- | --- |
| 1. | **ДобавлениеИзменениеЭлектронныхПодписейИШифрование** (+ **БазовыеПрава**)  Просмотр сертификатов, программ электронной подписи и шифрования, настроек подсистемы. Добавление заявления на выпуск сертификата. Выполнение подписания, проверки подписей, шифрования, расшифровки и остальных операций. Изменение некоторых свойств своих сертификатов в справочнике |
| 2. | **ПолныеПрава** (из подсистемы «Базовая функциональность»)  Изменение всех доступных свойств любых сертификатов в справочнике.  Изменение административных настроек подсистемы:  ● включение/отключение подписания/шифрования;  ● настройка списка программ электронной подписи и шифрования;  ● включение/отключение выполнения операций на сервере |

## Использование при разработке конфигурации

Программный интерфейс подсистемы представлен экспортными процедурами и функциями общих модулей **ЭлектроннаяПодпись<\*>** в стандартных областях **ПрограммныйИнтерфейс**.

Код проверки подписи реализуется по месту, т. к. зависит от прикладных характеристик подписываемого объекта. Пример реализации кода можно посмотреть в справочнике **Файлы** – в команде **ПроверитьВсе** в форме элемента.

### Настройка обмена данными

Для настройки обмена данными следует руководствоваться общими правилами.

В планы обмена распределенной информационной базы (РИБ) рекомендуется включать все объекты метаданных подсистемы, кроме:

● константы **ПроверятьЭлектронныеПодписиНаСервере**,

● константы **СоздаватьЭлектронныеПодписиНаСервере**,

● регистра сведений **ПутиКПрограммамЭлектроннойПодписиИШифрованияНаСерверахLinux**,

● объектов метаданных, имена которых начинаются с **Удалить**.

# 3.59. Особые случаи внедрения подсистем

В данном разделе описаны особые случаи внедрения подсистем. Как правило, эти случаи возникают из-за взаимосвязей некоторых подсистем друг с другом – когда для внедрения одной подсистемы требуется выполнить донастройку соседней и, наоборот, при внедрении одной подсистемы без другой требуется отменить ранее выполненную настройку в этой другой подсистеме.

На инструкции данного раздела приведены ссылки из инструкций по настройке соответствующих подсистем раздела 3. Инструкции приведены как для случая совместного, так и для случаев раздельного внедрения – в зависимости от способа поставки подсистем из библиотеки по умолчанию. Например, если подсистемы поставляются по умолчанию не связанными друг с другом, то дается инструкция по настройке связи. При необходимости «разорвать» связь между подсистемами необходимо выполнить обратные действия согласно инструкции.

## Внедрение подсистем «Бизнес-процессы и задачи» и «Управление доступом»

По умолчанию для реализации ограничения доступа на уровне записей к данным подсистемы «[Бизнес-процессы и задачи](https://its.1c.ru/db/content/bsp23doc/src/3.6.%20%D0%B1%D0%B8%D0%B7%D0%BD%D0%B5%D1%81-%D0%BF%D1%80%D0%BE%D1%86%D0%B5%D1%81%D1%81%D1%8B%20%D0%B8%20%D0%B7%D0%B0%D0%B4%D0%B0%D1%87%D0%B8.htm?_=1504708101#_бизнес-процессы_и_задачи)» используются возможности подсистемы «[Управление доступом](https://its.1c.ru/db/content/bsp23doc/src/3.53.%20%D1%83%D0%BF%D1%80%D0%B0%D0%B2%D0%BB%D0%B5%D0%BD%D0%B8%D0%B5%20%D0%B4%D0%BE%D1%81%D1%82%D1%83%D0%BF%D0%BE%D0%BC.htm?_=1504708101#_управление_доступом)». При внедрении подсистемы «Бизнес-процессы и задачи» без подсистемы «Управление доступом» необходимо удалить шаблоны и тексты ограничений доступа, которые используют возможности подсистемы «Управление доступом», в следующих ролях:

● **ДобавлениеИзменениеЗаданий**,

● **ИзменениеВыполнениеЗадач**,

● **ЧтениеЗаданий**.

Если подсистема «Управление доступом» внедряется без подсистемы «Бизнес-процессы и задачи», то при первоначальном внедрении (и обновлении) конфигуратор выдаст предупреждение о наличии неразрешимых ссылок из-за отсутствия объектов метаданных подсистемы «Бизнес-процессы и задачи». В этом случае в окне **Неразрешимые ссылки** необходимо нажать кнопку **Продолжить**.

## Внедрение подсистем «Взаимодействия» и «Управление доступом»

По умолчанию для реализации ограничения доступа на уровне записей к объектам подсистемы «[Взаимодействия](https://its.1c.ru/db/content/bsp23doc/src/3.10.%20%D0%B2%D0%B7%D0%B0%D0%B8%D0%BC%D0%BE%D0%B4%D0%B5%D0%B9%D1%81%D1%82%D0%B2%D0%B8%D1%8F.htm?_=1504708101#_взаимодействия)» используются возможности подсистемы «[Управление доступом](https://its.1c.ru/db/content/bsp23doc/src/3.53.%20%D1%83%D0%BF%D1%80%D0%B0%D0%B2%D0%BB%D0%B5%D0%BD%D0%B8%D0%B5%20%D0%B4%D0%BE%D1%81%D1%82%D1%83%D0%BF%D0%BE%D0%BC.htm?_=1504708101#_управление_доступом)».

