
G3AppAdministrationGuide

Выпуск 0.0.12

"Бизнес Технологии" ООО

июн. 20, 2024

Содержание

1	Предисловие	2
1.1	На кого ориентировано руководство	2
2	Введение	2
2.1	Пользователь	2
2.2	Администрируемый объект	2
2.3	Роль	7
2.4	Профиль пользователя	7
2.5	Замещение прав	7
2.6	Индексация привилегий пользователей	7
2.7	Предустановленные профили	7
3	Приложение администратор	8
3.1	Список пользователей	8
3.2	Карточка пользователя	8
3.3	Детализация пользователя	8
3.4	Список ролей	10
3.5	Карточка роли	10
3.6	Групповая настройка привилегий ролей	12
3.7	Индексация привилегий для пользователей	14
3.8	Аудит прав доступа	14
3.9	Трассировка прав доступа	16
4	Дискретный доступ	17
4.1	Настройка привилегии	17
4.2	Примеры скриптов	22
5	Приложение 1	28
5.1	Функции отображения	28
5.2	Функции проверки привилегий	28
6	Связь выборок с администрируемыми объектами	29
6.1	основные понятия	29
6.2	Принцип построения адм. объектов	30
6.3	Определение связи выборки и адм. объекта	31

1 Предисловие

Добро пожаловать в Руководство администратора GlobalFramework. Цель руководства предоставить информацию о принципах и способах администрирования решений на основе GlobalFramework.

1.1 На кого ориентировано руководство

Руководство ориентировано на:

- Разработчиков системы
- Аналитиков, настраивающих доступ к системе на проектах

2 Введение

Действие пользователя в системе требует наличия у него тех или иных привилегий. Администрирование доступа производится путем выдачи пользователям прав на **объектные привилегии** или **элементарные привилегии**.

2.1 Пользователь

Для работы в системе необходимо зайти под каким-либо пользователем. С точки зрения системы пользователем является учетная запись, на которую выдается перечень прав, дающий возможность юридическому или физическому лицу выполнять действия в системе.

Список пользователей можно открыть из пункта меню:

- Администратор \ Доступ > Пользователи

Отключение администрирования

В случае если у пользователя стоит признак **Супер-пользователь**, настройки администрирования не применяются, под таким пользователем можно совершать в системе любые действия. Данный признак удобно использовать для разработки и тестирования, так как это позволяет отделить администрирование от разработки.

2.2 Администрируемый объект

Администрируемый объект - это логически связанный для массовой раздачи прав набор привилегий. Административный объект генерируется по объектам системы и позволяет единым образом выдавать права на них.

Администрируемый объект генерируется в момент установки или обновления системы по таким объектам как:

- Произвольная выборка
- Класс
 - Справочник
 - Настройка

- Журнал
- Документы
- Пакеты
- Приложения

Список административных объектов можно открыть из пункта меню:

- Администратор \ Настройки > Администрируемые объекты

Отключение администрирования

Возможно отключить администрирование в разрезе административного объекта, данным механизмом удобно пользоваться в случае поэтапной настройки прав, когда администрирование включается только для тех объектов, на которых оно уже настроено. Для отключения администрирования на административном объекте существуют следующие поля:

- **Не распространяются настройки администрирования**
Если флаг установлен, на объекты представляющие состав административных объектов не распространяются настройки администрирования
- **Не требуется настройка прав доступа на состояния**
Если флаг установлен, на документе данного административного объекта выключается контроль доступных переходов между состояниями.

Узел администрирования

Группирует элементы администрируемого объекта для массовой раздачи прав, по умолчанию создается одна основная группа на каждый администрируемый объект.

Узел администрирования зарезервирован на случай, если в системе потребуется настраивать права в зависимости от **типа объекта** или **состояния**

Элементы администрируемого объекта

Элемент административного объекта создается на объекты которые требуют раздачи прав на элементарные привилегии.

Выборка

По умолчанию выборка со всеми отображениями регистрируется в один **элемент администрируемого объекта** с именем {SelName}#Default. При этом на каждую операцию и атрибут в разрезе системного имени создается **элементарная привилегия**.

Для переноса отображения в отдельный **элемент администрируемого объекта** существует аннотация:

```
@AcItemRep(name = "{Name}")
```

В таком случае элемент администрируемого объекта получит имя {SelName}#{Name} Так же элемент администрируемого объекта можно задать в avm.xml через атрибут отображения acItemRep

Примечание: Изменения вступят в силу после обновления административного объекта (см. главу Групповая настройка привилегий ролей)

Для создания элементов администрируемого объекта по выборкам, не имеющим класса, в `avm` файле обязательно должен быть тег `<acObject/>`. Для выборок без класса формируется администрируемый объект и элемент администрируемого объекта с суффиксом `Avi` в имени.

Дискретные привилегии

Дискретные привилегии используют правила дискретного доступа при проверке прав, что позволяет раздавать доступ по объектно (в зависимости от значений атрибутов строки). В интерфейсе настройки ролей дискретные привилегии можно выдать с ограничением по дискретному правилу.

Элементарные привилегии

Элементарные привилегии используются для разграничения действий пользователя и создаются следующим образом:

- по выборке на:
 - Операции
Создаются дискретные привилегии
 - Атрибуты
 - Сеттеры атрибута
Создаются дискретные привилегии

Типы элементарных привилегий

- Чтение
Содержит настройки видимости атрибутов.
- Редактирование
Содержит сеттеры атрибутов.
- Добавление
Содержит операции вставки и копирования.
- Удаление
Содержит операция удаления.
- Интерактивные права
Содержит остальные операции.

Для изменения типа привилегии, которой соответствует операция выборки, существует аннотация `@AcPrivilegeType`. Пример:

```
@AcPrivilegeType(value = PrivilegeTypes.Read)
def myOperation(): Unit = {}
```

Примечание: Изменения вступят в силу после обновления административного объекта (см. главу Групповая настройка привилегий ролей)

Объектные привилегии

Объектные привилегии позволяют раздавать права на бизнес логику.

