

Руководство администратора uBase

ООО «Ю-груп»

3 мая 2016

Оглавление

Введение	vi
1 Установка системы	1
1.1 Требования к оборудованию и программному обеспечению	1
1.1.1 Требования серверной части системы	1
1.1.2 Требования клиентской части системы	1
1.1.3 Требования клиентской части системы при доступе через веб-интерфейс	3
1.2 Установка серверной части системы	4
1.2.1 Установка на ОС Linux	4
1.2.2 Установка на ОС Windows Server	5
1.3 Первоначальная настройка сервера	6
1.4 Настройка веб-сервера	7
1.5 Настройка резервного копирования	8
1.5.1 Вводная часть	8
1.5.2 Резервное копирование на ОС Linux	9
Пример	10
1.5.3 Восстановление из резервной копии на ОС Linux .	11
1.5.4 Резервное копирование на ОС Windows Server . .	11
Примеры	12
1.6 Установка клиентской части системы	14
1.6.1 Установка на ОС Windows	14
1.6.2 Установка на ОС Linux	15
2 Менеджер системы uBase	16
2.1 Настройки сервера	16
2.2 Настройки библиотек	17
2.2.1 Создание новой библиотеки	18
2.3 Список подключений	19

3	Данные системы	20
3.1	Основные понятия	20
3.1.1	Идентификаторы	21
3.2	Классы сущностей и справочники	21
3.2.1	Древовидные справочники	23
3.2.2	Табличные части	23
3.3	Типы данных полей	23
3.3.1	Отношения	24
3.4	Права доступа	26
3.4.1	Специальные пользователи	26
3.4.2	Специальные группы	26
3.4.3	Права доступа руководителей	27
3.5	Права доступа объектов системы	27
3.5.1	Права доступа сущностей	27
3.5.2	Права доступа классов сущностей	27
3.5.3	Права доступа реестров	28
3.5.4	Права доступа к моделям процессов	29
3.5.5	Права доступа процессов	29
3.5.6	Права доступа к вложениям процесса из задачи	29
3.5.7	Права доступа отчётов	30
3.6	Реестры	30
3.6.1	Шаблон для генерации номера записи реестра и путей к папкам	31
3.7	Маршрутизация	33
3.8	Отчёты	35
3.9	Настройки рабочего места	36
3.9.1	Пользовательские настройки	36
3.9.2	Локальные настройки	37
4	Пользователи и группы	39
4.1	Описание данных пользователей и групп	39
4.1.1	Аккаунты пользователей	39
4.1.2	Группы	39
4.2	Управление пользователями и группами	40
4.2.1	Замещение пользователей	41
5	Метаданные системы	43
5.1	Классы сущностей и свойства	43
5.2	Карточка свойств	45
5.3	Редактор карточек	47

6	База данных системы	49
6.1	Консоль базы данных	49
6.2	Таблицы	50
6.2.1	ACL	50
6.2.2	AUTHENTICATION	50
6.2.3	CATALOG	51
6.2.4	CERTIFICATE	51
6.2.5	CERTIFICATE_STATE	52
6.2.6	CHECKOUT	52
6.2.7	ENTITY_CLASS	52
6.2.8	ENTITY_PHONES	53
6.2.9	ENTITY_REVISION	53
6.2.10	EVENT	54
6.2.11	EVENT_PARTICIPANT	55
6.2.12	EVENT_REMINDER	55
6.2.13	LOGIN_AS	55
6.2.14	OBJECT_LINK	56
6.2.15	PREVIEW	56
6.2.16	PREVIEW_CONTENT	57
6.2.17	PROCESS	57
6.2.18	PROCESS_ATTACHMENT	58
6.2.19	PROCESS_COMMENT	58
6.2.20	PROCESS_ITERATION	58
6.2.21	PROCESS_LOG	59
6.2.22	PROCESS_PROPERTIES	59
6.2.23	PROCESS_REVISION	59
6.2.24	READ_COMMENTS	60
6.2.25	REGISTER_ENTRY	60
6.2.26	REGISTER_LINK	61
6.2.27	REVISION	61
6.2.28	SIGNATURE	61
6.2.29	STATE	62
6.2.30	SYSTEM	62
6.2.31	TASK	62
6.2.32	TELEPHONY_ACCOUNT	63
6.2.33	USER_CERTIFICATE	64
6.2.34	USER_SETTING	64
6.2.35	X500NAME	64

7	Расположение конфигурации и данных	66
7.1	Сервер системы	66
7.1.1	Директория с конфигурацией и данными системы	66
7.1.2	Файл server.xml	67
7.1.3	Файл server.conf (только для Linux)	67
7.1.4	Расположение данных	68
7.2	Клиент системы	69
7.2.1	Файл client.xml	70
7.2.2	Файл блокировки	70
7.2.3	Кэш файлового хранилища	71
7.2.4	Данные сервиса просмотра, редактирования и интеграции	71
7.2.5	Путь установки клиента системы	71
7.2.6	Журналы работы клиента	72
A	Шаблоны библиотеки	73
A.1	Шаблоны общего назначения	73
A.1.1	Базовый шаблон uBase	73
	Аккаунт пользователя	73
	Валюта	74
	Вид активности	75
	Город	75
	Группа	76
	Дерево	76
	Договор	77
	Документ	78
	Должность	78
	Дополнительное соглашение	78
	Класс сущности	79
	Контейнер	80
	Метаданные	81
	Модель маршрутизации	81
	Наша организация	82
	Обычное дерево	83
	Организация	84
	Отчёт	85
	Папка	85
	Первичный документ	86
	Персона	87
	Письмо	88
	Письмо входящее	88

Письмо исходящее	89
Подразделение	90
Пользователь или группа	91
Приказ	91
Приказ по личному составу	92
Приказ по основной деятельности	93
Регион	93
Реестр	94
Реестр организации	95
Свойство	96
Сотрудник	97
Справочник	98
Ссылки на реестр	98
Страна	99
Строгое дерево	99
Сущность	100
Табличная часть	100
Тип договора	101
Участник	101

Введение

Платформа uBase предназначена для управления документами и бизнес-процессами организации. Платформу uBase можно использовать в качестве системы электронного документооборота (СЭД) предприятия.

Системы электронного документа предназначены для хранения документов и обеспечения их движения по бизнес-процессам. В частности, такие системы обеспечивают перевод в электронную форму процессов согласования и корректировки документов; предоставляют средства контроля исполнительской дисциплины; обеспечивают архивное хранения документов с возможностью быстрого поиска нужных данных.

Как правило, такие системы не используются «из коробки», поскольку процессы делопроизводства сложны и специфичны для каждой конкретной организации. Системы в чистом виде предоставляют инструмент для создания готового решения, а не само решение. В поставке СЭД могут идти те или иные типовые решения, однако типовое решение может покрыть потребности организации лишь в редких случаях.

Настройка системы под нужды конкретного предприятия представляет собой большой трудоёмкий процесс. Его целью является не просто внедрение СЭД, как таковой, а перевод делопроизводства в электронный вид с оптимизацией бизнес-процессов. Система выступает здесь в роли ключевого средства для достижения этой цели, но она не заменяет весь цикл работ по автоматизации предприятия. Рекомендуется, чтобы процессы автоматизации проводились квалифицированными специалистами, имеющими опыт в этой сфере деятельности. Обычно это означает, что для получения результата требуется привлечение компании, оказывающей соответствующие услуги. Изучение и реализация всего цикла работ по автоматизации собственными силами, в большинстве случаев, будут неэффективными.

Автоматизация предполагает изучение деятельности предприятия;

выработку новых схем с учётом перехода на автоматизированную систему; разработку структуры библиотеки документов, маршрутов движения документов, отчётов и прочего; обучение персонала; перенос имеющихся данных в систему; опытную эксплуатацию; корректировки по её итогам и переход к промышленной эксплуатации.

Дальнейшая поддержка работы системы значительно менее трудоёмка и не требует больших вложений.

Руководство администратора iBase предназначено для изучения системными администраторами предприятия и покрывает соответствующий круг вопросов. Данное руководство не является пособием по переводу организации на электронный документооборот и не заменяет соответствующую литературу и руководства для разработчиков специализированных решений.

Глава 1

Установка системы

Система состоит из серверной и клиентской части.

1.1. Требования к оборудованию и программному обеспечению

1.1.1. Требования серверной части системы

Для промышленной эксплуатации системы рекомендуется использование выделенного сервера или выделенной виртуальной машины на сервере. Это позволит избежать нежелательного взаимодействия с другими службами и программами и повысит надёжность работы системы и безопасность ваших данных.

Список поддерживаемых операционных систем для серверной части системы приведён в таблице 1.1.

Также существует возможность запуска серверной части на всех 64-битных платформах, поддерживаемых клиентской частью системы; а также на других платформах Linux x86_64, соответствующих стандарту LSB 4.0 и использующих systemd для управления службами. Однако корректная работа на этих платформах не гарантируется и техническая поддержка на эти конфигурации не распространяется.

1.1.2. Требования клиентской части системы

Список поддерживаемых операционных систем для клиентской части системы приведён в таблице 1.2.

Оборудование для клиентской части системы должно соответствовать требованиям установленной операционной системы. Кроме этого,

Таблица 1.1: Поддерживаемые операционные системы для серверной части системы

Платформа и версия	Архитектура
SUSE Linux Enterprise Server 12	x86_64 (64-битная)
openSUSE Linux 13.1	x86_64 (64-битная)
openSUSE Linux 13.2	x86_64 (64-битная)
openSUSE Leap 42.1	x86_64 (64-битная)
openSUSE Tumbleweed	x86_64 (64-битная)
Windows Server 2008 R2 SP1	x64 (64-битная)
Windows Server 2012	x64 (64-битная)
Windows Server 2012 R2	x64 (64-битная)

Таблица 1.2: Поддерживаемые операционные системы для клиентской части системы

Платформа и версия	Архитектура
Windows 7 SP1	x86 (32-битная)
Windows 7 SP1	x64 (64-битная)
Windows 8.x	x86 (32-битная)
Windows 8.x	x64 (64-битная)
Windows 10	x86 (32-битная)
Windows 10	x64 (64-битная)
SUSE Linux Enterprise Desktop 12	x86_64 (64-битная)
openSUSE Linux 13.1	x86_64 (64-битная)
openSUSE Linux 13.2	x86_64 (64-битная)
openSUSE Leap 42.1	x86_64 (64-битная)
openSUSE Tumbleweed	x86_64 (64-битная)

для всех 32-битных систем объём оперативной памяти не должен быть меньше 1 ГиБ, а для 64-битных — 2 ГиБ. При совместном использовании различных приложений требуется повышенный объём оперативной памяти.

Для установки и работы клиента требуется порядка 300 МиБ памяти в профиле пользователя.

Клиентская часть системы предоставляет пользователю современный интерфейс. Для обеспечения оптимальной производительности рекомендуется, чтобы компьютеры с установленной системой имели видеокарты, перечисленные в таблице 1.3, а в операционных системах были установлены драйверы актуальных версий.

Таблица 1.3: Рекомендуемые видеокарты для клиентской части системы

Производитель	Модели
NVIDIA	Мобильные GeForce серий 8M и 100M и выше, NVS серий 2100M и выше, Mobility Quadro FX серий 300M и выше Настольные GeForce серий 8 и 100 и выше Рабочие станции Quadro FX серий 300 и выше
ATI	Мобильные Mobility Radeon HD серий 3000, 4000 и 5000 Настольные Radeon HD серий 2400, 3000, 4000, 5000 и 6000
Intel	Мобильные GMA 4500MHD и GMA HD Настольные GMA 4500 и GMA HD

Допускается использование иных видеокарт, при этом возможно некоторое снижение производительности интерфейса системы.

1.1.3. Требования клиентской части системы при доступе через веб-интерфейс

Поддерживаемые веб-браузеры приведены в таблице 1.4. Для настольных компьютеров и ноутбуков рекомендуется разрешение экрана не ниже 1366x768. Для мобильных устройств рекомендуется разрешение экрана не ниже 480x800.

Таблица 1.4: Поддерживаемые браузеры для веб-интерфейса системы

Браузер	Версии
Chromium	Последняя версия
Google Chrome	Последняя версия
Firefox	Последняя версия
Internet Explorer	11
Microsoft Edge	Последняя версия
Opera	Последняя версия
Safari	5, 6

1.2. Установка серверной части системы

1.2.1. Установка на ОС Linux

Для запуска сервера системы используется `systemd`. Использование традиционных методов запуска, таких как `SysVinit`, не поддерживается. Ваша операционная система на сервере должна быть настроена на использование `systemd`. В поддерживаемых серверных системах Linux `systemd` используется по умолчанию и не требует дополнительной настройки.

Для установки системы предоставляется репозиторий с необходимыми пакетами RPM. Для работы в минимальном режиме достаточно пакета `ubase-server` и требуемого им пакета `ugo-cp-jre`. Для генерации предпросмотра в зависимостях прописаны и другие зависимости от пакетов, поставляемых вместе с поддерживаемыми операционными системами. При наличии доступных системных репозиториях их доустановка будет выполнена автоматически. Для генерации предпросмотра офисных документов требуется LibreOffice версий 4.0–5.1, предоставляемый The Document Foundation. Сервер также может использовать LibreOffice, входящий в состав операционной системы, однако корректная работа с ним не гарантируется.

Пакеты при установке создают пользователя `ubase-server` и группу `ubase-server` с идентификаторами 325. Сервис будет запускаться под этими пользователем и группой.

После первоначальной установки пакета в конфигурационный файл `/opt/ubase/server/etc/ubase/server.conf`

следует записать правильную локаль для службы. Для русского языка

следует отредактировать строку с параметром `LANG` следующим образом:

```
LANG=ru_RU.UTF-8
```

Брандмауэр системы должен разрешать доступ к сервису или быть отключенным. Сервис принимает удалённые подключения на порты 8519 и 8520. При необходимости в `server.xml` (см. раздел 7.1.2) можно в параметре `miPort` указать другой порт. Сервис будет использовать указанный порт и следующий за ним. В этом случае записанный в конфигурационном файле порт нужно будет указывать при установке клиентов системы вместо стандартного.

1.2.2. Установка на ОС Windows Server

Для установки сервера системы следует запустить установочный файл под учётной записью администратора и следовать его инструкциям.

Для генерации предпросмотра следует при установке оставить включённым соответствующий модуль.

После первоначальной установки будет запущена утилита `uBase Server Manager` для дальнейшей настройки. После появления в окне установки сообщения о её завершённости мастер установки следует закрыть и приступить к настройке сервера в менеджере системы.

После обновления системы, если ранее сервер был настроен, менеджер не запускается, однако его можно запустить вручную из меню.

Для генерации предпросмотра офисных документов следует дополнительно установить LibreOffice версий 4.0–5.1 от The Document Foundation стандартным образом. LibreOffice не входит в состав дистрибутива сервера `uBase`.

В настройках брандмауэра Windows следует разрешить запуск программы

```
%ProgramFiles%\U-Group\uBase\Server\VERSION\ubase-server.exe
```

Здесь `%ProgramFiles%` — путь к Program Files, а `VERSION` — версия системы. Если брандмауэр отключен, данное разрешение не требуется.

1.3. Первоначальная настройка сервера

Для настройки сервера используется графическая утилита uBase Server Manager. Эта утилита должна быть запущена под пользователем, имеющим привилегии администратора системы.

В окне uBase Server Manager имеется три вкладки.

Вкладка Сервер предназначена для управления сервисом. На серверах, имеющих несколько IP-адресов, в выпадающем списке следует выбрать адрес, который будут использовать клиенты системы. Этот же адрес впоследствии должен быть указан при установке клиентов системы. Вместо адреса можно указать корректное доменное имя сервера.

Если для пользователей будет использоваться аутентификация на сервере Kerberos V (например, в домене Windows), то следует указать имя или IP-адрес сервера Kerberos и имя домена. Следует обратить внимание на то, что имя домена чувствительно к регистру.

Вкладка Библиотеки предназначена для настройки библиотек. На этой вкладке нужно добавить новую библиотеку. Будут предложены настройки по умолчанию. Следует указать имя библиотеки, которое будет отображено пользователям при входе в систему и пути к данным хранилища.

При нажатии кнопки Применить появится диалог создания библиотеки. В нём следует выбрать подходящий шаблон и нажать ОК.

После сообщения об успешном создании библиотеки будет предложено сохранить конфигурацию сервера. Следует сохранить конфигурацию, так как в противном случае сервер не сможет использовать эту библиотеку и её нужно будет добавлять повторно.

Сервер может предоставлять несколько библиотек одновременно, но в промышленной эксплуатации, как правило, должна быть только одна библиотека.

После сохранения настроек с вкладки Сервер следует запустить сервис. Сервис будет работать под системной учётной записью и запускаться автоматически вместе с операционной системой.

При создании библиотеки из одного из стандартных шаблонов в ней создаётся пользователь `root`, имеющий права администратора. Этот пользователь используется для дальнейшей настройки библиотеки из клиента системы. При первом входе в систему под этой учётной записью система запомнит для него использованный при входе пароль.

Вкладка Подключения предназначена для просмотра активных сеансов пользователей.

1.4. Настройка веб-сервера

В состав сервера uBase входит веб-сервер. По умолчанию веб-сервер принимает подключения по протоколу HTTP на порт 8080 и, при наличии сертификата, по протоколу HTTPS на порт 8181.

Сервер uBase принимает сертификаты и ключи в формате JKS. Можно экспортировать имеющийся сертификат и ключ в этот формат в два этапа. Сначала сохранить их в контейнер PKCS#12 с непустым паролем при помощи OpenSSL (в команде следует указать пути к своим файлам):

```
openssl pkcs12 -export -in servercertificate.crt -inkey \  
servercertificatekey.key -out ubase.p12 -name ubase \  
-CAfile CA.crt -caname root -chain
```

Затем конвертировать PKCS#12 в JKS с помощью keytool:

```
keytool -importkeystore -deststorepass new_password \  
-destkeypass new_password -destkeystore ubase.keystore \  
-srckeystore ubase.p12 -srcstoretype PKCS12 \  
-srcstorepass openssl_password -alias ubase
```

Здесь в качестве `openssl_password` следует указать пароль, указанный команде `openssl` для сохранения контейнера. В качестве `new_password` указать пароль для контейнера JKS. Следует указать одинаковый пароль после аргументов `-deststorepass` и `-destkeypass`.