При внедрении (или обновлении) подсистемы «Управление доступом» без подсистемы «Взаимодействия» конфигуратор выдаст предупреждение о наличии неразрешимых ссылок из-за отсутствия справочников и документов подсистемы «Взаимодействия» В этом случае в окне **Неразрешимые ссылки** необходимо нажать кнопку **Продолжить**.

## Внедрение подсистемы «Взаимодействия» без подсистемы «Электронная подпись»

По умолчанию подсистема «[Электронная подпись](https://its.1c.ru/db/content/bsp23doc/src/3.56.%20%D1%86%D0%B5%D0%BD%D1%82%D1%80%20%D0%BC%D0%BE%D0%BD%D0%B8%D1%82%D0%BE%D1%80%D0%B8%D0%BD%D0%B3%D0%B0.htm?_=1504708101#_электронная_подпись)» расширяет поведение подсистемы «[Взаимодействия](https://its.1c.ru/db/content/bsp23doc/src/3.10.%20%D0%B2%D0%B7%D0%B0%D0%B8%D0%BC%D0%BE%D0%B4%D0%B5%D0%B9%D1%81%D1%82%D0%B2%D0%B8%D1%8F.htm?_=1504708101#_взаимодействия)». Поэтому при внедрении подсистемы «Взаимодействия» без подсистемы «Электронная подпись» требуется:

● включить возможность внесения изменений в каждый из справочников:

* ● **ВстречаПрисоединенныеФайлы**;
* ● **ЗапланированноеВзаимодействиеПрисоединенныеФайлы**;
* ● **СообщениеSMSПрисоединенныеФайлы**;
* ● **ТелефонныйЗвонокПрисоединенныеФайлы**;
* ● **ЭлектронноеПисьмоВходящееПрисоединенныеФайлы**;
* ● **ЭлектронноеПисьмоИсходящееПрисоединенныеФайлы**;

● в перечисленных выше справочниках:

* ● удалить реквизиты **Зашифрован** и **ПодписанЭП**;
* ● удалить табличные части **ЭлектронныеПодписи** и **СертификатыШифрования**.

## Внедрение подсистем «Дополнительные отчеты и обработки» и «Управление доступом»

По умолчанию для реализации ограничения доступа на уровне записей к дополнительным отчетам и обработкам в подсистеме [«Дополнительные отчеты и обработки»](https://its.1c.ru/db/content/bsp23doc/src/3.59.%20%D0%BE%D1%81%D0%BE%D0%B1%D1%8B%D0%B5%20%D1%81%D0%BB%D1%83%D1%87%D0%B0%D0%B8%20%D0%B2%D0%BD%D0%B5%D0%B4%D1%80%D0%B5%D0%BD%D0%B8%D1%8F%20%D0%BF%D0%BE%D0%B4%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC.htm?_=1504708101#_свойства_2) используются возможности подсистемы [«Управление доступом»](https://its.1c.ru/db/content/bsp23doc/src/3.59.%20%D0%BE%D1%81%D0%BE%D0%B1%D1%8B%D0%B5%20%D1%81%D0%BB%D1%83%D1%87%D0%B0%D0%B8%20%D0%B2%D0%BD%D0%B5%D0%B4%D1%80%D0%B5%D0%BD%D0%B8%D1%8F%20%D0%BF%D0%BE%D0%B4%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC.htm?_=1504708101#_управление_доступом_).

При внедрении подсистемы «Дополнительные отчеты и обработки» в конфигурацию без подсистемы «Управление доступом» следует внести в конфигурацию изменение:

● в роли **ЧтениеДополнительныхОтчетовИОбработок** удалить шаблон и текст ограничений доступа к справочнику **ДополнительныеОтчетыИОбработки**.

## Внедрение подсистемы «Пользователи» без подсистемы «Контактная информация»

По умолчанию подсистема «[Контактная информация](https://its.1c.ru/db/content/bsp23doc/src/3.24.%20%D0%BA%D0%BE%D0%BD%D1%82%D0%B0%D0%BA%D1%82%D0%BD%D0%B0%D1%8F%20%D0%B8%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%86%D0%B8%D1%8F.htm?_=1504708101#_контактная_информация)» расширяет поведение подсистемы «[Пользователи](https://its.1c.ru/db/content/bsp23doc/src/3.37.%20%D0%BF%D0%BE%D0%BB%D1%8C%D0%B7%D0%BE%D0%B2%D0%B0%D1%82%D0%B5%D0%BB%D0%B8.htm?_=1504708101#_пользователи)». Поэтому при внедрении подсистемы «Пользователи» без подсистемы «Контактная информация» требуется:

● включить возможность внесения изменений в справочник **Пользователи**,

● удалить табличную часть **КонтактнаяИнформация**.

## Внедрение подсистемы «Пользователи» без подсистемы «Свойства»

По умолчанию подсистема «[Свойства](https://its.1c.ru/db/content/bsp23doc/src/3.48.%20%D1%81%D0%B2%D0%BE%D0%B9%D1%81%D1%82%D0%B2%D0%B0.htm?_=1504708101#_свойства)» расширяет поведение подсистемы «[Пользователи](https://its.1c.ru/db/content/bsp23doc/src/3.37.%20%D0%BF%D0%BE%D0%BB%D1%8C%D0%B7%D0%BE%D0%B2%D0%B0%D1%82%D0%B5%D0%BB%D0%B8.htm?_=1504708101#_пользователи)». Поэтому при внедрении подсистемы «Пользователи» без подсистемы «Свойства» требуется:

● включить возможность внесения изменений в справочник **Пользователи**:

* ● удалить табличную часть **ДополнительныеРеквизиты**;

● включить возможность внесения изменений в справочник **ВнешниеПользователи**:

* ● удалить табличную часть **ДополнительныеРеквизиты**.