Авто генерируемые объектные привилегии

- edit#
Создаются дискретные привилегии
- view#
Создаются дискретные привилегии
- viewReport#
Создаются дискретные привилегии

Привилегии на переход состояний

Привилегии на переход состояний ограничивают доступные переходы состояния. Ограничения на переходы реализовано в стандартном выпадающем списке для состояний `ru.bitec.app.btk.Btk_ClassStateAvi.Lookup_Class`

Имя привилегии формируется по правилу:

```
s"StateChange_OT:${idObjectType}_SS:${idStartState}_FS:${idFinishState}"
```

Функция для определения возможности перехода: `ru.bitec.app.btk.Btk_AdminPkg#hasStateChangePriv.`

Ручное создание объектных привилегий

Для ручного создания объектных привилегий на классе(`odm.xml`) или пакете(`pkg.xml`) используются xml разметка, пример:

```
<admin>
  <privileges>
    <privilege name="SbtManage" caption="Управление решением"/>
  </privileges>
</admin>
```

Примечание: Изменения вступят в силу после обновления административного объекта (см. главу Групповая настройка привилегий ролей)

Приложение

Для администрирования приложений создается адм. объект **Приложения**(Btk_ApplicationsListPkg). Объектные привилегии импортируются в него из xml файлов META-INF/applications.xml всех подключенных модулей.

Выданные права на такую привилегию обеспечивают отображение приложения в списке в меню выбора приложений.

Дискретные ограничения

Дискретные ограничения позволяют задавать выражения для создания правил выдачи дискретных привилегий на конкретной роли. Подробности смотрите в главе **Дискретный доступ** Принцип настройки дискретных ограничений:

1. Администратор определяет дискретные ограничения на административном объекте
2. Администратор определяет дискретные правила в роли
Каждое правило содержит дискретное ограничение и набор параметров
3. Администратор выдает грант на дискретную привилегию роли
При этом при синхронизации пользователей проходит расчет дискретных привилегий данных пользователю по алгоритму объединения грантов

При проверки привилегии в бизнес логике, в функцию проверки передается перечень строк, для которых выполняется проверка. При ограничении запроса по дискретной привилегии, каждое ограничение добавляется в where условие.

Алгоритм объединение грантов:

1. Пусть g_1 и g_2 гранты пользователю на объектную привилегию p
2. Если g_1 или g_2 с типом полный доступ, вернуть грант на p с полным доступом
3. Иначе вернуть грант на p объединив ограничения

Алгоритм объединения ограничений:

1. Пусть A массив ограничений для грантов g_1 и g_2
2. Пусть S пустое множество ограничений(параметры ограничения не входят в ключ множества)
3. Для каждой g из A
 1. Если g существует в S , добавить новые параметры к ограничению в S
 2. Иначе добавить g к S
4. Вернуть S

Скрипт фильтрации

Используется в универсальной фильтрации для ограничения видимости по объектам.

Скрипт проверки объектного кэша

Используется в автономной бизнес логики для проверки возможности действия со строкой.

2.3 Роль

Роль - административная единица для выдачи прав. Для раздачи дискретных привилегий на роли могут быть настроены дискретные правила, которые определяют дискретные ограничения и передаваемые в них на гранте параметры.

В роле выдаются гранты на привилегии.

Список ролей можно открыть из пункта меню:

- Администратор \ Доступ > Роли

2.4 Профиль пользователя

Профили пользователя используется для группировки ролей для более удобной раздачи их пользователю.

Список профилей можно открыть из пункта меню:

- Администратор \ Доступ > Профили

2.5 Замещение прав

Позволяет указать какой пользователь будет обладать всеми правами другого и на какое время. Настройки замещения прав учитываются при выполнении индексации прав пользователя. Для отслеживания окончания периода действия замещения каждую ночь выполняется задание, которые выполняет индексацию пользователей.

Список замещений можно открыть из пункта меню:

- Администратор \ Доступ > Замещение прав

2.6 Индексация привилегий пользователей

Для быстрой проверки привилегий пользователя используются **индексы привилегий**. Индексы привилегий необходимо пересчитывать после завершения редактирования ролей и их выдаче пользователю.

2.7 Предустановленные профили

Профиль «Отладчик»

Профиль "Отладчик" предназначен для пользователей, для которых нужно предоставить доступ к инструментам отладки. Таким как:

- доступ к операциям управления решением (управление SBT)
- возможность открыть дебаггер
- выполнение jexl скриптов в дебагере
- доступ к инструментам меню Сервиса

- доступ к инструментам отладки клиента

Профиль «Разработчик»

Профиль "Разработчик" предназначен для пользователей, которые являются разработчиками. Включает в себя все права профиля «Отладчик». А также имеет доступ к следующим инструментам:

- доступ к редактированию справки

3 Приложение администратор

Приложение Администратор используется для администрирования пользователей.

3.1 Список пользователей

Список пользователей можно открыть из пункта меню:

- Администратор \ Доступ > Пользователи

Операции списка пользователей:

- **Пересчитать индексацию привилегий для текущего пользователя**
Операция предназначена для обновления индексов системных привилегий для текущего пользователя. В результате операции обновятся записи в индексе(Btk_AcUserItemPrivFlat) по всем элементам администрируемых объектов на привилегии, которых у пользователя есть права, а также записи в индексе(Btk_AcUserObjPrivFlat) по всем администрируемым объектам, на объектные права которых у пользователя есть права.
- **Пересчитать индексацию привилегий для всех пользователей**
Операция предназначена для обновления индексов системных привилегий для всех пользователей.

3.2 Карточка пользователя

Операции карточки пользователя:

- **Изменить пароль**
Операция предназначена для установки пароля для входа в систему для пользователя.
- **Пересчитать индексацию привилегий для текущего пользователя**
Операция предназначена для обновления индексов привилегий для текущего пользователя.

3.3 Детализация пользователя

В карточке и списке доступна детализация, содержащая закладки:

- **Профили доступа**
Список записей в классе-развязке(Btk_AcUserGrant) со ссылкой на профиль. Для закладки **Профили доступа** доступна детализация с закладкой **Роли доступа профиля**, отображающая список записей в классе-развязке(Btk_AcProfileGrant) со ссылкой на роль
- **Роли доступа**
Список ролей, связанных с пользователем через профиль

- **Индексация элементарных привилегий**
Список записей в `Btk_AcUserItemPrivFlat` для текущего пользователя.
- **Индексация объектных привилегий**
Список записей в `Btk_AcUserObjPrivFlat` для текущего пользователя.