При отсутствии сертификата сервера можно сгенерировать собственный самоподписанный сертификат, однако такой сертификат не будет приниматься веб-браузерами без добавления в браузере соответствующего исключения. Такой сертификат может быть создан непосредственно с помощью утилиты `keytool`:

```
keytool -genkey -alias ubase -keyalg RSA \  
-keystore ubase.keystore
```

Полученный тем или иным способом контейнер с сертификатом следует поместить в подходящее место и настроить права доступа к файлу таким образом, чтобы он был доступен серверу uBase.

В конфигурационном файле `server.xml` следует указать свои значения следующим атрибутам тега `server`:

<code>keyAlias</code>	<code>ubase</code>
<code>keystoreFile</code>	Путь к файлу <code>ubase.keystore</code>
<code>keystorePass</code>	Пароль контейнера JKS

Значением атрибута `webPort` следует указать порт для HTTP-подключений. При указании значения 0 веб-интерфейс uBase будет отключен.

Значением атрибута `webSecurePort` следует указать порт для безопасных HTTPS-подключений. При указании значения 0 веб-интерфейс uBase через протокол HTTPS будет отключен. Доступ по HTTPS также будет отключен при отсутствии сертификата сервера.

Следует учесть, что на UNIX-подобных системах порты с номерами, меньшими 1024, не будут доступны для сервера uBase.

Изменения в конфигурационном файле вступят в силу при перезапуске службы uBase.

Для работы через незащищённую сеть рекомендуется использовать исключительно протокол HTTPS. Доступ по протоколу HTTP можно закрыть при помощи брандмауэра.

1.5. Настройка резервного копирования

1.5.1. Вводная часть

Для минимизации рисков потери важной информации резервное копирование должно быть настроено после установки платформы uBase и осуществляться регулярно.

Методы резервного копирования определяются системным администратором в соответствии с техническими возможностями и особенностями компьютерной инфраструктуры, а также с учётом политики информационной безопасности организации.

Метод резервного копирования по созданию цельного образа сервера (дисков, операционной системы) зависит от платформы, на которой развёрнут сервер и от используемого программного обеспечения. Данный метод не рассматривается в данном руководстве, но является предпочтительным по надёжности и простоте восстановления работоспособного сервера на момент создания последней резервной копии.

Метод резервного копирования данных платформы uBase является более избирательным, но в большинстве случаев позволяет сэкономить дисковое пространство, а также использовать свободное ПО и встроенные инструменты ОС.

Резервное копирование данных включает три вида информации:

- Конфигурационные файлы — файлы настроек сервера и библиотек, файлы лицензий. Особо важным в этих конфигурационных файлах является сведения об учётной записи, от имени которой

осуществляется работа с базой данных, а также файлы лицензий, определяющие количество доступных подключений.

- База данных — библиотека платформы uBase, содержащая сведения об объектах системы: атрибутах, связях, правах доступа, а также о задачах и процессах. В случае использования СУБД отличной от встроенной H2 резервное копирование БД необходимо выполнять в соответствии с правилами используемой СУБД.
- Хранилище файлов — содержимое всех документов, хранящихся в системе. При планировании резервного копирования файлового хранилища необходимо учитывать, что в платформе uBase каждый файл после создания не меняется, все изменения в документах иницируют создание нового файла, не изменяя старые.

Любое резервное копирование настоятельно рекомендуется производить на носитель информации, логически и физически независимый от сервера платформы uBase, и желательно территориально отдалённый (как минимум в разных кабинетах, как максимум на разных континентах).

1.5.2. Резервное копирование на ОС Linux

Конфигурационные файлы платформы uBase по умолчанию создаются и хранятся в следующей директории:

```
/opt/ubase/server/etc/ubase/
```

Изменения этих файлов, как правило, производятся только администратором системы. При первоначальной настройке и при последующих изменениях администратору следует создавать резервные копии этой директории или изменённых файлов.

Полная копия базы данных платформы uBase без остановки сервера создаётся следующей командой:

```
ubase-server backup
```

Файл резервной копии появляется в виде zip-архива с меткой даты и времени в папке

```
/opt/ubase/server/backup/
```

Созданный резервный файл необходимо перенести на сервер резервных копий.

Для файлового хранилища рекомендуется сделать зеркало на сервере резервных копий с использованием программы синхронизации файлов и каталогов `rsync`.

Автоматизация операций по резервному копированию по расписанию может быть выполнена с использованием скриптовых команд и сервиса `cron`.

Пример

1. Раздел сервера резервного копирования примонтирован к файловой системе сервера платформы `uBase` в режиме записи как директория:

```
/mnt/backups/
```

В этой директории будут созданы поддиректории `db` и `files` для хранения копий базы данных и для зеркала файлового хранилища соответственно.

2. Резервные копии базы данных при создании сохраняются в директорию по умолчанию:

```
/opt/ubase/server/backup/
```

3. Файловое хранилище располагается в директории:

```
/opt/ubase/server/library/main/files/
```

4. Создаём следующий файл скрипта:

```
/opt/ubase/server/bin/backup_command
```

в него записываем следующие команды:

```
# make db backup
/opt/ubase/server/bin/ubase-server backup
# move backup files to archive server
mv /opt/ubase/server/backup/* /mnt/backups/db/
# synchronize file storage and archive copy
rsync -r /opt/ubase/server/library/main/files /mnt/backups/
```

и делаем созданный файл исполняемым:

```
chmod a+x /opt/ubase/server/bin/backup_command
```

5. Чтобы данный файл выполнялся каждый день в два часа ночи, для сервиса cron в файле:

```
/etc/crontab
```

добавляем соответствующую строку:

```
# backup uBase  
0 2 * * * root /opt/ubase/server/bin/backup_command
```

и перезапускаем сервис cron:

```
systemctl restart cron.service
```

1.5.3. Восстановление из резервной копии на ОС Linux

Восстановление данных системы uBase можно разделить на два основных варианта: восстановление на существующем сервере и перенос на новый сервер.

Восстановление на существующем сервере требует остановки сервера, перезаписи файла базы данных и файлового хранилища. Важно после восстановления данных поменять владельца восстановленных файлов, иначе сервер не сможет обратиться к восстановленной библиотеке. Пример смены владельца при условии, что база данных и файловое хранилище находятся в директориях по умолчанию:

```
chown -R ubase-server:ubase-server \  
/opt/ubase/server/library/<Имя библиотеки>
```

1.5.4. Резервное копирование на ОС Windows Server

В реестре Windows в ключе

```
HKLM\SOFTWARE\U-group\uBase
```

под именем DataRoot записано значение с путём к директории с конфигурацией и данными системы. По умолчанию это директория C:\Users\.

В этой директории в её подкаталоге `etc\` хранятся конфигурационные файлы системы. Изменения этих файлов, как правило, производятся только администратором системы. При первоначальной настройке и при последующих изменениях администратору следует создавать резервные копии этой директории или изменённых файлов.

Полная копия базы данных платформы uBase без остановки сервера создаётся из графической утилиты uBase Server Manager. Кроме того для автоматического создания полной копии базы данных платформы uBase из скрипта или из командной строки можно воспользоваться командным файлом:

```
%ProgramFiles%\U-group\uBase\ВЕРСИЯ\backup.cmd <ПУТЬ>
```

где `%ProgramFiles%` — путь к Program Files, `ВЕРСИЯ` — версия системы, `<ПУТЬ>` — необязательный параметр, задающий путь к директории для сохранения резервной копии; если этот параметр отсутствует, то файл резервной копии будет создан в виде zip архива с меткой даты и времени в подкаталоге `backup` каталога `DataRoot`.

Созданный резервный файл необходимо перенести на сервер резервных копий.

Каталог файлового хранилища при первоначальной настройке библиотек сервера рекомендуется создавать на отдельном логическом диске. Это позволит выполнять резервное копирование библиотеки uBase средствами Windows Server Backup отдельно от резервного копирования файлов самой операционной системы.

Настроить периодическое резервное копирование по расписанию можно с помощью графической утилиты Windows Server Backup.

Для запуска процесса создания резервной копии из скрипта (и/или создания резервной копии в сетевой папке) можно воспользоваться утилитой командной строки `wbadmin`:

```
wbadmin start backup -backupTarget:\\[server]\[share]  
-include:C: -vssFull -quiet
```

Для автоматизации операций по резервному копированию на сетевой диск можно использовать приведённую выше команду, запуская её через Планировщик заданий (Task Scheduler).

Примеры

В качестве примера рассмотрим настройку резервного копирования в ОС Windows 2008 Server.

Предположим, что в соответствии с рекомендациями под хранение данных uBase выделен отдельный диск (например, E:), и сервер uBase настроен на хранение данных в директории E:\uBase, а в качестве хранилища резервных копий используется диск X:.

1. Настроим резервное копирование БД с помощью графической утилиты Планировщика заданий (Task Scheduler).

Из-под учётной записи Администратора запускаем GUI планировщика заданий из меню Пуск — Администрирование — Планировщик заданий.

Нажимаем кнопку Создать задачу, заполняем имя и описание задачи.

На вкладке Триггеры нажимаем кнопку Создать; выбираем По расписанию, Ежедневно и настраиваем время выполнения резервного копирования. Нажимаем ОК.

На вкладке Действия нажимаем кнопку Создать;

выбираем действие Запуск программы;

Нажимаем кнопку Обзор и указываем командный файл

```
%ProgramFiles%\U-group\uBase\VERSION\backup.cmd
```

В поле Необязательные аргументы вводим путь X:\backup. Нажимаем ОК.

Нажимаем ОК в окне Создание задачи. Задача по резервному копированию БД создана.

2. Настроим резервное копирование файлового хранилища с помощью Windows Server Backup.

Для успешного использования Windows Server Backup через графическую утилиту ваш сервер должен иметь как минимум один диск, не используемый операционной системой. На этот диск система будет создавать резервные копии.

Из-под учётной записи Администратора запускаем GUI Windows Server Backup из меню Пуск — Администрирование — Система архивации данных Windows Server.

Нажимаем действие Расписание архивации, нажимаем Далее;

выбираем тип конфигурации Настраиваемый, нажимаем Далее;

выделяем галочкой только диск на котором находится файловое хранилище (в нашем примере диск E:), нажимаем Далее;

настраиваем расписание архивации, нажимаем Далее;

в следующем окне добавляем диск назначения, то есть диск специально выделенный под резервные копии (например, X:),

выделяем его и подтверждаем удаление всех имеющихся на нём данных.

Нажимаем **Готово**. Задача по резервному копированию файлового хранилища создана.

В целях оптимизации рекомендуется переключить данную задачу в режим икрементного резервного копирования. Для этого в окне Windows Server Backup нажимаем действие **Настройка параметров производительности**, выбираем вариант **Выборочная**, и для тома с файловым хранилищем изменяем **Параметр архивации** со значения **Полная резервная копия** на значение **Добавочная архивация**.

1.6. Установка клиентской части системы

1.6.1. Установка на ОС Windows

Внимание! На сервере системы должен быть установлен пакет поддержки клиентов Windows, а сам сервер должен быть настроен, запущен и доступен по сети.

Первоначальная установка клиента осуществляется с помощью установочного файла, запущенного под учётной записью пользователя операционной системы, который будет работать с системой. Не следует запускать установку от имени другого пользователя или администратора. Для установки следует указать адрес сервера. Если сервер доступен и у него есть поддержка клиентов Windows, будет произведена установка клиента системы.

Обновления клиента системы при обновлении сервера будут производиться автоматически при запуске клиента системы пользователем.

На 64-битных системах Windows возможен запуск как 64-битной, так и 32-битной версии клиента uBase. По умолчанию на этих системах запускается 64-битная версия, но при установке можно указать принудительное использование 32-битной, если она в данном конкретном случае предпочтительнее. В частности, рекомендуется использование 32-битной версии при работе с *потокowymi* сканерами, чтобы иметь возможность работать через TWAIN вместо WIA. Для обычных сканеров функциональности WIA в общем случае должно быть достаточно и в этом случае целесообразнее использовать 64-битную версию. И 32-битная, и 64-битная версия клиента поддерживает как TWAIN, так и WIA, однако в состав ОС Windows входит только 32-битная библиотека TWAIN и производители принтеров обычно предоставляют только 32-битный драйвер TWAIN. Работа с WIA возможна из 32-битной и

64-битной версии в равной мере.

В некоторых случаях интеграция с внешними приложениями также может отличаться между 32-битной и 64-битной версиями; в работе же самой системы не должно наблюдаться отличий.

1.6.2. Установка на ОС Linux

Внимание! На сервере системы должен быть установлен пакет поддержки клиентов Linux, а сам сервер должен быть настроен, запущен и доступен по сети.

Первоначальная установка выполняется под учётной записью пользователя, который будет работать в системе в графическом режиме. Следует распаковать дистрибутив клиента системы для Linux и запустить файл `client-install`. Для установки следует указать адрес сервера. Если сервер доступен и у него есть поддержка клиентов Linux, будет произведена установка клиента системы.

Обновления клиента системы при обновлении сервера будут производиться автоматически при запуске клиента системы пользователем.

Глава 2

Менеджер системы uBase

Менеджер системы электронного документооборота uBase предназначен для изменения настроек сервера, создания библиотек и изменения их параметров, а также для просмотра существующих подключений к серверу.

Запуск менеджера необходимо производить от имени пользователя, имеющего права Администратора операционной системы сервера.

2.1. Настройки сервера

Вкладка **Сервер** содержит группу настроек данного сервера. После изменения настроек следует выполнить перезапуск серверной службы, чтобы новые настройки вступили в силу.

Настройка **Имя хоста RMI** содержит перечень IP-адресов сетевых интерфейсов сервера. Для корректного взаимодействия клиентов с сервером необходимо выбрать адрес, соответствующий внутренней локальной сети.

Настройки **Kerberos V KDC / Контроллер домена** и **Kerberos V realm / Домен** необходимы для корректной аутентификации с использованием доступного в локальной сети контроллера домена. При этом настройка **Kerberos V KDC / Контроллер домена** должна содержать разрешаемое доменное имя или IP-адрес доступного контроллера домена, а настройка **Kerberos V realm / Домен** — полное имя домена с учётом регистра (обычно прописными буквами).

После выбора параметров сервера и библиотек необходимо сохранение изменений, для этого в Менеджере присутствует кнопка **Сохранить**. Также сохранённый файл конфигурации доступен для загрузки в Менеджер с помощью кнопки **Загрузить**.

Статус сервера отображается цветной надписью слева от кнопки Запустить. При запущенном сервере отображается зеленая надпись Запущен, при остановленном сервере красная надпись Остановлен. Запуск и остановка сервера может выполняться с использованием кнопок Запустить и Остановить.

Кнопка Создать резервную копию доступна только при запущенном сервере. Функция данной кнопки — создание резервной копии только баз данных библиотек. Оно не включает в себя резервное копирование файлового хранилища. При нажатии на эту кнопку менеджер запускает процесс резервного копирования для каждой библиотеки на сервере. Если библиотеки имеют большой объём, процедура резервного копирования может занять значительное время.

Строка сервиса с кнопками Включить и Отключить управляют настройками службы в операционной системе. Если сервис включен, сервер uBase стартует при загрузке операционной системы.

2.2. Настройки библиотек

Вкладка Библиотеки содержит индивидуальные настройки для каждой библиотеки.

Для отображения и изменения настроек конкретной библиотеки необходимо выбрать требуемую библиотеку из выпадающего списка.

При выборе библиотеки все настройки доступны только для просмотра. Изменение настроек библиотеки становится доступным после нажатия кнопки Редактировать настройки. Кнопка при этом меняет надпись на Применить настройки. Изменение настроек существующих библиотек требуется выполнять при остановленном сервере. Настройки применяются при перезапуске сервера.

Кнопка Экспортировать шаблон выгружает данные из базы данных указанной библиотеки в файл с расширением `.ubaset` в формате XML. Экспортированный шаблон не содержит файлы содержимого документов библиотеки. Экспортированный шаблон может использоваться для создания новых библиотек.

Кнопка Добавить новую рассмотрена в подразделе 2.2.1.

Галочка Библиотека включена (запускается при старте сервера) отвечает за загрузку библиотеке при запуске сервера. Также из клиента uBase для входа будут доступны только включенные библиотеки.

Название библиотеки — последовательность символов, используемая для отображения в менеджере, в окне входа в систему, в заголовке окна клиента и при создании резервной копии. Рекомендуются

использовать в имени библиотеки символы, разрешённые для именованя файлов.

Схема или название базы данных — последовательность символов, используемая для именованя схемы используемой базы данных, или самой базы данных для некоторых баз данных. В зависимости от выбора СУБД могут предъявляться ограничения на используемый набор символов.

URL базы данных используется сервером для доступа к базе данных библиотеки. Начинается с указания драйвера, используемого для доступа к базе данных, и содержит сведения о местонахождении библиотеки. Формат этих сведений зависит от СУБД. Если по указанному пути схемы или базы данных не используется, то менеджер пытается её создать

Имя пользователя БД и Пароль БД используются для доступа к базе данных. Указанные учётные данные должны иметь право на создание и изменение схемы или базы данных.

Путь к данным библиотеки определяет путь к директории для хранения содержимого библиотеки.

Лимит пользовательских подключений определяет количество одновременных подключений к серверу. Количество одновременных подключений определяется количеством лицензий на сервер и этим лимитом.