## Внедрение подсистем «Работа с почтовыми сообщениями» и «Управление доступом»

По умолчанию для реализации ограничения доступа на уровне записей к данным подсистемы «[Работа с почтовыми сообщениями](https://its.1c.ru/db/content/bsp23doc/src/3.43.%20%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%B0%20%D1%81%20%D0%BF%D0%BE%D1%87%D1%82%D0%BE%D0%B2%D1%8B%D0%BC%D0%B8%20%D1%81%D0%BE%D0%BE%D0%B1%D1%89%D0%B5%D0%BD%D0%B8%D1%8F%D0%BC%D0%B8.htm?_=1504708101#_работа_с_почтовыми)» используются возможности подсистемы «[Управление доступом](https://its.1c.ru/db/content/bsp23doc/src/3.53.%20%D1%83%D0%BF%D1%80%D0%B0%D0%B2%D0%BB%D0%B5%D0%BD%D0%B8%D0%B5%20%D0%B4%D0%BE%D1%81%D1%82%D1%83%D0%BF%D0%BE%D0%BC.htm?_=1504708101#_управление_доступом)».

При внедрении подсистемы «Работа с почтовыми сообщениями» в конфигурацию без подсистемы «Управление доступом» следует внести в конфигурацию изменение:

● у роли **ЧтениеУчетныхЗаписейЭлектроннойПочты** удалить шаблоны и тексты ограничений доступа к справочнику **УчетныеЗаписиЭлектроннойПочты**.

## Внедрение подсистемы «Работа с файлами» вместе с подсистемой «Присоединенные файлы»

Для доступа к списку файлов в томе необходимо в критерии отбора **ФайлыВТоме** установить флажок **Использовать стандартные команды**. (Примечание: при использовании подсистемы «[Работа с файлами](https://its.1c.ru/db/content/bsp23doc/src/3.44.%20%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%B0%20%D1%81%20%D1%84%D0%B0%D0%B9%D0%BB%D0%B0%D0%BC%D0%B8.htm?_=1504708101#_работа_с_файлами)» совместно с подсистемой «[Присоединенные файлы](https://its.1c.ru/db/content/bsp23doc/src/3.39.%20%D0%BF%D1%80%D0%B8%D1%81%D0%BE%D0%B5%D0%B4%D0%B8%D0%BD%D0%B5%D0%BD%D0%BD%D1%8B%D0%B5%20%D1%84%D0%B0%D0%B9%D0%BB%D1%8B.htm?_=1504708101#_присоединенные_файлы)» доступ к списку файлов в томах осуществляется другими средствами).

Для исключения ошибочного понимания списка файлов команда **ВсеФайлы** справочника **Файлы** должна быть исключена из всех командных интерфейсов (доступ через **Все функции** к списку справочника **Файлы** остается и рассчитан на высококвалифицированных администраторов).

Для исключения ошибочного понимания списка присоединенных файлов следует использовать подсистему «Работа с файлами» только для справочника **Папки файлов**. Соответственно, типы владельцев файлов, использованные в подсистеме «Присоединенные файлы», должны быть исключены из типов владельцев файлов подсистемы «Работа с файлами».

## Внедрение подсистем «Работа с файлами» и «Управление доступом»

По умолчанию для реализации ограничения доступа на уровне записей к объектам подсистемы «[Работа с файлами](https://its.1c.ru/db/content/bsp23doc/src/3.44.%20%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%B0%20%D1%81%20%D1%84%D0%B0%D0%B9%D0%BB%D0%B0%D0%BC%D0%B8.htm?_=1504708101#_работа_с_файлами)» используются возможности подсистемы «[Управление доступом](https://its.1c.ru/db/content/bsp23doc/src/3.53.%20%D1%83%D0%BF%D1%80%D0%B0%D0%B2%D0%BB%D0%B5%D0%BD%D0%B8%D0%B5%20%D0%B4%D0%BE%D1%81%D1%82%D1%83%D0%BF%D0%BE%D0%BC.htm?_=1504708101#_управление_доступом)».

При внедрении (или обновлении) подсистемы «Управление доступом» без подсистемы «Работа с файлами» конфигуратор выдаст предупреждение о наличии неразрешимых ссылок из-за отсутствия справочников **ПапкиФайлов** и **Файлы**. В этом случае в окне **Неразрешимые ссылки** необходимо нажать кнопку **Продолжить**.

## Внедрение подсистемы «Работа с файлами» без подсистемы «Электронная подпись»

По умолчанию подсистема «[Электронная подпись](https://its.1c.ru/db/content/bsp23doc/src/3.56.%20%D1%86%D0%B5%D0%BD%D1%82%D1%80%20%D0%BC%D0%BE%D0%BD%D0%B8%D1%82%D0%BE%D1%80%D0%B8%D0%BD%D0%B3%D0%B0.htm?_=1504708101#_электронная_подпись)» расширяет поведение подсистемы «[Работа с файлами](https://its.1c.ru/db/content/bsp23doc/src/3.44.%20%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%B0%20%D1%81%20%D1%84%D0%B0%D0%B9%D0%BB%D0%B0%D0%BC%D0%B8.htm?_=1504708101#_работа_с_файлами)». Поэтому при внедрении подсистемы «Работа с файлами» без подсистемы «Электронная подпись» требуется:

● включить возможность внесения изменений в справочник **Файлы**:

* ● удалить реквизит **Зашифрован**;
* ● удалить табличную часть **СертификатыШифрования**;

● включить возможность внесения изменений в справочник **ВерсииФайлов**:

* ● удалить реквизиты **Зашифрован** и **ПодписанЭП**;
* ● удалить табличную часть **ЭлектронныеПодписи**.