Для закладок **Индексация элементарных привилегий** и **Индексация объектных привилегий** доступна операция

- **Посмотреть значение индекса**
Операция открывает в модальном окне содержимое поля(`cPriv`) с перечнем доступных привилегий для элемента или объекта.

Синхронизация с Active Directory

Для синхронизации:

1. Выполните операцию **Синхронизация с Active Directory**
Операция доступна в списке пользователей. При этом откроется интерфейс синхронизации отображающий список логов.
2. При необходимости укажите шаблон в шапке интерфейса
Вновь созданные пользователи будут созданы по этому шаблону
3. Нажмите кнопку **Синхронизировать**
При этом будет просканирован домен на наличие групп и пользователей. Будут созданы только группы которые настроены как основная группа пользователей. Произойдет синхронизация пользователей

Синхронизируемы поля пользователей:

- **Системное имя**
Атрибут `sAMAccountName`
- **Пароль**
Равен системному имени при создании пользователя
- **Группа**
Определяется через атрибут `primaryGroupID`
- **Для физ. лица:**
 - **Фамилия**
атрибут `sn`
 - **Имя**
1-е слово из атрибута `givenName`
 - **отчество**
Если атрибут `givenName` содержит 2 слова, то 2-ое слово – отчество.
 - **email**
Атрибут `mail`

Синхронизируемые поля групп:

- **Наименование**
Атрибут «`name`», если он не указан, то полное имя. Например: `CN=Domain users,CN=Users`
- **Системное имя**
Полное имя группы. Например: `CN=Domain users,CN=Users`

Пользователи создаются только в случае, если они не заблокированы в домене. Если пользователь заблокирован, и уже зарегистрирован в системе Global, то ему выставляется признак «Заблокирован».

Настройка синхронизации

Для настройки подключения используется универсальный класс настроек (Btk_Setting). Открыть настройки можно из пункта меню **Настройка системы \ Настройка > Общие настройки системы**

Необходимые параметры:

- url
сервера контроллера домена.
Например: `ldap://192.168.2.11:389`
- логин
- пароль
- домен
который требуется синхронизировать.

3.4 Список ролей

Список ролей можно открыть из пункта меню:

- Администратор \ Доступ > Роли

3.5 Карточка роли

Используется для выдачи привилегий для роли от дерева административных объектов.

Права роли

В данном интерфейсе отображается дерево административных объектов с уровнями:

- Группа администрируемых объектов(только чтение)
 - Объект администрирования
 - * Узел администрирования
 - Элемент администрируемого объекта

Совет: По умолчанию отображаются только административные объекты использующиеся в роли. Для отображения всех административных объектов выключите флаг **Только объекты, на которые у роли имеются права** в панели фильтрации.

В данной интерфейсе каждый тип элементарной привилегии отображен отдельным столбцом, что позволяет выдать права массово по типу для всех элементарных привилегий этого типа для текущего уровня и уровней ниже.

Расшифровка прав

В случае если выбран объект администрирования отображает:

- перечень типов элементарных привилегий
- перечень объектных привилегий

В случае если выбран элемент администрируемого объекта отображает:

- перечень элементарных привилегий для данной строки

Перевод состояний

Отображается в случае если выбран объект администрирования. Позволяет настроить возможные переходы между состояниями.

Выдача прав на объектные привилегии

1. Выберите требуемый адм. объект.
При этом в расшифровке прав отобразятся объектные привилегии
2. Раздайте права на объектные привилегии роли \
3. Обновите индексы системных привилегий

Выдача прав на тип элементарных привилегий

В случае если выдано право на тип элементарной привилегии, доступ будет дан всем элементарным привилегиям данного типа во всех элементах администрируемого объекта.

1. Выберите требуемый адм. объект
При этом в расшифровке прав отобразятся типы элементарных привилегий.
2. Раздайте права на типы элементарных привилегий
Права на типы можно раздавать либо в расшифровке пара либо в дереве административных объектов
3. Обновите индексы системных привилегий

Выдача прав на элементарную привилегию

1. Выберите требуемый элемент администрируемого объекта
При этом в расшифровке прав отобразятся типы элементарных привилегий.
2. Раздайте права на типы элементарных привилегий
Для этого в расшифровке прав выберите нужный элемент и предоставьте к нему доступ.
3. Обновите индексы системных привилегий

Примечание: Если установить признак «Запрещено», то пользователи, обладающие этой ролью, не будут иметь прав на запрещенную привилегию, даже если другие роли дают на нее доступ.

Выдача прав на вызов Api и Pkg методов в jexl-контексте

Примечание: Для того, чтобы при выполнении jexl-скриптов учитывались настроенные права, необходимо включить в общих настройках модулей btk - «Выполнение jexl-скриптов через безопасный диалект».

1. В карточке Роли на закладке Права вызовов через jexl-script слева отображен список Api и Pkg классов проекта. Справа - их методы.
2. Для выбора/снятия доступа ко всем методом класса используйте чек-бокс Доступны все методы, в списке методов можно настроить доступ к конкретному методу класса.
3. Обновите индексы системных привилегий

Выдача прав на использование Rest-пакетов

1. В карточке Роли на закладке Права на Rest-пакеты отображен список пакетов. Чекбокс Rest-пакет указывает, является ли пакет Rest-пакетом.
2. Для выдачи\отзыва доступа используйте чекбокс Имеет доступ.
3. Обновите индексы системных привилегий по роли.

Примечание: Обновление списка Rest-пакетов происходит при обновлении адм.объектов по модулю. С пакетов, которые перестали быть Rest-пакетами, чекбокс Rest-пакет автоматически снимается.

3.6 Групповая настройка привилегий ролей

Выборка Групповая настройка привилегий ролей предназначена для массовой раздачи привилегий по ролям от административного объекта.