2.2.1. Создание новой библиотеки

Кнопка **Добавить новую** создаёт набор настроек для новой библиотеки. Настройки сразу доступны для изменения. После нажатия кнопки **Применить настройки** откроется окно с выбором параметров для генерации базы данных библиотеки. Параметры **Хеш-алгоритм библиотеки** и **Хеш-алгоритм хранилища документов** для выбора предлагают список доступных алгоритмов. Рекомендуется использовать какой-либо из вариантов алгоритма SHA2 (SHA-256, SHA-384 или SHA-512) или SHA3 (SHA3-256, SHA3-384 или SHA3-512). От размера хеша в некоторой степени зависит размер базы данных библиотеки, поэтому не следует выбирать алгоритмы с большим размером хеша без особых на то соображений. По умолчанию в обоих полях выбирается алгоритм SHA-256.

Шаблон выбирается из списка шаблонов, которые располагаются в папке конфигурации `etc/ubase/ubasetl`. По умолчанию выбран первый шаблон по порядку. Шаблон используется для создания начального содержимого библиотеки и может содержать необходимые справочники, карточки, группы, структуру папок и другие данные.

Клонирование библиотеки — дополнительный инструмент, который может использоваться в особых случаях. Для корректного клонирования библиотеки необходимо выбирать шаблон, на основе которого была создана исходная библиотека.

2.3. Список подключений

Вкладка Подключения отображает список активных подключений. Для обновления информации на вкладке располагается соответствующая кнопка. В таблице отображаются сведения о подключении: реальный и эффективный пользователь, библиотека, хост с которого производится подключение, дата и время подключения пользователя.

Глава 3

Данные системы

3.1. Основные понятия

Одним из основных понятий, с которым работает система, является *сущность*. И документы, и записи в справочниках, и многие другие данные являются сущностями. Каждая сущность принадлежит определённому *классу сущностей*. Класс сущностей определяет назначение сущности и то, какие *поля* в них доступны для заполнения. Эти поля у сущностей могут быть заполнены теми или иными *значениями*. Для полей определены *типы данных*, от которых зависит, какого рода значения в них можно сохранять, а название поля обозначает его смысл.

В системе существует некоторое количество базовых классов сущностей. Дополнительные классы сущностей могут быть доступны из предопределённой конфигурации системы, а также созданы администратором системы вручную.

Одной из отличительных черт системы является работа с *состояниями* системы. В любой момент времени изменяемые данные в системе могут находиться в том или ином состоянии. Когда эти данные меняются, например редактируется документ или добавляется новый, образуется новое состояние системы. Помимо *текущего* (актуального) состояния, при необходимости можно просмотреть и старые состояния системы, чтобы поднять старую версию документа или записи в справочнике, или посмотреть историю их изменения. Сущность, изменяясь с течением времени, порождает новые *редакции*. Старые редакции при необходимости могут быть восстановлены.

В системе существуют т. н. *реестры*. Реестры используются для регистрации писем, договоров, приказов, служебных записок и прочего. Каждый создаваемый реестр предназначается для ведения запи-

сей определённого рода. Каждая запись в пределах реестра имеет свой уникальный номер. Как правило, номера генерируются автоматически по шаблону. Помимо номера у записи есть дата и время регистрации. К записи может быть прикреплена какая-либо сущность. Например, в реестре входящих писем к записям может прикрепляться документ с отсканированным письмом.

Записи реестра невозможно отредактировать или удалить.

Записи могут иметь *связанные записи*. Связи могут быть *направленными* и *ненаправленными*. Направленные связи можно использовать, например, для обозначения, что исходящее письмо отправлено в ответ на определённое входящее письмо. Ненаправленные связи используются, когда между записями не прослеживается определённый порядок или направление, но они имеют какое-либо отношение друг к другу.

3.1.1. Идентификаторы

Каждая сущность, появляясь в системе, получает свой *идентификатор сущности*. Все ссылки на данную сущность внутри системы используют этот идентификатор. Если сущность изменяется, её идентификатор остаётся неизменным. Однако у каждой *редакции* сущности есть свой собственный *идентификатор редакции*. Если сущность возвращается к одной из своих прежних редакций, то тот же самый идентификатор редакции используется повторно.

Свои идентификаторы также есть у записей реестров, процессов и задач.

3.2. Классы сущностей и справочники

Каждая сущность в системе должна принадлежать к одному или нескольким *классам сущностей*. Класс сущностей определяет смысловое значение сущности, такое как Группа, Папка, Город и тому подобные. Классы сущностей могут наследовать другие классы сущностей, а все они наследуют базовый класс Сущность. Поддерживается множественное наследование.

Классы сущностей делятся на *обычные*, *первичные*, *абстрактные* и *абстрактные первичные*. Далее под абстрактными мы будем подразумевать абстрактные и абстрактные первичные классы, а под первичными — первичные и абстрактные первичные классы. Сущность не может принадлежать к абстрактным классам, но может принадлежать к их потомкам.

Первичные классы используются для определения смысловых категорий сущностей. Если сущность принадлежит к какому-либо классу сущностей, то либо этот класс должен быть первичным, либо один из родительских для него классов должен быть первичным. Сущность не может принадлежать прямо или косвенно к нескольким первичным классам, за исключением случая, когда один первичный класс наследует напрямую или через другие классы в иерархии другой или другие первичные классы. Например, сущность может принадлежать одновременно к классам Пользователь и Сотрудник, так как в их иерархии первичным является только класс Персона. Но сущность не может одновременно принадлежать к классам Пользователь и Документ, так как классы Персона (который наследует класс Пользователь) и класс Документ — первичные, но ни один из классов Пользователь и Документ не наследует другой.

Сущность может изменять классы, к которым она принадлежит в своих новых редакциях. Однако при всех таких изменениях сущность не может примкнуть прямо или косвенно к новому первичному классу или покинуть какой-либо первичный класс. Например, можно из Персоны сделать Сотрудника, но нельзя из той же Персоны сделать Документ.

Возможность одновременной принадлежности сущности к нескольким классам позволяет, например, создать сущность, реализующую классы Сотрудник и Пользователь одновременно, вместо того, чтобы заводить две разных сущности в разных справочниках и дублировать тем самым данные.

У классов сущностей есть поля. Сущность, принадлежащая к классу, в своих свойствах может содержать значения этих полей. Каждый класс наследует также все поля от родительских классов. Более того, одно и то же поле может быть использовано в неродственных классах, чтобы не заводить однотипные поля для разноплановых классов, которые нет смысла или возможности группировать в вышестоящий общий класс-предок с этим полем.

Часть классов сущностей используется для пользователей и групп, документов, папок, метаданных и прочих служебных целей. Другие же классы используются для создания справочников. В системе есть абстрактный класс Справочник, который должны наследовать добавляемые в систему специальные справочники.

Если у класса-справочника есть классы-потомки, то в таком справочнике могут находиться записи, принадлежащие и к этому классу, и к его потомкам, т. е. в одном справочнике могут находиться сущности несколько отличающегося друг от друга вида. Разумеется, не следует

пытаться группировать в общий справочник разнородные не используемые совместно объекты.

3.2.1. Древоподобные справочники

Содержащие большое количество элементов справочники рекомендуется делать древоподобными. Такая организация справочника может упростить и ускорить работу с данными. Также могут существовать особые виды справочников, в которых их древоподобное представление является наглядным и естественным.

Древоподобные справочники делятся на:

- Справочники со строгой структурой дерева. В таком справочнике каждый элемент может находиться в дереве только в одном узле, а в дереве не может быть циклов.
- Обычные древоподобные справочники. В этом случае в дереве элементов справочника также не может быть циклов, но элементы можно включать в несколько мест в дереве. В некоторых случаях эта возможность может быть полезна.

3.2.2. Табличные части

Особым видом справочников являются табличные части.

Табличная часть — набор полей, принадлежащих одной сущности, позволяющие определять более одной записи в качестве свойств сущности.

Можно представить табличные части в виде справочника, строки которого должны обязательно ссылаться на какую-либо сущность определённого класса. Доступ к этому справочнику осуществляется через свойства сущности, отдельного вызова для справочника со всеми табличными частями нет.

3.3. Типы данных полей

Для каждого поля в классе сущностей указан *тип данных*. В системе используются следующие типы данных:

STRING Строка символов. Строка может быть пустой. Максимальная длина строки может зависеть от используемой базы данных. В конфигурации со встроенной базой данных нет специфических

ограничений на длину строки, её размер ограничен объёмом доступной памяти и другими системными параметрами.

TEXT Используется для записи текста в несколько строк. На карточках для полей этого типа используется соответствующий многострочный компонент для ввода текста. Ограничение на длину не менее такового для типа `STRING`.

INTEGER Используется для представления целых чисел. Значения поля могут лежать в диапазоне $[-2\,147\,483\,648, 2\,147\,483\,647]$.

FLOAT Для рациональных чисел. Представимые значения соответствуют формату `double` стандарта IEC 60559.

BOOLEAN Используется для булевых (логических) значений.

DATE Для записи даты.

TIMESTAMP Используется для сохранения отметок времени с высокой точностью. Значения данного типа можно отображать с учётом часовых поясов.

ENUMERATION Хранит одно значение из списка predefined. Как правило, не следует создавать новые поля такого типа, а использовать вместо этого справочники. Тем не менее, есть несколько predefined системных свойств такого типа.

REGISTER_ENTRY Хранит ссылку на запись в реестре.

REGISTER_ENTRY_SET Хранит множество ссылок на записи в реестрах.

PHONE_NUMBERS Хранит список телефонных номеров.

Помимо этих типов данных существуют специальные типы данных для представления *отношений* между сущностями, а также внутрисистемные типы.

3.3.1. Отношения

Система поддерживает различные виды отношений между сущностями. Для хранения отношений в системе существуют восемь специальных типов данных для полей. Они делятся по *виду отношения* на четыре группы:

O2O Один-к-одному. Представляет собой *ссылку* на сущность.

M2O Многие-к-одному. Также представляет собой *ссылку* на сущность.

O2M Один-ко-многим. Является *множеством ссылок* на сущности.

M2M Многие-ко-многим. Аналогично является множеством ссылок на сущности.

Обилие видов отношений связано в том числе и с тем, что отношения могут храниться как только с одной из сторон отношения, так и с обеих. Если отношение хранится с обеих сторон, то при изменении его с одной из них, ссылки с другой стороны автоматически обновляются правильным образом. Такое отношение удобно тем, что с любой из его сторон можно быстро получить связанные сущности, просто запросив значение соответствующего свойства. Однако изменение отношения с одной стороны вызывает порождение новых редакций и в связанных сущностях, что может быть как полезно, так и вредно. В отношениях, хранящихся с одной стороны, получение информации о связанных сущностях с другой стороны затруднено, поскольку для хранения этой информации соответствующее поле отсутствует. Эту информацию можно получить, используя поиск по библиотеке.

Помимо этого, каждый из этих четырёх видов делится на два под-вида: *главенствующий* (MASTER) и *подчинённый* (SLAVE). Это деление важно для двусторонних отношений и сказывается на проверке прав доступа для изменения отношения. Важно заметить, что любая из сторон отношения может быть как того, так и другого вида, в том числе обе стороны могут быть главенствующими или подчинёнными. Особенность проверки доступа заключается в том, что при изменении двустороннего отношения, вторая сторона, в том случае, если она обозначена подчинённой, может быть автоматически обновлена даже при отсутствии на неё прав доступа. Если же вторая сторона является главенствующей, то наличие прав доступа на изменение будет проверено и для связанных сущностей, затронутых в изменяемом отношении. Главенство или подчинённость с той стороны, с которой изменяется связь, значения не имеет.

Исходя из этого, мы получаем 8 типов данных для отношений:

MASTER_O2O	SLAVE_O2O
MASTER_M2O	SLAVE_O2M
MASTER_O2M	SLAVE_M2O
MASTER_M2M	SLAVE_M2M

При задании двустороннего отношения важно, чтобы типы данных для поля-отношения с обеих сторон сочетались друг с другом. К примеру, если с одной стороны отношение указано как один-ко-многим, то с другой стороны оно должно быть обозначено, как многие-к-одному.

При задании одностороннего отношения в качестве связанного поля указывается какое-либо иное поле связанного класса сущностей, не являющееся отношением, но имеющее какое-либо смысловое значение. В абсолютном большинстве случаев для этой цели выбирается свойство Полное имя.

3.4. Права доступа

Большинство объектов системы использует унифицированную схему настройки *прав доступа*. Права доступа назначаются для произвольных пользователей и групп пользователей и делятся на права на чтение, на запись, и на управление доступом. Право на запись включает в себя право на чтение, а право на управление доступом обеспечивает полные права доступа. Группы могут включать в себя другие группы, при этом права доступа, назначенные на группу, распространяются не только на непосредственных участников группы, но и на участников других входящих в неё напрямую или через другие группы групп.

3.4.1. Специальные пользователи

В системе существует три специальных пользователя, имеющих полные права доступа к любому объекту системы независимо от настроенных на этот объект прав доступа. Это системный администратор root и внутренние пользователи самой системы документооборота и маршрутизатора.

3.4.2. Специальные группы

В системе есть две специальные группы пользователей. Это администраторы системы и глобальные читатели. Участники группы администраторы могут, как и пользователь root, выполнять любые действия, даже если для них не настроены какие-либо права доступа. Учётные записи, имеющие права доступа администраторов, не должны использоваться в повседневной работе в системе. Участники группы глобальные читатели имеют доступ на чтение ко всем объектам системы, независи-

мо от настроенных на них правах доступа и могут иметь дополнительные права доступа согласно этим настроенным правам.

Специальные группы и специальные пользователи имеют соответствующие права ко всем объектам системы, поэтому не рекомендуется задавать для этих групп явные права к каким-либо объектам.

3.4.3. Права доступа руководителей

Руководитель какой-либо группы пользователей также имеет права доступа, которые есть хотя бы у одного из участников руководимых групп. Эти права также распространяются на руководителя этого руководителя и далее по цепочке. Это означает, что руководитель вправе выполнить любое действие, которое вправе выполнять какие-либо из его подчинённых.

Если у пользователь является специальным пользователем (Системный администратор) или входит в специальную группу (Администраторы, Глобальные читатели), то его специальные права не распространяются на его руководителя (управляющего группой, в которую входит пользователь). Однако, если специальным пользователям или группам заданы явные права к объекту, то эти права распространяются и на руководителя пользователя.

3.5. Права доступа объектов системы

3.5.1. Права доступа сущностей

Каждая сущность в системе имеет свои права доступа. Сущность сама по себе доступна независимо от прав доступа, но непосредственно в ней не содержится никакой информации, кроме её уникального идентификатора. Все свойства сущности находятся в её редакции. Права доступа определяют возможность работы с редакциями сущности.

Наличие этих прав не всегда означает, что редакция сущности доступна. Классы сущностей могут накладывать дополнительные ограничения на работу с сущностями, относящимися к данным классам.

3.5.2. Права доступа классов сущностей

Каждый класс сущностей сам по себе является сущностью. Его собственные права доступа, как и права доступа свойств, не должны настраиваться без особой необходимости. По умолчанию они могут ре-

дактироваться только администраторами системы и могут читаться всеми. Не гарантируется корректная работа системы при наличии в ней классов сущностей или свойств, недоступных всем пользователям на чтение.

У класса сущностей есть несколько дополнительных свойств, в которых задаются дополнительные права доступа.

Права доступа к содержимому ограничивают права доступа пользователей к сущностям, реализующим этот класс. Какие бы права доступа не были настроены на конкретную сущность, результирующие права не будут выше, чем указанные в классе в этом свойстве.

Права по умолчанию указывают, какие права будут у новых создаваемых сущностей, принадлежащих к этому классу. Пользователь обычно (но не всегда) может дополнительно откорректировать права доступа перед сохранением новой сущности. Если сущность создаётся в папке или в реестре, то она также получит права доступа от этой папки или от этого реестра.

Права доступа к содержимому по умолчанию имеют смысл только для классов сущностей, наследующих класс Контейнер. Для новых создаваемых сущностей, принадлежащих этому классу, права доступа к содержимому изначально будут настроены в соответствии со значением этого свойства.

В справочниках обычно нет смысла в индивидуальной настройке к каждому конкретному элементу справочника. В этом случае у класса-справочника задаются правильные права доступа к содержимому (записям) справочника. При этом нежелательно давать кому-либо полные права на управление доступом. Для тех, кто создаёт новые записи справочника и редактирует существующие следует ограничиться правом доступа на запись. Это не позволит пользователям ошибочно изменять права доступа на записи этого справочника. В правах доступа по умолчанию этого класса можно указать такие же права, но правильнее задать полные права доступа Пользователям. Поскольку в любом случае без прав доступа к содержимому справочника доступ предоставлен не будет, это не создаст угрозу безопасности. Зато впоследствии, при необходимости изменить права доступа к справочнику, достаточно будет изменить права доступа к содержимому класса сущности этого справочника и не трогать права доступа записей справочника.

3.5.3. Права доступа реестров

Собственные права доступа реестров не подлежат корректировке, значения по умолчанию корректны.

Права доступа к записям реестра настраиваются в свойстве Права доступа к содержимому.

При создании новых сущностей из реестра к их начальным правам доступа будут добавляться права, указанные в значении свойства Права доступа по умолчанию. Например, туда можно добавить доступ на чтение для группы Пользователи, чтобы все зарегистрированные из данного реестра сущности по умолчанию были доступны для всех на чтение; и добавить доступ на запись тем, кто должен быть в состоянии редактировать зарегистрированные в реестре сущности.

3.5.4. Права доступа к моделям процессов

Как обычно, собственные права доступа моделей не следует изменять.

Права на запуск процесса определяют, какие пользователи могут запускать процессы по этой модели. Конкретные указанные права доступа значения не имеют, достаточно доступа на чтение.

Права доступа к содержимому определяют, какие пользователи и как могут просматривать процессы по этой модели и управлять ими. Помимо этого права доступа для работы с конкретным процессом требуются и права доступа на сам процесс.