## Внедрение подсистемы «Работа с файлами» без подсистемы «Свойства»

По умолчанию подсистема «[Свойства](https://its.1c.ru/db/content/bsp23doc/src/3.48.%20%D1%81%D0%B2%D0%BE%D0%B9%D1%81%D1%82%D0%B2%D0%B0.htm?_=1504708101#_свойства)» расширяет поведение подсистемы «[Работа с файлами](https://its.1c.ru/db/content/bsp23doc/src/3.44.%20%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%B0%20%D1%81%20%D1%84%D0%B0%D0%B9%D0%BB%D0%B0%D0%BC%D0%B8.htm?_=1504708101#_работа_с_файлами)». Поэтому при внедрении подсистемы «Работа с файлами» без подсистемы «Свойства» требуется:

● включить возможность внесения изменений в справочник **Файлы**:

* ● удалить табличную часть **ДополнительныеРеквизиты**;

● включить возможность внесения изменений в справочник **ПапкиФайлов**:

* ● удалить табличную часть **ДополнительныеРеквизиты**.

## Внедрение подсистем «Свойства» и «Управление доступом»

По умолчанию для реализации ограничения доступа на уровне записей к дополнительным сведениям в подсистеме «[Свойства](https://its.1c.ru/db/content/bsp23doc/src/3.48.%20%D1%81%D0%B2%D0%BE%D0%B9%D1%81%D1%82%D0%B2%D0%B0.htm?_=1504708101#_свойства)» используются возможности подсистемы «[Управление доступом](https://its.1c.ru/db/content/bsp23doc/src/3.53.%20%D1%83%D0%BF%D1%80%D0%B0%D0%B2%D0%BB%D0%B5%D0%BD%D0%B8%D0%B5%20%D0%B4%D0%BE%D1%81%D1%82%D1%83%D0%BF%D0%BE%D0%BC.htm?_=1504708101#_управление_доступом)».

При внедрении подсистемы «Свойства» в конфигурацию без подсистемы «Управление доступом» следует внести в конфигурацию изменение: у ролей **БазовыеПрава**, **БазовыеПраваВнешнегоПользователя**,**ИзменениеДополнительныхСведений**, **ЧтениеДополнительныхСведений**у удалить шаблоны и тексты ограничений доступа к объектам метаданных – ПВХ **ДополнительныеРеквизитыИСведения**, регистр сведений **ДополнительныеСведения**.

## Внедрение подсистемы «Свойства» без подсистемы «Взаимодействия»

При внедрении подсистемы «Свойства» в конфигурацию без подсистемы «Взаимодействия» следует внести в конфигурацию изменение – удалить предопределенные элементы справочника **Наборы дополнительных реквизитов и сведений**:

● **Документ\_Встреча**,

● **Документ\_ЗапланированноеВзаимодействие**,

● **Документ\_ТелефонныйЗвонок**,

● **Документ\_ЭлектронноеПисьмоВходящее**,

● **Документ\_ЭлектронноеПисьмоИсходящее**.

## Внедрение подсистемы «Свойства» без подсистемы «Работа с файлами»

При внедрении подсистемы «Свойства» в конфигурацию без подсистемы «Работа с файлами» следует внести в конфигурацию изменение – удалить предопределенные элементы справочника **Наборы дополнительных реквизитов и сведений**:

● **Справочник\_ПапкиФайлов**,

● **Справочник\_Файлы**.

## Внедрение подсистемы «Шаблоны сообщений» без подсистемы «Управление доступом»

По умолчанию для реализации ограничения доступа на уровне записей к шаблонам сообщений в подсистеме [«Шаблоны сообщений»](https://its.1c.ru/db/content/bsp23doc/src/3.59.%20%D0%BE%D1%81%D0%BE%D0%B1%D1%8B%D0%B5%20%D1%81%D0%BB%D1%83%D1%87%D0%B0%D0%B8%20%D0%B2%D0%BD%D0%B5%D0%B4%D1%80%D0%B5%D0%BD%D0%B8%D1%8F%20%D0%BF%D0%BE%D0%B4%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC.htm?_=1504708101#_свойства_2) используются возможности подсистемы [«Управление доступом»](https://its.1c.ru/db/content/bsp23doc/src/3.59.%20%D0%BE%D1%81%D0%BE%D0%B1%D1%8B%D0%B5%20%D1%81%D0%BB%D1%83%D1%87%D0%B0%D0%B8%20%D0%B2%D0%BD%D0%B5%D0%B4%D1%80%D0%B5%D0%BD%D0%B8%D1%8F%20%D0%BF%D0%BE%D0%B4%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC.htm?_=1504708101#_управление_доступом_).

При внедрении подсистемы «Шаблоны сообщений» в конфигурацию без подсистемы «Управление доступом» следует внести в конфигурацию изменение:

● в роли **БазовыеПрава** удалить тексты ограничений доступа к справочникам **ШаблоныСообщений и  ШаблоныСообщенийПрисоединенныеФайлы**.

## Внедрение подсистемы «Шаблоны сообщений» без подсистемы «Присоединенные файлы»

По умолчанию подсистема «[Присоединенные файлы](https://its.1c.ru/db/content/bsp23doc/src/3.48.%20%D1%81%D0%B2%D0%BE%D0%B9%D1%81%D1%82%D0%B2%D0%B0.htm?_=1504708101#_свойства)» расширяет поведение подсистемы «Шаблоны сообщений». Поэтому при внедрении подсистемы «Шаблоны сообщений» без подсистемы «[Присоединенные файлы](https://its.1c.ru/db/content/bsp23doc/src/3.48.%20%D1%81%D0%B2%D0%BE%D0%B9%D1%81%D1%82%D0%B2%D0%B0.htm?_=1504708101#_свойства)» требуется:

● удалить справочник **ШаблоныСообщенийПрисоединенныеФайлы**.