Выборку можно открыть из пункта меню:

- Администратор \ Доступ > Групповая настройка привилегий ролей

Общий принцип работы с выборкой:

1. Выберите требуемый административный объект
2. Выберите требуемые роли
Для этого в панели фильтрации привилегий откройте список подбора для поля Перечень ролей
3. Проставьте гранты на требуемых привилегиях
4. Пересчитайте индексы привилегий
Для этого выполните операцию Пересчитать индексацию привилегий в интерфейсе Перечень ролей

Дерева администрируемых объектов

Дерево администрируемых объектов состоит из следующих уровней:

- Группа администрируемых объектов (Опционально)
 - Объект администрирования
 - * Узел администрирования
 - Элемент администрируемого объекта

Объекты администрирования могут не входить в группу, тогда они будут считаться корневыми записями.

Операции дерева администрируемых объектов

- Обновить выбранный объект
Операция предназначена для обновления выделенного объекта или элемента адм. объекта. При обновлении административного объекта происходит сканирование исходного кода по которому добавляются недостающие привилегии и элементы.
- Обновить адм. объект по имени
При этом произойдет запрос имени и произойдет обновление административного объекта по выбранному имени
- Обновить адм. объекты модуля
Операция предназначена для обновления адм. объектов по всему модулю

Примечание: Классы-коллекции не имеют собственных объектов администрирования и входят в объект мастера. Поэтому для обновления настроек для коллекций необходимо обновлять адм. объект по имени класса мастера.

Настройка прав

Закладка отображает настройку прав для ролей в зависимости от выделенной записи в дереве администрируемых объектов:

1. Для административного объекта:
 - объектные привилегии
1. Для элемента администрирования:
 - типы элементарных привилегий
 - элементарные привилегии

Колонки ролей в настройке прав формируются динамически по ролям из фильтра.

Операции настройки прав:

- **Подобрать роли**
Операция предназначена для подбора в фильтр ролей, для которых есть какие-либо права для текущего объекта или элемента.

Пользователи

Отображаются пользователи для роли в активной ячейки.

Профили

Отображаются профили для роли в активной ячейки.

3.7 Индексация привилегий для пользователей

Индексация пользователя

Для пересчета индекса по 1-му пользователю существует операция **Пересчитать индексацию привилегий для текущего пользователя**. Операция доступна:

- в списке и карточке **Пользователей**
- на закладке **Пользователи** роли в списке и карточке **Ролей**.

Индексация всех пользователей

Для индексации привилегий всех пользователей предназначена операция **Пересчитать индексацию привилегий для всех пользователей**. Операция доступна:

- в списке **Пользователей**.

3.8 Аудит прав доступа

Историю изменений прав доступа можно посмотреть в **Аудит прав доступа**, находящемся в пункте меню **Аудит**.

Примечание: Для включения аудита изменений прав доступа необходимо выполнить операцию **Включить аудит для структур прав доступа** в этом меню.

В самом типе необходимо выбрать **Вид объекта** (Пользователь, Профиль или Роль), **Объект** и, если необходимо, период времени.

Пользователи

Для пользователей в аудите отображаются **Профили**, которые были выданы пользователю или отобраны у него, дата изменения и изменивший пользователь.

Профили

Для профилей в аудите отображаются **Роли**, которые включали в профиль или убрали из него, и **Пользователи**, которым выдавался профиль или его отбирали, а также дата изменения и изменивший пользователь.

Примечание: В списке изменений для **Пользователей** и **Профилей** есть операция **Изменения по объекту**, которая позволяет перейти к аудиту другого объекта, который является выделенным.

Роли

Для ролей список изменений представляет собой две выборки:

- **Дерево администрируемых объектов**
Отображаются администрируемые объекты или элементы администрирования, по которым для данной роли были совершены изменения
- **Список привилегий конкретного администрируемого объекта**
Список привилегий конкретного адм. объекта или элемента администрирования, которые изменялись для данной роли

Список привилегий представляет из себя:

1. Имя адм. объекта или элемента администрирования
2. Тип привилегии:
 - Изменение атрибута (например, при простановке **Не распространяются настройки администрирования** для адм. объекта),
 - Объектная привилегия,
 - Элементарная привилегия,
 - Группа элементарных привилегий (когда изменяются доступ не на конкретные элементарные привилегии, а на всю группу, например, Редактирование)
 - Перевод состояния
1. Системное имя привилегии
2. Привилегия - наименование привилегии
3. Изменивший пользователь
4. Дата изменения
5. Системное имя атрибута
6. Наименование атрибута
7. Старое значение - старое значение атрибута
8. Новое значение - новое значение атрибута

Примечание: Для списка привилегий есть фильтр **С входящими**, который в случае, когда в дереве адм. объектов выбран адм. объект, а не элемент администрирования, отображает изменения не только администрируемого объекта, но и элементов администрирования, входящих в этот объект.

3.9 Трассировка прав доступа

Запуск трассировки прав пользователя

Для запуска трассировки прав пользователя, выполните следующие шаги:

- Перейдите в раздел «Сервис» в меню системы.
- В меню «Инструменты» найдите опцию «Начать трассировку прав» и выберите её.

После начала трассировки, система будет отслеживать все ваши действия. Выполняйте необходимые действия с данными в системе, так как вам требуется.

Чтобы завершить трассировку прав пользователя, выполните следующие шаги:

- Перейдите снова в раздел «Сервис» в меню системы.
- В меню «Инструменты» выберите опцию «Закончить трассировку прав».

Отчет сессии трассировки прав

Отчет «сессии трассировки прав пользователей» выведен в приложении «Администратор», в меню «Отчеты». Отображает список сессий трассировки. При открытии карточки сессии в детализации будет отображена более подробная информация по проверкам прав, которые вызывались в ходе сессии трассировки.

Очистка записей

Сессии трассировки сохраняются в таблицу `Btk_AcTraceSession`, а детализация сессии в `Btk_AcTraceJournal`. Для очистки этих таблиц предусмотрен по умолчанию регистрируется задание очистки данных по классу «`Btk_AcTraceSession`», удаляющий записи, хранящиеся больше 30 дней.