Права доступа по умолчанию определяют, какие права доступа будут изначально получать новые процессы, запускаемые по этой модели. Они могут быть откорректированы пользователем при необходимости.

3.5.5. Права доступа процессов

Процесс в системе не является обычной сущностью, но, тем не менее, имеет обычные права доступа. Право доступа на чтение означает, что можно просматривать информацию о процессе. Право доступа на запись даёт возможность изменять вложения процесса. Права на управление доступом дополнительно дают возможность изменять права доступа процесса, переназначать задачи по процессу и принудительно прерывать процесс.

3.5.6. Права доступа к вложениям процесса из задачи

У каждой задачи индивидуально настроены права доступа к вложениям процесса для пользователя, выполняющего эту задачу. Права настраиваются отдельно для основных и дополнительных вложений.

У пользователя может не быть доступа к вложениям, может быть доступ на просмотр списка вложений, на просмотр и добавление новых элементов в список и полный доступ, включая удаление существующих вложений.

Эти права доступа указываются автоматически скриптом модели процесса при создании задачи и не подлежат дальнейшему изменению.

Примечание. Пользователь, имеющий доступ к процессу, может управлять его вложениями также с карточки самого процесса. В этом случае используются права доступа пользователя на этот процесс. В то же время права доступа на процесс игнорируются при управлении вложениями с карточки задачи, на ней актуальны только права доступа к вложениям, указанные для этой задачи.

3.5.7. Права доступа отчётов

Собственные права доступа отчётов не следует изменять.

Права доступа к содержимому определяют, какие пользователи могут генерировать данный отчёт. Для этого достаточно доступа на чтение. Важно заметить, что в древовидном справочнике Отчёты некоторые узлы дерева могут применяться лишь для группировки других отчётов в подузлы. Если на такой узел нет прав доступа к содержимому, то ни этот узел, ни его подузлы не будут показаны пользователю. Таким образом на группирующие узлы-отчёты также следует настраивать права доступа к содержимому правильным образом.

3.6. Реестры

Реестры создаются из справочника Реестры. Помимо имени, у каждого реестра должны быть указаны формат генерации номера и период.

Как правило, внутри полного номера есть порядковый номер записи. Этот номер может быть со сквозной нумерацией или может быть с нумерацией, действующей в течение определённого периода, а потом начинающейся с начала. Это поведение определяется значением свойства Период.

Общий формат номера определяется шаблоном. Формат шаблонов приведён в разделе 3.6.1. Для реестров с несквозной нумерацией в нём, помимо порядкового номера, должны быть защиты и сведения о периоде.

С каждой записью в реестре может быть ассоциирована какая-либо сущность, как могут существовать и записи без связанных сущностей.

Система умеет автоматически раскладывать регистрируемые сущности по папкам согласно значению свойства Путь, представляющего собой шаблон имён генерируемых папок, аналогичный шаблону самого номера. Можно генерировать вложенные папки, для их разделения в шаблоне используется символ /. Корневая папка, внутри которой будут созданы папки для помещения в неё сущности, указывается в свойстве Папка. Таким образом, для раскладывания регистрируемых сущностей по папкам свойства Путь и Папка заполняются совместно.

Кроме того, можно раскладывать по папкам и сущности, которые в своих свойствах ссылаются на записи реестра. Для этого на карточке реестра есть табличная часть. В её элементах указывается имя очередного правила, классы сущностей ссылок и шаблон пути. Можно завести несколько правил для разных классов сущностей. Например, договоры могут попадать в папку с соответствующим номером, письма, ссылающиеся на них в подпапку Письма, а дополнительные соглашения — в подпапку Дополнительные соглашения.

3.6.1. Шаблон для генерации номера записи реестра и путей к папкам

Шаблон для генерации номера по реестру задается администратором системы для каждого реестра и определяет, в каком виде будет представлен пользователю номер по данному реестру. Шаблон для генерации номера по реестру представляет собой строку, содержащую *выражения* и строковые символы. При формировании номера *выражения* заменяются своими значениями, а строковые символы остаются в своём первоначальном виде. В системе предусмотрены следующие виды *выражений*:

$\{\text{number:n}\}$ заменяется на последовательно генерируемый в рамках периода нумерации номер по данному реестру. **n** — целое число, определяющее количество символов в номере. Номер при необходимости дополняется ведущими нулями до **n** символов.

$\{\text{property:UUID}\}$ (для отличных от даты/времени и от ссылочных свойств) заменяется на значение свойства регистрируемой сущности, имеющего идентификатор **UUID**

Например,

$\{\text{property:00000000-0000-0001-0000-000000000003}\}$

заменится на полное имя регистрируемой сущности.

`#{property:UUID:format}` (только для свойств типа дата/время) заменяется на значение свойства (имеющего идентификатор **UUID**) регистрируемой сущности согласно шаблона, **format** — строка-шаблон, определяющая, какой компонент даты(времени) должен быть выведен на месте данного *выражения*.

Например,

`#{property:00000000-0001-0016-0000-000000000001:MMMM}`

заменится на полное название месяца из свойства дата документа регистрируемой сущности.

`#{property:UUID1~UUID2}` (только для ссылочных свойств) заменяется на значение свойства (имеющего идентификатор **UUID2**) сущности, связанной со свойством (имеющим идентификатор **UUID1**) регистрируемой сущности.

Например,

`#{property:00000000-0001-0017-0000-000000000001~00000000-0001-0006-0000-000000000004}`

заменится на ИНН организации, указанной в качестве свойства контрагент у регистрируемой сущности.

`#{register:UUID}` (для отличных от даты/времени и от ссылочных свойств) заменяется на значение свойства (имеющего идентификатор **UUID**) используемого реестра.

Например,

`#{register:00000000-0000-0001-0000-000000000003}`

заменится на полное имя реестра.

`#{register:UUID1~UUID2}` (только для ссылочных свойств) заменяется на значение свойства (имеющего идентификатор **UUID2**) сущности, связанной со свойством (имеющим идентификатор **UUID1**) используемого реестра.

Например,

`#{register: 00000000-0001-0015-0000-000000000001~00000000-0000-0001-0000-000000000003}`

заменится на полное имя организации, к которой привязан реестр.

`#{date:format}` заменяется на текущую дату (или компонент даты (времени)) согласно шаблона, **`format`** — строка-шаблон, определяющая, какой компонент даты (времени) должен быть выведен на месте данного *выражения*.

Доступные форматы приведены в таблице 3.1.

Таблица 3.1: Форматы даты (времени) для шаблонов реестров

Шаблон	Выходная информация	Пример
dd.MM.yuuu	Дата в формате день.месяц.год	28.03.2012
HH:mm:ss	Время (часы:минуты:секунды)	15:05:19
uuuu	Год четырьмя цифрами	2012
uu	Год двумя цифрами	12
MMMM	Полное название месяца (родительный падеж)	марта
LLLL	Полное название месяца	Март
MMM	Краткое название месяца (родительный падеж)	мар
LLL	Краткое название месяца	Мар.
MM	Месяц двумя цифрами	03
dd	День месяца, всегда двумя цифрами	28
d	День месяца одной или двумя цифрами	28
D	День в году	088
W	Номер недели в месяце	5
w	Номер недели в году	14
e	Номер дня недели (1 — Пн, ..., 7 — Вс)	3
eee или E	Краткое название дня недели	Ср
eeee	Полное название дня недели	среда

3.7. Маршрутизация

Движение документов в системе осуществляется с помощью маршрутизации. Оно осуществляется в рамках отдельных *процессов*. Процесс создаёт некоторое количество *задач* для пользователей системы. В ходе выполнения или просрочки выполнения задач процесс создаёт новые задачи. Правила, по которым работает процесс, определяются в *модели процесса*. *Виды активности* опционально объединяют модели и

принадлежащие к ним процессы по категориям, а также при необходимости могут использоваться для определения правил автоматической нумерации процессов в рамках конкретного вида активности.

Модели процессов создаются и настраиваются при разработке библиотеки. Эти модели отражают те или иные процессы делопроизводства.

Пользователи, имеющие соответствующие права доступа, могут создавать новые процессы по модели. При этом пользователем указываются специфичные для конкретной модели параметры запуска и прикрепляются необходимые документы.

Запущенные по модели процессы выполняются автоматически согласно прописанным в модели правилам. Выполнение процесса в целом при этом зависит от выполнения пользователями задач по процессу.

Процесс также может быть запущен без модели. Задачи в таких процессах создаются только вручную.

При наличии прав доступа можно наблюдать и управлять запущенным процессом, переназначать его задачи, изменять статус процесса, а также перезапускать его по той же самой или иной модели.

У каждого процесса есть списки основных и дополнительных *вложений процесса*. Вложениями могут являться любые сущности. Если сущность находится в основных вложениях каких-либо процессов, можно получить информацию об этих процессах из панели детализации сущности.

В дополнительные вложения помещаются сущности, поиск процессов по которым не имеет смысла и которые носят вспомогательный характер. Например, туда можно помещать различные регламентные документы, стандарты и иные сущности, работа с которыми осуществляется в рамках данного процесса, но которые, как правило, не являются результатом выполнения этого процесса, а существуют независимо от него.

Пользователь, получивший задачу, может видеть и изменять вложения процесса с карточки задачи, если модель процесса дала ему такой доступ. В зависимости от рода задачи списки вложений могут быть недоступны, доступны только на просмотр, на просмотр и добавление новых вложений и с полным доступом, включая удаление. Права доступа настраиваются для основных и дополнительных вложений отдельно. Это позволяет разделять роли пользователей. Имеющие доступ к самому процессу пользователи также могут управлять вложениями процесса с карточки самого процесса.

В системе также присутствуют *комментарии процесса*. Они доступны как пользователям, имеющим задачи по процессу, так и наблюдате-

лям самих процессов. Пользователи могут добавлять новые комментарии, но не могут изменять и редактировать существующие. Комментарии предназначены для обсуждения рабочих вопросов по конкретному процессу. Маршрутизатор также может оставлять комментарии от своего имени для информирования пользователей о статусе процесса и о произошедших в нём значимых событиях.

Полученная по процессу задача может иметь крайний срок выполнения, но существуют и бессрочные задачи. За некоторое определённое моделью время до наступления крайнего срока выполнения задачи она становится критической, чтобы проинформировать пользователя о необходимости срочного завершения задачи.

Некоторые задачи после истечения крайнего срока могут допускать выполнение с опозданием, при этом задача находится в льготном периоде. После его истечения задача становится неактуальной.

Задачи могут быть завершены пользователем с тем или иным статусом, определяющим дальнейшие действия по процессу. Например, можно согласовать документ или отправить его на доработку. Задача может быть прервана самим маршрутизатором в результате её просрочки, выполнения других задач пользователями или по наступлению иных событий. Задача также будет прервана в результате её переназначения и при завершении породившего его процесса.

Как правило, по достижении своей цели, процесс завершается автоматически согласно правилам модели. Процесс также может быть остановлен вручную путём перевода его в одно из остановленных состояний пользователем, имеющим достаточные права доступа к процессу. Остановленный процесс может быть запущен повторно.

Руководители групп пользователей (в качестве группы может выступать в том числе и подразделение компании) также в состоянии проводить наблюдение за выполнением задач руководимыми ими пользователями.

3.8. Отчёты

Новые виды отчётов настраиваются в системе в справочнике Отчёты. Построение интерфейса запроса параметров отчёта у пользователя, выборка данных для отчёта и генерация документов осуществляется с помощью языка JavaScript. Выборка данных осуществляется на стороне сервера под учётной записью системы, поэтому в отчёт могут быть включены все нужные объекты системы, в том числе и недоступные пользователю, выгружающему данный отчёт. Это нужно для кор-

ректности и однозначности полученного отчёта, но для отчётов, которые могут содержать необщедоступные данные требуется правильная настройка прав доступа к содержимому, чтобы не имеющие допуска пользователи не могли получать конфиденциальную информацию через отчёты.

Справочник Отчёты — древовидный. Это позволяет удобно сгруппировать большое количество разнообразных отчётов. При этом узлы дерева могут генерировать самостоятельные общие отчёты, а могут и использоваться только для группировки отчётов по категориям.

3.9. Настройки рабочего места

В данном разделе описаны хранимые настройки рабочего места пользователя.

Настройки рабочего места необходимы для сохранения и восстановления предпочитаемых пользователем размеров окон приложения, настроек столбцов (их порядка и сортировок) в таблицах, фильтров справочников и прочего. Все настройки сохраняются автоматически. Например, при изменении размеров некоего окна его размеры автоматически сохраняются, и при последующем открытии данное окно откроется с теми размерами, которые пользователь установил в прошлый раз.

В большинстве случаев работа с настройками полностью автоматизирована и не требует вмешательства пользователя или администратора системы. Тем не менее, в клиенте предусмотрена возможность чтения имеющихся сохранённых настроек и, в случае необходимости, их редактирования, либо удаления. Обратите внимание на то, что после удаления настройки система будет использовать вместо этой настройки значения по умолчанию. Так, если удалить настройки размеров некоторого диалога — данный диалог откроется с размерами по умолчанию, но при последующем изменении его размеров настройка вновь будет сохранена с новым значением размеров.

Просмотр/редактирование/удаление настроек доступны из главного меню клиента: Меню — Дополнительно — Сохранённые настройки

3.9.1. Пользовательские настройки

Пользовательские настройки — это настройки связанные с некоторой учётной записью пользователя. Этот тип настроек хранится на сервере и не зависит от рабочего места. Т. е. два клиента, запущен-

ные на разных рабочих станциях под одной учётной записью пользователя, имеют общие пользовательские настройки. Этот тип настроек удобен для тех настроек, которые относятся непосредственно к работе с данными.

К этому типу настроек относятся:

- сохранённые поисковые запросы;
- фильтры справочников и фильтры ссылочных компонентов;
- предпочитаемые пункты сворачиваемых меню;
- сохраненные значения периодов (для реестров, для завершённых задач и процессов);
- предпочитаемое действие по двойному щелчку мышью;

а также некоторые другие.

Пользователи системы могут просмотреть/редактировать/удалять только свои настройки. Пользователи с правами администратора могут просмотреть/редактировать/удалять настройки любого из выбранных пользователей. Кроме того, администраторам доступен пункт меню Очистить пользовательские настройки всех пользователей, который рекомендуется применять только лишь в крайних случаях.

3.9.2. Локальные настройки

Локальные настройки — это настройки связанные с конкретным рабочим местом пользователя. Этот тип настроек хранится на конкретной рабочей станции и привязан скорее к представлению данных, чем к самим данным. Так например, предпочитаемые размеры окон могут различаться в зависимости от размеров экрана, а каталог, открываемый в диалоге открытия/сохранения, отличается для разных рабочих станций. Поэтому лучший вариант хранения таких настроек именно рабочая станция.

К этому типу настроек относятся:

- размеры всевозможных окон (тут необходимо заметить, что размеры одних и тех же диалогов для разных классов сущностей хранятся отдельно. Так например, для окна карточки Письма может потребоваться окно большего размера, чем для карточки Папки, ввиду того, что карточка Письма имеет большее количество атрибутов);

- настройки отображения таблиц (отображаемые/скрытые столбцы, их порядок, их ширина, их сортировка);
- путь к каталогу, открываемому в диалогах открытия/сохранения;
- включение/отключение предпросмотра в главном окне клиента, в окне редактирования карточек;

и другие подобные.

Так как этот тип настроек индивидуален для каждой станции, то у администратора нет никаких дополнительных инструментов по их редактированию. Просмотр/редактирование/удаление этих настроек доступны только с локального рабочего места.

Глава 4

Пользователи и группы

4.1. Описание данных пользователей и групп

4.1.1. Аккаунты пользователей

Класс Аккаунты пользователей наследует класс Персона. У этого класса имеются дополнительные свойства Имя входа и Активен. Также для пользователя задаётся пароль или включается внешняя аутентификация. Имя входа используется в окне входа в систему. У каждого пользователя значение этого свойства должно быть уникально. Имя входа чувствительно к регистру. Для удобства рекомендуется использовать в имени только символы в нижнем регистре.

Новые пользователи автоматически добавляются в группу Пользователи.

Для отключения учётной записи пользователя (например, при увольнении сотрудника) следует снять галочку у значения свойства Активен. Следует заметить, что если пользователя исключить из группы Пользователи вручную, то учётная запись также будет заблокирована независимо от того, отмечено ли свойство Активен.

У каждого пользователя также должен быть задан пароль для входа или указана внешняя аутентификация.

4.1.2. Группы

Группы применяются для структурирования и упрощения работы с пользователями. У группы есть дополнительные свойства Руководители группы и Члены группы. Классы организации и подразделения

являются потомками класса Группа и могут быть использованы для построения организационной структуры предприятия.

Членами группы могут быть другие группы, любые персоны и потомки этих классов. Настраиваемые для группы права доступа применяются на всех членов группы.

Руководителями группы могут быть персоны и потомки класса Персона. Руководителей может быть несколько. Персона, являясь руководителем группы, получает набор прав, включающий множество прав доступа каждого члена группы. Иначе говоря, руководитель группы видит всё, что видит хотя бы один из группы.

4.2. Управление пользователями и группами

Управление осуществляется из клиента системы. Для этого следует зайти в систему uBase под учётной записью одного из администраторов.

На вкладке Справочники аккордеона следует нажать кнопку Учётные записи. Обратите внимание, что все операции над пользователями и группами не будут сохранены, пока в этом диалоге не будет нажата кнопка ОК.

В верхней части диалога находится таблица пользователей, в нижней — таблица групп пользователей. Пользователей и группы пользователей можно добавлять и редактировать в них обычным образом.