**Глава 4. Программный интерфейс**

Программный интерфейс инструментария разработчика 1С:Библиотека стандартных подсистем включает в себя все экспортные процедуры и функции, которые размещены в областях кода ПрограммныйИнтерфейс. При разработке собственных библиотек и прикладных решений настоятельно рекомендуется использовать только процедуры и функции программного интерфейса библиотеки.

При выпуске новых версий библиотеки в этих процедурах и функциях обеспечивается обратная совместимость, что не требует от прикладных разработчиков пересматривать свой код и адаптировать объекты метаданных своих конфигураций под новые требования и возможности каждой новой версии библиотеки. Это не гарантируется для всех прочих служебных экспортных процедур и функций.

Для удобства поиска и изучения программный интерфейс сгруппирован по подсистемам и разделен на две основные категории:

**Интерфейс** - экспортные процедуры и функции, которые предназначены для вызова из прикладного кода;

**Переопределение** - экспортные процедуры переопределяемых модулей, содержимое которых может или должно быть изменено в конфигурации-потребителе. С их помощью решаются задачи изменения поведения библиотечной функциональности, ее параметризации спецификой конфигурации-потребителя, а также для подключения библиотечной функциональности к объектам конфигурации-потребителя. Они не пред-назначены для вызова из прикладного кода.

Подробнее о программном интерфейсе и переопределяемых модулях [см. Систему стандартов и методик разработки конфигураций для платформы 1С:Предприятие 8](https://its.1c.ru/db/v8std), раздел «Разработка и использование библиотек».