Реализация трассировки

При запуске трассировки

- в параметрах рабочей сессии сохраняется параметр активности трассировки;
- создается новая запись в `Btk_AcTraceSession` в логирующей транзакции;
- в параметрах рабочей сессии сохраняется параметр с идентификатором запущенной сессии трассировки;
- для уже открытых выборок подключается подписка на события, так же она подключается на `onLoadAdminMeta` при открытии новой выборки; При действиях пользователя во время трассировки
- на методах проверки прав доступа, вызов операции/сеттера на администрируемых выборках, создается запись в логирующей транзакции в `Btk_AcTraceJournal`; При завершении трассировки
- вызывается коммит логирующих транзакций со всех запущенных сессий выборок;

- в сессию трассировки Btk_AcTraceSession сохраняется дата завершения;
- отключается подписка на события, подключенная для логирования;
- из параметров рабочей сессии удаляются параметр активности трассировки и идентификатор сессии трассировки.

4 Дискретный доступ

Дискретный доступ позволяет выдавать привилегии в рамках объектов классов или записей в администрируемой выборке.

4.1 Настройка привилегии

Административный объект

1. Открыть карточку Адм. объекта
2. Установить признак Дискретный доступ
3. Перейти на закладку Дискретные ограничения доступа
4. Создать Дискретное ограничение.

Дискретное ограничение доступа

1. Создать новую запись или встать на запись для редактирования
2. Указать уникальное в рамках адм. объекта имя и наименование
3. Настроить тип правила
4. Настроить параметры правила.
5. На закладке Скрипт для фильтрации объектных привилегий написать sql-запрос, который будет фильтровать списки объектов
6. На закладке Скрипт проверки строк по объектному кешу написать jexl-скрипт, который будет по гор-у объекта или строке выборки возвращать признак, что условие выполнилось для этого объекта.

Тип правила

Доступны 3 варианта типа правила:

- Примитивное
- Составное
- Без параметров

Примитивное правило

Состоит из одного параметра.

При анализе прав пользователя со всех его ролей будут собраны значения параметров и собраны в один массив примитивных значений. Например для числового правила будет собран массив вида: [10, 20, 30]

Составное правило

Состоит из нескольких параметров.

При анализе прав пользователя используется следующий алгоритм:

1. Для каждого набора значений, указанных на роли, формируется json-объект вида:

```
{
  <ИД параметра 1>: [<Массив указанных значений>],
  <ИД параметра 2>: [<Массив указанных значений>],
  ...
  <ИД параметра n>: [<Массив указанных значений>]
}
```

2. Каждая роль может обладать несколькими наборами значений на одно правило. По этому результатом агрегации настроенных значений на роли будет массив, содержащий объекты, указанные в п.1:

```
[
  {json-объект для набора 1},
  {json-объект для набора 2},
  ...
  {json-объект для набора n},
]
```

3. При агрегации прав пользователя:

- со всех ролей анализируются массивы, описанные в п.2
- в json-е заменяются ИД параметров на их имена
- каждый элемент массива приводится к строке
- собирается результирующий массив, содержащий уникальные строки.

Таким образом исключаются дублирующие настройки на ролях.

Пример агрегации прав пользователя:

```
[
  {
    "paramName1": ["a", "b", "c"],
    "paramName2": [1, 2, 3]
  },
  {
    "paramName1": ["c", "d", "e"],
    "paramName2": [3, 4, 5]
  }
]
```

(continues on next page)

```
}
]
```

Правило без параметров

Не содержит параметров.

При анализе прав пользователя, если хотя на одной роли будет правило без параметров, оно будет проверяться.

Параметры правила

Позволяют настроить тип данных и ссылочность для значений, которые будут указаны на ролях.

Если правило примитивное, то у него доступен для настройки только один параметр, для составного правила доступны несколько параметров.

Скрипт для фильтрации объектных привилегий

Используется для фильтрации списков объектов, через наложение макроса универсального фильтра `&DefUniFltMacros`.

Доступные макросы внутри текста скрипта:

- `(<Имя атрибута>)`
Будет заменено на указанное имя атрибута. Например, `t.id`
Для фильтрации доступны все атрибуты дата-сета выборки шапки БО.
- `(¶ms)`
Массив строк, в котором будут параметры, собранные со всех ролей пользователя.
- `(&CurrentUserID)`
Будет заменено на ИД текущего пользователя.
- `(&CurrentUserName)`
Будет заменено на имя текущего пользователя.

Запрос чаще всего представляет из себя `exist`, в котором джойнится таблица класса, и через массив значений накладываются условия фильтрации.

Скрипт проверки строк по объектному кешу

Используется для проверки ограничения по объектам, используя объектный кеш.

Представляет из себя `jexl`-скрипт, который возвращает `true`, если объект проходит условие, и `false` - если не проходит.

Доступные переменные:

- `row \`
 - Если адм. объект создан по классу, то в этой переменной будет `rop` проверяемого объекта. Это не исходный `rop`, а его обертка `JexlRop`, которая позволяет обращаться к полям объекта.

Совет: Для получения исходного `rop` воспользуйтесь следующим примером:

```
//в переменной jexlRop находится объект класса JexlRop  
var rop = jexlRop.data()
```

- Если адм. объект создан не по классу, то здесь будет json-представление строки дата-сета выборки.
- **params**
массив значений параметров, собранных со всех ролей пользователя.
 - если правило примитивное - то будет содержать примитивы (строка или число)
 - если правило составное - то будет содержать строки, каждая из которых - json

Роль

1. Открыть карточку роли или выбрать ее в списке;
2. перейти на закладку **Дискретный доступ**;
3. создать новую запись, выбрав адм. объект;
4. указать значение параметров для ограничений;
5. сопоставить объектные и элементарные привилегии с нужным значением параметров ограничений, заполнив поле **Ограничение дискретного доступа**

Ограничение доступа к объектам класса.

Выдается через объектные привилегии с системными именами `edit#` и `view#`

Совет: Если таких привилегий нет у адм. объекта, то выполните его синхронизацию.

Дискретные ограничения применяются только, если:

- это главная выборка адм. объекта (шапка БО);
- это не супер-пользователь;
- к адм. объекту применяются настройки администрирования;
- адм. объект имеет признак **Дискретный доступ**.

Определение объектов для проверки дискретных прав

При проверке дискретных привилегий строки, в контексте которых требуется проверить наличие прав, определяются методом `acRows`

Алгоритм работы метода:

1. Если выборка принадлежит классу (`thisApi() != null`), то для всех выделенных записей через поле первичного ключа загружаются **провайдеры строк**, из них через поле `gidRoot_dz` определяются шапки БО

2. Если выборка не принадлежит классу (`thisApi() == null`), то по всем выделенным строкам собирается json-массив, содержащий на каждую строку json-объект, в котором ключ - имя атрибута, значение - значение атрибута.