При добавлении нового аккаунта пользователя в систему нельзя сразу назначить ему пароль или назначить иную аутентификацию. Следует добавить и настроить всех нужных пользователей и сохранить эти изменения. После этого аутентификацию пользователей можно будет задавать из их контекстного меню. Для этого можно открыть этот диалог повторно или найти пользователей поиском или иным образом. Настройки аутентификации пользователей могут изменять администраторы системы. Для каждого пользователя администратор задаёт пароль или указывает Kerberos-аутентификацию. Для Kerberos-аутентификации автоматически подставляется имя входа пользователя из системы uBase. Если у данного пользователя имя входа в uBase отличается от системного, то следует отредактировать имя для аутентификации на внешнем сервере. При этом для входа в uBase по-прежнему будет использоваться указанное на карточке пользователя имя входа. Kerberos-аутентификация не может быть использована для системного пользователя root.

Пользователи, имеющие настроенный в самой системе uBase пароли, могут также изменять свои пароли самостоятельно из клиента системы. Эта возможность недоступна для пользователей, у которых настроена внешняя аутентификация.

Также в аккаунт пользователя можно преобразовать любую зарегистрированную в системе персону. Например, можно открыть справочник Сотрудники и в контекстном меню нужного сотрудника, не являющегося ещё пользователем, вызвать команду Реализуемые классы. В диалоге выбора к имеющимся классам добавить класс Аккаунт пользователя. Если среди имеющихся классов есть класс Персона, то этот класс следует убрать из выбранных, поскольку любой пользователь также является и персоной. Однако при преобразовании сотрудника в аккаунт пользователя не следует удалять из выбранных класс Сотрудник, так как пользователь не является автоматически сотрудником. После этого нужно будет доопределить недостающие значения свойств. Как минимум, пользователь должен быть активен, состоять в группе Пользователи, а также ему следует после сохранения изменений задать новый пароль из контекстного меню.

Работа с группами, подразделениями и организациями ведётся аналогичным образом. Группы создаются в нижней части таблицы, и каждая группа может быть преобразована в Подразделение, Организацию и Нашу организацию.

Сущности классов Группа и Наша организация отображаются в списке доступных узлов для настройки прав доступа. Они являются корневыми узлами, и разворачиваются с отображением дочерних членов группы. Остальные классы, наследующие класс Персона, располагаются в узле Дополнительно.

Наша организация может включать дочерние организации и подразделения с необходимым количеством уровней. Выстроенная организационная структура предприятия может использоваться при настройке прав доступа. Настройка прав доступа к объектам может осуществляться в хранилище документов или с использованием средств маршрутизации. Маршрутизация позволяет назначать права доступа к вложениям с восходящим движением по организационной структуре.

4.2.1. Замещение пользователей

Замещение позволяет входить в систему от имени другого пользователя. Имя входа и пароль этого пользователя не требуются. Все действия в системе будут производиться вошедшим пользователем от имени замещаемого пользователя. В системе есть два вида замещения.

Первый вид замещения доступен учетной записи root и членам группы Администраторы. Вход от имени другого пользователя осуществляется из пункта контекстного меню "Войти как..." аккаунта пользователя.

Второй вид замещения доступен для любого пользователя системы. Создание такого замещения доступно в маршрутизации. Модель процесса создания замещения должна соответствовать регламентам предприятия. В модели доступно создание новых замещений и их аннулирование. Для каждого замещения определяются Замещающий, Замещаемый, Дата начала замещения и Дата окончания.

После создания замещения, замещающий может войти от имени замещаемого с использованием пункта главного меню "Войти как...". При наличии действующих замещений пользователю предлагается для выбора список замещаемых, при отсутствии выдаётся соответствующее предупреждение.

Глава 5

Метаданные системы

5.1. Классы сущностей и свойства

Различные виды документов и справочники в системе uBase являются классами сущностей.

Их добавление осуществляется из клиента системы, запущенного под учётной записью одного из администраторов системы. На вкладке Справочники аккордеона следует нажать кнопку Управление.

Никакие изменения не будут сохранены в систему, пока не будет нажата кнопка ОК в открывшемся диалоге.

Для добавления нового вида документа следует в дереве выбрать вид-предок и нажать кнопку добавить. Например, различные виды приказов, вероятно, должны наследовать класс Приказ. Для создания принципиально нового вида документа следует в качестве предка выбрать класс Документ.

Для добавления нового справочника в дереве, в большинстве случаев, следует выбрать узел Справочник. В отдельно взятых случаях новый справочник может наследовать другой справочник, в этом случае в дереве выбирается справочник-предок.

Важно понимать, что в данных классов сущностей и их свойств некоторые значения можно задать только при их создании и нельзя изменить впоследствии. Во всех случаях следует создать необходимые классы сначала на тестовом сервере или в тестовой библиотеке на основном сервере и проверить их соответствие замыслу. На рабочую библиотеку добавляются только проверенные решения.

У каждого класса сущностей нужно заполнить Полное имя, под которым он будет отображаться в системе, Тип класса сущностей, Права доступа к содержимому, Права доступа по умолчанию.

Для справочников обычно применяется первичный тип класса. Для видов документов может применяться первичный тип, а может и обычный, если у создаваемых документов такого вида требуется возможность последующего изменения вида. В сложных конфигурациях с наследованием могут применяться и другие типы классов.

В правах доступа к содержимому для справочников обычно указывается чтение для группы пользователи и запись для тех групп пользователей, которые будут заполнять этот справочник. Для справочников нежелательно разрешать для кого-либо управление доступом. В правах по умолчанию для справочников обычно указывается то же, что и в правах доступа к содержимому.

Права доступа к содержимому для видов документов, как правило, должны задаваться на изменение для всей группы Пользователи. Однако если новый вид документа сам по себе конфиденциальный, то ограничение для них может задаваться и на уровне класса в этом свойстве. В правах по умолчанию задаются необходимые начальные права доступа для всех документов такого вида. При создании документа, тем не менее, эти начальные права дополняются правами из папки или реестра при создании документа из них. Также на документ получает доступ сам создавший его пользователь.

После заполнения свойств класса в него следует добавить необходимые свойства. На карточке класса сущности в таблице для наглядности отображаются все его свойства, включая унаследованные от родительских классов. Если нажать кнопку редактирования, то в диалоге выбора свойств будут отображаться только настроенные непосредственно для этого класса свойства, без свойств родительских классов. В нём можно переносить в правую часть уже существующие в системе свойства и можно создавать и добавлять новые свойства.

После настройки свойств, вернувшись на карточку класса, можно продолжить его настройку.

Для классов-справочников имеет смысл настроить отображение таблицы. Эта настройка будет использоваться при просмотре справочника по умолчанию. Она должна подходить большинству пользователей. Каждый пользователь может для себя переопределить эту настройку под свои нужды.

Также желательно настроить внешний вид карточки класса в редакторе карточек 5.3.

5.2. Карточка свойств

Карточка свойств определяет расположение свойств сущности в отображаемом пользователю представлении.

По умолчанию все свойства сущности расположены в столбик, одно под другим. Но это расположение можно изменить для более удобного и наглядного представления данных на карточке с помощью редактора карточек 5.3.

Внимание! Приведённая ниже информация является справочной. В большинстве случаев настройку форм следует производить через конструктор форм. Но в случае необходимости возможен вариант редактирования настройки формы вручную через текстовый редактор.

Описание формы представляет собой XML-текст, хранящийся в свойстве Форма настраиваемого класса сущности, и выглядит примерно следующим образом:

```
<?xml version='1.0' encoding='UTF-8' standalone='no'?>
<form>
  <tab caption='Основные' order='0'>
    <property caption='Полное имя' column='0' row='0'
      width='3' height='1' enabled='true'
      uuid='00000000-0000-0001-0000-000000000003'/>
    <property caption='Фамилия' column='0' row='1'
      width='1' height='1' enabled='true'
      uuid='00000000-0000-000b-0000-000000000003'/>
    <property caption='Имя' column='1' row='1'
      width='1' height='1' enabled='true'
      uuid='00000000-0000-000b-0000-000000000001'/>
    <property caption='Отчество' column='2' row='1'
      width='1' height='1' enabled='true'
      uuid='00000000-0000-000b-0000-000000000004'/>
    <property caption='Телефоны' column='0' row='2'
      width='1' height='1' enabled='true'
      uuid='00000000-0001-000d-0000-000000000002'/>
    <property caption='E-mail' column='1' row='2'
      width='1' height='1' enabled='true'
      uuid='00000000-0001-000d-0000-000000000001'/>
    <property caption='Примечание' column='0' row='3'
      width='3' height='1' enabled='true'
      uuid='00000000-0000-0001-0000-000000000005' />
  </tab>
```

```
<tab caption='Адреса' order='1'>
  <property caption='Полное имя' column='0' row='0'
    width='1' height='1' enabled='true'
    uuid='00000000-0000-0001-0000-000000000003'>/>
  <property caption='Фактический адрес' column='0' row='1'
    width='1' height='1' enabled='true'
    uuid='00000000-0001-0020-0000-000000000001'>/>
  <property caption='Почтовый адрес' column='0' row='2'
    width='1' height='1' enabled='true'
    uuid='00000000-0001-0020-0000-000000000003'>/>
  <property caption='Юридический адрес' column='0' row='3'
    width='1' height='1' enabled='true'
    uuid='00000000-0001-0020-0000-000000000002'>/>
</tab>
</form>
```

`form` — корневой элемент XML-документа. Атрибутов не имеет.

`tab` описывает вкладки карточки свойств. В общем случае вкладок у карточки может быть более одной. Как правило, для большинства карточек достаточно единственной вкладки, но в некоторых случаях бывает удобно разнести группы свойств на несколько вкладок. В этом случае есть смысл группировать свойства во вкладки согласно смысловой нагрузки этих свойств. Кроме того, возможно отображать один и тот же компонент сразу на нескольких вкладках (в приведенном выше примере свойство Полное имя отображается на обеих вкладках).

Элемент `tab` имеет 2 атрибута. Значение атрибута `caption` указывает заголовок вкладки, а целочисленное значение атрибута `order` задаёт порядок вкладок для многовкладочных форм.

В случае с формами, имеющими одну вкладку, эти атрибуты игнорируются ввиду того, что закладки для единственной вкладки не отображаются.

`property` описывает расположение свойства внутри карточки/вкладки карточки. Имеет следующие атрибуты:

- `uuid` — идентификатор свойства (3.1.1), с которым ассоциирован данный компонент (данный атрибут является обязательным);
- `caption` — текстовая метка, с которой отображается данный компонент (если данный атрибут отсутствует, то по умолчанию будет использовано имя свойства данного компонента);

- `column` — колонка таблицы, в которой расположен данный компонент. Нумерация строк начинается с 0. То есть строка расположенная в n -ой колонке будет иметь значением данного свойства $n - 1$ (по умолчанию — 0);
- `row` — строка таблицы, в которой расположен данный компонент;
- `width` — количество занимаемых компонентом столбцов (по умолчанию — 1);
- `height` — количество занимаемых компонентом строк (по умолчанию — 1);
- `enabled` — позволяет деактивировать компонент, выставив в данном атрибуте значение `false` (по умолчанию — `true`).

5.3. Редактор карточек

Редактор карточек предназначен для настройки внешнего вида карточек определённого класса сущностей.

Открыть редактор карточки можно кнопкой Редактировать карточку на карточке класса сущности в редакторе метаданных 5.1.

Редактор карточки состоит из 4-х панелей: в верхней части формы расположена панель управления, слева — список доступных компонентов, по центру формы — панель редактора формы, и справа — панель свойств выделенного компонента.

Панель управления позволяет импортировать настройку формы во внешний файл /импортировать настройки формы из внешнего файла. Кроме того, здесь же находятся элементы для добавления, удаления, переименования и изменения порядка вкладок многовкладочной формы.

Список доступных компонентов отображает список свойств сущности, для которых на данную форму/вкладку ещё не вынесен компонент.

Панель редактора формы отображает представление формы с соответствующей компоновкой.

Панель свойств выделенного компонента отображает и позволяет редактировать атрибуты компоновки выделенного компонента.

Для помещения компонентов свойств на форму/удаления компонентов с формы можно перетаскивать компоненты из списка доступных свойств на форму и обратно с помощью мыши, либо двойным щелчком по компоненту.

Расположение элементов на форме можно изменять с помощью перетаскивания мышью. Для помещения компонента в новое место следует просто перетащить его мышью в любую свободную позицию. Для помещения компонента в новую строку/столбец между двумя существующими необходимо перетащить его в узкую область, находящуюся между соседними компонентами. При этом в указанную область будет добавлена новая строка/столбец, и компонент будет помещен в неё. Для изменения количества столбцов, занимаемых компонентом, следует ухватить компонент мышью за его правую границу и перетащить границу вправо для увеличения ширины компонента, либо влево для уменьшения его ширины.

Если компонент (в основном это касается компонентов занимающих более одного столбца в ширину/более одной строки в высоту) не может быть помещен в желаемую позицию, то он удаляется с формы и помещается в список доступных. При этом в панели свойств выделенного компонента отображается сообщение о невозможности размещения компонента в указанную позицию. В таком случае следует либо уменьшить ширину компонента до требуемых, либо поместить компонент в другую (свободную) позицию формы.

Панель свойств выделенного компонента позволяет задавать такие атрибуты компонента, как текстовая метка, с которой компонент отображается в карточке свойств; можно сделать компонент нередактируемым; для ссылочных компонентов возможно задать свойство связанной сущности, отображаемое в компоненте. Кроме того имеется возможность задавать размещение компонента на форме, с помощью установки свойств: колонка, строка, ширина, высота.

Глава 6

База данных системы

Внимание! Данная информация приведена в справочных целях. Система не поддерживает внесение изменений в базу данных администратором системы или внешними программными средствами. Любое стороннее изменение может нарушить целостность данных и привести к невозможности дальнейшей работы с библиотекой.

Модель данных системы не соответствует напрямую структуре базы данных. Для корректной работы с данными для партнёров — разработчиков решений на базе uBase предоставляется API системы. Внешние изменения данных производятся только через API.

6.1. Консоль базы данных

При использовании базы данных H2 на сервере доступна её консоль. Для её открытия нужно запустить на сервере веб-браузер и ввести адрес

```
http://127.0.0.1:15888/
```

В некоторых конфигурациях используется протокол https, в этом случае адрес изменится на

```
https://127.0.0.1:15888/
```

Если вы не уверены, какой именно протокол используется, попробуйте оба варианта.

При использовании сервера с ОС Linux без запущенного графического сервера следует удалённо подключиться к серверу с помощью `ssh`

с параметром `-X` и запустить на нём браузер удалённо, или использовать сервер VNC. По соображениям безопасности консоль базы данных не принимает подключений по сети.

При подключении следует указать URL базы данных, имя пользователя и пароль. Эти данные можно посмотреть в конфигурации конкретной библиотеки в менеджере системы.

Эта консоль может также применяться при использовании внешних баз данных, но в этом случае, как правило, предпочтительнее средства, предоставляемые используемой базой данных.

6.2. Таблицы

Типы данных и атрибуты столбцов таблиц приведены для встроенной базы данных. При использовании внешних баз данных возможны небольшие отличия.

6.2.1. ACL

Предназначена для хранения прав доступа.

```
CREATE CACHED TABLE ACL(  
    ACL BIGINT NOT NULL AUTO_INCREMENT PRIMARY KEY,  
    HASH BINARY(32) NOT NULL,  
    DATA VARBINARY NOT NULL  
);  
CREATE UNIQUE INDEX REVISION_HASH_IDX  
ON REVISION(HASH);
```

6.2.2. AUTHENTICATION

Предназначена для хранения данных о способе входа пользователей в систему.

```
CREATE TABLE AUTHENTICATION(  
    IDENTITY BIGINT NOT NULL AUTO_INCREMENT PRIMARY KEY,  
    USER UUID NOT NULL,  
    AUTHENTICATION_DATA VARCHAR NOT NULL,  
    REAL_EDITOR UUID NOT NULL,  
    EFFECTIVE_EDITOR UUID NOT NULL,  
    TIMESTAMP TIMESTAMP NOT NULL
```

```
);  
CREATE UNIQUE INDEX AUTHENTICATION_USER_TIMESTAMP_IDX  
ON AUTHENTICATION(USER, TIMESTAMP DESC);
```

6.2.3. CATALOG

Содержит записи о счётчиках реестров и нумеруемых в рамках вида активности процессов. Реестры и виды активности, в которых ещё не производилась регистрация с генерацией новых номеров не имеют записей в этой таблице.