[Адресный классификатор](https://its.1c.ru/db/content/bsp23doc/src/%D0%B0%D0%B4%D1%80%D0%B5%D1%81%D0%BD%D1%8B%D0%B9%20%D0%BA%D0%BB%D0%B0%D1%81%D1%81%D0%B8%D1%84%D0%B8%D0%BA%D0%B0%D1%82%D0%BE%D1%80.htm?_=1504708101)  
[Базовая функциональность](https://its.1c.ru/db/content/bsp23doc/src/%D0%B1%D0%B0%D0%B7%D0%BE%D0%B2%D0%B0%D1%8F%20%D1%84%D1%83%D0%BD%D0%BA%D1%86%D0%B8%D0%BE%D0%BD%D0%B0%D0%BB%D1%8C%D0%BD%D0%BE%D1%81%D1%82%D1%8C.htm?_=1504708101)  
[Банки](https://its.1c.ru/db/content/bsp23doc/src/%D0%B1%D0%B0%D0%BD%D0%BA%D0%B8.htm?_=1504708101)  
[Бизнес-процессы и задачи](https://its.1c.ru/db/content/bsp23doc/src/%D0%B1%D0%B8%D0%B7%D0%BD%D0%B5%D1%81-%D0%BF%D1%80%D0%BE%D1%86%D0%B5%D1%81%D1%81%D1%8B%20%D0%B8%20%D0%B7%D0%B0%D0%B4%D0%B0%D1%87%D0%B8.htm?_=1504708101)  
[Валюты](https://its.1c.ru/db/content/bsp23doc/src/%D0%B2%D0%B0%D0%BB%D1%8E%D1%82%D1%8B.htm?_=1504708101)  
[Варианты отчетов](https://its.1c.ru/db/content/bsp23doc/src/%D0%B2%D0%B0%D1%80%D0%B8%D0%B0%D0%BD%D1%82%D1%8B%20%D0%BE%D1%82%D1%87%D0%B5%D1%82%D0%BE%D0%B2.htm?_=1504708101)  
[Версионирование объектов](https://its.1c.ru/db/content/bsp23doc/src/%D0%B2%D0%B5%D1%80%D1%81%D0%B8%D0%BE%D0%BD%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5%20%D0%BE%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BE%D0%B2.htm?_=1504708101)  
[Взаимодействия](https://its.1c.ru/db/content/bsp23doc/src/%D0%B2%D0%B7%D0%B0%D0%B8%D0%BC%D0%BE%D0%B4%D0%B5%D0%B9%D1%81%D1%82%D0%B2%D0%B8%D1%8F.htm?_=1504708101)  
[Графики работы](https://its.1c.ru/db/content/bsp23doc/src/%D0%B3%D1%80%D0%B0%D1%84%D0%B8%D0%BA%D0%B8%20%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D1%8B.htm?_=1504708101)  
[Групповое изменение объектов](https://its.1c.ru/db/content/bsp23doc/src/%D0%B3%D1%80%D1%83%D0%BF%D0%BF%D0%BE%D0%B2%D0%BE%D0%B5%20%D0%B8%D0%B7%D0%BC%D0%B5%D0%BD%D0%B5%D0%BD%D0%B8%D0%B5%20%D0%BE%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BE%D0%B2.htm?_=1504708101)  
[Даты запрета изменения](https://its.1c.ru/db/content/bsp23doc/src/%D0%B4%D0%B0%D1%82%D1%8B%20%D0%B7%D0%B0%D0%BF%D1%80%D0%B5%D1%82%D0%B0%20%D0%B8%D0%B7%D0%BC%D0%B5%D0%BD%D0%B5%D0%BD%D0%B8%D1%8F.htm?_=1504708101)  
[Дополнительные отчеты и обработки](https://its.1c.ru/db/content/bsp23doc/src/%D0%B4%D0%BE%D0%BF%D0%BE%D0%BB%D0%BD%D0%B8%D1%82%D0%B5%D0%BB%D1%8C%D0%BD%D1%8B%D0%B5%20%D0%BE%D1%82%D1%87%D0%B5%D1%82%D1%8B%20%D0%B8%20%D0%BE%D0%B1%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%BA%D0%B8.htm?_=1504708101)  
[Завершение работы пользователей](https://its.1c.ru/db/content/bsp23doc/src/%D0%B7%D0%B0%D0%B2%D0%B5%D1%80%D1%88%D0%B5%D0%BD%D0%B8%D0%B5%20%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D1%8B%20%D0%BF%D0%BE%D0%BB%D1%8C%D0%B7%D0%BE%D0%B2%D0%B0%D1%82%D0%B5%D0%BB%D0%B5%D0%B9.htm?_=1504708101)  
[Загрузка данных из файла](https://its.1c.ru/db/content/bsp23doc/src/%D0%B7%D0%B0%D0%B3%D1%80%D1%83%D0%B7%D0%BA%D0%B0%20%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85%20%D0%B8%D0%B7%20%D1%84%D0%B0%D0%B9%D0%BB%D0%B0.htm?_=1504708101)  
[Заметки пользователя](https://its.1c.ru/db/content/bsp23doc/src/%D0%B7%D0%B0%D0%BC%D0%B5%D1%82%D0%BA%D0%B8%20%D0%BF%D0%BE%D0%BB%D1%8C%D0%B7%D0%BE%D0%B2%D0%B0%D1%82%D0%B5%D0%BB%D1%8F.htm?_=1504708101)  
[Заполнение объектов](https://its.1c.ru/db/content/bsp23doc/src/%D0%B7%D0%B0%D0%BF%D0%BE%D0%BB%D0%BD%D0%B5%D0%BD%D0%B8%D0%B5%20%D0%BE%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BE%D0%B2.htm?_=1504708101)  
[Запрет редактирования реквизитов объектов](https://its.1c.ru/db/content/bsp23doc/src/%D0%B7%D0%B0%D0%BF%D1%80%D0%B5%D1%82%20%D1%80%D0%B5%D0%B4%D0%B0%D0%BA%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F%20%D1%80%D0%B5%D0%BA%D0%B2%D0%B8%D0%B7%D0%B8%D1%82%D0%BE%D0%B2%20%D0%BE%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BE%D0%B2.htm?_=1504708101)  
[Защита персональных данных](https://its.1c.ru/db/content/bsp23doc/src/%D0%B7%D0%B0%D1%89%D0%B8%D1%82%D0%B0%20%D0%BF%D0%B5%D1%80%D1%81%D0%BE%D0%BD%D0%B0%D0%BB%D1%8C%D0%BD%D1%8B%D1%85%20%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85.htm?_=1504708101)  