Ограничение списков

Используя значения параметров всех ролей и настроенные «Скрипты для фильтрации объектных привилегий» формируется условие ограничения и накладывается фильтрация через макрос `&DefUniFltMacros`.

Ограничение открытия карточек

На открытие карточки используя значения параметров всех ролей выполняется `jexl-скрипт`, настроенный в Скрипт проверки строк по объектному кешу. Если у пользователя нет прав на объект, то будет выдана ошибка.

Ограничение на редактирование объектов

На `beforeEdit` или `delete` проверяется наличие привилегии `edit#` (выполняется `jexl-скрипт`, настроенный в Скрипт проверки строк по объектному кешу). Если у пользователя нет прав на объект, то будет выдана ошибка.

Ограничение на объектные привилегии

Если для объектной привилегии указано **Ограничение дискретного доступа**, то такая привилегия может быть проверена только в контексте какого-либо объекта. Для проверки объектных привилегий используется метод `Btk_AdminPkg().hasObjPriv`.

Пример:

```
if (Btk_AdminPkg().hasObjPriv("SomeObjectName", "SomePrivName", Seq(rop))) {  
    //код, выполняемый при наличии привилегии  
}
```

Если для привилегии, имеющей ограничение, выполнить проверку без передачи объектов, то будет выдана ошибка.

Ограничение на элементарные привилегии

Для элементарной привилегии может быть указано ограничение дискретного доступа. Если ограничение указано, то при выполнении операции будут проверены права в контексте записей в выборке.

На загрузку выборки или перезагрузку `ClassLoader`-а происходит подписка на событие выполнения операции.

При выполнении операции проверяются дискретные права на эту элементарную привилегию:

1. Получение `acRows`
2. Для полученных строк выполняется `jexl-скрипт`, настроенный в Скрипт проверки строк по объектному кешу

3. Если прав нет, то выдается ошибка.

4.2 Примеры скриптов

Примитивное правило по атрибуту класса

Правило значимое, тип данных строка. Фильтрация по like атрибута sCode

Скрипт для фильтрации объектных привилегий:

```
select 1
  from RplTst_ClassTree tt
 join (
    select cast(json_array_elements_text(cast((&params) as json)) as varchar) as sCode
      ) as codes
  on tt.sCode like codes.sCode
where tt.id = (&id)
```

Скрипт проверки строк по объектному кешу:

```
for (p: params) {
  if (row.sCode != null && p != null) {
    if (like(row.sCode, p)) {
      return true;
    }
  }
}
```

Примитивное правило по атрибуту коллекции

Правило ссылочное на Btk_Group. Фильтрация по прямому вхождению объекта в группу (фильтрация коллекции Btk_ObjectGroup).

Скрипт для фильтрации объектных привилегий:

```
select 1
  from RplTst_TestGroup tt
 join Btk_ObjectGroup og
    on tt.gid = og.gidSrcObject
 join (
    select cast(json_array_elements_text(cast((&params) as json)) as int8) as id
      ) as params
  on og.idGroup = params.id
where tt.id = (&id)
```

Скрипт проверки строк по объектному кешу:

```
for (p: params) {
  var rops = toJRops(Btk_ObjectGroupApi.byParent(row.data())).asList();
  if (p != null) {
    for (r: rops) {
      if (r.idGroup == p) {
        return true;
      }
    }
  }
}
```

(continues on next page)

(продолжение с предыдущей страницы)

}

Примитивное правило по адм. объекту, созданному не по классу (произвольная выборка)

Правило значимое, тип данных строка. Фильтрация регистронезависимая по like поля sClass

Скрипт для фильтрации объектных привилегий:

```
select 1
  from json_array_elements_text(cast((&params) as json)) as p
 where upper((&sClass)) like upper(p)
```

Скрипт проверки строк по объектному кешу:

```
for (p: params) {
    if (row.sClass != null && p != null) {
        if (like(row.sClass.toUpperCase(), p.toUpperCase())) {
            return true;
        }
    }
}
```

Составное правило по 3 атрибутам класса

Параметры правила:

1. **dDate** - Значимый, тип данных дата.
2. **nNumber** - Значимый, тип данных число
3. **sCaption** - Значимый, тип данных строка

Фильтрация отбирает записи у которых дата меньше указанной, число равно указанному, и Наименование по like регистронезависимо совпадает с указанной

Скрипт для фильтрации объектных привилегий:

```
select 1
  from RplTst_AllDbTypes tt
 where tt.id = (&id)
    and exists (
      select 1
        from (
          select v -> 'dDate' as dDate
                ,v -> 'nNumber' as nNumber
                ,v -> 'sCaption' as sCaption
          from json_array_elements(cast((&params) as json))
        ) as v
      ) p
  where exists (
```

(continues on next page)

```

select 1
  from json_array_elements_text(p.dDate) pp
 where tt.dDate <= to_timestamp(pp, 'DD.MM.YYYY HH24:MI:SS')
)
and exists (
select 1
  from json_array_elements_text(p.nNumber) pp
 where tt.nNumber = cast(pp as numeric)
)
and exists (
select 1
  from json_array_elements_text(p.sCaption) pp
 where upper(tt.sCaption) like upper(pp)
)
)

```

Скрипт проверки строк по объектному кешу:

```

for (p: params) {
  var jsonObj = toJsonObject(p);
  var res = true

  //проверка даты
  if (res) {
    res = false
    var aValues = jsonObj.childJSONArray("dDate");
    var i = 0
    while(i < aValues.size() && !res) {
      var value = aValues.getDate(i);
      if (row.dDate <= value) {
        res = true;
      }
      i = i + 1;
    }
  }

  //проверка числа
  if (res) {
    res = false
    var aValues = jsonObj.childJSONArray("nNumber");
    var i = 0
    while(i < aValues.size() && !res) {
      var value = aValues.getNumber(i);
      if (row.nNumber == value) {
        res = true;
      }
      i = i + 1;
    }
  }

  //проверка наименования
  if (res) {

```

(continues on next page)

```

res = false
var aValues = jsonObj.childJSONArray("sCaption");
var i = 0
while(i < aValues.size() && !res) {
    var value = aValues.getString(i);
    if (row.sCaption != null && value != null) {
        if (like(row.sCaption.toUpperCase(), value.toUpperCase())) {
            res = true;
        }
    }
    i = i + 1;
}

if (res) {
    return true;
}
}

```

Составное правило по атрибуту класса и коллекции

Параметры правила:

1. sCaption - Значимый, тип данных строка
2. idGroup - ссылочный на Btk_Group

Фильтрация отбирает записи, у которых Наименование по like регистронезависимо совпадает с указанным, и есть прямое вхождение в группу (фильтрация коллекции Btk_ObjectGroup).

