**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, БазовыеПраваБСП 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 ПриДобавленииОбработчиковУстановкиПараметровСеанса общего модуля ОбщегоНазначенияПереопределяемый. При этом обработчик инициализации должен принимать два параметра:

● ИмяПараметра – строка – имя инициализируемого параметра;

● УстановленныеПараметры – массив – имена параметров, которые были инициализированы.

**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/bsp301doc/src/3.57.%20удаление%20помеченных%20объектов.htm?_=1542292879#_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.

В случаях, когда конфигурация поддерживает работу в модели сервиса и есть объекты, видимость которых должна зависеть от текущего режима работы программы, то их необходимо включить в состав следующих функциональных опций:

● РаботаВАвтономномРежиме

● РаботаВЛокальномРежиме

● РаботаВМоделиСервиса

Подробнее см. в разделе «[Работа в модели сервиса](https://its.1c.ru/db/content/bsp301doc/src/3.47.%20работа%20в%20модели%20сервиса.htm?_=1542292879#_работа_в_модели)».

**Использование при разработке конфигурации**

Программный интерфейс подсистемы описан в соответствующем разделе главы 4 «[Программный интерфейс](https://its.1c.ru/db/content/bsp301doc/src/глава%204.%20программный%20интерфейс.htm?_=1542292879)».

**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 и отладке 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 ОбновлениеВспомогательныхДанных.epf, которая входит в состав дистрибутива библиотеки.

● 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.

Такие случаи специально отмечены в тексте документации соответствующих подсистем и выделены префиксом Внимание.

В остальных случаях обновление вспомогательных данных выполняется автоматически при каждом изменении номера версии конфигурации – в процессе обновления или первоначального заполнения информационной базы.

**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 функций ИдентификаторОбъектаМетаданных и ИдентификаторыОбъектовМетаданных общего модуля ОбщегоНазначения.

Внимание. 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 справочника ИдентификаторыОбъектовМетаданных можно также обновлять вручную, без увеличения номера версии конфигурации, с помощью обработки ОбновлениеВспомогательныхДанных.epf, которая входит в состав дистрибутива библиотеки (см. раздел «[Обновление вспомогательных данных во время разработки](https://its.1c.ru/db/content/bsp301doc/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?_=1542292879#_обновление_вспомогательных_данных)»). 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.

Рассогласования идентификаторов объектов метаданных с объектами метаданных конфигурации также 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. | БазовыеПраваБСП  Роль, предоставляющая доступ к тем объектам информационной базы, которые должны быть всегда доступны всем пользователям.  Назначается всем пользователям |
| 2. | БазовыеПраваВнешнихПользователейБСП  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, БазовыеПраваБСП и БазовыеПраваВнешнихПользователейБСП).

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/bsp301doc/src/3.58.%20управление%20доступом.htm?_=1542292879#_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/bsp301doc/src/3.57.%20удаление%20помеченных%20объектов.htm?_=1542292879#_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,  ● БазовыеПраваБСП,  ● ДобавлениеИзменениеАдресныхСведений,  ● ДобавлениеИзменениеБанков,  ● ДобавлениеИзменениеКалендарныхГрафиков,  ● ДобавлениеИзменениеДополнительныхРеквизитовИСведений,  ● ДобавлениеИзменениеКурсовВалют,  ● ДобавлениеИзменениеГрафиковРаботы,  ● ДобавлениеИзменениеВидовКонтактнойИнформации,  ● <Раздел нормативно-справочная информация> |

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

При [поиске и удалении дублей](https://its.1c.ru/db/content/bsp301doc/src/3.40.%20поиск%20и%20удаление%20дублей.htm?_=1542292879#_поиск_и_удаление), программном вызове ЗаменитьСсылки общего модуля ОбщегоНазначения (и в любых других случаях замены ссылок в документах), 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.

**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 АдресПубликацииИнформационнойБазыВИнтернете,

● константа АдресПубликацииИнформационнойБазыВЛокальнойСети,

● константа ГлавныйУзел,

● константа ЗаголовокСистемы,

● константа ИдентификаторИнформационнойБазы,

● константа ИспользоватьРазделениеПоОбластямДанных,

● константа НеИспользоватьРазделениеПоОбластямДанных,

● константа ЭтоАвтономноеРабочееМесто,

● справочник ВерсииРасширений,

● справочник ИдентификаторыОбъектовРасширений,

● регистр сведений БезопасноеХранилищеДанных,

● регистр сведений БезопасноеХранилищеДанныхОбластейДанных,

● регистр сведений ЗапросыРазрешенийНаИспользованиеВнешнихРесурсов,

● регистр сведений ИдентификаторыОбъектовВерсийРасширений,

● регистр сведений ПараметрыРаботыВерсийРасширений,

● регистр сведений СеансыВерсийРасширений,

● регистр сведений КэшПрограммныхИнтерфейсов.

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/bsp301doc/src/3.30.%20обмен%20данными.htm?_=1542292879#_особенности_создания_начального)»):

● регистр сведений ПараметрыРаботыПрограммы.

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

● справочник ВерсииРасширений,

● справочник ИдентификаторыОбъектовМетаданных,

● справочник ИдентификаторыОбъектовРасширений,

● регистр сведений ПараметрыРаботыПрограммы,

● 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.