[Интеграция c 1С-Бухфон](https://its.1c.ru/db/content/bsp23doc/src/%D0%B8%D0%BD%D1%82%D0%B5%D0%B3%D1%80%D0%B0%D1%86%D0%B8%D1%8F%20c%201%D1%81-%D0%B1%D1%83%D1%85%D1%84%D0%BE%D0%BD.htm?_=1504708101)  
[Информация при запуске](https://its.1c.ru/db/content/bsp23doc/src/%D0%B8%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%86%D0%B8%D1%8F%20%D0%BF%D1%80%D0%B8%20%D0%B7%D0%B0%D0%BF%D1%83%D1%81%D0%BA%D0%B5.htm?_=1504708101)  
[Календарные графики](https://its.1c.ru/db/content/bsp23doc/src/%D0%BA%D0%B0%D0%BB%D0%B5%D0%BD%D0%B4%D0%B0%D1%80%D0%BD%D1%8B%D0%B5%20%D0%B3%D1%80%D0%B0%D1%84%D0%B8%D0%BA%D0%B8.htm?_=1504708101)  
[Контактная информация](https://its.1c.ru/db/content/bsp23doc/src/%D0%BA%D0%BE%D0%BD%D1%82%D0%B0%D0%BA%D1%82%D0%BD%D0%B0%D1%8F%20%D0%B8%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%86%D0%B8%D1%8F.htm?_=1504708101)  
[Напоминания пользователя](https://its.1c.ru/db/content/bsp23doc/src/%D0%BD%D0%B0%D0%BF%D0%BE%D0%BC%D0%B8%D0%BD%D0%B0%D0%BD%D0%B8%D1%8F%20%D0%BF%D0%BE%D0%BB%D1%8C%D0%B7%D0%BE%D0%B2%D0%B0%D1%82%D0%B5%D0%BB%D1%8F.htm?_=1504708101)  
[Настройка порядка элементов](https://its.1c.ru/db/content/bsp23doc/src/%D0%BD%D0%B0%D1%81%D1%82%D1%80%D0%BE%D0%B9%D0%BA%D0%B0%20%D0%BF%D0%BE%D1%80%D1%8F%D0%B4%D0%BA%D0%B0%20%D1%8D%D0%BB%D0%B5%D0%BC%D0%B5%D0%BD%D1%82%D0%BE%D0%B2.htm?_=1504708101)  
[Обмен данными](https://its.1c.ru/db/content/bsp23doc/src/%D0%BE%D0%B1%D0%BC%D0%B5%D0%BD%20%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D0%BC%D0%B8.htm?_=1504708101)  
[Обновление версии ИБ](https://its.1c.ru/db/content/bsp23doc/src/%D0%BE%D0%B1%D0%BD%D0%BE%D0%B2%D0%BB%D0%B5%D0%BD%D0%B8%D0%B5%20%D0%B2%D0%B5%D1%80%D1%81%D0%B8%D0%B8%20%D0%B8%D0%B1.htm?_=1504708101)  
[Обновление конфигурации](https://its.1c.ru/db/content/bsp23doc/src/%D0%BE%D0%B1%D0%BD%D0%BE%D0%B2%D0%BB%D0%B5%D0%BD%D0%B8%D0%B5%20%D0%BA%D0%BE%D0%BD%D1%84%D0%B8%D0%B3%D1%83%D1%80%D0%B0%D1%86%D0%B8%D0%B8.htm?_=1504708101)  
[Отправка SMS](https://its.1c.ru/db/content/bsp23doc/src/%D0%BE%D1%82%D0%BF%D1%80%D0%B0%D0%B2%D0%BA%D0%B0%20sms.htm?_=1504708101)  
[Оценка производительности](https://its.1c.ru/db/content/bsp23doc/src/%D0%BE%D1%86%D0%B5%D0%BD%D0%BA%D0%B0%20%D0%BF%D1%80%D0%BE%D0%B8%D0%B7%D0%B2%D0%BE%D0%B4%D0%B8%D1%82%D0%B5%D0%BB%D1%8C%D0%BD%D0%BE%D1%81%D1%82%D0%B8.htm?_=1504708101)  
[Печать](https://its.1c.ru/db/content/bsp23doc/src/%D0%BF%D0%B5%D1%87%D0%B0%D1%82%D1%8C.htm?_=1504708101)  
[Поиск и удаление дублей](https://its.1c.ru/db/content/bsp23doc/src/%D0%BF%D0%BE%D0%B8%D1%81%D0%BA%20%D0%B8%20%D1%83%D0%B4%D0%B0%D0%BB%D0%B5%D0%BD%D0%B8%D0%B5%20%D0%B4%D1%83%D0%B1%D0%BB%D0%B5%D0%B9.htm?_=1504708101)  
[Полнотекстовый поиск](https://its.1c.ru/db/content/bsp23doc/src/%D0%BF%D0%BE%D0%BB%D0%BD%D0%BE%D1%82%D0%B5%D0%BA%D1%81%D1%82%D0%BE%D0%B2%D1%8B%D0%B9%20%D0%BF%D0%BE%D0%B8%D1%81%D0%BA.htm?_=1504708101)  
[Получение файлов из Интернета](https://its.1c.ru/db/content/bsp23doc/src/%D0%BF%D0%BE%D0%BB%D1%83%D1%87%D0%B5%D0%BD%D0%B8%D0%B5%20%D1%84%D0%B0%D0%B9%D0%BB%D0%BE%D0%B2%20%D0%B8%D0%B7%20%D0%B8%D0%BD%D1%82%D0%B5%D1%80%D0%BD%D0%B5%D1%82%D0%B0.htm?_=1504708101)  
[Пользователи](https://its.1c.ru/db/content/bsp23doc/src/%D0%BF%D0%BE%D0%BB%D1%8C%D0%B7%D0%BE%D0%B2%D0%B0%D1%82%D0%B5%D0%BB%D0%B8.htm?_=1504708101)  
[Префиксация объектов](https://its.1c.ru/db/content/bsp23doc/src/%D0%BF%D1%80%D0%B5%D1%84%D0%B8%D0%BA%D1%81%D0%B0%D1%86%D0%B8%D1%8F%20%D0%BE%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BE%D0%B2.htm?_=1504708101)  
[Присоединенные файлы](https://its.1c.ru/db/content/bsp23doc/src/%D0%BF%D1%80%D0%B8%D1%81%D0%BE%D0%B5%D0%B4%D0%B8%D0%BD%D0%B5%D0%BD%D0%BD%D1%8B%D0%B5%20%D1%84%D0%B0%D0%B9%D0%BB%D1%8B.htm?_=1504708101)  
[Проверка легальности получения обновления](https://its.1c.ru/db/content/bsp23doc/src/%D0%BF%D1%80%D0%BE%D0%B2%D0%B5%D1%80%D0%BA%D0%B0%20%D0%BB%D0%B5%D0%B3%D0%B0%D0%BB%D1%8C%D0%BD%D0%BE%D1%81%D1%82%D0%B8%20%D0%BF%D0%BE%D0%BB%D1%83%D1%87%D0%B5%D0%BD%D0%B8%D1%8F%20%D0%BE%D0%B1%D0%BD%D0%BE%D0%B2%D0%BB%D0%B5%D0%BD%D0%B8%D1%8F.htm?_=1504708101)  