Скрипт для фильтрации объектных привилегий:

```

select 1
  from RplTst_TestGroup tt
 where tt.id = (&id)
    and exists (
      select 1
      from (
        select v -> 'sCaption' as sCaption
              ,v -> 'idGroup' as idGroup
              from json_array_elements(cast((&params) as json)) as v
        ) p
    where exists (
      select 1
      from json_array_elements_text(p.sCaption) pp
      where upper(tt.sCaption) like upper(pp)
    )
    and exists (
      select 1
      from Btk_ObjectGroup og
      join json_array_elements_text(p.idGroup) pp
      on og.idGroup = cast(pp as int8)
    )

```

(continues on next page)

```

        where og.gidSrcObject = tt.gid
    )
)

```

Скрипт проверки строк по объектному кешу:

```

for (p: params) {
    var jObj = toJObject(p);
    var res = true

    //проверка наименования
    if (res) {
        res = false
        var aValues = jObj.childJArray("sCaption");
        var i = 0
        while(i < aValues.size() && !res) {
            var value = aValues.getString(i);
            if (row.sCaption != null && value != null) {
                if (like(row.sCaption.toUpperCase(), value.toUpperCase())) {
                    res = true;
                }
            }
            i = i + 1;
        }
    }

    //проверка группы
    if (res) {
        res = false
        var rops = toJRops(Btk_ObjectGroupApi.byParent(row.data())).asList();
        var aValues = jObj.childJArray("idGroup");
        var i = 0
        while(i < aValues.size() && !res) {
            var value = aValues.getLong(i);
            for (r: rops) {
                if (r.idGroup == value) {
                    res = true;
                }
            }
            i = i + 1;
        }
    }

    if (res) {
        return true;
    }
}

```

Составное правило по 2 атрибутам адм. объекта, созданного не по классу (произвольная выборка)

Параметры правила:

1. sClass - Значимый, тип данных строка
2. idRefClass - ссылочный на Btk_Class

Фильтрация отбирает записи, у которых поле sClass по like регистронезависимо совпадает с указанным, и поле idRefClass совпадает с указанным

Скрипт для фильтрации объектных привилегий:

```
select 1
  from (
    select v -> 'sClass' as sClass
           ,v -> 'idRefClass' as idRefClass
    from json_array_elements(cast((&params) as json)) as v
  ) p
where exists (
  select 1
    from json_array_elements_text(p.sClass) pp
   where upper((&sClass)) like upper(pp)
)
and exists (
  select 1
    from json_array_elements_text(p.idRefClass) pp
   where (&idRefClass) = cast(pp as int8)
)
```

Скрипт проверки строк по объектному кешу:

```
for (p: params) {
  var jsonObj = toJsonObject(p);
  var res = true

  //проверка sClass
  if (res) {
    res = false
    var aValues = jsonObj.childJSONArray("sClass");
    var i = 0
    while(i < aValues.size() && !res) {
      var value = aValues.getString(i);
      if (row.sClass != null && value != null) {
        if (like(row.sClass.toUpperCase(), value.toUpperCase())) {
          res = true;
        }
      }
      i = i + 1;
    }
  }

  //проверка idRefClass
  if (res) {
    res = false
```

(continues on next page)

```

var aValues = jsonObj.childJSONArray("idRefClass");
var i = 0
while(i < aValues.size() && !res) {
    var value = aValues.getLong(i);
    if (row.idRefClass == value) {
        res = true;
    }
    i = i + 1;
}
}

if (res) {
    return true;
}
}

```

5 Приложение 1

5.1 Функции отображения

Для работы системы администрирования в отображении объявлены следующие функции:

- **onLoadAdminMeta**
Применяет настройки администрирования;
- **acObject**
Возвращает имя администрируемого объекта;
- **acObjectBundle**
Имя узла администрирования;
- **acObjectItem**
Имя элемента администрирования;
- **isAdministrated**
Признак, что выборка администрируется.

5.2 Функции проверки привилегий

- **ru.bitec.app.btk.Btk_AdminPkg#hasObjPriv**
Проверяет наличие объектной привилегии у пользователя
- **ru.bitec.app.btk.Btk_AdminPkg#hasStateChangePriv**
Проверяет наличие привилегии на перевод состояния
- **ru.bitec.app.btk.Btk_AdminPkg#hasRole**
Проверяет наличие у пользователя роли

6 Связь выборок с администрируемыми объектами

6.1 основные понятия

Элемент администрирования

Элемент администрирования (класс `Btk_AcItem`) - логический объект, который соответствует одной выборке.

На каждую выборку, обладающую своей `avm.xml`, создается отдельный элемент администрирования, обладает перечнем элементарных привилегий

Элементарная привилегия

Элементарная привилегия (класс `Btk_AcItemPrivilege`, коллекция к `Btk_AcItem`) - логический объект, который соответствует одному из элементов выборки: атрибуту или операции.

Элементарные привилегии обладают типом привилегии. Например, атрибуты относятся к типу Чтение.

Тип элементарной привилегии

Определяет группу Элементарной привилегии. Позволяет управлять доступом к группам привилегий, а не по отдельности к каждой элементарной привилегии

Объект администрирования

Объект администрирования (класс `Btk_AcObject`) - логический объект, который соответствует Бизнес объекту, и обладает перечнем элементов администрирования, входящих в этот объект.

Бизнес объект

Бизнес объект (класс `Btk_BoEntity`) - логическое объединение нескольких классов в единую сущность.

Бизнес объект создается на корневые классы, и включает все его коллекции.