Допускается увеличить значение счётчика (поле `COUNTER`), если требуется начать генерацию номеров не со значения 1. Сброс значения счётчика для начала нового периода выполняется автоматически сервером системы, для этого в реестре или виде активности должен быть указана соответствующая периодичность. Сброс будет производиться только при генерации номера в новом периоде.

```
CREATE TABLE CATALOG(  
    CATALOG UUID NOT NULL PRIMARY KEY,  
    COUNTER INTEGER NOT NULL  
);
```

6.2.4. CERTIFICATE

Предназначена для хранения известных системе сертификатов X.509.

```
CREATE TABLE CERTIFICATE(  
    IDENTITY BIGINT NOT NULL AUTO_INCREMENT PRIMARY KEY,  
    CERTIFICATE BLOB NOT NULL,  
    ISSUER BIGINT NOT NULL,  
    SERIAL VARCHAR NOT NULL  
);  
CREATE INDEX CERTIFICATE_ISSUER_IDX  
ON CERTIFICATE(ISSUER);  
CREATE INDEX CERTIFICATE_SERIAL_IDX  
ON CERTIFICATE(SERIAL);  
CREATE INDEX CERTIFICATE_ISSUER_SERIAL_IDX  
ON CERTIFICATE(ISSUER, SERIAL);
```

6.2.5. CERTIFICATE_STATE

Предназначена для хранения состояния доверия сертификатов.

```
CREATE TABLE CERTIFICATE_STATE(  
    IDENTITY BIGINT NOT NULL AUTO_INCREMENT PRIMARY KEY,  
    CERTIFICATE BIGINT NOT NULL,  
    STATE VARCHAR NOT NULL,  
    REAL_USER UUID NOT NULL,  
    TIMESTAMP TIMESTAMP NOT NULL  
);  
CREATE UNIQUE INDEX CERTIFICATE_STATE_CERTIFICATE_TIMESTAMP_IDX  
ON CERTIFICATE_STATE(CERTIFICATE, TIMESTAMP DESC);
```

6.2.6. CHECKOUT

Предназначена для хранения данных о выписанных документах.

```
CREATE TABLE CHECKOUT(  
    IDENTITY BIGINT NOT NULL AUTO_INCREMENT PRIMARY KEY,  
    ENTITY UUID NOT NULL,  
    STATE BIGINT NOT NULL,  
    REAL_EDITOR UUID NOT NULL,  
    EFFECTIVE_EDITOR UUID NOT NULL,  
    TIMESTAMP TIMESTAMP NOT NULL,  
    HOST UUID NOT NULL,  
    PATH VARCHAR,  
    REVOKED_BY_REAL UUID,  
    REVOKED_BY_EFFECTIVE UUID,  
    REVOKED TIMESTAMP  
);  
CREATE INDEX CHECKOUT_ENTITY_IDX  
ON CHECKOUT(ENTITY);  
CREATE INDEX CHECKOUT_REAL_EDITOR_EFFECTIVE_EDITOR_IDX  
ON CHECKOUT(REAL_EDITOR, EFFECTIVE_EDITOR);
```

6.2.7. ENTITY_CLASS

Вспомогательная таблица, содержащая информацию о классах сущностей в текущем состоянии.

```
CREATE TABLE ENTITY_CLASS(  
    IDENTITY BIGINT NOT NULL AUTO_INCREMENT PRIMARY KEY,  
    ENTITY UUID NOT NULL,  
    CLASS UUID NOT NULL,  
    ROOT BOOLEAN NOT NULL  
);  
CREATE UNIQUE INDEX ENTITY_CLASS_ENTITY_CLASS_IDX  
    ON ENTITY_CLASS(ENTITY, CLASS);  
CREATE INDEX ENTITY_CLASS_CLASS_ROOT_IDX  
    ON ENTITY_CLASS(CLASS, ROOT);
```

6.2.8. ENTITY_PHONES

Вспомогательная таблица для индексирования телефонных номеров.

```
CREATE TABLE ENTITY_PHONES(  
    IDENTITY BIGINT NOT NULL AUTO_INCREMENT PRIMARY KEY,  
    ENTITY UUID NOT NULL,  
    NUMBER VARCHAR NOT NULL,  
    ADDITIONAL VARCHAR  
);  
CREATE INDEX ENTITY_PHONES_ENTITY_IDX  
    ON ENTITY_PHONES(ENTITY);  
CREATE INDEX ENTITY_PHONES_NUMBER_IDX  
    ON ENTITY_PHONES(NUMBER);  
CREATE INDEX ENTITY_PHONES_NUMBER_ADDITIONAL_IDX  
    ON ENTITY_PHONES(NUMBER, ADDITIONAL);
```

6.2.9. ENTITY_REVISION

Предназначена для хранения списка редакций сущностей.

```
CREATE TABLE ENTITY_REVISION(  
    IDENTITY BIGINT NOT NULL AUTO_INCREMENT PRIMARY KEY,  
    ENTITY UUID NOT NULL,  
    STATE BIGINT NOT NULL,  
    REVISION BIGINT NOT NULL,  
    ACL BIGINT NOT NULL
```

```
);  
CREATE UNIQUE INDEX ENTITY_REVISION_ENTITY_STATE_IDX  
ON ENTITY_REVISION(ENTITY, STATE DESC);
```

6.2.10. EVENT

Предназначена для хранения редакций событий.

```
CREATE TABLE EVENT(  
    IDENTITY BIGINT NOT NULL AUTO_INCREMENT PRIMARY KEY,  
    EVENT UUID NOT NULL,  
    PARENT UUID,  
    USER UUID NOT NULL,  
    TYPE VARCHAR NOT NULL,  
    DESCRIPTION VARCHAR NOT NULL,  
    BEGIN_TIMESTAMP NOT NULL,  
    END_TIMESTAMP NOT NULL,  
    PLACE VARCHAR,  
    INCOMING BOOLEAN,  
    PHONE_NUMBER VARCHAR,  
    OUR_EXTERNAL_PHONE_NUMBER VARCHAR,  
    OUR_INTERNAL_PHONE_NUMBER VARCHAR,  
    ANSWERED_TIMESTAMP,  
    EXTERNAL_IDENTITY VARCHAR,  
    COMMENTS VARCHAR,  
    STATE VARCHAR NOT NULL,  
    SIGNIFICANCE VARCHAR NOT NULL,  
    PRIVATE BOOLEAN NOT NULL,  
    TIMESTAMP_TIMESTAMP NOT NULL,  
    ACTIVE BOOLEAN NOT NULL  
);  
CREATE INDEX EVENT_EVENT_ACTIVE_IDX  
ON EVENT(EVENT, ACTIVE);  
CREATE INDEX EVENT_USER_ACTIVE_TYPE_IDX  
ON EVENT(USER, ACTIVE, TYPE);  
CREATE INDEX EVENT_PARENT_IDX  
ON EVENT(PARENT);  
CREATE INDEX EVENT_PHONE_NUMBER_ACTIVE_IDX  
ON EVENT(PHONE_NUMBER, ACTIVE);
```

6.2.11. EVENT_PARTICIPANT

Предназначена для хранения информации об участниках редакций событий.

```
CREATE TABLE EVENT_PARTICIPANT(  
    IDENTITY BIGINT NOT NULL AUTO_INCREMENT PRIMARY KEY,  
    EVENT_REVISION BIGINT NOT NULL,  
    PARTICIPANT UUID NOT NULL  
);  
CREATE INDEX EVENT_PARTICIPANT_EVENT_REVISION_IDX  
ON EVENT_PARTICIPANT(EVENT_REVISION);
```

6.2.12. EVENT_REMINDER

Предназначена для хранения напоминаний о событиях.

```
CREATE TABLE EVENT_REMINDER(  
    IDENTITY BIGINT NOT NULL AUTO_INCREMENT PRIMARY KEY,  
    EVENT_REVISION BIGINT NOT NULL,  
    TIMESTAMP TIMESTAMP NOT NULL,  
    RELATIVE BOOLEAN NOT NULL  
);  
CREATE INDEX EVENT_REMINDER_EVENT_REVISION_IDX  
ON EVENT_REMINDER(EVENT_REVISION);  
CREATE INDEX EVENT_REMINDER_TIMESTAMP_IDX  
ON EVENT_REMINDER(TIMESTAMP);
```

6.2.13. LOGIN_AS

Эта таблица используется для хранения информации о замещениях сотрудников.

```
CREATE TABLE LOGIN_AS(  
    IDENTITY UUID NOT NULL,  
    REAL_USER UUID NOT NULL,  
    EFFECTIVE_USER UUID NOT NULL,  
    'FROM' TIMESTAMP NOT NULL,  
    TILL TIMESTAMP NOT NULL,  
    CREATED TIMESTAMP NOT NULL,  
    CREATED_BY_REAL UUID NOT NULL,
```

```
    CREATED_BY_EFFECTIVE UUID NOT NULL,  
    CREATED_BY_PROCESS UUID,  
    REVOKED TIMESTAMP,  
    REVOKED_BY_REAL UUID,  
    REVOKED_BY_EFFECTIVE UUID,  
    REVOKED_BY_PROCESS UUID  
);
```

6.2.14. OBJECT_LINK

Вспомогательная таблица для хранения истории появления связей между объектами.

```
CREATE TABLE OBJECT_LINK(  
    IDENTITY BIGINT NOT NULL AUTO_INCREMENT PRIMARY KEY,  
    A UUID NOT NULL,  
    B UUID NOT NULL,  
    CREATED TIMESTAMP NOT NULL  
);  
CREATE INDEX OBJECT_LINK_A_CREATED_IDX  
    ON OBJECT_LINK(A, CREATED DESC);  
CREATE UNIQUE INDEX OBJECT_LINK_A_B_IDX  
    ON OBJECT_LINK(A, B);
```

6.2.15. PREVIEW

Хранит записи о сгенерированном предпросмотре для документов. Информация о конкретных сгенерированных изображениях для предпросмотра хранятся в таблице PREVIEW_CONTENT.

В виде исключения допускается удалять из этой таблицы строки со значением поля PAGES равным нулю. Это записи о документах, генерация предпросмотра для которых не удалась. Если на сервер были доустановлены новые модули для генерации предпросмотра, имеет смысл удалить такие записи, чтобы при последующих обращениях к этим документам была произведена ещё одна попытка генерации предпросмотра.

```
CREATE TABLE PREVIEW(  
    PREVIEW BIGINT NOT NULL AUTO_INCREMENT PRIMARY KEY,  
    CONTENT BINARY(32) NOT NULL,
```



```
        CONTENT_TYPE VARCHAR NOT NULL,  
        PAGES INTEGER NOT NULL  
    );  
CREATE UNIQUE INDEX PREVIEW_CONTENT_CONTENT_TYPE_IDX  
    ON PREVIEW(CONTENT, CONTENT_TYPE);
```

6.2.16. PREVIEW_CONTENT

Хранит записи о сгенерированных изображениях для предпросмотра. Количество записей на один документ соответствует количеству страниц в нём.

```
CREATE TABLE PREVIEW_CONTENT(  
    IDENTITY BIGINT NOT NULL AUTO_INCREMENT PRIMARY KEY,  
    PREVIEW BIGINT NOT NULL,  
    PAGE INTEGER NOT NULL,  
    CONTENT BINARY(32) NOT NULL  
);  
CREATE INDEX PREVIEW_CONTENT_PREVIEW_IDX  
    ON PREVIEW_CONTENT(PREVIEW);
```

6.2.17. PROCESS

Содержит процессы системы.

```
CREATE TABLE PROCESS(  
    IDENTITY UUID NOT NULL PRIMARY KEY,  
    NUMBER VARCHAR,  
    KIND UUID,  
    REAL_AUTHOR UUID NOT NULL,  
    EFFECTIVE_AUTHOR UUID NOT NULL,  
    CREATED TIMESTAMP NOT NULL  
);  
CREATE INDEX PROCESS_EFFECTIVE_AUTHOR_CREATED_IDX  
    ON PROCESS(EFFECTIVE_AUTHOR, CREATED);  
CREATE INDEX PROCESS_KIND_NUMBER_IDX  
    ON PROCESS(KIND, NUMBER);
```

6.2.18. PROCESS_ATTACHMENT

Содержит информацию о вложениях в процесс.

```
CREATE TABLE PROCESS_ATTACHMENT(  
    IDENTITY BIGINT NOT NULL AUTO_INCREMENT PRIMARY KEY,  
    PROCESS UUID NOT NULL,  
    ENTITY UUID NOT NULL,  
    ADDITIONAL BOOLEAN NOT NULL  
);  
CREATE INDEX PROCESS_ATTACHMENT_PROCESS_IDX  
    ON PROCESS_ATTACHMENT(PROCESS);  
CREATE INDEX PROCESS_ATTACHMENT_ENTITY_IDX  
    ON PROCESS_ATTACHMENT(ENTITY);
```

6.2.19. PROCESS_COMMENT

Содержит комментарии процессов.

```
CREATE TABLE PROCESS_COMMENT(  
    IDENTITY BIGINT NOT NULL AUTO_INCREMENT PRIMARY KEY,  
    PROCESS UUID NOT NULL,  
    USER UUID NOT NULL,  
    TIME TIMESTAMP NOT NULL,  
    COMMENT VARCHAR NOT NULL,  
    QUESTION_FOR UUID,  
    ANSWER_NEEDED BOOLEAN  
);  
CREATE INDEX PROCESS_COMMENT_USER_IDX  
    ON PROCESS_COMMENT(USER);  
CREATE INDEX PROCESS_COMMENT_QUESTION_FOR_IDX  
    ON PROCESS_COMMENT(QUESTION_FOR);
```

6.2.20. PROCESS_ITERATION

Содержит итерации процессов.

```
CREATE TABLE PROCESS_ITERATION(  
    IDENTITY UUID NOT NULL PRIMARY KEY,  
    PROCESS UUID NOT NULL,  
    MODEL UUID NOT NULL
```

```
);  
CREATE INDEX PROCESS_ITERATION_PROCESS_IDX  
ON PROCESS_ITERATION(PROCESS);
```

6.2.21. PROCESS_LOG

Содержит журнал ошибок аварийно завершившихся процессов.

```
CREATE TABLE PROCESS_LOG(  
    ITERATION UUID NOT NULL PRIMARY KEY,  
    LOG VARCHAR NOT NULL,  
    TIMESTAMP TIMESTAMP NOT NULL  
);
```

6.2.22. PROCESS_PROPERTIES

Содержит свойства процессов, запущенных по модели.

```
CREATE TABLE PROCESS_PROPERTIES(  
    PROCESS UUID NOT NULL PRIMARY KEY,  
    PROPERTIES VARBINARY NOT NULL  
);
```

6.2.23. PROCESS_REVISION

Содержит редакции процессов.

```
CREATE TABLE PROCESS_REVISION(  
    IDENTITY BIGINT NOT NULL PRIMARY KEY,  
    PROCESS UUID NOT NULL,  
    NAME VARCHAR NOT NULL,  
    IMPORTANCE VARCHAR NOT NULL,  
    ACL VARBINARY,  
    STATE VARCHAR NOT NULL,  
    ITERATION UUID,  
    REAL_AUTHOR UUID NOT NULL,  
    EFFECTIVE_AUTHOR UUID NOT NULL,  
    TIMESTAMP TIMESTAMP NOT NULL  
);
```

```
CREATE UNIQUE INDEX PROCESS_REVISION_PROCESS_TIMESTAMP_IDX
ON PROCESS_REVISION(PROCESS, TIMESTAMP DESC);
```

6.2.24. READ_COMMENTS

Содержит записи о прочитанных пользователями комментариях в процессе.

```
CREATE TABLE READ_COMMENTS(
    IDENTITY BIGINT NOT NULL AUTO_INCREMENT PRIMARY KEY,
    PROCESS UUID NOT NULL,
    USER UUID NOT NULL,
    READ_COMMENTS INTEGER NOT NULL
);
CREATE UNIQUE INDEX READ_COMMENTS_PROCESS_USER_IDX
ON READ_COMMENTS(PROCESS, USER);
```

6.2.25. REGISTER_ENTRY

Содержит записи реестров.

```
CREATE TABLE REGISTER_ENTRY(
    IDENTITY UUID NOT NULL PRIMARY KEY,
    REGISTER UUID NOT NULL,
    VALUE VARCHAR NOT NULL,
    ENTITY UUID NOT NULL,
    REAL_AUTHOR UUID NOT NULL,
    EFFECTIVE_AUTHOR UUID NOT NULL,
    TIMESTAMP TIMESTAMP NOT NULL
);
CREATE UNIQUE INDEX REGISTER_ENTRY_VALUE_IDX
ON REGISTER_ENTRY(VALUE);
CREATE INDEX REGISTER_ENTRY_REGISTER_TIMESTAMP_IDX
ON REGISTER_ENTRY(REGISTER, TIMESTAMP);
CREATE INDEX REGISTER_ENTRY_ENTITY_IDX
ON REGISTER_ENTRY(ENTITY);
```

6.2.26. REGISTER_LINK

Хранит информацию о связанных записях реестров. Удалённые связи также хранятся в этой таблице с соответствующими пометками.

```
CREATE TABLE REGISTER_LINK(  
    IDENTITY BIGINT NOT NULL AUTO_INCREMENT PRIMARY KEY,  
    LINK_FROM UUID NOT NULL,  
    LINK_TO UUID NOT NULL,  
    REAL_AUTHOR UUID NOT NULL,  
    EFFECTIVE_AUTHOR UUID NOT NULL,  
    TIMESTAMP TIMESTAMP NOT NULL,  
    REAL_REMOVER UUID,  
    EFFECTIVE_REMOVER UUID,  
    REMOVED TIMESTAMP  
);  
CREATE INDEX REGISTER_LINK_LINK_FROM_IDX  
    ON REGISTER_LINK(LINK_FROM);  
CREATE INDEX REGISTER_LINK_LINK_TO_IDX  
    ON REGISTER_LINK(LINK_TO);
```

6.2.27. REVISION

Содержит редакции сущностей.

```
CREATE CACHED TABLE REVISION(  
    REVISION BIGINT NOT NULL AUTO_INCREMENT PRIMARY KEY,  
    HASH BINARY(32) NOT NULL,  
    PROPERTIES VARBINARY NOT NULL  
);  
CREATE UNIQUE INDEX REVISION_HASH_IDX  
    ON REVISION(HASH);
```

6.2.28. SIGNATURE

Хранит электронные подписи содержимого документов.

```
CREATE TABLE SIGNATURE(  
    IDENTITY BIGINT NOT NULL AUTO_INCREMENT PRIMARY KEY,  
    CONTENT BINARY(32) NOT NULL,
```

```
SIGNATURE BLOB NOT NULL,  
REAL_USER UUID NOT NULL,  
TIMESTAMP TIMESTAMP NOT NULL  
);  
CREATE INDEX SIGNATURE_CONTENT_IDX  
ON SIGNATURE(CONTENT);
```

6.2.29. STATE

Хранит записи о состояниях системы.

```
CREATE CACHED TABLE STATE(  
STATE BIGINT NOT NULL AUTO_INCREMENT PRIMARY KEY,  
REAL_AUTHOR UUID NOT NULL,  
EFFECTIVE_AUTHOR UUID NOT NULL,  
TIMESTAMP TIMESTAMP NOT NULL  
);
```

6.2.30. SYSTEM

Содержит одну строку с информацией о версии схемы данных и текущем состоянии системы.

```
CREATE TABLE SYSTEM(  
VERSION VARCHAR NOT NULL,  
HEAD BIGINT,  
HASH_ALGORITHM VARCHAR NOT NULL,  
CONTENT_HASH_ALGORITHM VARCHAR NOT NULL  
);
```