[Работа с контрагентами](https://its.1c.ru/db/content/bsp23doc/src/%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%B0%20%D1%81%20%D0%BA%D0%BE%D0%BD%D1%82%D1%80%D0%B0%D0%B3%D0%B5%D0%BD%D1%82%D0%B0%D0%BC%D0%B8.htm?_=1504708101)  
[Работа с почтовыми сообщениями](https://its.1c.ru/db/content/bsp23doc/src/%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%B0%20%D1%81%20%D0%BF%D0%BE%D1%87%D1%82%D0%BE%D0%B2%D1%8B%D0%BC%D0%B8%20%D1%81%D0%BE%D0%BE%D0%B1%D1%89%D0%B5%D0%BD%D0%B8%D1%8F%D0%BC%D0%B8.htm?_=1504708101)  
[Работа с файлами](https://its.1c.ru/db/content/bsp23doc/src/%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%B0%20%D1%81%20%D1%84%D0%B0%D0%B9%D0%BB%D0%B0%D0%BC%D0%B8.htm?_=1504708101)  
[Рассылка отчетов](https://its.1c.ru/db/content/bsp23doc/src/%D1%80%D0%B0%D1%81%D1%81%D1%8B%D0%BB%D0%BA%D0%B0%20%D0%BE%D1%82%D1%87%D0%B5%D1%82%D0%BE%D0%B2.htm?_=1504708101)  
[Регламентные задания](https://its.1c.ru/db/content/bsp23doc/src/%D1%80%D0%B5%D0%B3%D0%BB%D0%B0%D0%BC%D0%B5%D0%BD%D1%82%D0%BD%D1%8B%D0%B5%20%D0%B7%D0%B0%D0%B4%D0%B0%D0%BD%D0%B8%D1%8F.htm?_=1504708101)  
[Резервное копирование ИБ](https://its.1c.ru/db/content/bsp23doc/src/%D1%80%D0%B5%D0%B7%D0%B5%D1%80%D0%B2%D0%BD%D0%BE%D0%B5%20%D0%BA%D0%BE%D0%BF%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5%20%D0%B8%D0%B1.htm?_=1504708101)  
[Свойства](https://its.1c.ru/db/content/bsp23doc/src/%D1%81%D0%B2%D0%BE%D0%B9%D1%81%D1%82%D0%B2%D0%B0.htm?_=1504708101)  
[Склонение представлений объектов](https://its.1c.ru/db/content/bsp23doc/src/%D1%81%D0%BA%D0%BB%D0%BE%D0%BD%D0%B5%D0%BD%D0%B8%D0%B5%20%D0%BF%D1%80%D0%B5%D0%B4%D1%81%D1%82%D0%B0%D0%B2%D0%BB%D0%B5%D0%BD%D0%B8%D0%B9%20%D0%BE%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BE%D0%B2.htm?_=1504708101)  
[Структура подчиненности](https://its.1c.ru/db/content/bsp23doc/src/%D1%81%D1%82%D1%80%D1%83%D0%BA%D1%82%D1%83%D1%80%D0%B0%20%D0%BF%D0%BE%D0%B4%D1%87%D0%B8%D0%BD%D0%B5%D0%BD%D0%BD%D0%BE%D1%81%D1%82%D0%B8.htm?_=1504708101)  
[Текущие дела](https://its.1c.ru/db/content/bsp23doc/src/%D1%82%D0%B5%D0%BA%D1%83%D1%89%D0%B8%D0%B5%20%D0%B4%D0%B5%D0%BB%D0%B0.htm?_=1504708101)  
[Удаление помеченных объектов](https://its.1c.ru/db/content/bsp23doc/src/%D1%83%D0%B4%D0%B0%D0%BB%D0%B5%D0%BD%D0%B8%D0%B5%20%D0%BF%D0%BE%D0%BC%D0%B5%D1%87%D0%B5%D0%BD%D0%BD%D1%8B%D1%85%20%D0%BE%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BE%D0%B2.htm?_=1504708101)  
[Управление доступом](https://its.1c.ru/db/content/bsp23doc/src/%D1%83%D0%BF%D1%80%D0%B0%D0%B2%D0%BB%D0%B5%D0%BD%D0%B8%D0%B5%20%D0%B4%D0%BE%D1%81%D1%82%D1%83%D0%BF%D0%BE%D0%BC.htm?_=1504708101)  
[Файловые функции](https://its.1c.ru/db/content/bsp23doc/src/%D1%84%D0%B0%D0%B9%D0%BB%D0%BE%D0%B2%D1%8B%D0%B5%20%D1%84%D1%83%D0%BD%D0%BA%D1%86%D0%B8%D0%B8.htm?_=1504708101)  
[Центр мониторинга](https://its.1c.ru/db/content/bsp23doc/src/%D1%86%D0%B5%D0%BD%D1%82%D1%80%20%D0%BC%D0%BE%D0%BD%D0%B8%D1%82%D0%BE%D1%80%D0%B8%D0%BD%D0%B3%D0%B0.htm?_=1504708101)  
[Шаблоны сообщений](https://its.1c.ru/db/content/bsp23doc/src/%D1%88%D0%B0%D0%B1%D0%BB%D0%BE%D0%BD%D1%8B%20%D1%81%D0%BE%D0%BE%D0%B1%D1%89%D0%B5%D0%BD%D0%B8%D0%B9.htm?_=1504708101)  
[Электронная подпись](https://its.1c.ru/db/content/bsp23doc/src/%D1%8D%D0%BB%D0%B5%D0%BA%D1%82%D1%80%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F%20%D0%BF%D0%BE%D0%B4%D0%BF%D0%B8%D1%81%D1%8C.htm?_=1504708101)

# Адресный классификатор

[Интерфейс](https://its.1c.ru/db/content/bsp23doc/src/%D0%B8%D0%BD%D1%82%D0%B5%D1%80%D1%84%D0%B5%D0%B9%D1%81.htm?_=1504708101)

# ЗагрузитьАдресныйКлассификатор

Открывает форму загрузки классификатора.

**Синтаксис**

Процедура ЗагрузитьАдресныйКлассификатор(Параметры = Неопределено) Экспорт

**Параметры**

Параметры - ПараметрыВыполненияКоманды, Структура - параметры открытия формы.

**Пример вызова**

АдресныйКлассификаторКлиент.ЗагрузитьАдресныйКлассификатор(Параметры)

**Доступность**

Тонкий клиент, Толстый клиент