Базовая выборка

Выборка, которая соответствует классу. Например, выборка `Btk_ClassAvi` является базовой выборкой класса `Btk_Class`

Произвольная выборка

Выборка, которая не принадлежит какому-либо классу. В том числе и выборки-наследники от базовой выборки

Корневой класс

Класс, на который формируется отдельный бизнес объект. Является логическим отдельным объектом, который обладает своими коллекциями.

Это классы с типом Справочник, Документ, Настройка, Журнал.

6.2 Принцип построения адм. объектов

Формирование на основе структуры odm.xml

1. На каждый корневой класс создается свой администрируемый объект, имя которого равно имени класса.
2. В администрируемый объект включаются элементы администрирования, которые соответствуют выборкам всех классов входящий в бизнес объект (сам класс и все его коллекции).

Пример структуры бизнес объекта:

```
Документ Some_Document
+ Коллекция Some_DocumentPosition
+ Коллекция Some_DocumentPositionDet
+ V-коллекция Коллекция Some_DocumentLink
```

Пример сформированного администрируемого объекта:

```
Адм. объект Some_Document
+ Элемент Some_DocumentAvi
+ Элемент Some_Document\Some_DocumentPositionAvi
+ Элемент Some_Document\Some_DocumentPosition\Some_DocumentPositionDetAvi
+ Элемент Some_Document\Some_DocumentLinkAvi
```

Элементы администрирования в составе объекта администрирования имеют имя, которое соответствует их пути внутри бизнес объекта. Зачем используются такие пути будет описано в главе **Определение связи выборки и адм. объекта**

Подключение произвольной выборки в структуру адм. объекта

По мимо элементов администрирования, соответствующих базовым выборкам классов бизнес объектов, есть возможность добавить произвольную выборку в структуру администрируемого объекта. Для этого существует 2 способа:

1. Подключение в исходном коде

В avm.xml базовой выборки корневого класса добавить тег `acObject`, в котором указать дополнительные элементы администрирования.

Пример разметки:

```
<view xmlns="http://www.global-system.ru/xsd/global3-view-1.0"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:schemaLocation="http://www.global-system.ru/xsd/global3-view-1.0"
      name="Some_Document">

  <acObject>
```

(continues on next page)

```

        <acItems>
            <acItem name="Some_Document\Some_CustomAvi" caption="Произвольная выборка" /
↪>
        </acItems>
    </acObject>
</view>

```

В этом примере объявлен один дополнительный элемент администрирования в составе объекта «Some_Document», с именем «Some_Document\Some_CustomAvi» и наименованием «Произвольная выборка». Наименование используется для визуального отображения этого элемента в приложении администратор.

2. Подключение через приложение администратор

- Открыть карточку нужного объекта администрирования
- добавить новый элемент на закладке Элементы администрируемого объекта

6.3 Определение связи выборки и адм. объекта

Каждая выборка на открытии в событии `onLoadAdminMeta` через методы выборки `acObject` и `acObjectItem` определяет свою принадлежность к элементу администрирования и объекту администрирования. Все настройки прав доступа к выборке будут искаться по паре значений, которые вернули эти два метода.

Значение, которые вернули эти два метода для выборки, можно увидеть через окно отладки.

Т.е. чтобы к выборке корректно применялись настройки доступа, в приложении администратор должен быть объект администрирования, имя которого равно результату метода `acObject`, и в составе этого объекта должен быть элемент администрирования, имя которого равно результату метода `acObjectItem`.

Метод `acObject`

Возвращает системное имя объекта администрирования к которому относится выборка. Например, «Some_Document»

Основной принцип:

1. Если выборка принадлежит классу, и класс корневой, то имя этого класса = имя адм. объекта
2. Иначе вызывается метод `acObject` мастер-выборки.

Метод `acObjectItem`

Возвращает системное имя элемента администрирования в составе объекта. например, «Some_Document\Some_DocumentPositionAvi»

Основной принцип:

1. Вызывается метод `acObjectItem` мастер-выборки.
2. К полученному значению добавляется текущее имя выборки.

Базовые выборки классов

Для базовых выборок классов в большинстве случаев корректно отрабатывает стандартная логика определения элемента администрирования и объекта администрирования.

Исключением может служить базовая выборка v-коллекций, если она открывается вне формы, на которой расположена базовая выборка мастер-объекта. Например, если открыть список объектов v-коллекции, как главную выборку формы.

Произвольные выборки

Для произвольных выборок определить принадлежность к объекту администрирования автоматически не так просто, как для базовых выборок, которые соответствуют структуре бизнес объектов.

В большинстве случаев разработчику требуется самостоятельно переопределить работу методов `acObject` и `acObjectItem`.

Выборка как самостоятельный адм. объект

Если произвольную выборку предполагается использовать как самостоятельный объект администрирования, то требуется:

1. в `avm.xml` объявить тег `acObject`
2. в `Avi` переопределить методы `acObject` и `acObjectItem` следующим образом:

```
override def acObject: NString = {  
    baseAvi.simpleName  
}  
  
override def acObjectItem: NString = {  
    baseAvi.simpleName  
}
```

Выборка как часть другого адм. объекта

Если произвольную выборку предполагается использовать как часть существующего объекта администрирования, то требуется:

1. В `avm.xml` базовой выборки корневого класса этого бизнес объекта объявить тег `acObject` и внутри него добавить новый элемент администрирования

```
<view xmlns="http://www.global-system.ru/xsd/global3-view-1.0"  
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
    xsi:schemaLocation="http://www.global-system.ru/xsd/global3-view-1.0"  
    name="Some_Document">  
  
    <acObject>  
        <acItems>  
            <acItem name="Some_Document\Some_CustomAvi" caption="Произвольная выборка"/  
↪>  
        </acItems>
```

(continues on next page)

(продолжение с предыдущей страницы)

```
</acObject>  
</view>
```

2. в `Avi` произвольной выборки переопределить методы `acObject` и `acObjectItem` следующим образом:

```
override def acObject: NSString = {  
    //имя объекта  
    "Some_Document"  
}  
  
override def acObjectItem: NSString = {  
    //путь, который мы указали в теге acItem  
    "Some_Document\\Some_CustomAvi"  
}
```