6.2.31. TASK

Содержит информацию о задачах в процессе.

```
CREATE TABLE TASK(  
IDENTITY UUID NOT NULL PRIMARY KEY,  
USER UUID NOT NULL,  
TITLE VARCHAR NOT NULL,  
TEXT VARCHAR,  
REAL_INITIATOR UUID,
```

```

EFFECTIVE_INITIATOR UUID,
IMPORTANCE VARCHAR NOT NULL,
POSSIBLE_RESULTS VARBINARY,
CREATED TIMESTAMP NOT NULL,
DEADLINE TIMESTAMP,
CRITICAL_PERIOD BIGINT NOT NULL,
GRACE_PERIOD BIGINT NOT NULL,
ATTACHMENTS_ACCESS VARCHAR NOT NULL,
ADDITIONAL_ATTACHMENTS_ACCESS VARCHAR NOT NULL,
ON_SHOW VARCHAR,
READ TIMESTAMP,
STATE VARCHAR NOT NULL,
COMPLETED TIMESTAMP,
COMPLETE_ACTION VARCHAR,
REAL_COMPLETER UUID,
PROCESS UUID,
ON_COMPLETION VARCHAR,
ON_STATE_CHANGED VARCHAR,
ON_REASSIGNED VARCHAR,
REASSIGNED TIMESTAMP,
REASSIGNED_FROM UUID,
REASSIGNED_BY_REAL UUID,
REASSIGNED_BY_EFFECTIVE UUID
);
CREATE INDEX TASK_USER_COMPLETED_IDX
  ON TASK(USER, COMPLETED);
CREATE INDEX TASK_CREATED_IDX
  ON TASK(CREATED);
CREATE INDEX TASK_COMPLETED_IDX
  ON TASK(COMPLETED);
CREATE INDEX TASK_PROCESS_IDX
  ON TASK(PROCESS);

```

6.2.32. TELEPHONY_ACCOUNT

Содержит информацию об учётных записях телефонии.

```

CREATE TABLE TELEPHONY_ACCOUNT(
  IDENTITY BIGINT NOT NULL PRIMARY KEY,
  USER UUID NOT NULL,

```

```
    PEER VARCHAR NOT NULL,  
    'PRIMARY' BOOLEAN NOT NULL  
);  
CREATE UNIQUE INDEX TELEPHONY_ACCOUNT_USER_PEER_IDX  
    ON TELEPHONY_ACCOUNT(USER, PEER);  
CREATE INDEX TELEPHONY_ACCOUNT_PEER_IDX  
    ON TELEPHONY_ACCOUNT(PEER);
```

6.2.33. USER_CERTIFICATE

Содержит информацию о сертификатах пользователей.

```
CREATE TABLE USER_CERTIFICATE(  
    IDENTITY BIGINT NOT NULL AUTO_INCREMENT PRIMARY KEY,  
    USER UUID NOT NULL,  
    CERTIFICATE BIGINT NOT NULL,  
    ASSIGNED TIMESTAMP NOT NULL,  
    UNASSIGNED TIMESTAMP  
);  
CREATE INDEX USER_CERTIFICATE_USER_IDX  
    ON USER_CERTIFICATE(USER);
```

6.2.34. USER_SETTING

Содержит настройки пользователей.

```
CREATE TABLE USER_SETTING(  
    IDENTITY BIGINT NOT NULL AUTO_INCREMENT PRIMARY KEY,  
    USER UUID NOT NULL,  
    NAME VARCHAR NOT NULL,  
    VALUE VARCHAR NOT NULL  
);  
CREATE UNIQUE INDEX USER_SETTING_USER_NAME_IDX  
    ON USER_SETTING(USER, NAME);
```

6.2.35. X500NAME

Хранит информацию об именах X.500.


```
CREATE TABLE X500NAME(  
    IDENTITY BIGINT NOT NULL AUTO_INCREMENT PRIMARY KEY,  
    NAME VARCHAR NOT NULL  
);  
CREATE UNIQUE INDEX X500NAME_NAME_IDX  
ON X500NAME(NAME);
```

Глава 7

Расположение конфигурации и данных

7.1. Сервер системы

7.1.1. Директория с конфигурацией и данными системы

В операционной системе Linux сервер системы устанавливается в директорию

```
/opt/ubase/server
```

Конфигурация и данные хранятся в её поддиректориях, указанных ниже.

В операционной системе Windows сервер устанавливается в директорию

```
%ProgramFiles%\U-group\uBase\VERSION
```

где `%ProgramFiles%` — путь к Program Files, а `VERSION` — версия системы. Эта директория предназначена только для программного обеспечения uBase Server. Конфигурация и данные системы хранятся отдельно. В реестре Windows в ключе

```
HKLM\SOFTWARE\U-group\uBase
```

под именем `DataRoot` записано значение с путём к директории с конфигурацией и данными системы.

Обычно в значении `DataRoot` указан путь `C:\uBase`. Путь к директории с данными на Windows задаётся при установке сервера системы uBase.

7.1.2. Файл `server.xml`

Основная конфигурация сервера системы находится в файле `server.xml`. На операционной системе Linux он располагается по фиксированному пути

```
/opt/ubase/server/etc/ubase/server.xml
```

В операционной системе Windows конфигурация сервера обычно находится по пути

```
C:\uBase\etc\ubase\server.xml
```

Однако, если в `DataRoot` записан иной путь, то конфигурационный файл следует смотреть по пути

```
%DataRoot%\etc\ubase\server.xml
```

Здесь `%DataRoot%` следует заменить на указанный путь.

Содержимое файла выглядит примерно следующим образом:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<server dbConsolePort="15888" rmiPort="8519">
<library dbPassword="password" dbSchema="UBASE"
  dbURL="jdbc:h2:./library/library1/db/db" dbUser="sa"
  enabled="true" name="Library 1"
  path="./library/library1/files" userConnectionLimit="2"/>
</server>
```

Этот файл записывается менеджером системы и читается сервером при запуске.

7.1.3. Файл `server.conf` (только для Linux)

Файл `server.conf` на Linux расположен по пути

```
/opt/ubase/server/etc/ubase/server.conf
```

и выглядит примерно следующим образом:

```
# Set your language here. Encoding should remain UTF-8.
LANG=en_US.UTF-8
# Do not remove this line
```

Для русского языка, как уже указано в описании установки системы, значение переменной `LANG` заменяется на `ru_RU.UTF-8`. Это действие производится вручную после первой установки системы до запуска сервера. Наличие этого файла связано с тем, что в ряде систем `systemd` при запуске сервисов по соображениям безопасности не предоставляет им переменные среды и с тем, что в ряде систем изначально локаль для сервисов не настроена надлежащим образом.

7.1.4. Расположение данных

На сервере системы настроено некоторое количество библиотек. Как правило, для промышленной эксплуатации создаётся и настраивается только одна библиотека, но тестовых и иных серверах библиотек может быть больше. По умолчанию, для каждой библиотеки создаётся своя поддиректория внутри директории

```
/opt/ubase/server/library
```

на Linux и внутри директории

```
%DataRoot%\library
```

на Windows.

Внутри директории конкретной библиотеки создаются поддиректории для базы данных, файлового хранилища и прочих данных системы.

Внутри поддиректории `db` находится файл `db.h2.db` с базой данных системы. Помимо него в ней должен находиться файл блокировок `db.lock.db` и может находиться файл `db.trace.db`. Если система настроена на использование иной базы данных вместо встроенной, то директория `db`, скорее всего, не будет создана.

В поддиректории `files` располагается файловое хранилище с содержимым документов. Внутри этой директории может находиться скрытая поддиректория `.temp`, содержащая только временные файлы. Остальные поддиректории директории `files` содержат данные хранилища системы. До первого обращения системы к файловому хранилищу сама директория `files` не будет создана. Файловое хранилище используется для хранения содержимого документов, значков классов сущностей и т. п.

В поддиректории `index` находится индекс для поиска объектов в системе. Резервное копирование этой директории не требуется. Если при запуске сервера директория индекса не будет обнаружена, она будет создана и индекс будет сгенерирован заново, что займёт некоторое

время. При восстановлении библиотеки из резервных копий рекомендуется удалять директорию с индексом до запуска сервера, чтобы индекс был пересоздан под восстановленные из резервной копии данные.

В поддиректории `upload` временно хранится загружаемое на сервер содержимое документов. Резервное копирование этой директории не требуется.

В поддиректории `preview` находятся данные файлового хранилища с данными предварительного просмотра документов. Резервное копирование этой директории не требуется. При утере данных хранилища предварительного просмотра он будет регенерирован по мере необходимости.

7.2. Клиент системы

При описании путей на Windows ниже используются переменные среды `%APPDATA%` и `%LOCALAPPDATA%`.

На современных системах Windows `%APPDATA%` обычно соответствует

```
C:\Users\<имя пользователя>\AppData\Roaming
```

На устаревших системах Windows это обычно

```
C:\Documents and Settings\<имя пользователя>\Application Data
```

На современных системах Windows `%LOCALAPPDATA%` чаще всего располагается по пути

```
C:\Users\<имя пользователя>\AppData\Local
```

Для устаревших систем путь другой:

```
C:\Documents and Settings\<имя пользователя>\Local Settings\Application Data
```

При описании путей на Linux ниже используется символ `~`, обозначающий домашнюю директорию пользователя.

Если определена переменная среды `$XDG_CONFIG_HOME`, то `~/.config` в описаниях нужно заменить на её значение.

Если определена переменная среды `$XDG_CACHE_HOME`, то `~/.cache` в описаниях нужно заменить на её значение.

Если определена переменная среды `$XDG_DATA_HOME`, то `~/.local/share` в описаниях нужно заменить на её значение.

7.2.1. Файл `client.xml`

В файле `client.xml` находится информация об используемых серверах системы. Этот файл создаётся при установке клиента в соответствии с указанными установщику значениями. Установщик проверяет доступность сервера и наличие на нём поддержки клиентов для используемой операционной системы, поэтому начальные настройки в этом файле должны быть корректны.

В случае переноса сервера на другой адрес или при необходимости добавить ещё один сервер (например, у администратора системы может быть дополнительно настроен тестовый сервер), содержимое этого файла можно откорректировать.

На Windows этот файл расположен по пути

```
%APPDATA%\uBase\Client\client.xml
```

На Linux этот файл расположен по пути

```
~/.config/uBase/Client/client.xml
```

Содержимое этого файла выглядит приблизительно следующим образом:

```
<?xml version='1.0' encoding='UTF-8' standalone='no'?>
<client>
  <server url='10.0.0.1:8519'/>
</client>
```

Если в тестовых целях требуется использование дополнительных серверов, то добавляется соответствующее количество тегов `server`. В параметре `url` должны быть указаны адрес или имя сервера, двоеточие и порт.

7.2.2. Файл блокировки

Этот файл используется для предотвращения повторного запуска клиента. Наличие этого файла при незапущенном клиенте также не является проблемой. Клиент использует блокировки этого файла, а не просто проверяет его наличие.

На Windows этот файл расположен по пути

```
%LOCALAPPDATA%\uBase\Client\lock\.lock
```

На Linux этот файл расположен по пути

```
~/.config/uBase/Client/lock/.lock
```

7.2.3. Кэш файлового хранилища

Загружаемое с сервера содержимое документов и некоторые другие данные попадают в кэш файлового хранилища. Клиент системы следит за тем, чтобы кэш не занимал слишком много места, поэтому его обслуживание обычно не требуется.

Однако директорию кэша можно без опасения удалить целиком при незапущенном клиенте. При следующем запуске она будет создана повторно.

На Windows директория кэша расположена по пути

```
%LOCALAPPDATA%\uBase\Client\files
```

На Linux директория кэша расположена по пути

```
~/.cache/uBase/Client/files
```

7.2.4. Данные сервиса просмотра, редактирования и интеграции

На Windows директория сервиса расположена по пути

```
%LOCALAPPDATA%\U-group
```

На Linux директория сервиса расположена по пути

```
~/.local/share/U-group
```

В директории сервиса создаются отдельные поддиректории для каждой библиотеки.

Также в директории сервиса клиент создаёт файл `.userProfile`. Этот файл используется сервисом редактирования для идентификации рабочего места при выписке редактируемых документов. Его не следует удалять.

7.2.5. Путь установки клиента системы

На Windows установка происходит внутрь директории

```
%LOCALAPPDATA%\uBase\Client
```

На Linux установка происходит внутрь директории

```
~/local/share/uBase/Client
```

Внутри этой директории создаётся поддиректория с именем версии клиента. При автоматическом обновлении клиента появляется новая поддиректория, а старая удаляется.

Следует заметить, что клиент системы при запуске удаляет из этой директории все посторонние файлы и директории.

7.2.6. Журналы работы клиента

Отладочные сообщения клиент системы записывает в файлы в директории с журналами.

На Windows журналы записываются в директорию

```
%LOCALAPPDATA%\uBase\Client\log
```

На Linux журналы записываются в директорию

```
~/local/share/uBase/Client/log
```

При обращении за технической поддержкой вам может понадобиться содержимое этой директории с рабочего места, на котором возникла проблема.

Приложение А

Шаблоны библиотеки

А.1. Шаблоны общего назначения

А.1.1. Базовый шаблон uBase

Аккаунт пользователя

Наследует класс: Персона (стр. 87).

Имя свойства	Описание	Тип	Обязательное
E-mail		STRING	Нет
Активен		BOOLEAN	Да
Группы		SLAVE M2M	Нет
Значок		CONTENT HASH	Нет
Имя		STRING	Нет
Имя входа		STRING	Да
Инициалы		STRING	Нет
ИНН		STRING	Нет
Классы сущности		MASTER M2M	Да
Личная папка		MASTER O2O	Нет
Отчество		STRING	Нет
Папки		SLAVE M2M	Нет

Продолжение на следующей странице

Имя свойства	Описание	Тип	Обязательное
Полное имя		STRING	Да
Почтовый адрес		STRING	Нет
Префикс имени		STRING	Нет
Примечание		TEXT	Нет
Суффикс имени		STRING	Нет
Телефоны		STRING	Нет
Управляемые группы		MASTER M2M	Нет
Фактический адрес		STRING	Нет
Фамилия		STRING	Нет
Фотография		CONTENT HASH	Нет
Юридический адрес		STRING	Нет

Валюта

Наследует класс: Справочник (стр. 98).

Имя свойства	Описание	Тип	Обязательное
Классы сущности		MASTER M2M	Да
Код ISO		STRING	Нет
Папки		SLAVE M2M	Нет
Полное имя		STRING	Да
Примечание		TEXT	Нет

Вид активности

Наследует класс: Обычное дерево (стр. 83).

Имя свойства	Описание	Тип	Обязательное
Дочерние элементы		MASTER M2M	Нет
Классы сущности		MASTER M2M	Да
Модели		MASTER O2M	Нет
Папки		SLAVE M2M	Нет
Период		ENUM.	Нет
Полное имя		STRING	Да
Примечание		TEXT	Нет
Родительские элементы		MASTER M2M	Нет
Формат		STRING	Нет

Город

Наследует класс: Справочник (стр. 98).

Имя свойства	Описание	Тип	Обязательное
Классы сущности		MASTER M2M	Да
Организации		MASTER O2M	Нет
Папки		SLAVE M2M	Нет
Полное имя		STRING	Да
Примечание		TEXT	Нет
Регион		MASTER M2O	Нет

Группа

Наследует класс: Пользователь или группа (стр. 91).

Имя свойства	Описание	Тип	Обязательное
Группы		SLAVE M2M	Нет
Классы сущности		MASTER M2M	Да
Папки		SLAVE M2M	Нет
Полное имя		STRING	Да
Примечание		TEXT	Нет
Руководители группы		MASTER M2M	Нет
Члены группы		MASTER M2M	Нет

Дерево

Наследует класс: Сущность (стр. 100).

Имя свойства	Описание	Тип	Обязательное
Классы сущности		MASTER M2M	Да
Папки		SLAVE M2M	Нет
Полное имя		STRING	Да
Примечание		TEXT	Нет

Договор

Наследует класс: Документ (стр. 78).

Имя свойства	Описание	Тип	Обязательное
Валюта		MASTER M2O	Нет
Дата документа		DATE	Нет
Классы сущности		MASTER M2M	Да
Курс валюты		FLOAT	Нет
Наша организация		MASTER M2O	Нет
Номер документа		STRING	Нет
Папки		SLAVE M2M	Нет
Полное имя		STRING	Да
Примечание		TEXT	Нет
Содержимое		CONTENT HASH	Нет
Срок действия		DATE	Нет
Срок исполнения		DATE	Нет
Сумма		FLOAT	Нет
Сумма в валюте		FLOAT	Нет
Тип договора		MASTER M2O	Нет
Тип содержимого		STRING	Нет
Участник		MASTER M2O	Нет

Документ

Наследует класс: Сущность (стр. 100).

Имя свойства	Описание	Тип	Обязательное
Классы сущности		MASTER M2M	Да
Папки		SLAVE M2M	Нет
Полное имя		STRING	Да
Примечание		TEXT	Нет
Содержимое		CONTENT HASH	Нет
Тип содержимого		STRING	Нет

Должность

Наследует класс: Справочник (стр. 98).

Имя свойства	Описание	Тип	Обязательное
Классы сущности		MASTER M2M	Да
Папки		SLAVE M2M	Нет
Полное имя		STRING	Да
Примечание		TEXT	Нет

Дополнительное соглашение

Наследует класс: Договор (стр. 77).

Имя свойства	Описание	Тип	Обязательное
Валюта		MASTER M2O	Нет
Дата доку- мента		DATE	Нет
Договор		REGISTER ENTRY	Нет
Классы сущ- ности		MASTER M2M	Да
Курс валюты		FLOAT	Нет
Наша органи- зация		MASTER M2O	Нет
Номер доку- мента		STRING	Нет
Папки		SLAVE M2M	Нет
Полное имя		STRING	Да
Примечание		TEXT	Нет
Содержимое		CONTENT HASH	Нет
Срок дей- ствия		DATE	Нет
Срок исполне- ния		DATE	Нет
Сумма		FLOAT	Нет
Сумма в ва- люте		FLOAT	Нет
Тип договора		MASTER M2O	Нет
Тип содержи- мого		STRING	Нет
Участник		MASTER M2O	Нет

Класс сущности

Наследует классы: Метаданные (стр. 81), Обычное дерево (стр. 83).

Имя свойства	Описание	Тип	Обязательное
Дочерние элементы		MASTER M2M	Нет
Значок		CONTENT HASH	Нет
Классы сущности		MASTER M2M	Да
Папки		SLAVE M2M	Нет
Полное имя		STRING	Да
Права доступа к содержимому		ACL	Нет
Права доступа к содержимому по умолчанию		ACL	Нет
Права по умолчанию		ACL	Нет
Примечание		TEXT	Нет
Родительские элементы		MASTER M2M	Нет
Свойства		MASTER M2M	Нет
Таблица		TEXT	Нет
Тип класса сущности		ENUM.	Да
Форма		TEXT	Нет

Контейнер

Наследует класс: Сущность (стр. 100).

Имя свойства	Описание	Тип	Обязательное
Классы сущности		MASTER M2M	Да
Папки		SLAVE M2M	Нет
Полное имя		STRING	Да
Права доступа к содержимому		ACL	Нет
Примечание		TEXT	Нет

Метаданные

Наследует класс: Контейнер (стр. 80).

Имя свойства	Описание	Тип	Обязательное
Классы сущности		MASTER M2M	Да
Папки		SLAVE M2M	Нет
Полное имя		STRING	Да
Права доступа к содержимому		ACL	Нет
Примечание		TEXT	Нет

Модель маршрутизации

Наследует класс: Контейнер (стр. 80).

Имя свойства	Описание	Тип	Обязательное
Вид активности		MASTER M2O	Нет
Классы сущностей для запуска		MASTER M2M	Нет
Классы сущности		MASTER M2M	Да
Код пользователя		TEXT	Нет
Код сервера		TEXT	Нет
Папки		SLAVE M2M	Нет
Полное имя		STRING	Да
Права доступа к содержимому		ACL	Нет
Права на запуск процесса		ACL	Нет
Права по умолчанию		ACL	Нет
Примечание		TEXT	Нет

Наша организация

Наследует класс: Организация (стр. 84).

Имя свойства	Описание	Тип	Обязательное
E-mail		STRING	Нет
Город		MASTER M2O	Нет
Группы		SLAVE M2M	Нет
ИНН		STRING	Нет

Продолжение на следующей странице

Имя свойства	Описание	Тип	Обязательное
Классы сущности		MASTER M2M	Да
КПП		STRING	Нет
Папки		SLAVE M2M	Нет
Полное имя		STRING	Да
Почтовый адрес		STRING	Нет
Примечание		TEXT	Нет
Руководители группы		MASTER M2M	Нет
Телефоны		STRING	Нет
Фактический адрес		STRING	Нет
Члены группы		MASTER M2M	Нет
Юридический адрес		STRING	Нет
Юридическое наименование		STRING	Нет

Обычное дерево

Наследует класс: Дерево (стр. 76).

Имя свойства	Описание	Тип	Обязательное
Дочерние элементы		MASTER M2M	Нет
Классы сущности		MASTER M2M	Да
Папки		SLAVE M2M	Нет
Полное имя		STRING	Да
Примечание		TEXT	Нет

Продолжение на следующей странице

Имя свойства	Описание	Тип	Обязательное
Родительские элементы		MASTER M2M	Нет

Организация

Наследует классы: Подразделение (стр. 90), Справочник (стр. 98), Участник (стр. 101).

Имя свойства	Описание	Тип	Обязательное
E-mail		STRING	Нет
Город		MASTER M2O	Нет
Группы		SLAVE M2M	Нет
ИНН		STRING	Нет
Классы сущности		MASTER M2M	Да
КПП		STRING	Нет
Папки		SLAVE M2M	Нет
Полное имя		STRING	Да
Почтовый адрес		STRING	Нет
Примечание		TEXT	Нет
Руководители группы		MASTER M2M	Нет
Телефоны		STRING	Нет
Фактический адрес		STRING	Нет
Члены группы		MASTER M2M	Нет
Юридический адрес		STRING	Нет
Юридическое наименование		STRING	Нет

Отчёт

Наследует классы: Контейнер (стр. 80), Обычное дерево (стр. 83).

Имя свойства	Описание	Тип	Обязательное
Дочерние элементы		MASTER M2M	Нет
Классы сущности		MASTER M2M	Да
Код генератора отчёта		TEXT	Нет
Код интерфейса запроса		TEXT	Нет
Папки		SLAVE M2M	Нет
Полное имя		STRING	Да
Права доступа к содержимому		ACL	Нет
Примечание		TEXT	Нет
Родительские элементы		MASTER M2M	Нет

Папка

Наследует класс: Сущность (стр. 100).

Имя свойства	Описание	Тип	Обязательное
Классы сущности		MASTER M2M	Да
Папки		SLAVE M2M	Нет
Полное имя		STRING	Да

Продолжение на следующей странице

Имя свойства	Описание	Тип	Обязательное
Примечание		TEXT	Нет
Содержимое папки		MASTER M2M	Нет

Первичный документ

Наследует класс: Документ (стр. 78).

Имя свойства	Описание	Тип	Обязательное
Валюта		MASTER M2O	Нет
Дата документа		DATE	Нет
Договор		REGISTER ENTRY	Нет
Классы сущности		MASTER M2M	Да
Курс валюты		FLOAT	Нет
Наша организация		MASTER M2O	Нет
Номер документа		STRING	Нет
Папки		SLAVE M2M	Нет
Полное имя		STRING	Да
Примечание		TEXT	Нет
Содержимое		CONTENT HASH	Нет
Сумма		FLOAT	Нет
Сумма в валюте		FLOAT	Нет
Тип содержимого		STRING	Нет
Участник		MASTER M2O	Нет

Персона

Наследует классы: Пользователь или группа (стр. 91), Участник (стр. 101).

Имя свойства	Описание	Тип	Обязательное
E-mail		STRING	Нет
Группы		SLAVE M2M	Нет
Значок		CONTENT HASH	Нет
Имя		STRING	Нет
Инициалы		STRING	Нет
ИНН		STRING	Нет
Классы сущности		MASTER M2M	Да
Отчество		STRING	Нет
Папки		SLAVE M2M	Нет
Полное имя		STRING	Да
Почтовый адрес		STRING	Нет
Префикс имени		STRING	Нет
Примечание		TEXT	Нет
Суффикс имени		STRING	Нет
Телефоны		STRING	Нет
Управляемые группы		MASTER M2M	Нет
Фактический адрес		STRING	Нет
Фамилия		STRING	Нет
Фотография		CONTENT HASH	Нет

Продолжение на следующей странице

Имя свойства	Описание	Тип	Обязательное
Юридический адрес		STRING	Нет

Письмо

Наследует класс: Документ (стр. 78).

Имя свойства	Описание	Тип	Обязательное
Дата документа		DATE	Нет
Договоры		REGISTER ENTRY SET	Нет
Классы сущности		MASTER M2M	Да
Наша организация		MASTER M2O	Нет
Номер документа		STRING	Нет
Папки		SLAVE M2M	Нет
Полное имя		STRING	Да
Примечание		TEXT	Нет
Содержимое		CONTENT HASH	Нет
Тип содержимого		STRING	Нет

Письмо входящее

Наследует класс: Письмо (стр. 88).

Имя свойства	Описание	Тип	Обязательное
Дата документа		DATE	Нет
Договоры		REGISTER ENTRY SET	Нет
Классы сущности		MASTER M2M	Да
Наша организация		MASTER M2O	Нет
Номер документа		STRING	Нет
Папки		SLAVE M2M	Нет
Письмо исходящее		REGISTER ENTRY	Нет
Полное имя		STRING	Да
Примечание		TEXT	Нет
Содержимое		CONTENT HASH	Нет
Сотрудники		MASTER M2M	Нет
Тип содержимого		STRING	Нет
Участник		MASTER M2O	Нет

Письмо исходящее

Наследует класс: Письмо (стр. 88).

Имя свойства	Описание	Тип	Обязательное
Дата документа		DATE	Нет

Продолжение на следующей странице

Имя свойства	Описание	Тип	Обязательное
Договоры		REGISTER ENTRY SET	Нет
Классы сущности		MASTER M2M	Да
Наша организация		MASTER M2O	Нет
Номер документа		STRING	Нет
Папки		SLAVE M2M	Нет
Письмо входящее		REGISTER ENTRY	Нет
Полное имя		STRING	Да
Получатели		MASTER M2M	Нет
Примечание		TEXT	Нет
Содержимое		CONTENT HASH	Нет
Сотрудник		MASTER M2O	Нет
Тип содержимого		STRING	Нет

Подразделение

Наследует классы: Группа (стр. 76), Справочник (стр. 98).

Имя свойства	Описание	Тип	Обязательное
Группы		SLAVE M2M	Нет
Классы сущности		MASTER M2M	Да

Продолжение на следующей странице

Имя свойства	Описание	Тип	Обязательное
Папки		SLAVE M2M	Нет
Полное имя		STRING	Да
Примечание		TEXT	Нет
Руководители группы		MASTER M2M	Нет
Члены группы		MASTER M2M	Нет

Пользователь или группа

Наследует класс: Сущность (стр. 100).

Имя свойства	Описание	Тип	Обязательное
Группы		SLAVE M2M	Нет
Классы сущности		MASTER M2M	Да
Папки		SLAVE M2M	Нет
Полное имя		STRING	Да
Примечание		TEXT	Нет

Приказ

Наследует класс: Документ (стр. 78).

Имя свойства	Описание	Тип	Обязательное
Классы сущности		MASTER M2M	Да

Продолжение на следующей странице

Имя свойства	Описание	Тип	Обязательное
Наша организация		MASTER M2O	Нет
Папки		SLAVE M2M	Нет
Полное имя		STRING	Да
Примечание		TEXT	Нет
Содержимое		CONTENT HASH	Нет
Сотрудник		MASTER M2O	Нет
Сотрудники		MASTER M2M	Нет
Тип содержимого		STRING	Нет

Приказ по личному составу

Наследует класс: Приказ (стр. 91).

Имя свойства	Описание	Тип	Обязательное
Классы сущности		MASTER M2M	Да
Наша организация		MASTER M2O	Нет
Папки		SLAVE M2M	Нет
Полное имя		STRING	Да
Примечание		TEXT	Нет
Содержимое		CONTENT HASH	Нет
Сотрудник		MASTER M2O	Нет
Сотрудники		MASTER M2M	Нет

Продолжение на следующей странице

Имя свойства	Описание	Тип	Обязательное
Тип содержимого		STRING	Нет

Приказ по основной деятельности

Наследует класс: Приказ (стр. 91).

Имя свойства	Описание	Тип	Обязательное
Классы сущности		MASTER M2M	Да
Наша организация		MASTER M2O	Нет
Папки		SLAVE M2M	Нет
Полное имя		STRING	Да
Примечание		TEXT	Нет
Содержимое		CONTENT HASH	Нет
Сотрудник		MASTER M2O	Нет
Сотрудники		MASTER M2M	Нет
Тип содержимого		STRING	Нет

Регион

Наследует класс: Справочник (стр. 98).

Имя свойства	Описание	Тип	Обязательное
Города		MASTER O2M	Нет
Классы сущности		MASTER M2M	Да
Папки		SLAVE M2M	Нет
Полное имя		STRING	Да
Примечание		TEXT	Нет
Страна		MASTER M2O	Нет

Реестр

Наследует класс: Контейнер (стр. 80).

Имя свойства	Описание	Тип	Обязательное
Классы новых зарегистрированных сущностей		MASTER M2M	Нет
Классы сущности		MASTER M2M	Да
Папка		MASTER M2O	Нет
Папки		SLAVE M2M	Нет
Период		ENUM.	Да
Полное имя		STRING	Да
Права доступа к содержимому		ACL	Нет
Права по умолчанию		ACL	Нет
Примечание		TEXT	Нет

Продолжение на следующей странице

Имя свойства	Описание	Тип	Обязательное
Путь		STRING	Нет
Ссылки		MASTER O2M	Нет
Требуемые классы регистрируемых сущностей		MASTER M2M	Нет
Формат		STRING	Да

Реестр организации

Наследует класс: Реестр (стр. 94).

Имя свойства	Описание	Тип	Обязательное
Классы новых зарегистрированных сущностей		MASTER M2M	Нет
Классы сущности		MASTER M2M	Да
Наша организация		MASTER M2O	Нет
Папка		MASTER M2O	Нет
Папки		SLAVE M2M	Нет
Период		ENUM.	Да
Полное имя		STRING	Да
Права доступа к содержанию		ACL	Нет
Права по умолчанию		ACL	Нет
Примечание		TEXT	Нет

Продолжение на следующей странице

Имя свойства	Описание	Тип	Обязательное
Путь		STRING	Нет
Ссылки		MASTER O2M	Нет
Требуемые классы регистрируемых сущностей		MASTER M2M	Нет
Формат		STRING	Да

Свойство

Наследует класс: Метаданные (стр. 81).

Имя свойства	Описание	Тип	Обязательное
Доступ к значению		ENUM.	Да
Класс-перечисление		STRING	Нет
Классы		MASTER M2M	Нет
Классы сущности		MASTER M2M	Да
Неизменяемое значение		BOOLEAN	Да
Папки		SLAVE M2M	Нет
Полное имя		STRING	Да
Права доступа к содержанию		ACL	Нет
Примечание		TEXT	Нет
Специальный редактор		STRING	Нет

Продолжение на следующей странице

Имя свойства	Описание	Тип	Обязательное
Ссылается на свойство		MASTER M2O	Нет
Ссылается на сущность		MASTER M2O	Нет
Тип данных		ENUM.	Да
Требование значения		ENUM.	Да
Фильтр		STRING	Нет

Сотрудник

Наследует класс: Персона (стр. 87).

Имя свойства	Описание	Тип	Обязательное
E-mail		STRING	Нет
Группы		SLAVE M2M	Нет
Должность		STRING	Нет
Значок		CONTENT HASH	Нет
Имя		STRING	Нет
Инициалы		STRING	Нет
ИНН		STRING	Нет
Классы сущности		MASTER M2M	Да
Отчество		STRING	Нет
Папки		SLAVE M2M	Нет
Полное имя		STRING	Да
Почтовый адрес		STRING	Нет
Префикс имени		STRING	Нет
Примечание		TEXT	Нет

Продолжение на следующей странице

Имя свойства	Описание	Тип	Обязательное
Суффикс имени		STRING	Нет
Телефоны		STRING	Нет
Управляемые группы		MASTER M2M	Нет
Фактический адрес		STRING	Нет
Фамилия		STRING	Нет
Фотография		CONTENT HASH	Нет
Юридический адрес		STRING	Нет

Справочник

Наследует класс: Сущность (стр. 100).

Имя свойства	Описание	Тип	Обязательное
Классы сущности		MASTER M2M	Да
Папки		SLAVE M2M	Нет
Полное имя		STRING	Да
Примечание		TEXT	Нет

Ссылки на реестр

Наследует класс: Табличная часть (стр. 100).

Имя свойства	Описание	Тип	Обязательное
Классы сущностей ссылок		MASTER M2M	Да
Классы сущности		MASTER M2M	Да
Папки		SLAVE M2M	Нет
Полное имя		STRING	Да
Примечание		TEXT	Нет
Путь		STRING	Нет
Реестр		MASTER M2O	Да

Страна

Наследует класс: Справочник (стр. 98).

Имя свойства	Описание	Тип	Обязательное
Классы сущности		MASTER M2M	Да
Папки		SLAVE M2M	Нет
Полное имя		STRING	Да
Примечание		TEXT	Нет
Регионы		MASTER O2M	Нет

Строгое дерево

Наследует класс: Дерево (стр. 76).

Имя свойства	Описание	Тип	Обязательное
Дочерние элементы		MASTER O2M	Нет
Классы сущности		MASTER M2M	Да
Папки		SLAVE M2M	Нет
Полное имя		STRING	Да
Примечание		TEXT	Нет
Родительский элемент		MASTER M2O	Нет

Сущность

Имя свойства	Описание	Тип	Обязательное
Классы сущности		MASTER M2M	Да
Папки		SLAVE M2M	Нет
Полное имя		STRING	Да
Примечание		TEXT	Нет

Табличная часть

Наследует класс: Сущность (стр. 100).

Имя свойства	Описание	Тип	Обязательное
Классы сущности		MASTER M2M	Да
Папки		SLAVE M2M	Нет
Полное имя		STRING	Да

Продолжение на следующей странице

Имя свойства	Описание	Тип	Обязательное
Примечание		TEXT	Нет

Тип договора

Наследует класс: Справочник (стр. 98).

Имя свойства	Описание	Тип	Обязательное
Классы сущности		MASTER M2M	Да
Код		STRING	Да
Папки		SLAVE M2M	Нет
Полное имя		STRING	Да
Примечание		TEXT	Нет

Участник

Наследует класс: Справочник (стр. 98).

Имя свойства	Описание	Тип	Обязательное
E-mail		STRING	Нет
ИНН		STRING	Нет
Классы сущности		MASTER M2M	Да
Папки		SLAVE M2M	Нет
Полное имя		STRING	Да
Почтовый адрес		STRING	Нет
Примечание		TEXT	Нет
Телефоны		STRING	Нет

Продолжение на следующей странице

Имя свойства	Описание	Тип	Обязательное
Фактический адрес		STRING	Нет
Юридический адрес		STRING	Нет